11th International Conference on Geographic Information Science

GIScience 2021, September 27–30, 2021, Poznań, Poland

Part I

Edited by
Krzysztof Janowicz
Judith A. Verstegen

Editors

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

Judith A. Verstegen
University of Münster, Germany
j.a.verstegen@uni-muenster.de

ACM Classification 2012
Information systems → Geographic information systems; Human-centered computing → Geographic visualization; Theory of computation → Computational geometry; Computing methodologies → Machine learning; Information systems → Temporal data; Information systems → Spatial-temporal systems

ISBN 978-3-95977-166-5

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/978-3-95977-166-5.

Publication date
September, 2020

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.

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

ISBN 978-3-95977-166-5 **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.

ISSN 1868-8969

https://www.dagstuhl.de/lipics

Contents

Regular Papers

11th International Conference on Geographic Information Science (GIScience 2021) – Part I.
Editors: Krzysztof Janowicz and Judith A. Verstegen
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Preface

This first volume contains the full paper proceedings of the 11th International Conference on Geographic Information Science (GIScience 2020) that was scheduled to be held in Poznań, Poland, 15-18 September, 2020. The conference and its submission deadlines were affected by the widespread outbreak of COVID-19. Consequently, the program chairs decided to split the full paper track into two deadlines to accommodate authors impacted by the pandemic with one deadline mid of March and the second deadline roughly four weeks later. The short paper track was canceled approximately one month before the deadline.

Overall, we received 50 submissions, out of which 44 were full papers and six were short papers submitted before the track was canceled. While most papers received four reviews, the number of reviews varied between three and six, e.g., when reviewers provided their reviews well past the deadline while we had already assigned emergency reviewers. 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. In one case, we requested a second metareview. One manuscript was accepted conditionally at first, to undergo another round of editorial checks. In total, we accepted 17 submissions for this first volume.

The accepted papers represent a wide range of topics at the forefront of GIScience research including work on computational geometry, routing and traffic forecasting, the privacy of trajectories, the analysis of geo-social media, and (mobile) sensors.

In coordination with the GIScience series' Steering Committee, the organizers canceled the in-person part of GIScience 2020 due to the worsening pandemic and decided not to hold an online event. Instead, they voted to postpone the conference to 2021, departing from the usual 2-year rhythm, and to combine the papers in this volume with a second round of full papers (and short papers) with an anticipated submission deadline in March 2021 for the full papers and June 2021 for the short papers. GIScience 2021 will be held on September 27-30, 2021 in Poznań, Poland. The workshops and tutorials accepted for GIScience 2020 were given the opportunity to postpone until 2021 as well and a small number of workshops may be added for the 2021 edition to ensure that emerging topics can be covered appropriately.

The entire GIScience 2020 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 local organizing team in Poznan for their flexibility during these difficult times.

The GIScience 2020/21 Organizing Team

11th International Conference on Geographic Information Science (GIScience 2021) – Part I.
Editors: Krzysztof Janowicz and Judith A. Verstegen
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Conference Organization

General Chair
Piotr Jankowski, San Diego State University, USA & Adam Mickiewicz University, Poland
Program Chairs
Krzysztof Janowicz, University of California, Santa Barbara, USA
Judith Verstegen, University of Münster, Germany
Local Organization Chair
Zbigniew Zwolinski, Adam Mickiewicz University, Poland
Workshop and Tutorial Chairs
Marcin Winowski, Adam Mickiewicz University, Poland
Grant McKenzie, McGill University, Canada
Publicity Chairs
Clio Andris, Georgia Institute of Technology (Georgia Tech), USA
Alfred Stach, Adam Mickiewicz University, Poland
Sponsorship Chair
Bernd Resch, University of Salzburg, Austria
Arrangements and Logistics
Joanna Gudowicz, Adam Mickiewicz University, Poland
Robert Kruszyk, Adam Mickiewicz University, Poland
arosław Jasiewicz, Adam Mickiewicz University, Poland
Paweł Matulewski, Adam Mickiewicz University, Poland
Małgorzata Mazurek, Adam Mickiewicz University, Poland
Alicja Najwer, Adam Mickiewicz University, Poland
Justyna Weltrowska, Adam Mickiewicz University, 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, Poland

11th International Conference on Geographic Information Science (GIScience 2021) – Part I.
Editors: Krzysztof Janowicz and Judith A. Verstegen
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Program Committee

Ben Adams	University of Canterbury, New Zeeland
Sean Ahearn	City University of New York, USA
Natalia Andrienko	Fraunhofer Institute IAIS, Germany
Gennady Andrienko	Fraunhofer Institute IAIS, Germany
Clio Andris[m]	Georgia Tech, USA
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
Boyan Brodaric	Geological Survey of Canada, Canada
Ling Cai[e]	University of California, Santa Barbara, USA
Christophe Claramunt	Naval Academy Research Institute, France
Keith Clarke	University of California, Santa Barbara, USA
Eliseo Clementini	University of L'Aquila, Italy
Clodoveu Davis	Universidade Federal de Minas Gerais, Brazil
Somayeh Dodge[m]	University of California, Santa Barbara, USA
Matt Duckham	RMIT University, Australia
Ekaterina Egorova	University of Zürich, Switzerland
Sara Irina Fabrikant	University of Zürich, Switzerland
Christian Freksa	University of Bremen, Germany
Mark Gahegan	University of Auckland, New Zeeland
Song Gao[e]	University of Wisconsin-Madison, USA
Rina Ghose	University of Wisconsin-Milwaukee, USA
Ioannis Giannopoulos	TU Vienna, Austria
Darius Gotlib	Warsaw University of Technology, Poland
Amy Griffin	RMIT University, Australia
Tony Grubesic	Arizona State University, USA
Torsten Hahmann	University of Maine, USA
Francis Harvey	University of Leipzig, Germany
Jan-Henrik Haunert	University of Bonn, Germany
Gerard Heuvelink	Wageningen University, The Netherlands
Stephen Hirtle	University of Pittsburgh, USA
Hartwig Hochmair	University of Florida, USA
Bernhard Höfle	University of Heidelberg, Germany
Yingjie Hu[e,m]	University of Buffalo, USA
Marta Jankowska	University of California, San Diego, USA
Jaroslaw Jasiewicz	Adam Mickiewicz University, Poznań, Poland
Bin Jiang	University of Gävle, Sweden
Christopher Jones	Cardiff University, UK
Carsten Keßler	Aalborg University, Denmark
Peter Kiefer	ETH Zürich, Switzerland
Alexander Klippel	Pennsylvania State University, USA
Christian Kray	University of Münster, Germany
Marc van Kreveld	Utrecht University, The Netherlands
Phaedon Kyriakidis	Cyprus University of Technology, Greece
Shawn Laffan	University of New South Wales, Australia

Nina Lam	Louisiana State University, USA
Michael Leitner	Louisiana State University, USA
WenWen Li[m]	Arizona State University, USA
Gengchen Mai[e]	University of California, Santa Barbara, USA
Ed Manley[m]	University of Leeds, UK
Bruno Martins[m]	University of Lisbon, Portugal
Grant McKenzie[m]	McGill University, Canada
Liqiu Meng	TU München, Germany
Harvey Miller	The Ohio State University, USA
Jennifer Miller	University of Texas, Austin, USA
Daniel R. Montello	University of California, Santa Barbara, USA
Mir Abolfazl Mostafavi	Université Laval, Canada
Alan Murray	University of California, Santa Barbara, USA
Atsushi Nara[m]	San Diego State University, USA
Robert Olszewski	Warsaw University of Technology, Poland
David O'Sullivan	University of Wellington, New Zeeland
Edzer Pebesma	University of Münster, Germany
Karin Pfeffer	University of Twente, The Netherlands
Ross Purves	University of Zürich, Switzerland
Martin Raubal	ETH Zürich, Switzerland
Bernd Resch	University of Salzburg, Austria
Simon Scheider[e,m]	Utrecht University, The Netherlands
Oliver Schmitz	Utrecht University, The Netherlands
Johannes Scholz	Graz University of Technology, Austria
Raja Sengupta	McGill University, Canada
Monika Sester	Leibniz Universität Hannover, Germany
Shih-Lung Shaw	University of Tennessee, USA
Takeshi Shirabe	Royal Institute of Technology Stockholm, Sweden
Andre Skupin	San Diego State University, USA
Kathleen Stewart	University of Maryland, USA
Martin Swobodzinski	Portland State University, USA
Jean-Claude Thill	University of North Carolina Charlotte, USA
Sabine Timpf	University of Augsburg, Germany
Martin Tomko	The University of Melbourne, Australia
Ming-Hsiang Tsou	San Diego State University, USA
Monica Wachowicz	University of New Brunswick, Canada
John Wilson	University of Southern California, USA
Stephan Winter	University of Melbourne, Australia
Andreas Wytzisk	Bochum University of Applied Sciences, Germany
Ningchuan Xiao	The Ohio State University, USA
Bo Yan[e]	LinkedIn, USA
Phil Yang	George Mason University, USA
Xinyue Ye	Texas A&M University, USA
Eunhye Yoo	University at Buffalo, USA
May Yuan	University of Texas at Dallas, USA
Rui Zhu[e]	University of California, Santa Barbara, USA
Alexander Zipf	University of Heidelberg, Germany

[e]: (additionally as) emergency reviewer
[m]: (additionally as) meta reviewer

Using Georeferenced Twitter Data to Estimate Pedestrian Traffic in an Urban Road Network

Debjit Bhowmick[1]
Department of Infrastructure Engineering, The University of Melbourne, Australia
dbhowmick@student.unimelb.edu.au

Stephan Winter
Department of Infrastructure Engineering, The University of Melbourne, Australia
winter@unimelb.edu.au

Mark Stevenson
Melbourne School of Design, Department of Infrastructure Engineering, The University of Melbourne, Australia
mark.stevenson@unimelb.edu.au

Abstract

Since existing methods to estimate the pedestrian activity in an urban area are data-intensive, we ask the question whether just georeferenced Twitter data can be a viable proxy for inferring pedestrian activity. Walking is often the mode of the last leg reaching an activity location, from where, presumably, the tweets originate. This study analyses this question in three steps. First, we use correlation analysis to assess whether georeferenced Twitter data can be used as a viable proxy for inferring pedestrian activity. Then we adopt standard regression analysis to estimate pedestrian traffic at existing pedestrian sensor locations using georeferenced tweets alone. Thirdly, exploiting the results above, we estimate the hourly pedestrian traffic counts at every segment of the study area network for every hour of every day of the week. Results show a fair correlation between tweets and pedestrian counts, in contrast to counts of other modes of travelling. Thus, this method contributes a non-data-intensive approach for estimating pedestrian activity. Since Twitter is an omnipresent, publicly available data source, this study transcends the boundaries of geographic transferability and scalability, unlike its more traditional counterparts.

2012 ACM Subject Classification Information systems → Geographic information systems

Keywords and phrases Twitter, pedestrian traffic, location-based, regression analysis, correlation analysis

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.1

1 Introduction

1.1 Background and motivation

Urban traffic monitoring and analysis is increasingly important with the ever-growing urban population. Urban traffic consists of two major modes of travel, namely the vehicular mode (automobile, bicycle, transit) and the pedestrian mode [38]. Research on urban vehicular traffic has been popular for the past five or six decades owing to increased number of motorised vehicles on the road and the consequent challenges related to traffic congestion, competition for space and consumption of resources. On the contrary, the focus on pedestrians is more recent with the authorities now spending more resources on developing infrastructure for active mobility (non-motorised forms of transport such as biking and walking) to curb the detrimental impact of the motorised modes on public health and climate.

[1] Corresponding author

While walking is the most common and essential travel mode as almost all places are accessible on foot [39, 13], several studies have shown the benefits of walking with respect to a person's physical and mental health [3, 18, 14]. On a larger scale, walking generates indirect public health benefits by reducing the use of automobiles, consequently reducing traffic congestion, energy consumption, air and noise pollution, the overall level of traffic danger, and thus offerring more liveable communities [26, 37]. Besides public health, pedestrian activity is an important factor in urban planning, transportation management and decisions affecting land use and real estate. Analytical insights on pedestrian activity assists governing authorities to estimate demand with greater accuracy and allocate resources accordingly, which, in turn, improves operations.

Owing to the late rise in popularity of research relating to the pedestrian mode (being overlooked for decades), datasets representing pedestrian activity and movement are limited as compared to its motorised counterparts, more so in developing nations. As a result, most studies are still based on the traditional data collection methods of questionnaire surveys, manual counting, and tracking people's movements using GPS devices and smartphones [25, 2, 31, 9, 30, 5, 27]. While these methods are effective, they involve significant monetary costs and suffer from the issues of scalability and transferability in both space and time. Moreover, some of these forms of data collection such as GPS tracking, are highly privacy-sensitive [17]. On the other hand, traditional pedestrian volume estimation studies [22, 31, 9] are dependent on multiple, highly localised, predictor datasets which are not available in most places . Hence, there is a growing necessity for cheap, publicly available, omnipresent proxy data which transcends the boundaries of scalability and transferability. In this regard, location-based social media data, especially Twitter, has gained increasing attention by researchers who have used it to tackle a host of real-world problems including the detection and prediction of vehicular traffic levels [32], unusual levels of vehicular traffic congestion, accidents and disruptions [10, 8, 35], pedestrian congestion [34] and crowd movements at various spatio-temporal scales [7, 4].

1.2 Related work

There is limited systematic research comparing the nature of association of location-based social media data with varying urban travel modes to investigate the possibility that some modes are better represented by such data as compared to the others. While [32] have shown that Twitter and Instagram data can be used as a predictor for actual vehicular traffic by using Odds Ratio, Risk Ratio and RT-DBSCAN, the nature of association was drawn from comparing only the abnormalities of social media distribution and vehicular traffic volume. Since their work intends to identify traffic congestion in near-real time, their methods are limited to successfully differentiating between normal and anomalous traffic volumes. Although [7] and [4] have predicted crowd flow using Twitter, their study is also limited to outlier detection and anomalous behaviour resulting from events. On the other hand, [34] predicted pedestrian congestion using georeferenced tweet counts but did neither report any validation, nor any evidence of association between tweets and pedestrian congestion in the first place. In contrast to the aforementioned literature, this study investigates the possibility of using georeferenced tweets as a viable proxy for predicting pedestrian traffic and shows how tweet counts can be used for prediction of pedestrian traffic at high spatio-temporal resolution.

Other studies have used more traditional approaches of estimating pedestrian traffic at locations by quantifying the influence of multiple predictor variables. [22] presented a model to estimate the pedestrian volumes for street intersections. The study found that population

and job density, local transit access, and land use mix had the strongest explanatory power on variances of pedestrian volume. [31] used a log-linear regression model to identify statistically significant relationships between annual pedestrian volume at road intersections and predictor variables such as land use, transportation system, local environment and socio-economic characteristics surrounding the intersections. Similarly, [9] proposed a scalable approach by using regression models to predict pedestrian volume at road intersections using multiple infrastructural datasets and extrapolate at locations where count data was absent. All these studies are highly data-intensive and are dependent on the spatio-temporal granularity of the datasets representing the explanatory variables.

Among other techniques, the space syntax tool is stands out for being less data-intensive and relying only on street network data [11]. While this configurational approach has been used to predict pedestrian movement to an acceptable extent (roughly 60%) [21] in urban spaces, it is well-studied and has a well-developed methodology. But, existing literature has not employed social media data, or even investigated, in the first place, whether it can be used as a measure of estimating the number of pedestrians at a given location at a given time period. On the contrary, this study proposes a novel approach which is not only a scalable, but also transferable and non-data intensive by only using publicly available georeferenced Twitter data with fine granularity instead of employing multiple datasets.

1.3 Research hypothesis and objectives

This study aims to prove the hypothesis that georeferenced Twitter traffic can be used as a viable proxy for estimating pedestrian counts under specific conditions of space and time, with a certain degree of accuracy. The objectives of this paper are:

1. To show the existence of a strong positive correlation between georeferenced Twitter traffic and pedestrian traffic and that this correlation is stronger than vehicular traffic,
2. To develop a scalable and transferable method that predicts pedestrian traffic with reasonable accuracy, at any location in an urban network using only georeferenced tweet counts with high spatial and temporal granularity.

To attain the stated objectives, this study makes use of publicly available georeferenced Twitter data, pedestrian count data made available by the City of Melbourne, and vehicular traffic data from SCATS made available by the Victorian Government.

Overall, the contributions of this study are three-fold.

1. In the first step, this study implements correlation analysis to understand the association between georeferenced Twitter traffic and two major urban travel modes, pedestrian and vehicular traffic, by looking at the nature of correlations and the spatio-temporal patterns of variation of correlation. It compares the results across the two travel modes to understand whether one mode is more strongly associated with georeferenced tweets under certain conditions of space-time and hence tweets can be inferred as a viable proxy for that mode.
2. Based on the findings from the first step, which reveals the existence of a relatively stronger and more statistically significant correlation between Twitter traffic and pedestrian traffic as compared to vehicular traffic, the second step of this study uses standard regression analysis to estimate pedestrian traffic and the resultant estimation errors at existing pedestrian sensor locations, at any given hour of any day of the week. The method is geographically transferable and is able to estimate pedestrian traffic at the finest level of spatial granularity.
3. In the final step, this study predicts pedestrian traffic at every segment of the study area network (even where pedestrian counts are not available) at hourly intervals of any given date using georeferenced tweet counts.

2 Data overview

2.1 Pedestrian counts dataset

The City of Melbourne had developed an automated pedestrian counting system in 2009 to better understand pedestrian activity within the municipality. Using non-vision based sensors installed at multiple strategic locations in its administrative area (covering the Central Business District and its neighbouring suburbs), it collects counts of pedestrians on an hourly basis. The open dataset contains hourly pedestrian counts since 2009 and is updated on a monthly basis. The dataset is structured with the following information:

1. Sensor ID
2. Sensor location (street name)
3. Coordinates of sensor location (latitude, longitude)
4. Hourly pedestrian count
5. Detailed timestamp (date, day of the week, hour of the day)

During the period of data collection for this study (January to April 2018) there were 49 active sensor locations in the city. Figure 1 shows the locations of the pedestrian sensors (blue markers).

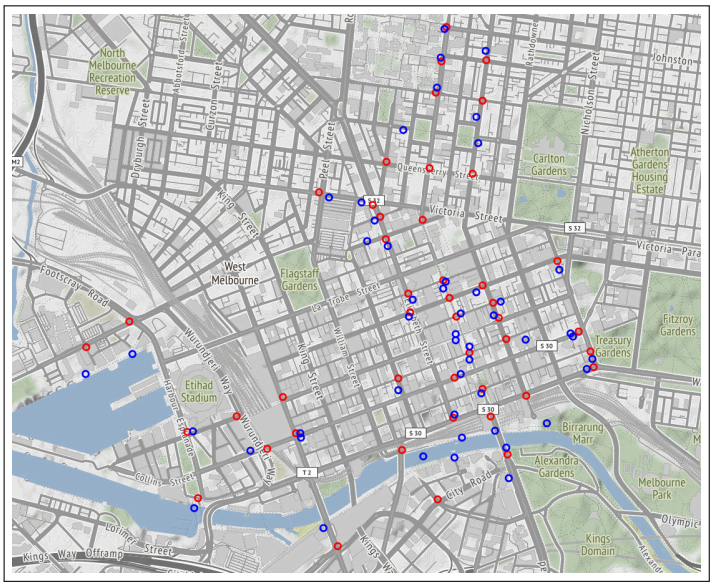


Figure 1 Locations of pedestrian sensors (blue) and SCATS sensors (red) in the City of Melbourne.

2.2 Vehicular counts dataset

The official source of the vehicle count data for the cities in Australia is based on the Sydney Coordinated Adaptive Traffic System (SCATS, `www.scats.com.au`). Recordings are made at intervals of 15 minutes and are available for download from the VicRoads website (`www.vicroads.gov.au`). Hourly vehicular traffic for each of the 45 sensor locations in terms of counts of vehicles were collected for the period of March 2018. Figure 1 shows the locations of the pedestrian sensors (red markers).

2.3 Twitter dataset

The Australian Urban Research Infrastructure Network (AURIN, `www.aurin.org.au`) has harvested tweets originating from all major cities of Australia from July 2014 to May 2018 using Twitter's Public Streaming API which allows for a real-time collection of a random sample of tweets. The collection of tweets was not continuous as the dataset contains significant time gaps (April to July 2015, March 2016, May 2016 to May 2017). Similar to previous studies conducted in the domain of spatial information using Twitter data, this research will rely only on precisely georeferenced tweets (tweets with explicit latitude/longitude information). In 2019, Twitter has turned off the option of precise georeferencing. Since then georeferenced tweets provide location information usually in terms of places of varying granularity. These coarse georeferences are also user selected, and hence there is no guarantee whether they were posted in that place at all. For this study, however, we rely on the precise georeferences which are system-generated and hence reliable. [24] compared the data from the Streaming API and the Firehose data set (the complete set of tweets available commercially) and stated that the 1% sample provided by the Streaming API almost returns the complete set of precisely georeferenced tweets despite the sampling. The challenge with the small number of precisely georeferenced tweets is that they only represent a set of self-selecting individuals supplying volunteered data. Thus they are highly unlikely to be representative of the entire pedestrian population of the study area. Regardless of this major caveat, it is accepted best practice in the literature to base investigations on this selective data. For this study of predicting pedestrian traffic, even if the tweets are non-representative, the base assumption of a correlation between persons' tweeting and their participation at activities still holds.

2.4 Study settings

The area chosen for this study is the City of Melbourne, one of the 32 local councils making up Greater Melbourne. City of Melbourne covers an area of roughly 37 km^2 and consists of metropolitan Melbourne's innermost suburbs, including the central business district. The study area includes an area specified by a bounding box, judiciously chosen to cover all sensor locations of the city's pedestrian counting system with adequate buffer zones. The coordinates of the bounding box are (-37.8359, 144.9269, -37.7860, 144.9903).

Since the Twitter dataset contains periods of gaps, this study makes use of the most recent and continuous time span for which the data was collected, from January 2018 to April 2018. Using the bounding box coordinates specified while defining the study area, 28197 precisely georeferenced tweets (a subset of the 10 million tweets extracted by AURIN in Greater Melbourne during the study period) were obtained for the study period. As this study aims to compare and investigate the nature of associations between pedestrian counts (which are relatively large numbers in the study area) and tweet counts (which are comparatively small numbers), any relationship between the two is highly sensitive. To avoid any further bias, we filtered out any consecutive tweets of the same user in our chosen time intervals, hence, counting Twitter users rather than tweets in each 1-hour period. Furthermore, tweeting activity was observed to be anomalous during 1st January and hence, it was removed from the dataset. This resulted in the final dataset of 25679 precisely georeferenced tweets.

3 Correlation analysis between tweet counts and pedestrian counts

The underlying assumption of this experiment is that highly populated outdoor spaces have a greater probability of experiencing high tweet counts, and in turn, high georeferenced tweet counts, than places with lower populations. This is in line with existing literature where social media has been used as proxy measure for urban land use [6], urban activity spaces [19, 23], ambient population [12], and most importantly pedestrian population [34, 7, 4]. While details of these studies [34, 7, 4] have been discussed in Section 1, all of them have assumed the latent existence of an association between tweets and pedestrians. This study bases its hypothesis on such findings and ventures deeper to investigate whether georeferenced Twitter data can actually be used as a viable proxy for inferring pedestrian traffic in an urban area. Since pedestrian count data is the most accurate representation of pedestrian traffic at a given location, this study aims to infer to what extent these counts are correlated with the count of georeferenced tweets. Additionally, this study makes use of vehicular traffic data obtained from SCATS locations to draw comparisons between pedestrian and vehicular travel mode in terms of the strength of correlation. It compares the results of the analyses across the two travel modes to understand whether georeferenced tweets can be inferred as a viable proxy for urban pedestrian traffic.

Using the location information of the active pedestrian sensors, an imaginary circular buffer (catchment) area was drawn with a sensor at the centre of a circle. Similar spatial querying was conducted before by [22] and [15, 16] in their studies of predicting pedestrian volume across intersections in San Francisco and New York City respectively. This spatial querying was performed to capture the number of precisely georeferenced tweets. The radius of the circle was varied from 100 to 1500 metres in steps of 100 metres. Using point-in-polygon analysis, each tweet was assigned to the pedestrian counter(s) in whose catchment area it fell. Correlation was drawn between the observed pedestrian count in each sensor and tweet count inside the catchment area of the same sensor, both at hourly and daily time intervals. For the vehicular traffic data, the SCATS location nearest to each pedestrian sensor was considered as the centre of a catchment circle and correlation was computed for the month of March 2018 only. A sample illustration of the aforementioned method has been shown in Figure 2.

The results of the correlation analysis between georeferenced tweet counts and pedestrian counts are shown in Figure 3. It can be observed that there exists a clear hourly pattern in the variation of correlation coefficient, while the daily pattern is less prominent. The resultant magnitude of correlation coefficients reduces drastically in between 5 AM to 10 AM, while remaining relatively high and statistically significant for the rest of the day. This could be attributed to the fact that Twitter traffic starts increasing more rapidly after 7 AM and reaches its peak quicker than pedestrian traffic, as observed from Twitter data of Melbourne (shown in Figure 4) and Australia [20]. Another possible cause could be that streets that are busy during those times do not cater to a lot of Twitter traffic (pedestrians not tweeting on busy streets before starting work). This indicates that pedestrian activities that are more tweet-productive, are found to be happening before 5 AM and after 10 AM. As far as days of week are concerned, weekends exhibit a slightly improved positive correlation coefficient value than weekdays. This maybe again attributed to the fact that weekend activities (which are more interesting) are more tweet-productive than weekday activities and that tweet counts are more reflective of actual pedestrian counts at places.

Also, the correlation coefficient varies significantly with the radius of the catchment area. It can be observed in Figure 3 that the correlation coefficient gradually increases with the increase in the radius of the circular catchment area, reaching its peak at 500 metres.

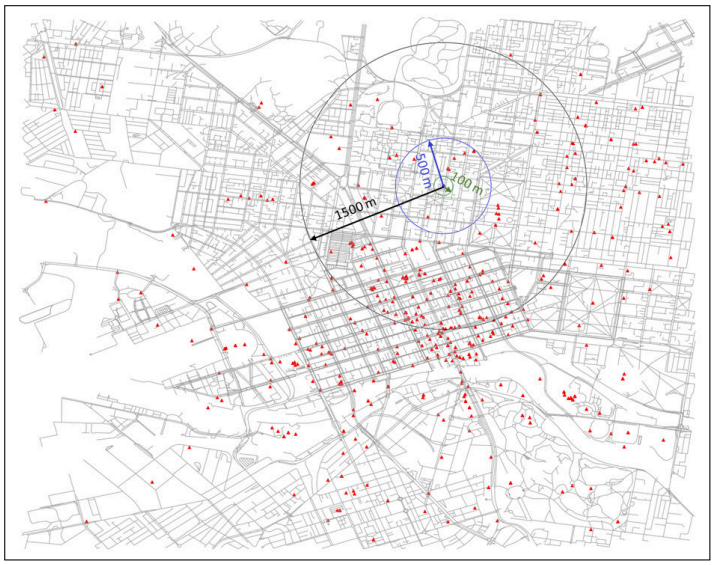
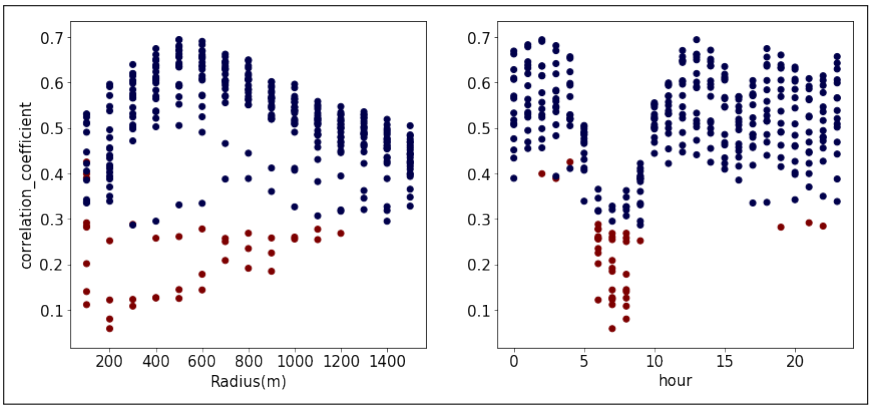


Figure 2 Spatial querying for georeferenced tweets made between 10 AM and 11 AM inside the study area (red triangular markers) using a circular buffer with radius 100, 500 and 1500 metres for a randomly chosen sensor (blue round marker).

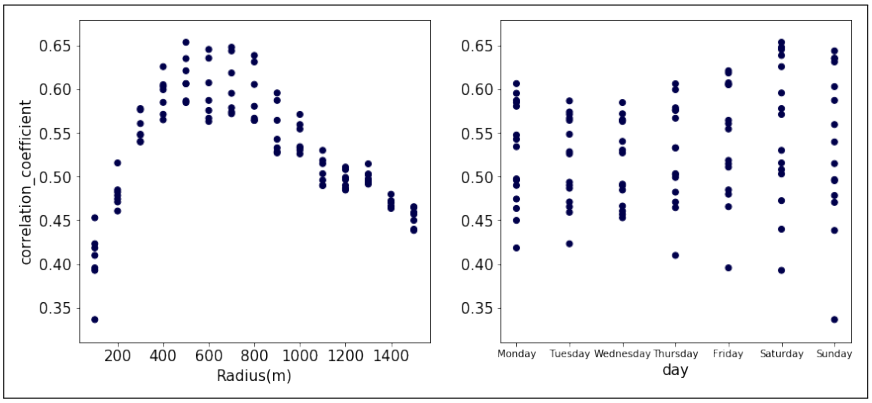
Interestingly, about 500 metres is the average walking range [1, 29, 33, 36, 28]. It then starts to reduce: Larger catchment areas lead to overlaps of catchment areas, and of tweets counted multiple times, thereby reducing the effectiveness of our experiment. The least correlation was observed at 1500 metres radius.

On the other hand, the association between vehicular traffic and georeferenced Twitter traffic appears to be substantially weaker. As shown in Figure 5, the magnitude of the resultant correlation coefficients is relatively less as compared to the ones obtained from pedestrian counts. Also, most of the coefficients are not statistically significant at 95% confidence level. This comparison throws up anticipated and intuitive results. It is more likely that tweets are made during a pedestrian activity as compared to a driving activity as walking is often the final mode of reaching an activity location, from where, presumably, the tweets originate. The results reaffirm this likelihood. Although there exists no binding definition of what a pedestrian activity exactly means (or is limited to), the general consensus is that it usually spans more than just the period of walking itself. On the contrary, vehicular activity is more limited and is confined to only driving a vehicle or being a passenger in one.

While this experiment was aimed more at inferring correlation as opposed to causation (which cannot be proven even with a statistically significant correlation coefficient), it adds the aspect of novelty to this study by establishing the correlation, its magnitude and temporal patterns, before proceeding to use tweets to measure pedestrian activity. Also, explanations can be speculated by observing fair correlations which helps in hypothesis generation. Hence, based on these findings, this study now argues with conviction that georeferenced tweet counts may be used as a viable proxy for estimating pedestrian count under given conditions. The following section aims at calculating errors arising while estimating pedestrian traffic from georeferenced tweet counts.



(a) With radius of catchment circle and hour of the day



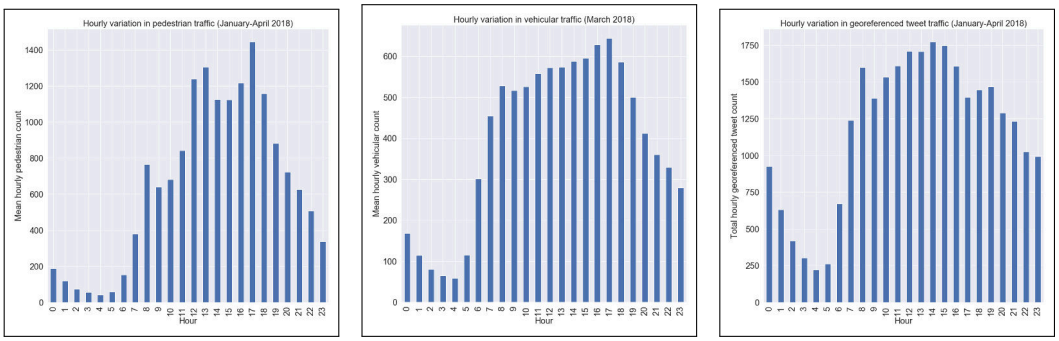
(b) With radius of catchment circle and day of the week

Figure 3 Variation of correlation coefficient (georeferenced tweet counts and pedestrian counts); dark blue points indicate the correlation is statistically significant at 95% confidence level.

4 Estimating pedestrian counts at existing sensor locations

Based on the findings from the first stage of this study, the second stage proposes to use standard regression analysis to estimate pedestrian traffic using tweet counts. It aims to investigate the one-to-one relationship between georeferenced tweet counts and pedestrian counts. It describes, in detail, the methodology of handling the dataset and computing the resultant errors obtained during estimation of pedestrian counts via regression in terms of common regression metrics such as Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). The only predictor variable used in regression is precisely georeferenced tweet count.

Using georeferenced tweet counts as the predictor variable (on the x-axis), an attempt was made to replicate its relationship with pedestrian counts, which is the predicted variable (on the y-axis). For this purpose, standard regression modelling was employed. For each hour of each day of the week (e.g., Thursday 1500 hours), a unique regression curve was developed, thus resulting in a total of 168 curves (24 hours multiplied by 7 days). Since this study is the first to investigate such one-to-one relationship between georeferenced tweet counts and pedestrian counts, it was made sure that the chosen regression model is comparatively better (in terms of standard regression metrics) and logical (non-overfitting) at the same time. Hence, trial regression analyses were performed using linear, log-linear, quadratic



(a) Hourly variation in pedestrian count

(b) Hourly variation in vehicular count

(c) Hourly variation in georeferenced tweet count

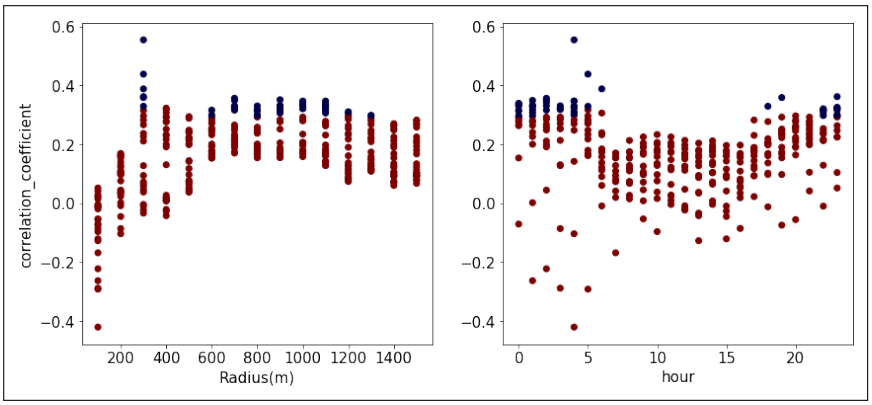
Figure 4 Variation in the City of Melbourne: (a) hourly pedestrian count per sensor, (b) hourly vehicular count per sensor and (b) aggregate hourly tweet count.

and cubic models. Previous studies related to pedestrian count prediction have used either linear or log-linear models to test statistical relationship between walkability measures and pedestrian volume [31, 15, 16]. Although these models have lesser accuracy, these models do not completely contradict this global behaviour, and hence either could be accepted as a viable representation. Quadratic and cubic models exhibited lesser errors but were prone to overfitting, and hence were not considered. The results of the regression analyses are shown in Table 1. Mean hourly values of the regression metrics (R-squared, MAE and MAPE) are obtained by taking arithmetic means of metric values over 168 cases (every hour of every day of the week). The temporal variation of the regression metrics have been shown in Figure 6.

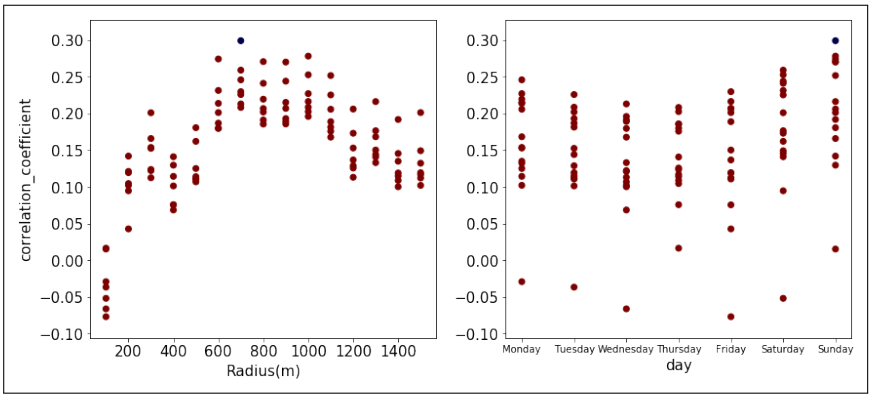
Table 1 Regression metrics for January-April 2018.

Regression model	**x-axis**	**y-axis**	**Mean hourly R-squared**	**Mean hourly MAE**	**Mean hourly MAPE**
Linear	Georeferenced tweet count	Pedestrian count	0.426	192.6	28.3
Log-linear	$\log_e$(Georeferenced tweet count)	Pedestrian count	0.424	212.9	26.9

The performance of the *mean hourly tweet count - mean hourly pedestrian count* regression curves obtained from this study were tested by applying the method to predict mean pedestrian counts of a different time period. For this purpose, data from November 2017 was chosen as it was the nearest month available from our study period devoid of known anomalies. They exhibit slightly greater estimation errors (linear: MAE = 274.3, MAPE = 35.83 and log-linear: MAE = 292.0, MAPE = 34.43), which could be due to seasonal variations in the *tweet count - pedestrian count* relationship that was not taken into account due to absence of continuous tweet collection periods. Nevertheless, the errors and temporal patterns are similar to the ones shown in Table 1 and Figure 6 respectively. Hence, the regression curves obtained in this study are acceptable.



(a) With radius of catchment circle and hour of the day



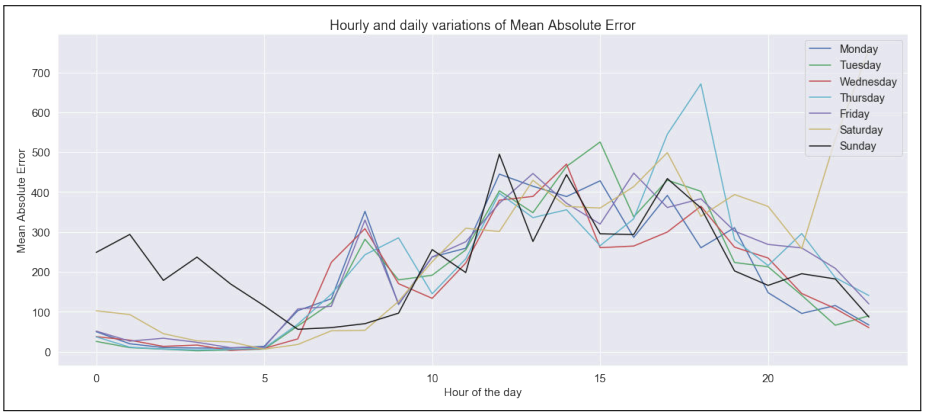
(b) With radius of catchment circle and day of the week

Figure 5 Variation of correlation coefficient (georeferenced tweet counts and vehicular counts); dark blue points indicate the correlation is statistically significant at 95% confidence level.

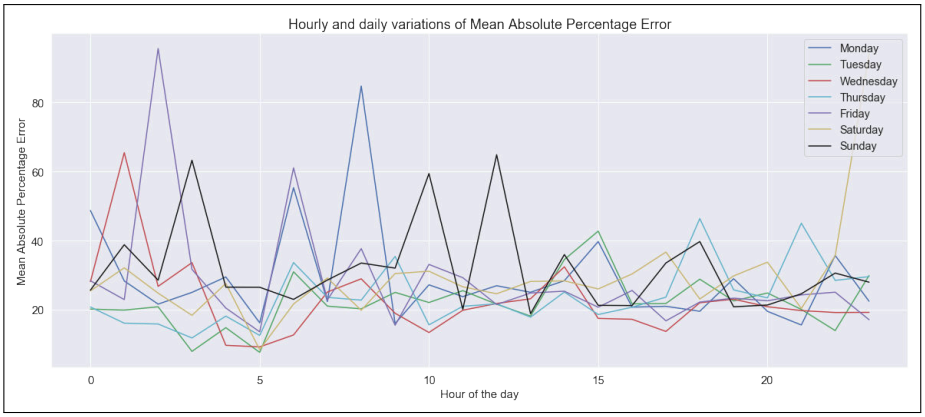
5 Predicting pedestrian counts at locations without any pedestrian count information

The third and final stage of this study is aimed at extrapolation of the developed methodology. The intention to develop a transferable (temporally and spatially) and scalable pedestrian count prediction methodology was based on the motive to predict pedestrian counts at high temporal (hour of the day of the week) and spatial resolution (point on the urban road network), even at locations without any pedestrian count information. The following method helps in predicting the pedestrian counts at any point in an urban pedestrian road network, given the date and time (at hourly granularity).

Pedestrian network data was obtained from OpenStreetMap using the bounding box coordinates specified in Section 2.4. For a given hour of any given date, georeferenced tweets were extracted from the AURIN dataset. Consequently, iterating over all edges of the network, centered on the mid point of an edge spatial querying was conducted using a 500 metre search radius to extract the number of georeferenced tweets. These tweet counts were associated with the corresponding edge. After obtaining the information on the queried hour of the day and day of the week , iterating over all edges of the network, the corresponding regression curve was referred to estimate the pedestrian count passing through an edge. A sample illustration of the spatial querying process to estimate pedestrian counts using georeferenced tweet counts for January 2, 2018 during 1000 to 1100 hours and the resultant estimation of pedestrian counts have been shown in Figure 7.



(a) Variation of MAE using linear regression with hour of the day and day of the week



(b) Variation of MAPE using linear regression with hour of the day and day of the week

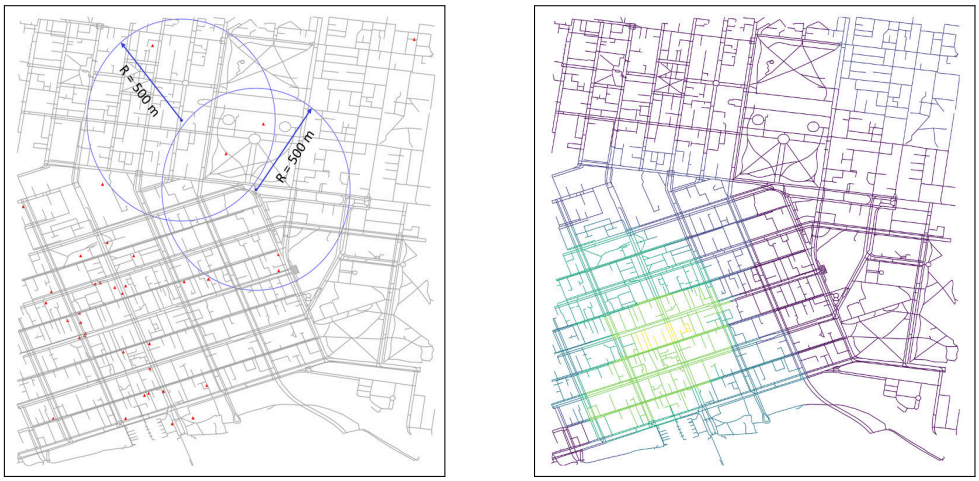
Figure 6 Temporal patterns of regression metrics.

6 Discussion

The three stages of analysis reported in this study makes novel contributions by finding moderate to high correlations between georeferenced tweet counts and pedestrian counts, and then developing a scalable, transferable and non-data-intensive methodology for estimating pedestrian counts from georeferenced tweet counts. Finally, using spatial querying, the study predicted pedestrian counts at high temporal and spatial resolution at locations devoid of sensors. Yet there are limitations of this study that need to be highlighted as well.

First, it was observed in Section 3 that the values of the resultant correlation coefficients during 5 AM to 10 AM were relatively lower in magnitude and statistically insignificant. Hence, predictions using regression curves during this time period are expected to be more erroneous. This is apparent from Figure 6 where the MAE and MAPE can be observed to reach high values (higher than the mean) during this time period, in most of the days. It can be argued that this time period is not ideal for making pedestrian volume prediction using georeferenced tweet counts alone.

Results in Section 4 showed that the proposed methodology produces significant estimation errors in both the study dataset as well as in the testing dataset. While this study argues about the benefits of employing a non-data-intensive approach to predict pedestrian counts in Section 1.2, the magnitude of errors indicate the drawbacks. The regression curve generalises



(a) Spatial querying for tweets (red triangular markers) using a circular buffer with radius 500 metres for 2 randomly chosen edge mid points (blue round markers)

(b) Edges of the network labelled as per predicted pedestrian counts. Lighter colours (yellow, green) indicate higher pedestrian volume, while darker colours (blue, purple) indicate lower pedestrian volume

Figure 7 Prediction of pedestrian counts using precisely georeferenced tweet counts by extrapolation during 1000 to 1100 hours on January 2, 2018.

all network segments with zero georeferenced tweet count as having one and the same pedestrian volume equal to the intercept of the regression curve, which overestimates the actual pedestrian volume in most cases. Furthermore, the betweenness centrality of the edges of the network was calculated to tally the results with the pedestrian count predictions. The dead ends, for example, have zero betweenness centrality but, it can be observed from Figure 7 that the proposed method is not differentiating between through roads and dead ends, in terms of pedestrian counts. These challenges will be addressed in a future study by analysing other network attributes (data-intensive) such as road width, road hierarchy, proximity to transit stops, number of Points-of-Interest which are proven factors known to influence pedestrian demand, to produce more representative results.

The underlying assumption of this study was that the entire study area is homogeneous in terms of every spatial attribute, apart from the spatial distribution of georeferenced tweets. Thus it assumed a existing one-to-one relationship between georeferenced tweet counts and pedestrian counts that varies temporally, but not spatially. The argument in favour of this assumption is that this study was conducted in a relatively homogeneous area in terms of land use. But the results indicate that this assumption is not robust, and there are multiple possible reasons. The relationship between georeferenced tweet counts and pedestrian counts is not independent of spatial variables. Different locations have different tweet count - pedestrian count relationships depending on location type. For example, the tweet productiveness of a railway station is possibly different from the tweet productiveness of an event location, both in magnitude and in temporal patterns. While both may experience pedestrian counts to the same degree, it is expected that an event will bring out greater number of georeferenced tweets than the lesser interesting public transit, for the same number of pedestrian counts. Hence, future work will address this shortcoming by incorporating the spatial variation of the relationship between tweet productivity and land use.

Finally, it must be noted that Twitter has removed the support for precise georeferencing of tweets since June 2019. Thus, the proposed method is applicable only on historic datasets. In terms of estimating pedestrian counts, this move impacts on any real-time interests, but long-term averages should not change quickly. To mitigate this, time-series modelling using historic tweet counts can be applied to predict future tweet counts, which can be used for predicting pedestrian counts using the same principle in future scenarios.

7 Conclusion and future work

Despite the highlighted limitations, this study contributes novel insights. The three-step methodology remains transferable due to use of an omnipresent data source. It can be applied if acceptable correlations are achieved. However, the regression equations in our case study will not hold true for another study area and need to be re-calculated for a different study area. Also, the size of the study area can be increased or decreased without any change in the methodology, with intuitive variations in estimation accuracy. Yet, analysis at micro-level and homogeneous land-use will need some strict assumptions. Also, a study area that is too small will have fewer geotagged tweets. Lastly, we attempted to mitigate population disparity between indoor spaces and adjoining outdoor spaces. We made justifiable assumptions that populated indoor spaces indicate popularity in its adjoining outdoor space, given a space-time buffer. Thus, we placed a 1-hour time buffer and a 500m radius distance buffer around pedestrian counters to capture tweets. We assume to catch most of the tweets and pedestrians in the same buffer although some exceptions will always be there. This uncertainty flattens out as the study area grows in size. Not only does this study investigate the nature of existing correlation, but also proposes an approach to estimate pedestrian counts from georeferenced tweet counts, even at places devoid of pedestrian sensors. By doing so, this study shows the extent to which this non-data-intensive approach can predict pedestrian counts (in terms of estimation errors) and thus brings out the limitations of such an approach, which need to be addressed in future to achieve more accurate and representative results.

References

1 V Thamizh Arasan, VR Rengaraju, and KV Krishna Rao. Characteristics of trips by foot and bicycle modes in Indian city. *Journal of Transportation Engineering*, 120(2):283–294, 1994.

2 Hieronymus C Borst, Sanne I de Vries, Jamie MA Graham, Jef EF van Dongen, Ingrid Bakker, and Henk ME Miedema. Influence of environmental street characteristics on walking route choice of elderly people. *Journal of Environmental Psychology*, 29(4):477–484, 2009.

3 JE Donnelly, DJ Jacobsen, K Snyder Heelan, R Seip, and S Smith. The effects of 18 months of intermittent vs continuous exercise on aerobic capacity, body weight and composition, and metabolic fitness in previously sedentary, moderately obese females. *International Journal of Obesity*, 24(5):566, 2000.

4 Ana Fernández Vilas, Rebeca P Díaz Redondo, and Mohamed Ben Khalifa. Analysis of crowds' movement using Twitter. *Computational Intelligence*, 35(2):448–472, 2019.

5 Sheila Ferrer and Tomás Ruiz. The impact of the built environment on the decision to walk for short trips: Evidence from two Spanish cities. *Transport Policy*, 67:111–120, 2018.

6 Vanessa Frias-Martinez and Enrique Frias-Martinez. Spectral clustering for sensing urban land use using Twitter activity. *Engineering Applications of Artificial Intelligence*, 35:237–245, 2014.

7 Gary Goh, Jing Yu Koh, and Yue Zhang. Twitter-informed crowd flow prediction. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 624–631. IEEE, 2018.

8 Yikai Gong, Fengmin Deng, and Richard O Sinnott. Identification of (near) Real-time Traffic Congestion in the Cities of Australia through Twitter. In *Proceedings of the ACM First International Workshop on Understanding the City with Urban Informatics*, pages 7–12. ACM, 2015.

9 Amir Hajrasouliha and Li Yin. The impact of street network connectivity on pedestrian volume. *Urban Studies*, 52(13):2483–2497, 2015.

10 Jingrui He, Wei Shen, Phani Divakaruni, Laura Wynter, and Rick Lawrence. Improving traffic prediction with tweet semantics. In *23rd International Joint Conference on Artificial Intelligence*, pages 1387–1393, 2013.

11 Bill Hillier, Alan Penn, Julienne Hanson, Tadeusz Grajewski, and Jianming Xu. Natural Movement: or, Configuration and Attraction in Urban Pedestrian Movement. *Environment and Planning B: planning and design*, 20(1):29–66, 1993.

12 John R Hipp, Christopher Bates, Moshe Lichman, and Padhraic Smyth. Using Social Media to Measure Temporal Ambient Population: Does it Help Explain Local Crime Rates? *Justice Quarterly*, 36(4):718–748, 2019.

13 Jinhyun Hong and Cynthia Chen. The role of the built environment on perceived safety from crime and walking: Examining direct and indirect impacts. *Transportation*, 41(6):1171–1185, 2014.

14 Marcus Johansson, Terry Hartig, and Henk Staats. Psychological benefits of walking: Moderation by company and outdoor environment. *Applied Psychology: Health and Well-Being*, 3(3):261–280, 2011. doi:10.1111/j.1758-0854.2011.01051.x.

15 Yuan Lai and Constantine Kontokosta. Analyzing the Drivers of Pedestrian Activity at High Spatial Resolution. In *2017 International Conference on Sustainable Infrastructure: Methodology, ICSI 2017*, pages 303–314. American Society of Civil Engineers (ASCE), 2017.

16 Yuan Lai and Constantine E Kontokosta. Quantifying place: Analyzing the drivers of pedestrian activity in dense urban environments. *Landscape and Urban Planning*, 180:166–178, 2018.

17 J. K. Laurila, Daniel Gatica-Perez, I. Aad, Blom J., Olivier Bornet, Trinh-Minh-Tri Do, O. Dousse, J. Eberle, and M. Miettinen. The Mobile Data Challenge: Big Data for Mobile Computing Research. *Infoscience : EPFL Scientific Publications*, 2012.

18 I-Min Lee and David M Buchner. The importance of walking to public health. *Medicine & Science in Sports and Exercise*, 40(7 Suppl):S512–8, 2008.

19 Jae Hyun Lee, Adam W Davis, Seo Youn Yoon, and Konstadinos G Goulias. Activity Space Estimation with Longitudinal Observations of Social Media Data. *Transportation*, 43(6):955–977, 2016.

20 Kevan Lee. The Biggest Social Media Science Study: What 4.8 Million Tweets Say About the Best Time to Tweet, 2016. URL: https://buffer.com/resources/best-time-to-tweet-research.

21 Yoav Lerman, Yodan Rofè, and Itzhak Omer. Using Space Syntax to Model Pedestrian Movement in Urban Transportation Planning. *Geographical Analysis*, 46(4):392–410, 2014.

22 XiaoHang Liu and Julia Griswold. Pedestrian volume modeling: A case study of San Francisco. *Yearbook of the Association of Pacific Coast Geographers*, pages 164–181, 2009.

23 Nick Malleson and Mark Birkin. New insights into individual activity spaces using crowd-sourced big data. In *2014 ASE BigData/SocialCom/CyberSecurity Conference, Stanford University*. Academy of Science and Engineering (ASE), USA, 2014.

24 Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen M Carley. Is the sample good enough? Comparing data from Twitter's streaming api with Twitter's Firehose. In *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

25 J Michael Oakes, Ann Forsyth, and Kathryn H Schmitz. The effects of neighborhood density and street connectivity on walking behavior: The twin cities walking study. *Epidemiologic Perspectives & Innovations*, 4(1):16, 2007.

26 John Pucher and Ralph Buehler. Walking and cycling for healthy cities. *Built Environment*, 36(4):391–414, 2010.

27 LSC Pun-Cheng and CWY So. A comparative analysis of perceived and actual walking behaviour in varying land use and time. *Journal of Location Based Services*, pages 1–20, 2019.

28 TM Rahul and Ashish Verma. A study of acceptable trip distances using walking and cycling in Bangalore. *Journal of Transport Geography*, 38:106–113, 2014.

29 S. Robertson. *Usability of pedestrian crossings: Further results from fieldwork Contemporary Ergonomics 2005*. Taylor & Francis, 2005.

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.

31 Robert J Schneider, Todd Henry, Meghan F Mitman, Laura Stonehill, and Jesse Koehler. Development and Application of Volume Model for Pedestrian Intersections in San Francisco, California. *Transportation Research Record*, 2299(1):65–78, 2012.

32 Richard O Sinnott, Yikai Gong, Shiping Chen, and Paul Rimba. Urban traffic analysis using social media data on the cloud. In *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, pages 134–141. IEEE, 2018.

33 State of Victoria. Pedestrian access strategy : A strategy to increase walking for transport in Victoria. Technical report, State of Victoria, 2010.

34 Shoko Wakamiya, Yukiko Kawai, Hiroshi Kawasaki, Ryong Lee, Kazutoshi Sumiya, and Toyokazu Akiyama. Crowd-sourced prediction of pedestrian congestion for bike navigation systems. In *5th ACM SIGSPATIAL International Workshop on GeoStreaming*, pages 25–32. Association for Computing Machinery, Inc, 2014.

35 S. Wongcharoen and T. Senivongse. Twitter analysis of road traffic congestion severity estimation. In *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, page 6 pp. IEEE, 2016.

36 Yong Yang and Ana V Diez-Roux. Walking distance by trip purpose and population subgroups. *American Journal of Preventive Medicine*, 43(1):11–19, 2012.

37 Xuan Zhang and Lan Mu. The perceived importance and objective measurement of walkability in the built environment rating. *Environment and Planning B: Urban Analytics and City Science*, page 2399808319832305, 2019.

38 Yinan Zheng, Lily Elefteriadou, Thomas Chase, Bastian Schroeder, and Virginia Sisiopiku. Pedestrian Traffic Operations in Urban Networks. *Transportation Research Procedia*, 15:137–149, 2016.

39 Dennis Zielstra and Hartwig Hochmair. Using free and proprietary data to compare shortest-path lengths for effective pedestrian routing in street networks. *Transportation Research Record: Journal of the Transportation Research Board*, 2299(1):41–47, 2012.

Estimation of Moran's *I* in the Context of Uncertain Mobile Sensor Measurements

Dominik Bucher[1]
Institute of Cartography and Geoinformation, ETH Zurich, Switzerland
dobucher@ethz.ch

Henry Martin[1]
Institute of Cartography and Geoinformation, ETH Zurich, Switzerland
martinhe@ethz.ch

David Jonietz
HERE Technologies Switzerland, Zurich, Switzerland
david.jonietz@here.com

Martin Raubal
Institute of Cartography and Geoinformation, ETH Zurich, Switzerland
mraubal@ethz.ch

René Westerholt[1]
School of Spatial Planning, TU Dortmund University, Germany
rene.westerholt@tu-dortmund.de

Abstract

Measures of spatial autocorrelation like Moran's I do not take into account information about the reliability of observations. In a context of mobile sensors, however, this is an important aspect to consider. Mobile sensors record data asynchronously and capture different contexts, which leads to considerable heterogeneity. In this paper we propose two different ways to integrate the reliability of observations with Moran's I. These proposals are tested in the light of two case studies, one based on real temperatures and movement data and the other using synthetic data. The results show that the way reliability information is incorporated into the Moran's I estimates has a strong impact on how the measure responds to volatile available information. It is shown that absolute reliability information is much less powerful in addressing the problem of differing contexts than relative concepts that give more weight to more reliable observations, regardless of the general degree of uncertainty. The results presented are seen as an important stimulus for the discourse on spatial autocorrelation measures in the light of uncertainties.

2012 ACM Subject Classification Information systems → Geographic information systems; Information systems → Sensor networks; Mathematics of computing → Statistical paradigms; Applied computing → Earth and atmospheric sciences

Keywords and phrases mobile sensors, Moran's I, uncertainty, probabilistic forecasting

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.2

Funding This research was supported by the Swiss Data Science Center (SDSC) and by the Swiss Innovation Agency Innosuisse within the Swiss Competence Center for Energy Research (SCCER) Mobility.

1 Introduction

Recent technological advances accompanied by price reductions of sensor hardware have propelled the emergence of mobile sensor networks. Mobile sensor data is widely collected using smartphones [27, 56], sensor-equipped cars and public transport vehicles [31, 28], boats [2], animals [52], or semi-stationary objects like buoys [33]. Such mobile sensors make

[1] These authors contributed equally to this work.

it possible to increase the spatial coverage of data collection while deploying relatively few additional devices compared to static sensor networks [15, 44]. Mobile sensors hence allow monitoring of our social and physical environments at a so far unprecedented scale.

Increasing the spatial coverage of sensor networks using mobile instead of static devices can entail a loss of homogeneity in the collected data. Mobile measurements are often recorded at different locations, at different points in time, and in an asynchronous mode. They thus capture differing contextual conditions [38]. Mobile sensor data therefore may represent different processes of dynamic geographic phenomena. For instance, measuring air temperature at different times of the day may result in collecting samples representative of different processes such as urban heating at midday or cooling at night due to atmospheric radiation losses into outer space. Such processes may behave very differently in varying geographic regions despite involving the same phenomenon. Respective sensed data may therefore be characterised by differing mean levels, dispersal mechanisms, and spatial structures.

The outlined variations can distort the interpretation of measures and statistics obtained from mobile sensor data. One example for this is the assessment of spatial autocorrelation, which can be described as the quantification of spatial interaction, or as the "coincidence of value similarity with locational similarity" [4, p. 241]. Computing the popular Moran's I index [43], for instance, establishes a relation between geographically close observations. The statistic is thus highly sensitive to asynchronously sensed value pairs attached with uncertainty, in particular when no or little prior knowledge is available about the underlying processes. Novel ways are thus needed to incorporate this kind of uncertainty attached to sensor measurements in the estimation of spatial measures.

This paper puts forward two approaches for estimating Moran's I using uncertain mobile measurements. Both approaches presented make use of weights reflecting the certainty attached to pairs of sensor measurements. The certainty measures used are calculated through a non-parametric probabilistic forecast of the measured values, with the underlying model being constantly refitted from incoming sensor data. The certainty factors obtained this way are then included in Moran's I through two different kinds of matrices of pair-wise terms, which rescale the spatial weights used in the statistic. The advantages of using empirical forecasts to quantify uncertainty are that the temporal correlation does not have to be modeled explicitly, that it allows treating problems that do not fall into a geostatistical category (where we could model the spatio-temporal dependencies explicitly) and that it naturally captures both uncertainties arising from the measured phenomenon itself as well as from the sensors in use. We evaluate our concepts by applying them to two case studies: One study contains data from sensor-equipped cars measuring air temperature in Switzerland over a three day period, whereas the second one is based on controlled, synthetic data. The latter study is used to investigate the capacity of our introduced certainty matrices to also outweigh non-stationarity, that is, a temporally varying spatial process.

2 Related Work

2.1 Spatial Non-Stationarity

One characteristic causing uncertainty is spatial non-stationarity. This may be reflected in variation in the mean, the variance, or higher-order moments. Ord & Getis have recently put forward a measure called Local Spatial Heteroscedasticity (LOSH) [48, 72, 22]. It quantifies spatially inhomogeneous variation allowing to disclose spatial boundaries separating regimes and to characterise the internal stability of clusters [1]. Westerholt *et al.* [68] have modified LOSH towards an entirely local test for identifying the role of spatial structure in local variance characterisations. Varying mean levels are commonly investigated using residuals above

trend surfaces [7, 25], defining the mean as a function of the coordinates levelling out spatial trends. Some kinds of data lead to non-stationarity, for instance, through uncontrolled data acquisition procedures. One example for this is georeferenced social media data. Such data are prone to uncertainty because people contribute in different ways simultaneously, including varying cognitive (e.g., [54, 66]), demographic (e.g., [65, 57]), idiosyncratically subjective (e.g., [12, 29]), and other factors pertaining to spatial perception and communication. Recent works have investigated the impact of this uncontrolled uncertainty on the estimation of spatial structure, and initial proposals were made to address related issues [69, 70, 67].

2.2 Spatiotemporal Autocorrelation

Uncertainty can enter estimations temporally, for example, when phenomena are not stable over time. The notion of spatial autocorrelation is a way to address this issue. One way to achieve this is to incorporate explicitly temporal notions of autocorrelation in the calculation of spatial measures. First discussed by [11] and [41], various approaches to measure spatiotemporal autocorrelation have been proposed, such as [37], who incorporate temporal trends through time-lagged correlation measures into the calculation of Moran's I. Another approach is to estimate Moran's I using spatiotemporal weight matrices, with exemplary studies including [16], who focus explicitly on how to build such matrices; [30], who, focusing on the related concept of geographically weighted regression (GWR), construct weight matrices from spatiotemporal (x, y, t)-coordinates; and [35], who, based on the assumption that spatiotemporal effects can be calculated as a product of spatial and temporal effects, integrate the according weights in a combined matrix, and compute both global and local spatiotemporal Moran's I. A slightly different approach is taken by [53], whose approach eliminates certain time effects by temporally detrending spatially referenced time series.

2.3 Investigation of Rates

Rate variables are commonly attached with varying uncertainty levels. This is caused by varying underlying populations like populations at risk or varying numbers of people counted in aggregation units [63, 64]. Rates have a higher propensity of being extreme when the underlying reference quantity is small [5]. In order to correct for these distortions, several approaches have been proposed including empirical Bayes correction [17, 40, 5, 32], omission of local population sizes by re-basing rates on the overall population size [46], and weighting deviations of residual rates by the inverse of the size of the local population at risk [62]. Methodically, our approach proposed below is closest to the adjustment proposed by Waldhör [62], but we focus on a different kind of uncertainty in this paper.

3 Methodology

3.1 Assumptions

Our work presented below is based on certain assumptions concerning our uncertainty assessment and the spatial method Moran's I that we modify. Let o_{il} and o_{jm} be elements of a set of observations O of a spatial phenomenon Q taken at geographic locations i and j, and at different points in time t_l and t_m, respectively. The following assumptions are assumed to hold true for the remainder:

- Observations O obtained from mobile sensors provide an incomplete representation of the phenomenon Q studied.
- A higher spatial coverage of observations O of Q can lead to an improved representation of Q, even if taken at different points in time.

- The certainty $u_{o_{il},o_{jm}}$ shared between two observations depends on the forecast horizon Δt_{lm} comprising a certain number of preceding observations. Predictions of the nearer future are considered more certain than distant ones.
- Phenomenon Q is assumed to show relatively stable spatial second-order characteristics over the time points observed. This facilitates meaningful interpretation of Moran's I.
- Although Q is geostatistical in the case study example, our proposed solution is free of model assumptions to ensure transferability to social science domains such as social media analysis or georeferenced surveys [6, 55].

3.2 Spatial Autocorrelation and Moran's *I*

Tobler's *first law of geography* states that "everything is related to everything else, but near things are more related than distant things" [60, p. 234]. This characteristic can be utilised for spatial interpolation, to detect pockets of non-stationarity, or to characterise spatial heterogeneity [19]. Spatial autocorrelation operationalises this empirical law [42] to quantify spatial associations [21] disclosing spatially clustered (positive), dispersed (negative), or random behaviour (close to zero autocorrelation) [21]. A number of global and local measures of spatial autocorrelation are available, including Moran's I [43, 10, 3], Geary's c [18, 3], Rogerson's R [50], and Getis and Ord's G hotspot statistics [47, 23].

Moran's I is often preferred over other measures because of its superior statistical power properties and its robustness against unfavourable configurations of spatial units, that is, outliers in the spatial weights matrix [9, 21]. Let x_i be measured values with arithmetic mean $\bar{x}$. Moran's I and its feasible range are then given as

$$I = \frac{n}{\sum\limits_{i,j\neq i}^{n} w_{ij}} \cdot \frac{\sum\limits_{i,j\neq i}^{n} w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum\limits_{i}^{n}(x_i - \bar{x})^2}, \quad I \in \left[\frac{n}{\sum\limits_{i,j\neq i}^{n} w_{ij}} \cdot \lambda_{\min}, \quad \frac{n}{\sum\limits_{i,j\neq i}^{n} w_{ij}} \cdot \lambda_{\max} \right]. \tag{1}$$

Matrix $\boldsymbol{W}$ holds spatial weights w_{ij}. These establish pairwise connections between the n spatial units based on their inverse distance, spatial contiguity, or other characteristics [20]. The measure strongly depends on the spatial weights structure chosen [13, 59]. Therefore, the range of I depends on the smallest and largest eigenvalues λ_{min} and λ_{max} of the centred symmetric part of the spatial weights matrix given as $(\boldsymbol{I} - \mathbf{1}\mathbf{1}^T/n)((1/2)(\boldsymbol{W} + \boldsymbol{W}^T))(\boldsymbol{I} - \mathbf{1}\mathbf{1}^T/n)$. Thereby, $\boldsymbol{I}$ is the $n \times n$ identity matrix and $\mathbf{1}$ denotes the $n \times 1$ all-ones vector. Values of global Moran's I below its expected value $E[I] = -1/(n-1)$ indicate negative spatial autocorrelation. Values for I larger than $E[I]$ hint on the opposite case [43, 10].

We argue that the effects of observations o_{il} and o_{jm} made at different points in time of a temporally non-static phenomenon Q should be explicitly considered in the calculation of the global Moran's I. Our approach consists of extending traditional Moran's I with measures of pair-wise certainty $u_{o_{il},o_{jm}}$ (abbreviated to $u_{il,jm}$ hereafter), which represent the influence of past time intervals on the reliability of measurements. The measures of certainty that we use are equivalent to projecting values observed at time t_l to more recent points in time t_m, and hence to their hypothetical re-measurement. We propose two different ways to include pair-wise certainty measures in Moran's I. Let Δt_{lm} denote a temporal forecast horizon. The indicators proposed are then given as

$$I^1_{\Delta t_{lm}} = \frac{n}{\sum\limits_{i,j\neq i}^{n} w_{il,jm}u_{il,jm}} \cdot \frac{\sum\limits_{i,j\neq i}^{n} w_{il,jm}u_{il,jm}(o_{il} - \bar{o})(o_{jm} - \bar{o})}{\sum\limits_{i}^{n}(o_i - \bar{o})^2}, \tag{2}$$

$$I^2_{\Delta t_{lm}} = \frac{n}{\sum\limits_{i,j\neq i}^{n} w_{il,jm}(1 + u_{il,jm} - \bar{u})} \cdot \frac{\sum\limits_{i,j\neq i}^{n} w_{il,jm}(1 + u_{il,jm} - \bar{u})(o_{il} - \bar{o})(o_{jm} - \bar{o})}{\sum\limits_{i}^{n}(o_i - \bar{o})^2}. \quad (3)$$

In the indicator proposed in Equation 2 the spatial weights are rescaled proportional to the joint certainties shared by neighboured locations. In practice, most weights will be affected, but some weights more than others depending on their joint certainty. The terms $u_{il,jm}$ range in the interval $[0, 1]$. Indicator $I^1_{\Delta t_{lm}}$ equals standard Moran's I only when no uncertainty is present. In all other cases, $I^1_{\Delta t_{lm}}$ shall be interpreted in the light of its feasible range given in Equation 1 but with the eigenvalues of $\boldsymbol{W}$ substituted by those of the Hadamard product $\boldsymbol{W} \circ \boldsymbol{U}$ and with the normalising factor replaced with $n(\sum_{i,j\neq i}^{n} w_{ij}u_{il,jm})^{-1}$.

The second indicator defined in Equation 3 presumes that uncertainty is acceptable as long as it is distributed evenly across the map. Whenever the joint certainty of two observations is above average, their relative importance in the spatial analysis increases. Analogously, when the mutual certainty of a pair of observations is below average, their joint spatial weight is penalised. The terms $1 + (u_{il,jm} - \bar{u})$, with $\bar{u}$ being the mean certainty estimate, range in the interval $[0, 2]$. $I^2_{\Delta t_{lm}}$ equates to Moran's I when either all certainties are close to their own average, or when there is at least a balance between above and below average certainties in the map. Like with $I^1_{\Delta t_{lm}}$, we shall consider the respective eigenvalue spectrum determining the feasible range of $I^2_{\Delta t_{lm}}$ to assess the impact of the uncertainty modelling proposed on the range of Moran's I values.

3.3 Uncertainty Estimation using Empirical Prediction Intervals

We make use of the quantiles of empirical prediction intervals. Such intervals are a form of probabilistic forecasting, expressing predictions of the future in the form of probability distributions over all possible outcomes [24]. Empirical prediction intervals thus allow to assign a degree of certainty (or uncertainty) to each of those potential events [34]. The method is based on the historical forecast errors of an existing deterministic forecast [36, 71]. Empirical prediction intervals cannot be conditioned on known variables like model or ensemble-based probabilistic forecasts. They are, however, straightforward and do not require a priori assumptions about the distribution of random variables or the distribution of forecast errors [36].

An empirical prediction interval can be constructed as follows [36]: Given observations $O = \{O_t : t \in \mathbb{T}\}$ of a random process, with $\mathbb{T}$ being an interval of $\mathbb{R}$ describing a set of time stamps [26], O_{t_i} is an observation at time t_i. We say that all observations O with $t < t_i$ are in the past of t_i and all observations O with $t > t_i$ are in the future of t_i. Now with $t_n = t_i + h$, let

$$\hat{O}_{t_n,h} = f(O_{t \leq t_i}) \quad (4)$$

be the h-step deterministic forecast of O_{t_n} created at time $t_i = t_n - h$ using a function f utilising all observations that are in the past of or at time t_i. Thus,

$$e_{t_n,h} = O_{t_n} - \hat{O}_{t_n,h} \quad (5)$$

gives the forecast error for observation O_{t_n} with forecast horizon h. For k available forecast errors $e_{t,h}$ with forecast horizon h we define the forecast horizon specific empirical cumulative distribution function as

$$\hat{F}_h(e) = k^{-1} \sum_{t=1}^{k} \mathbb{I}(e_{t,h} \leq e) \tag{6}$$

with e indicating a fixed threshold of some still acceptable error, and $\mathbb{I}(S)$ referring to the indicator function of some set S [36]. This distribution allows to draw conclusions on the uncertainty of the model. The deterministic forecast can then be enhanced by "dressing" the error distribution around it. In order to smooth the empirical cumulative distribution function, a kernel density estimation can be used.

Quantifying the pairwise joint certainty of the projection of past sensor observations to the present time would require deriving the joint probability distribution of the prediction of the two random variables involved. As we can not assume independence, this is not simply the product of their individual probabilities. The derivation of joint CDFs of two dependent variables can be achieved by assuming specific distributions or by using copula models [45]. We try to avoid both the complexity of copula models and the need to make rigid assumptions. Instead, we use a method developed in [39] and [51] to estimate the (sharp) lower and upper bounds $b_l(e)$ and $b_u(e)$ for the probability that the sum of two dependent random variables exceeds a certain threshold e. Let $X+Y$ be the sum of the forecast errors from two locations. We are looking for bounds such that

$$b_l(e) \leq P(X + Y \leq e) \leq b_u(e). \tag{7}$$

The bounds $b_l(e)$ and $b_u(e)$ define the possible range of the probability that the sum of the error of two variables does not exceed a specified threshold. To be sure not to overestimate this probability, we are interested in the the lower bound of the possible range of $P(X + Y \leq e)$. This can be calculated by the equation given in [14]:

$$b_l(e) = \sup_{x \in \mathbb{R}} \max\{F_1^-(x) + F_2^-(e - x) - 1, 0\}, \tag{8}$$

whereby $X+Y$ is to be substituted for x. Equation 8 determines the lower stochastic bound b_l that represents the lowest probability with which the sum of two dependent random variables exceeds a specified value e. For the calculation of a certainty measure, we calculate the distributions of the absolute forecast errors of the o_{il}, o_{jm}. X_{in} and X_{jn} are then distributions of absolute forecast errors for the projections of observations o_{il}, o_{jm} to a later point in time t_n. These distributions allow to estimate the joint uncertainty of both projected observations by calculating the lowest probability that the sum of the absolute forecast errors is below a specific threshold e:

$$u_{o_{il \to n}, o_{jm \to n}} = b_l(e) \leq P(\mathbf{X} + \mathbf{Y} \leq e). \tag{9}$$

Finally, $u_{o_{il \to n}, o_{jm \to n}}$ is the lowest probability that the sum of the absolute errors is below the threshold e when projecting o_{il}, o_{jm} to t_n. As described in Section 3.2 these terms are used to rescale the spatial weights attached to pairs of sensor measurements o_{il}, o_{jm} when calculating the extended versions of Moran's I presented in Equations 2 and 3 at time $t_I = t_n$. It is important to note that we take the lowest possible probability that the error is in an acceptable range $P(X + Y \leq e)$ in order not to underestimate the joint uncertainty of two measurements.

3.4 Case Studies

We apply our proposed solutions to two case studies. The first one is based on real temperature and mobility data, which we combined to engineer a dataset that could realistically have been generated by mobile temperature sensors on cars, yet for which we know the ground truth of the phenomenon (i.e., we know the temperature at every location and time in the study area). The temperatures are obtained from the COSMO Regional Reanalysis Project[2] [61]. They were measured hourly in the years 2007–2013 at 2 metres above the ground and are available in a 0.018° cell grid, which in Central Europe corresponds to a spatial resolution of about 2×2 km. We use these grid cells as discrete locations. For the mobility data, we use car trajectories obtained from customers of a Mobility-as-a-Service offer operated by the Swiss Federal Railways[3]. We have thus cropped the temperatures to a subset of 320×150 cells covering Switzerland. Also, because the trajectories were recorded in 2016, we have reset their timestamps to early July 2013 to match the temperatures available. We further restricted the GPS points of the trajectories to one point per cell maximum in order to harmonise the different spatial resolutions. Figure 1a illustrates the temperature data and Figure 1b shows the number of samples in each grid cell in one month.

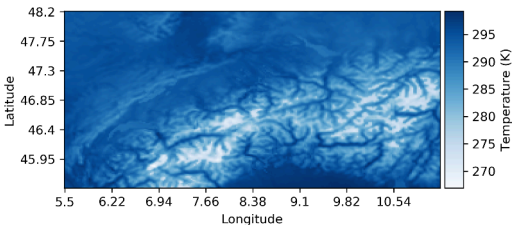
The second case study uses controlled synthetic data and introduces non-stationarity by varying the scale of the generative spatial process, allowing us to study the potential of probabilistic models to infer spatial autocorrelation of non-stationary phenomena (e.g., student location check-ins). We generate $i = 1, \ldots, 60$ grids (representing 60 time intervals) of 60×60 cells each (representing 3 600 spatial locations). The grids are populated using Simple Kriging based on a Gaussian spatiotemporal variogram [8] with sill $s = 1$, nugget $n = 0$, and a time-dependent range r_i:

$$\gamma(h, r_i) = (s - n)\left(1 - \exp\left(-\frac{h^2}{\frac{1}{3}r_i^2}\right)\right) + n. \tag{10}$$

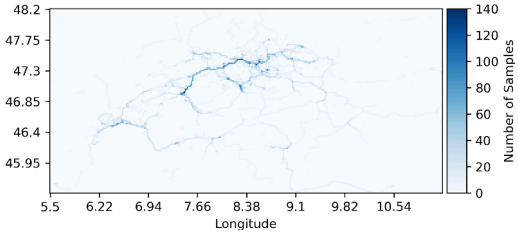
The range parameter is calculated using a sinusoidal function to simulate periodicity in the level of autocorrelation as $r_i = 0.5 + |10 \cdot \sin(2i/2\pi)|$, which is our way to impute non-stationarity. The temporal correlatedness is modelled analogously to Equation 10 using a temporal range of 1 (i.e., $r_i = 1$, $\forall i$ in the case of temporal correlatedness) and both the spatial and temporal correlatedness are weighted with 50% each. The simulation of values for individual cells is based on the approach outlined in [49, p. 27]: following a random sequence through the grid, the conditional distribution (based on previously simulated values) is calculated for each visited cell, and a new value is drawn from this distribution. In our case, this distribution is always assumed to be Gaussian (as this represents a wide range of naturally occurring phenomena and is a well-studied distribution), and the mean and variance are taken from the Kriging interpolation estimate and error. Once all cells in a grid i have been assigned a value, the process is repeated for grid $i + 1$. To simulate sensor measurements, each grid is finally sampled at 25 random locations.

Both case studies exemplify two different forms of sensory data: The first one is arguably the most well-known, where sensors sample temporally and spatially dependent phenomena at single points in space and time. The second one could be seen as sampling the aggregated movements of entities who periodically gather (e.g., students who go to campus during the day and use a location-based service to "check-in" at certain locations). Within the context

[2] This dataset can be retrieved from http://reanalysis.meteo.uni-bonn.de.
[3] www.sbb-greenclass.ch



(a) The temperature (in Kelvin) at 09:00 on the 2nd of July 2013.



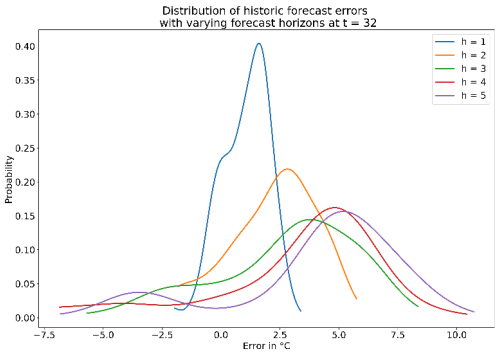
(b) The number of temperature measurements in July for each grid cell.

Figure 1 The temperature dataset used within this study. One can easily spot mountainous regions, where the temperatures (in Kelvin) are lower. We sampled this dataset along various real trajectories, leading to the measurements depicted in the right figure. Most samples can be found along major traffic axes as well as in bigger cities such as Zurich, Bern or Lausanne.

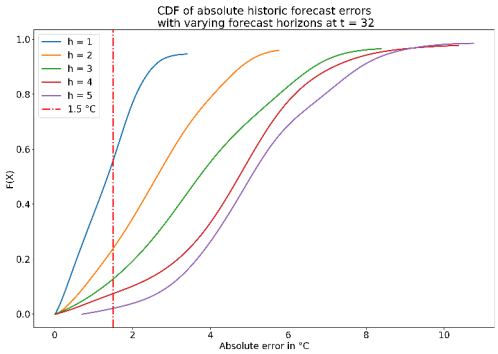
of this work, those are the sensor types of primary interest: they sample a phenomenon with unknown temporal dependency at different points in space and time. For both case studies, pairwise uncertainty values $u_{il,jm}$ are needed. As described in Section 3.3, we construct empirical prediction intervals from errors available at previous hours. We apply persistence prediction, assuming a temperature value observed like o_{il} to not have changed during Δt_{ln}, so it would still be the same at that respective location i. In order to log our forecast errors, whenever an observation is made at time t_n in a certain raster cell i, we check if an earlier persistence forecast is available for the respective time horizon (e.g., 2 hours into the future). If this is not the case, a deterministic persistence forecast is made for this location and the next 24 hours. Instead, if a forecast for this location and time horizon is available, we can calculate the forecast error from the absolute difference of the forecasted (persistence) and the actually measured value. Figures 2a and 2b illustrate errors and their distributions for one point in time of the temperature dataset. The spatial weights matrix is constructed from k-nearest-neighbour relations, whereby we use $k = 5$ (in combination with a 30-cell maximum distance in case of the temperature case study) as threshold (primarily to reduce the computational complexity), and a weighting function of $1/r$ (where r is the Euclidean distance). Increasing k does not substantially change the outcomes of the case studies, while decreasing it towards zero leads to non-interpretable results. As most of the associated weights thus are zero, we use sparse matrix representations for all computations. We use $e = 3°\text{C}$ as a threshold for the tolerable error in the first case study and $e = 0.5$ in the second case study for the calculation for the certainty values.

4 Results

As displayed in Figure 3, our results for the temperature case study show an improvement in Moran's I estimation when using the approach proposed in Equation 3 (shown in green) compared to both the baseline, which only uses values sampled within the same hour (leading to gaps when an insufficient number of samples is available), as well as to simply ignoring different time intervals and confidence values (denoted by Moran's I and shown in red in Figure 3; this essentially considers all samples recorded during previous hours as if they were recorded during the hour under investigation). The estimated values are consistently higher than those using plain spatial weights, and thus closer to the ground truth calculated from the measured temperatures (i.e., the non-sampled data shown in Figure 1a). The results are particularly promising for time windows in the early morning hours, which follow



(a) Marginal distributions of the forecast errors for forecast horizons $1 \leq h \leq 5$ hours.



(b) Cumulative distribution function of the absolute forecast errors at different horizons. The red dotted line marks an exemplary threshold value $e = 1.5$°C.

Figure 2 Uncertainty and cumulative distribution functions of errors at different horizons for the temperature dataset at $t = 32$.

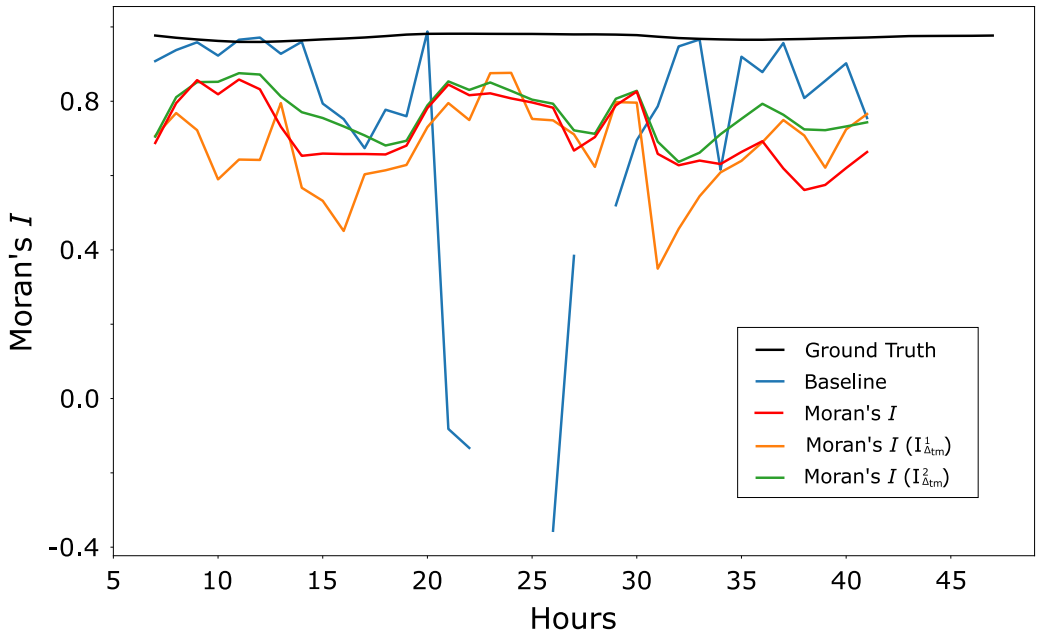


Figure 3 Different versions of Moran's I calculated for 45 hours of the temperature case study. The ground truth indicator is calculated from the measured temperatures. The baseline approach is based on forecasts using data from the respective previous hour only but without taking account of time differences or certainty values. The gaps in the plot for the baseline are caused by data gaps in the night time where no trajectories are available.

periods without data availability. The latter occurs at night, when no drivers use cars from the fleet and so there are no trajectories available. A look at the way Equation 3 contains certainty information shows that the method is not susceptible to large increases or decreases in the amount of available information, since it is based on reliability relative to the mean confidence level. This relative notion of including certainty values means that the most reliable observations are relied upon more than others, even if the overall average certainty of the information available decreases. Similarly, the proposed method improves on the baseline which heavily relies on a large number of samples and thus fails to provide an accurate estimate during the night and in the morning hours.

The other method proposed in Equation 2 (shown in orange) also leads to an improvement compared to the baseline, but not compared to the exclusive use of spatial weights. The Moran's I estimates shown in Figure 3 show that this method leads to a greater systematic underestimation of actual spatial associations in the data. More importantly, this method of incorporating certainty information is less stable and more volatile than the alternative presented in Equation 3. The obtained Moran's I values fluctuate more and show a more erratic behaviour. One reason for this is that the method is much more prone to missing information and simple prediction methods. The time windows after the nights described above are much more affected by the lack of available information, which is reflected in a sudden drop in Moran's I values. The reason for this is the immediacy of the method. Absolute rather than relative certainty information is used, and therefore a general decline in the overall confidence in the available information has a direct effect on the Moran's I estimates. This is a major limitation of the approach presented in Equation 2.

The above paragraphs describe the behaviour of the proposed approaches when a spatial pattern is present. Figure 4 shows the empirical distributions of Moran's I generated by Monte Carlo repetitions under spatial randomisations. For this purpose, the temperatures were randomised within their respective time periods and then Moran's I was repeatedly calculated ($n = 1000$). The graph of z-score standardised values contains not only the empirical distributions in the null hypothesis, but also the z-score standardized eigenvalues of the underlying matrices (i.e., either $\boldsymbol{W}$ or $\boldsymbol{W} \circ \boldsymbol{U}$). The latter eigenvalues give an indication of the shape of the distribution of Moran's I values [58]. What we see is that the distribution using Equation 3 has a slight right skew, which is indicated by the clustering of eigenvalues on the left margin of the distribution. This may complicate the determination of p-values and the interpretation of Moran's I. The method from Equation 2 behaves more similar to the usual spatial weights matrix, which is an advantage of this method.

The results obtained for the case study of synthetic observations indicate that both of our approaches proposed in this paper are not suitable for dealing with non-stationarity (Figure 5). Recall the temporal periodicity present in the level of autocorrelation in this case study, implying that observations are not necessarily related over time. Therefore, disregarding

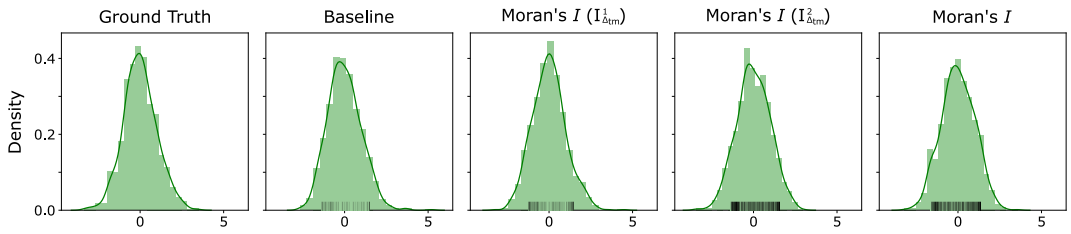


Figure 4 Histogram and density estimates of the null distributions for the different Moran's I values calculated for the temperature case study. The black bars at the bottom of each plot indicate the locations of the eigenvalues of each of the corresponding matrices used to calculate the respective measures.

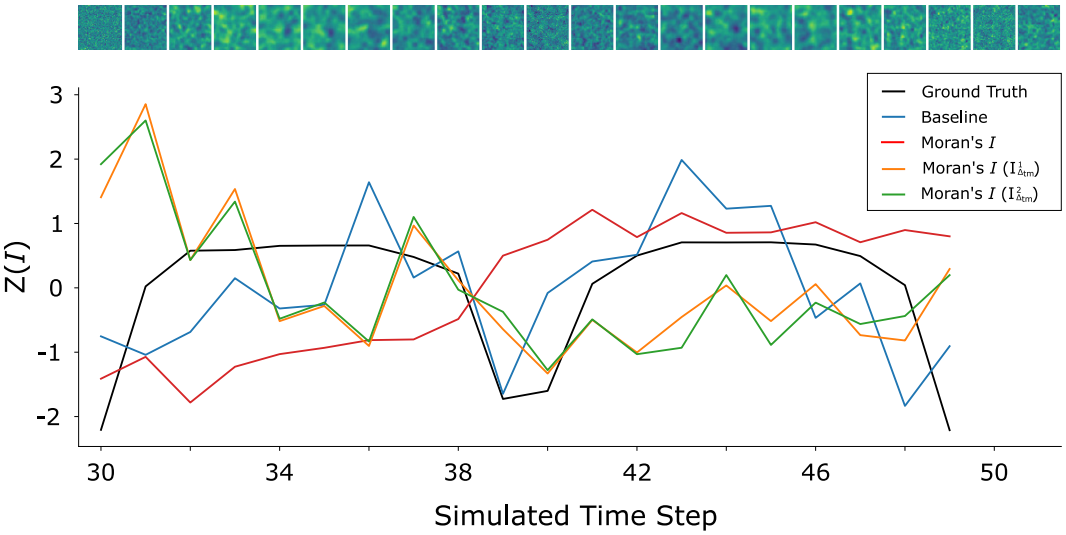


Figure 5 Different versions of Moran's I calculated for two temporal periodic cycles of the simulated data case study. All Moran's I values are given in standardised form to facilitate the readability of the figure. The bar at the top of the plot shows the simulated data.

certainty information of potential forecasts and simply using the baseline approach of only taking into account samples taken during the same hour (resp. time interval) has led to the best results for this case study, though also these are far from optimal (shown in blue in Figure 5). This finding demonstrates the importance of complying with the assumptions of Moran's I. Otherwise, the non-stationarity may lead to the disclosure of spurious patterns, which, in turn, may then lead to drawing wrong conclusions about geographic phenomena.

5 Conclusions

We put forward two ways of incorporating certainty information about sensor observations in the estimation of Moran's I. One of these approaches (Equation 2) uses an absolute notion of incorporating raw certainty scores. The alternative approach (Equation 3) proposed is based on the certainty of observations relative to others, that is, to the mean level of confidence in forecasts. These approaches were applied to two case studies. One study uses real-world temperatures and depicts one spatial process. The other one is based on synthetic values and simulates a succession of temporally varying spatial processes, which is realised by alternating the scale of the spatial patterns.

The results obtained show that using the best information available (relatively speaking) and weighting them accordingly performs better than using only good information in an absolute sense. The respective approach put forward in this paper (Equation 3) has, in comparison to ignoring time and reliability, led to a reduction of the systematic underestimation of Moran's I. The other approach presented here (Equation 2) is volatile and depends strongly on a sufficient amount of trustworthy data being available. These results are informative for the wider scholarly discussion on how to incorporate uncertainty in spatial measures like Moran's I. For future research, we recommend using certainty measures that work in a relative manner by giving more weight to those observations which are above-average reliable. In practice, researchers may use more sophisticated forecasting mechanisms, which may lead to further improvements like pushing Moran's I closer to the ground truth reference. Another

important result of this study is that it was shown that non-stationarity is a source of uncertainty that cannot be addressed by the approaches presented (or similar ones). This type of uncertainty needs to be addressed differently and corresponding attempts should be targeted in future research. Similarly, while the two presented case studies represent commonly found phenomena, evaluating the methods on a wider range of sensor measurements and synthetic data is required to further understand the impact of uncertainties arising due to different spatial and temporal distributions and individual (inaccurate or faulty) sensors.

References

1 J Aldstadt, M Widener, and N Crago. Detecting irregular clusters in big spatial data. In N. Xiao, M.-P. Kwan, M. F. Goodchild, and S. Shekhar, editors, *Proceedings of the 7th International Conference on Geographic Information Science (GIScience 2012)*, Columbus, OH, 2012.

2 Lilia Angelova, Puck Flikweert, Panagiotis Karydakis, Daniël Kersbergen, Roos Teeuwen, Kotryna Valečkaitė, Edward Verbree, Martijn Meijers, and Stefan van der Spek. Using a dynamic sensor network to obtain spatiotemporal data in an urban environment. In Peter Kiefer, Haosheng Huang, Nico Van de Weghe, and Martin Raubal, editors, *Adjunct Proceedings of the 14th International Conference on Location Based Services*, pages 13–18, Zürich, Switzerland, 2018. ETH Zurich.

3 Luc Anselin. Local indicators of spatial association—LISA. *Geographical Analysis*, 27(2):93–115, 1995.

4 Luc Anselin and Anil K Bera. Spatial dependence in linear regression models with an introduction to spatial econometrics. In Aman Ullah and David E. A. Giles, editors, *Handbook of Applied Economic Statistics*, pages 237–290. Marcel Dekker AG, New York, NY, 1998.

5 Renato M Assuncao and Edna A Reis. A new proposal to adjust Moran's I for population density. *Statistics in Medicine*, 18(16):2147–2162, 1999.

6 Matthias Bluemke, Bernd Resch, Clemens Lechner, René Westerholt, and Jan-Philipp Kolb. Integrating geographic information into survey research: Current applications, challenges and future avenues. *Survey Research Methods*, 11(3):307–327, 2017.

7 Jean-Pierre Bocquet-Appel and Robert R Sokal. Spatial autocorrelation analysis of trend residuals in biological data. *Systematic Zoology*, 38(4):333–341, 1989.

8 Jean-Paul Chiles and Pierre Delfiner. *Geostatistics: Modeling spatial uncertainty*. John Wiley & Sons, New York, NY, 2009.

9 Yongwan Chun and Daniel A Griffith. *Spatial statistics and geostatistics: Theory and applications for geographic information science and technology*. Sage, London, UK, 2013.

10 AD Cliff and JK Ord. The problem of spatial autocorrelation. *London Papers in Regional Science*, pages 25–55, 1969.

11 AD Cliff and John K Ord. Space-time modelling with an application to regional forecasting. *Transactions of the Institute of British Geographers*, pages 119–128, 1975.

12 Jens S Dangschat. Raumkonzept zwischen struktureller Produktion und individueller Konstruktion. *Ethnologie und Raum*, 9(1):24–44, 2007.

13 P De Jong, C Sprenger, and F Van Veen. On extreme values of Moran's I and Geary's c. *Geographical Analysis*, 16(1):17–24, 1984.

14 Michel Denuit, Christian Genest, and Étienne Marceau. Stochastic bounds on sums of dependent risks. *Insurance: Mathematics and Economics*, 25(1):85–104, 1999.

15 Mario Di Francesco, Sajal K Das, and Giuseppe Anastasi. Data collection in wireless sensor networks with mobile elements: A survey. *ACM Transactions on Sensor Networks*, 8(1):7, 2011.

16 Jean Dubé and Diègo Legros. A spatio-temporal measure of spatial dependence: An example using real estate data. *Papers in Regional Science*, 92(1):19–30, 2013.

17 Bradley Efron and Carl Morris. Stein's estimation rule and its competitors – an empirical Bayes approach. *Journal of the American Statistical Association*, 68(341):117–130, 1973.

18 Robert C Geary. The contiguity ratio and statistical mapping. *The Incorporated Statistician*, 5(3):115–146, 1954.

19 Arthur Getis. Reflections on spatial autocorrelation. *Regional Science and Urban Economics*, 37(4):491–496, 2007.

20 Arthur Getis. Spatial weights matrices. *Geographical Analysis*, 41(4):404–410, 2009.

21 Arthur Getis. Spatial autocorrelation. In Manfred M. Fischer and Arthur Getis, editors, *Handbook of Applied Spatial Analysis*, pages 255–278. Springer, Heidelberg, Germany, 2010.

22 Arthur Getis. Analytically derived neighborhoods in a rapidly growing west african city: The case of Accra, Ghana. *Habitat International*, 45:126–134, 2015.

23 Arthur Getis and J. K. Ord. The analysis of spatial association by use of distance statistics. *Geographical Analysis*, 24(3):189–206, 1992.

24 Tilmann Gneiting. Probabilistic forecasting. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, pages 319–321, 2008.

25 Daniel A Griffith. A linear regression solution to the spatial autocorrelation problem. *Journal of Geographical Systems*, 2(2):141–156, 2000.

26 Bruce Hajek. *Random processes for engineers*. Cambridge University Press, Cambridge, UK, 2015.

27 David Hasenfratz, Olga Saukh, Silvan Sturzenegger, and Lothar Thiele. Participatory air pollution monitoring using smartphones. In *Proceedings of the 2nd International Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data*, Cambridge, MA, 2012. Academic Press.

28 David Hasenfratz, Olga Saukh, Christoph Walser, Christoph Hueglin, Martin Fierz, Tabita Arn, Jan Beutel, and Lothar Thiele. Deriving high-resolution urban air pollution maps using mobile sensor nodes. *Pervasive and Mobile Computing*, 16:268–285, 2015.

29 Mary Hegarty, Daniel R Montello, Anthony E Richardson, Toru Ishikawa, and Kristin Lovelace. Spatial abilities at different scales: Individual differences in aptitude-test performance and spatial-layout learning. *Intelligence*, 34(2):151–176, 2006.

30 Bo Huang, Bo Wu, and Michael Barry. Geographically and temporally weighted regression for modeling spatio-temporal variation in house prices. *International Journal of Geographical Information Science*, 24(3):383–401, 2010.

31 Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. Cartel: a distributed mobile sensor computing system. In Andrew Campbell, Philippe Bonnet, and John S. Heidemann, editors, *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, pages 125–138. ACM, 2006.

32 Paul H Jung, Jean-Claude Thill, and Michele Issel. Spatial autocorrelation statistics of areal prevalence rates under high uncertainty in denominator data. *Geographical Analysis*, 51(3):354–380, 2019.

33 Teruyuki Kato, Yukihiro Terada, Masao Kinoshita, Hideshi Kakimoto, Hiroshi Isshiki, Masakatsu Matsuishi, Akira Yokoyama, and Takayuki Tanno. Real-time observation of tsunami by RTK-GPS. *Earth, Planets and Space*, 52(10):841–845, 2000.

34 Roman Krzysztofowicz. The case for probabilistic forecasting in hydrology. *Journal of Hydrology*, 249(1-4):2–9, 2001.

35 Jay Lee and Shengwen Li. Extending Moran's index for measuring spatiotemporal clustering of geographic events. *Geographical Analysis*, 49(1):36–57, 2017.

36 Yun Shin Lee and Stefan Scholtes. Empirical prediction intervals revisited. *International Journal of Forecasting*, 30(2):217–234, 2014.

37 Fernando A López-Hernández and Coro Chasco-Yrigoyen. Time-trend in spatial dependence: Specification strategy in the first-order spatial autoregressive model. *Estudios de Economia Aplicada*, 25(2), 2007.

38 Kevin M Lynch, Ira B Schwartz, Peng Yang, and Randy A Freeman. Decentralized environmental modeling by mobile sensor networks. *IEEE Transactions on Robotics*, 24(3):710–724, 2008.

39 GD Makarov. Estimates for the distribution function of a sum of two random variables when the marginal distributions are fixed. *Theory of Probability & its Applications*, 26(4):803–806, 1982.

40 Roger J Marshall. Mapping disease and mortality rates using empirical Bayes estimators. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 40(2):283–294, 1991.

41 Russell L Martin and JE Oeppen. The identification of regional forecasting models using space: Time correlation functions. *Transactions of the Institute of British Geographers*, pages 95–118, 1975.

42 Harvey J Miller. Tobler's first law and spatial analysis. *Annals of the Association of American Geographers*, 94(2):284–289, 2004.

43 Patrick AP Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950.

44 Enrico Natalizio and Valeria Loscrí. Controlled mobility in mobile sensor networks: Advantages, issues and challenges. *Telecommunication Systems*, 52(4):2411–2418, 2013.

45 Roger B. Nelsen. *An introduction to copulas*. Springer, New York, NY, 1999.

46 Neal Oden. Adjusting Moran's *I* for population density. *Statistics in Medicine*, 14(1):17–26, 1995.

47 J Keith Ord and Arthur Getis. Local spatial autocorrelation statistics: Distributional issues and an application. *Geographical Analysis*, 27(4):286–306, 1995.

48 J Keith Ord and Arthur Getis. Local spatial heteroscedasticity (LOSH). *The Annals of Regional Science*, 48(2):529–539, 2012.

49 Edzer J Pebesma and Cees G Wesseling. Gstat: A program for geostatistical modelling, prediction and simulation. *Computers & Geosciences*, 24(1):17–31, 1998.

50 Peter A Rogerson. The detection of clusters using a spatial version of the chi-square goodness-of-fit statistic. *Geographical Analysis*, 31(2):130–147, 1999.

51 Ludger Rüschendorf. Random variables with maximum sums. *Advances in Applied Probability*, pages 623–632, 1982.

52 Yasar Guneri Sahin. Animals as mobile biological sensors for forest fire detection. *Sensors*, 7(12):3084–3099, 2007.

53 Chenhua Shen, Chaoling Li, and Yali Si. Spatio-temporal autocorrelation measures for nonstationary series: A new temporally detrended spatio-temporal Moran's index. *Physics Letters A*, 380(1-2):106–116, 2016.

54 Steven M Smith and Edward Vela. Environmental context-dependent memory: A review and meta-analysis. *Psychonomic Bulletin & Review*, 8(2):203–220, 2001.

55 Enrico Steiger, René Westerholt, and Alexander Zipf. Research on social media feeds–A GIScience perspective. In Christina Capineri, Muki Haklay, Haosheng Huang, Vyron Antoniou, Juhani Kettunen, Frank Ostermann, and Ross Purves, editors, *European Handbook of Crowdsourced Geographic Information*, pages 237–254. Ubiquity Press, London, UK, 2016.

56 Xing Su, Hanghang Tong, and Ping Ji. Activity recognition with smartphone sensors. *Tsinghua Science and Technology*, 19(3):235–249, 2014.

57 Mila Sugovic and Jessica K Witt. An older view on distance perception: Older adults perceive walkable extents as farther. *Experimental Brain Research*, 226(3):383–391, 2013.

58 Michael Tiefelsdorf and Barry Boots. The exact distribution of Moran's *I*. *Environment and Planning A*, 27(6):985–999, 1995.

59 Michael Tiefelsdorf, Daniel A Griffith, and Barry Boots. A variance-stabilizing coding scheme for spatial link matrices. *Environment and Planning A*, 31(1):165–180, 1999.

60 Waldo R Tobler. A computer movie simulating urban growth in the Detroit region. *Economic Geography*, 46(sup1):234–240, 1970.

61 Sabrina Wahl, Christoph Bollmeyer, Susanne Crewell, Clarissa Figura, Petra Friederichs, Andreas Hense, Jan D Keller, and Christian Ohlwein. A novel convective-scale regional reanalysis COSMO-REA2: Improving the representation of precipitation. *Meteorologische Zeitschrift*, 26(4):345–361, 2017.

62 Thomas Waldhör. The spatial autocorrelation coefficient Moran's I under heteroscedasticity. *Statistics in Medicine*, 15(7-9):887–892, 1996.

63 SD Walter. The analysis of regional patterns in health data: I. Distributional considerations. *American Journal of Epidemiology*, 136(6):730–741, 1992.

64 SD Walter. The analysis of regional patterns in health data: II. The power to detect environmental effects. *American Journal of Epidemiology*, 136(6):742–759, 1992.

65 Elisabeth M Weiss, Georg Kemmler, Eberhard A Deisenhammer, W Wolfgang Fleischhacker, and Margarete Delazer. Sex differences in cognitive functions. *Personality and Individual Differences*, 35(4):863–875, 2003.

66 Karl F Wender, Daniel Haun, Björn Rasch, and Matthias Blümke. Context effects in memory for routes. In Christian Freksa, Wilfried Brauer, Christopher Habel, and Karl F. Wender, editors, *International Conference on Spatial Cognition (Spatial Cognition III)*, pages 209–231, Heidelberg, Germany, 2002. Springer.

67 René Westerholt. The impact of the spatial superimposition of point based statistical configurations on assessing spatial autocorrelation. In A. Mansourian, P. Pilesjö, L. Harrie, and R. von Lammeren, editors, *Geospatial Technologies for All: Short Papers, Posters and Poster Abstracts of the 21th AGILE Conference on Geographic Information Science*, Lund, Sweden, 2018.

68 René Westerholt, Bernd Resch, Franz-Benjamin Mocnik, and Dirk Hoffmeister. A statistical test on the local effects of spatially structured variance. *International Journal of Geographical Information Science*, 32(3):571–600, 2018.

69 René Westerholt, Bernd Resch, and Alexander Zipf. A local scale-sensitive indicator of spatial autocorrelation for assessing high-and low-value clusters in multiscale datasets. *International Journal of Geographical Information Science*, 29(5):868–887, 2015.

70 René Westerholt, Enrico Steiger, Bernd Resch, and Alexander Zipf. Abundant topological outliers in social media data and their effect on spatial analysis. *PLOS ONE*, 11(9):e0162360, 2016.

71 WH Williams and ML Goodman. A simple method for the construction of empirical confidence limits for economic forecasts. *Journal of the American Statistical Association*, 66(336):752–754, 1971.

72 Min Xu, Chang-Lin Mei, and Na Yan. A note on the null distribution of the local spatial heteroscedasticity (LOSH) statistic. *The Annals of Regional Science*, 52(3):697–710, 2014.

Generalizing Deep Models for Overhead Image Segmentation Through Getis-Ord Gi* Pooling

Xueqing Deng
EECS, University of California, Merced, CA, USA
xdeng7@ucmerced.edu

Yuxin Tian
EECS, University of California, Merced, CA, USA
ytian8@ucmerced.edu

Shawn Newsam
EECS, University of California, Merced, CA, USA
snewsam@ucmerced.edu

Abstract

That most deep learning models are purely data driven is both a strength and a weakness. Given sufficient training data, the optimal model for a particular problem can be learned. However, this is usually not the case and so instead the model is either learned from scratch from a limited amount of training data or pre-trained on a different problem and then fine-tuned. Both of these situations are potentially suboptimal and limit the generalizability of the model. Inspired by this, we investigate methods to inform or guide deep learning models for geospatial image analysis to increase their performance when a limited amount of training data is available or when they are applied to scenarios other than which they were trained on. In particular, we exploit the fact that there are certain fundamental rules as to how things are distributed on the surface of the Earth and these rules do not vary substantially between locations. Based on this, we develop a novel feature pooling method for convolutional neural networks using Getis-Ord G_i^* analysis from geostatistics. Experimental results show our proposed pooling function has significantly better generalization performance compared to a standard data-driven approach when applied to overhead image segmentation.

2012 ACM Subject Classification Computing methodologies → Neural networks

Keywords and phrases Remote sensing, convolutional neural networks, pooling function, semantic segmentation, generalization

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.3

Funding This work was funded in part by a National Science Foundation grant, #IIS-1747535.

Acknowledgements We gratefully acknowledge the support of NVIDIA Corporation through the donation of the GPU card used in this work. And we thank Yi Zhu for providing helpful discussion.

1 Introduction

Research in remote sensing has been steadily increasing since it is an important source for Earth observation. Overhead imagery can easily be acquired using low-cost drones and no longer requires access to expensive high-resolution satellite or airborne platforms. Since the data provides convenient and large-scale coverage, people are using it for a number of societally important problems such as traffic monitoring [20], land cover segmentation [16], building extraction [10, 37], geolocalization [31], image retrieval [27], etc.

Recently, the analysis of overhead imagery has benefited greatly from deep learning thanks to the significant advancements made by the computer vision community on regular (non-overhead) images. However, there still often remains challenges when adapting these deep learning techniques to overhead image analysis, such as the limited availability of labeled overhead imagery, the difficulty of the models to generalize between locations, etc.

Annotating overhead imagery is labor intensive so existing datasets are often not large enough to train effective convolutional neural networks (CNNs) from scratch. A common practice therefore is to fine-tune an ImageNet pre-trained model on a small amount of annotated overhead imagery. However, the generalization capability of fine-tuned models is limited as models trained on one location may not work well on others. This is known as the *cross-location generalization* problem and is not necessarily limited to overhead image analysis as it can also be a challenge for ground-level imagery such as cross-city road scene segmentation [9]. Deep models are often overfit due to their large capacity yet generalization is particularly important for overhead images since they can look quite different due to variations in the seasons, position of the sun, location variation, etc. For regular image analysis, two widely adopted approaches to overcome these so-called domain gaps include domain adaptation [11, 12, 33–35] and data fusion. Both approaches have been adapted by the remote sensing community [2] to improve performance and robustness.

In this paper, we take a different, novel approach to address the domain gap problem. We exploit the fact that things are not laid out at random on the surface of the Earth and that this structure does not vary substantially between locations. In particular, we pose the question of how prior knowledge of this structure or, more interestingly, how the fundamental rules of geography, might be incorporated into general CNN frameworks. Inspired by work on physics-guided neural networks [14], we develop a framework in which spatial hotspot analysis informs the feature map pooling. We term this geo-constrained pooling strategy *Getis-Ord G_i^* pooling* and show that it significantly improves the semantic segmentation of overhead imagery particularly in cross-location scenarios. To our knowledge, ours is the first work to incorporate geo-spatial knowledge directly into the fundamental mechanisms of CNNs.

Our contributions are summarized as follows:

1. We propose Getis-Ord G_i^* pooling, a novel pooling method based on spatial Getis-Ord G_i^* analysis of CNN feature maps. Getis-Ord G_i^* pooling is shown to significantly improve model generalization for overhead image segmentation.
2. We establish more generally that using geospatial knowledge in the design of CNNs can improve the generalizability of the models.

2 Related Work

Semantic segmentation. Fully connected neural networks (FCN) were recently proposed to improve the semantic segmentation of non-overhead imagery [19]. Various techniques have been proposed to boost their performance, such as atrous convolution [5–7, 40], skip connections [25] and preserving max pooling index for unpooling [3]. And, recently, video has been used to scale up training sets by synthesizing new training samples [42]. Remote sensing research has been driven largely by adapting advances in regular image analysis to overhead imagery. In particular, deep learning approaches to overhead image analysis have become a standard practice for a variety of tasks, such as land use/land cover classification [16], building extraction [37], road segmentation [22], car detection [8], etc. More literature can be found in a recent survey [41]. And various segmentation networks have been proposed, such relation-augmentation networks [23] and ScasNet [18]. However, these methods only adapt deep learning techniques and networks from regular to overhead images–they do not incorporate geographic structure or knowledge.

Knowledge guided neural networks. Analyzing overhead imagery is not just a computer vision problem since the principles of the physical world such as geo-spatial relationships can help. For example, knowing the road map of a city can improve tasks like building extraction or land cover segmentation. While there are no works directly related to ours, there have been some initial attempts to incorporate geographic knowledge into deep learning [4, 39]. Chen et al. [4] develop a knowledge-guided golf course detection approach using a CNN fine-tuned on temporally augmented data. They also apply area-based rules during a post-processing step. Zhang et al. [39] propose searching for adjacent parallel line segments as prior spatial information for the fast detection of runways. However, these methods simply fuse prior knowledge from other sources. Our proposed method is novel in that we incorporate geo-spatial rules into the CNN mechanics. We show later how this helps regularize the model and leads to better generalization.

Pooling functions. A number of works have investigated different pooling methods for image classification and segmentation tasks. The L_p norm has been proposed to extend max pooling where intermediate pooling functions are manually selected between max and average pooling to better fit the distribution of the input data. [17] generalizes pooling methods by using a learned linear combination of max and average pooling. Detail-Preserving Pooling (DPP) [26] learns weighted summations of pixels over different pooling regions. Salient pixels are considered more important and thus given higher weighting. Strided convolution has been used to replace all max pooling layers and activation functions in a small classification model that is trained from scratch and has shown to improve performance [30]. Strided convolution is common in segmentation tasks. For example, the DeepLab series of networks [6, 7] use strided convolutional layers for feature down-sampling rather than max pooling. To enhance detail preservation in segmentation, a recent polynomial pooling approach is proposed in [36]. However, all these pooling methods are based on non-spatial statistics. We instead incorporate geo-spatial rules/knowledge to perform the pooling and downsampling.

3 Methods

In this section, we investigate how geo-spatial knowledge can be incorporated into standard deep CNNs. We discuss some general rules from geography to describe geo-spatial patterns on the Earth. Then we propose using Getis-Ord G_i^* analysis, a common technique for geo-spatial clustering, to encapsulate these rules. This then informs our pooling function which is general and can be used in most network architectures.

3.1 Getis-Ord G_i^* pooling ($\mathcal{G}$-pooling)

We take inspiration from the well-known first law of geography: *everything is related to everything else, but near things are more related than distant things* [32]. While this rule is very general and abstract, it motivates a number of quantitative frameworks that have been shown to improve geospatial data analysis. For example, it motivates spatial autocorrelation which is the basis for spatial prediction models like kriging. It also motivates the notion of spatial clustering wherein similar things that are spatially nearby are more significant than isolated things. Our proposed framework exploits this to introduce a novel feature pooling method which we term Getis-Ord G_i^* pooling.

Pooling is used to spatially downsample the feature maps in deep CNNs. In contrast to standard image downsampling methods which seek to preserve the spatial envelope of pixel values, pooling selects feature values that are more significant in some sense. The most

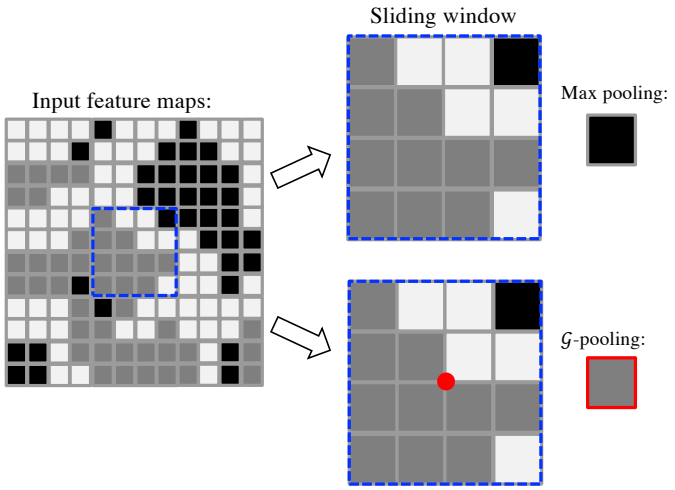


Figure 1 Given a feature map as an input, max pooling (top right) and the proposed $\mathcal{G}$-pooling (bottom right) produce different downsampled output feature maps. $\mathcal{G}$-pooling exploits spatial clusters of input feature map values. For example, the feature map within the sliding window (dotted blue line) indicates a spatial cluster. Max pooling takes the max value ignoring the spatial cluster, while our $\mathcal{G}$-pooling takes the interpolated value at the center location. (White, gray and black represent three ranges of feature map values, from low to high.)

standard pooling method is max pooling in which the maximum feature value in a window is propagated. Other pooling methods have been proposed. Average pooling is an obvious choice and is used in [13, 38] for image classification. Strided convolution [15] has also been used. However, max pooling remains by far the most common as it has the intuitive appeal of extracting the maximum activation and thus the most prominent features.

However, we postulate that isolated high feature values might not be the most informative and instead develop a method to propagate clustered values. Specifically, we use a technique from geostatistics termed hotspot analysis to identify clusters of large positive values and then propagate a representative from these clusters. Hotspot analysis uses the Getis-Ord G_i^* [24] statistic to find locations that have either high or low values and are surrounded by locations also with high or low values. These locations are the so-called hotspots. The Getis-Ord G_i^* statistic is computed by comparing the local sum of a feature and its neighbors proportionally to the sum of all features in a spatial region. When the local sum is different from the expected local sum, and when that difference is too large to be the result of random noise, it will lead to a high positive or low negative G_i^* value that is statistically significant. We focus on locations with high positive G_i^* values since we want to propagate activations.

3.2 Definition

We now describe our $\mathcal{G}$-pooling algorithm in detail. Please see Figure 1 for reference. Similar to other pooling methods, we use a sliding window to downsample the input. Given a feature map within the window, in order to compute its G_i^*, we first need to define the weight matrix based on the spatial locations.

We denote the feature values within the sliding window as $\mathbf{X} = x_1, x_2, ..., x_n$ where n is the number of pixels (locations) within the sliding window. We assume the window is rectangular and compute the G_i^* statistic at the center of the window. Let the feature value at the center be x_i. (If the center does not fall on a pixel location then we compute x_i

as the average of the adjacent values.) The G_i^* statistic uses weighed averages where the weights are based on spatial distances. Let $p^x(x_j)$ and $p^y(x_j)$ denote the x and y positions of feature value x_j in the image plane. A weight matrix w that measures the Euclidean distance on the image plane between x_i and the other locations within the sliding window is then computed as

$$w_{i,j} = \sqrt{(p^x(x_i) - p^x(x_j))^2 + (p^y(x_i) - p^y(x_j))^2}. \tag{1}$$

The Getis-Ord G_i^* value at location i is now computed as

$$G_i^* = \frac{\sum_{j=1}^{n} w_{i,j}x_j - \bar{X}\sum_{j=1}^{n} w_{i,j}}{S\sqrt{\frac{[n\sum_{j=1}^{n} w_{i,j}^2 - (\sum_{j=1}^{n} w_{i,j})^2]}{n-1}}}. \tag{2}$$

where $\bar{X}$ and S are as below,

$$\bar{X} = \frac{\sum_{j=1}^{n} x_j}{n}, \tag{3}$$

$$S = \sqrt{\frac{\sum_{j=1}^{n} x_j^2}{n} - (\bar{X})^2}. \tag{4}$$

Spatial clusters can be detected based on the G_i^* value. The higher the value, the more significant the cluster is. However, the G_i^* value just indicates whether there is a spatial cluster or not. To achieve our goal of pooling, we need to summarize the local region of the feature map by extracting a representative value. We use a threshold to do this. If the computed G_i^* is greater than or equal to the threshold, a spatial cluster is detected and the value x_i is used for pooling; otherwise the maximum value in the window is used:

$$\mathcal{G} - pooling(\mathbf{x}) = \begin{cases} x_i & \text{if } G_i^* \geq \text{threshold} \\ max(\mathbf{x}) & \text{if } G_i^* < \text{threshold} \end{cases} \tag{5}$$

G_i^* is in range [-2.8,2.8] for our particular range of feature map values. A positive value indicates a hotspot which is a cluster of positive values, a negative value indicates a coldspot which is a cluster of negative values, and values near zero indicate scatter. The absolute value $|G_i^*|$ indicates the significance of the cluster. For example, a high positive G_i^* value indicates the location is more likely to be a spatial cluster of high positive values.

The output feature map produced by $\mathcal{G}$-pooling is $\mathcal{G}$-pooling($\mathbf{X}$) which results after sliding the window over the entire input feature map. We experiment with three threshold values: 1.0, 1.5, 2.0. A higher threshold value results in fewer spatial clusters and so max pooling will be applied more often. A lower threshold value results in more spatial clusters and so max pooling will be applied less often. As the threshold ranges from 1.0 to 1.5 to 2.0, fewer spatial clusters/hotspots will be detected. We find that a threshold of 2.0 results in few hostpots being detected and max pooling primarily being used.

3.3 Network Architecture

A pretrained VGG network [29] is used in our experiments. VGG has been widely used as a backbone in various semantic segmentation networks such as FCN [19], U-net [25], and SegNet [3]. In VGG, the standard max pooling is a 2×2 window size with a stride of 1. Our

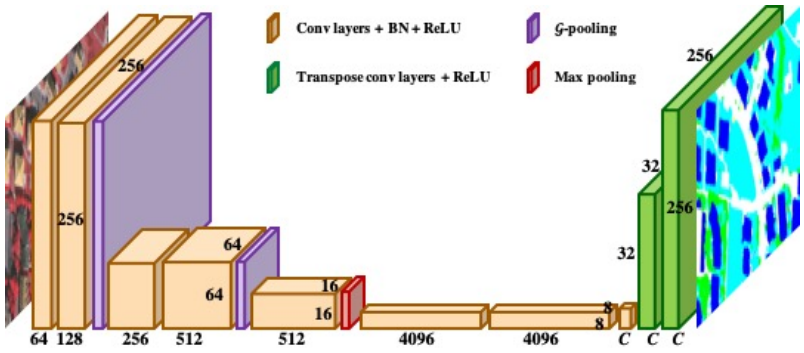


Figure 2 FCN network architecture with $\mathcal{G}$-pooling.

proposed $\mathcal{G}$-pooling uses a 4×4 window size with a stride of 4. Therefore, in standard pooling, the feature maps are reduced by a factor of 2, while in our $\mathcal{G}$-pooling, they are reduced by a factor of 4. A larger window is used in our proposed $\mathcal{G}$-pooling since Getis-Ord G_i^* analysis is not as meaningful for small regions. However, we evaluated the scenario in which standard pooling is also performed with a 4×4 sliding window and the performance is only slightly different from using a standard 2×2 window. In general, segmentation networks using VGG16 as the backbone have 5 max pooling layers, each of which downsamples by a factor of 2. So, when we replace max pooling with our proposed $\mathcal{G}$-pooling, our architecture has two $\mathcal{G}$-pooling and one max pooling layers in order to produce the same sized final feature map.

Table 1 Training and test data are from the same location. These results are for a FCN using VGG-16 as the backbone. Stride conv, $\mathcal{P}$-pooling and our approach, $\mathcal{G}$-pooling, are used to replace the standard max/average pooling. The per class results are reported as IoU. mIoU is the average across classes. Pixel Acc. is the overall pixel accuracy. Higher is better for all results.

Potsdam							
Methods	Roads	Buildings	Low Veg.	Trees	Cars	mIoU	Pixel Acc.
Max	70.62	74.28	65.94	61.36	61.40	66.72	79.55
Average	69.34	74.49	63.94	60.06	60.28	65.62	78.08
Stride	67.22	73.97	63.01	60.09	59.39	64.74	77.54
$\mathcal{P}$-pooling	**71.97**	75.55	66.80	62.03	62.39	67.75	81.02
$\mathcal{G}$-pooling-1.0 (ours)	68.59	**77.39**	67.48	55.56	62.18	66.24	79.43
$\mathcal{G}$-pooling-1.5 (ours)	70.06	76.12	**67.67**	**62.12**	**63.91**	**67.98**	**81.63**
$\mathcal{G}$-pooling-2.0 (ours)	70.99	74.89	65.34	61.57	60.77	66.71	79.46
Vaihingen							
Max	70.63	80.42	51.57	70.12	55.32	65.61	81.88
Average	70.54	79.86	50.49	69.18	54.83	64.98	79.98
Strde conv	68.36	77.65	49.21	67.34	53.29	63.17	79.44
$\mathcal{P}$-pooling	71.06	**80.52**	51.70	**70.93**	53.65	65.57	**82.44**
$\mathcal{G}$-pooling-1.0 (ours)	**72.15**	79.69	**53.28**	70.89	**53.72**	**65.95**	81.78
$\mathcal{G}$-pooling-1.5 (ours)	71.61	78.74	48.18	68.53	55.64	64.54	80.42
$\mathcal{G}$-pooling-2.0 (ours)	71.09	78.88	50.62	68.32	54.01	64.58	80.75

4 Experiments

4.1 Dataset

ISPRS dataset. We evaluate our method on two image datasets from the ISPRS 2D Semantic Labeling Challenge [1]. These datasets are comprised of very high resolution aerial images over two cities in Germany: Vaihingen and Potsdam. While Vaihingen is a relatively small village with many detached buildings and small multi-story buildings, Potsdam is a typical historic city with large building blocks, narrow streets and dense settlement structure. The goal is to perform semantic labeling of the images using six common land cover classes: buildings, impervious surfaces (e.g. roads), low vegetation, trees, cars and clutter/background. We report test metrics obtained on the held-out test images.

Vaihingen. The Vaihingen dataset has a resolution of 9 cm/pixel with tiles of approximately 2100×2100 pixels. There are 33 images, for which 16 have a public ground truth. Even though the tiles consist of Infrared-Red-Green (IRRG) images and DSM data extracted from the Lidar point clouds, we use only the IRRG images in our work. We select five images for validation (IDs: 11, 15, 28, 30 and 34) and the remaining 11 for training, following [21, 28].

Potsdam. The Potsdam dataset has a resolution of 5 cm/pixel with tiles of 6000×6000 pixels. There are 38 images, for which 24 have public ground truth. Similar to Vaihingen, we only use the IRRG images. We select seven images for validation (IDs: 2_11, 2_12, 4_10, 5_11, 6_7, 7_8 and 7_10) and the remaining 17 for training, again following [21, 28].

4.2 Experimental Settings

We first compare our $\mathcal{G}$-pooling to standard max-pooling, average-pooling, strided convolution and the recently proposed $\mathcal{P}$-pooling [36], all using an FCN semantic segmentation network with a VGG backbone. We later perform experiments using other semantic segmentation networks. We compare to max/average pooling as they are commonly used for downsampling semantic segmentation networks that have VGG as a backbone. Strided convolution has been used to replace max pooling in recent semantic segmentation frameworks such as the DeepLab series [5–7] and PSPNet [40]. Detail preserving pooling (DPP) has also been used to replace standard pooling in works such as DDP [26] and $\mathcal{P}$-pooling [36]. We compare to the most recent, $\mathcal{P}$-pooling, as it has been shown to outperform other detail preserving methods.

4.3 Evaluation Metrics

We have two goals in this work, the model's segmentation accuracy and its generalization performance. We report model accuracy as the performance on a test/validation set when the model is trained using training data from the same location (the same dataset). We report model generalizability as the performance on a test/validation set when the model is trained using training data from a different location (a different dataset). In general, the domain gap between the training and test/validation sets is small when they are from the same location/dataset. However, cross-location/dataset testing can result in large domain shifts.

Model accuracy. The commonly used per class intersection over union (IoU) and mean IoU (mIoU) as well as the pixel accuracy are adopted for evaluating segmentation accuracy. IoU is commonly used to measure the performance in semantic segmentation. IoU is the

Table 2 Cross-location evaluation. We compare the generalization capability of $\mathcal{G}$-pooling with domain adaptation using an AdaptSegNet model which exploits unlabeled data.

	Imp. Surf.	Buildings	Low Veg.	Trees	Cars	mIoU	Pixel Acc.
Potsdam → Vaihingen							
Max-pooling	28.75	51.10	13.48	**56.00**	25.99	35.06	47.48
stride conv	28.66	50.98	12.76	55.02	24.81	34.45	46.51
$\mathcal{P}$-pooling	32.87	50.43	13.04	55.41	25.60	35.47	48.94
Ours ($\mathcal{G}$-pooling)	**37.27**	**54.53**	**14.85**	54.24	**27.35**	**37.65**	**55.20**
AdaptSegNet	41.54	40.74	21.68	50.45	36.87	38.26	57.73
Vaihingen → Potsdam							
Max-pooling	20.36	24.51	19.19	9.71	3.65	15.48	45.32
stride conv	20.65	23.22	16.57	8.73	8.32	15.50	42.28
$\mathcal{P}$-pooling	23.97	27.66	14.03	**10.30**	12.07	19.61	44.98
Ours ($\mathcal{G}$-pooling)	**27.05**	**29.34**	**33.57**	9.12	**16.01**	**23.02**	**45.54**
AdaptSegNet	40.28	37.97	46.11	15.87	20.16	32.08	50.28

area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth. This metric ranges from 0–100% with 0% indicating no overlap and 100% indicating perfect overlap with the ground truth. Therefore, higher IoU scores indicate better segmentation performance. We compute IoU for each class as well as the mean IoU (mIoU) over all classes. Pixel accuracy is simply the percentage of pixels labeled correctly.

Model generalizability. To evaluate model generalizability, we apply a model trained on ISPRS Vahingen to ISPRS Potsdam (Potsdam→Vaihingen), and vice versa (Vaihingen→Potsdam).

4.4 Implementation Details

Implementation of $\mathcal{G}$-pooling. The models are implemented using the PyTorch framework. Max-pooling, average-pooling and strided convolution are provided in PyTorch, and we utilize open-source code for $\mathcal{P}$-pooling. We implement our $\mathcal{G}$-pooling in C and use an interface to connect to PyTorch for network training. We adopt an FCN [19] network architecture with a pretrained VGG-16 [29] as the backbone. The details of the FCN using our $\mathcal{G}$-pooling can be found in Section 3.3. The results in Table 1 are reported using FCN with a VGG-16 backbone.

Training settings. Since the image tiles are too large to be fed through a deep CNN due to limited GPU memory, we randomly extract image patches of size of 256×256 pixels as the training set. Following standard practice, we only use horizontal and vertical flipping as data augmentation during training. For testing, the whole image is split into 256×256 patches with a stride of 256. Then, the predictions of all patches are concatenated for evaluation.

We train all our models using Stochastic Gradient Descent (SGD) with an initial learning rate of 0.1, a momentum of 0.9, a weight decay of 0.0005 and a batch size of 5. If the validation loss plateaus for 3 consecutive epochs, we divide the learning rate by 10. If the

validation loss plateaus for 6 consecutive epochs or the learning rate is less than 1e-8, we stop the model training. We use a single TITAN V GPU for training and testing. We observe that $\mathcal{G}$-pooling takes about twice the time for training and inference as standard max pooling.

Table 3 The average percentage of detected spatial clusters per feature map with different thresholds.

Threshold	1.0	1.5	2.0
Potsdam	15.87	9.85	7.65
Vaihingen	14.99	10.44	7.91

5 Effectiveness of $\mathcal{G}$-pooling

In this section, we first show that incorporating geospatial knowledge into the pooling function of standard CNNs can improve segmentation accuracy even when the training and test sets are from the same location. We then demonstrate that our proposed $\mathcal{G}$-pooling results in improved generalization by training and testing with different locations.

The performance of the various pooling options for when the training and test sets are from the same location is shown in Table 1. For $\mathcal{G}$-pooling, we experiment with 3 different thresholds, 1.0, 1.5 and 2.0. The range of G_i^* is [-2.8, 2.8]. As explained in Section 3.2, a higher G_i^* value results in increased max pooling. If we set the G_i^* to 2.8 then only max pooling is performed. Qualitative results are shown in Figure 3. The results of the cross-location case are shown in Table 2.

Non-spatial vs geospatial statistics. Standard pooling techniques are non-spatial, for example, finding the max/average value. Instead, our approach uses geospatial statistics to discover how things are related based on their location. Here, we pose the question, "*is this knowledge useful for training and deploying deep CNNs?*". As mentioned in Section 3, incorporating such knowledge has the potential to improve model generalizability. As shown in Table 1, our approach outperforms $\mathcal{P}$-pooling on most classes but not for all threshold values, indicating that threshold selection is important. The qualitative results in Figure 3 show our proposed $\mathcal{G}$-pooling results in less pepper-and-salt artefacts. In particular, there is less noise inside the objects compared to the other methods. This demonstrates our proposed $\mathcal{G}$-pooling is better able to model the geospatial distributions and results in more compact object predictions. The effect of the threshold on the number of spatial clusters that are detected is shown in Table 3. As described in Section 3, higher threshold values result in fewer clusters.

Domain adaptation vs knowledge incorporation. Table 2 compares the various pooling functions to unsupervised domain adaptation (UDA) for the case when the training and test sets are from different locations. We note that the UDA method AdaptSegNet [33] uses a large amount of unlabeled data from the target dataset to adapt the model which has been demonstrated to help generalization. Direct comparison with this method is therefore unfair since the other methods do not exploit this unlabeled data. As shown in Table 2, our proposed $\mathcal{G}$-pooling achieves the best overall performance among the pooling methods. For Potsdam→Vaihingen, $\mathcal{G}$-pooling outperforms $\mathcal{P}$-pooling by more than 2% in mIoU. For Vaihingen→Potsdam, the improvement is even more significant at 3.41%. Our

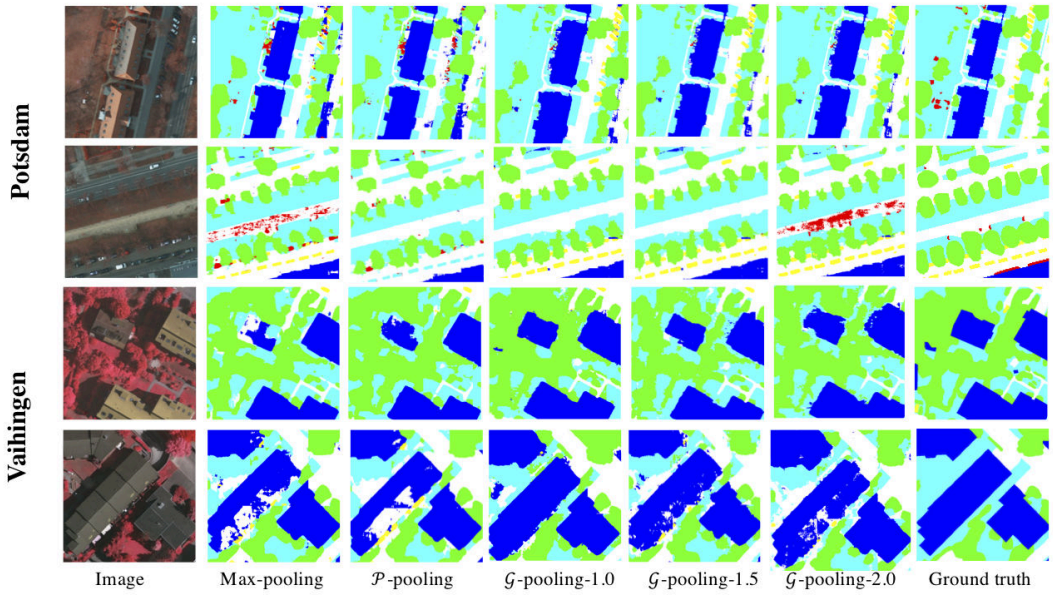


Figure 3 Qualitative results. White: impervious surfaces, blue: building, cyan: low vegetation, green: trees, yellow: cars, red: clutter.

method even performs almost as well as domain adaptation using AdaptSegNet, especially for Potsdam→Vaihingen where the gap is only 0.61%. Overall, these results confirm our assertion that incorporating geospatial knowledge into the model architecture can improve generalization performance. We note that our proposed $\mathcal{G}$-pooling can be combined with domain adaptation techniques, such as AdaptSegNet, to provide even better generalization.

6 $\mathcal{G}$-pooling and state-of-the-art methods

In order to verify that our proposed $\mathcal{G}$-pooling is able to provide improvement to state-of-the-art segmentation approaches in addition to FCN, we select DeepLab [5] and SegNet [3] as additional network architectures. As mentioned above, the models in Section 5 use FCN as the network architecture and VGG-16 as the backbone. For fair comparison with FCN, VGG-16 is also used as the backbone in DeepLab and SegNet.

DeepLab [5] uses large receptive fields through dilated convolution. For the baseline DeepLab itself, *pool4* and *pool5* from the backbone VGG-16 are removed and the the dilated conv layers with a dilation rate of 2 are replaced with *conv*5 layers. For the $\mathcal{G}$-pooling version, *pool1* and *pool2* are replaced with $\mathcal{G}$-pooling and we keep *pool3*. Thus there are three max pooling layers in the baseline and one $\mathcal{G}$-pooling layer and one max pooling layer in our proposed version. SegNet uses an encoder-decoder architecture and preserves the max pooling index for unpooling in the decoder. Similar to Deeplab, there are 5 max pooling layers in total in the encoder of SegNet so *pool1* and *pool2* are replaced with the proposed $\mathcal{G}$*_pool1* and *pool3* and *pool4* are replaced with $\mathcal{G}$*_pool2*, and *pool5* is kept. This leads us to use a 4×4 unpooling window to recover the spatial resolution where the original one is just 2×2. Thus there are two $\mathcal{G}$-pooling and one max pooling layers in our SegNet version.

As can be seen in Table 4, $\mathcal{G}$-pooling improves the model accuracy for Potsdam from 67.97% to 68.33% for DeepLab. And the improvement on the generalization test Potsdam→Vaihingen is even more obvious: $\mathcal{G}$-pooling improves mIoU from 38.57% to 40.04% for DeepLab. Similar

observations can be made for SegNet and FCN. For Vaihingen, even though the model accuracy is not as high as the baseline, the difference is small. The mIoUs of our versions of DeepLab, SegNet and FCN are less than 1% lower. We note that Vaihingen is an easier dataset than Potsdam since it only contains urban scenes while Potsdam contains both urban and nonurban. However, the generalizability of our model using $\mathcal{G}$-pooling is much better. When testing on Potsdam using a model trained on Vaihingen, FCN with $\mathcal{G}$-pooling is able to achieve 23.02% mIoU which is an improvement of 7.54%. The same observations can be made for DeepLab and SegNet.

Table 4 Experimental results comparing w/o and w/ proposed $\mathcal{G}$-pooling for the state-of-the-art segmentation networks. Potsdam→Vaihingen indicates the model is trained on Potsdam and tested on Vaihingen.

	Potsdam			Potsdam→Vaihingen	
Network	$\mathcal{G}$-Pooling	mIoU	Pixel Acc.	mIoU	Pixel Acc.
DeepLab	×	67.97	81.25	38.57	58.47
	✓	68.33	80.67	**40.04**	**63.21**
SegNet	×	69.47	82.53	35.98	53.69
	✓	70.17	83.27	**39.04**	**56.42**
FCN	×	66.72	79.55	35.06	47.48
	✓	67.98	81.63	**37.65**	**55.20**
	Vaihingen			Vaihingen→Potsdam	
DeepLab	×	70.80	83.74	18.44	33.96
	✓	70.11	83.09	**19.26**	**36.17**
SegNet	×	66.04	81.79	16.77	45.90
	✓	66.71	82.66	**25.64**	**48.08**
FCN	×	65.61	81.88	15.48	45.32
	✓	65.95	81.87	**23.02**	**45.54**

7 Discussion

Incorporating knowledge is not a novel approach for neural networks. Before deep learning, there was work on rule-based neural networks which required expert knowledge to design the network for specific applications. Due to the large capacity of deep models, deep learning has become the primary approach to address vision problems. However, deep learning is a data-driven approach which relies significantly on the amount of training data. If the model is trained with a large amount of data then it will have good generalization. But the case is often, particularly in overhead image segmentation, that the dataset is not large enough like it is in ImageNet/Cityscapes. This causes overfitting. Early stopping, cross-validation, etc. can help to avoid overfitting. Still, if significant domain shift exists between the training and test sets, the deep models do not perform well. In this work, we propose a knowledge-incorporated approach to reduce overfitting. We address the question of how to incorporate the knowledge directly into the deep models by proposing a novel pooling method for overhead image segmentation. But some issues still need discussing as follows.

Scenarios using $\mathcal{G}$-pooling. As mentioned in section 3, $\mathcal{G}$-pooling is developed using Getis-Ord G_i^* analysis which quantifies spatial correlation. Our approach is potentially therefore specific to geospatial data and might not be appropriate for other image datasets. This is a

general restriction of incorporating domain knowledge into machine learning models. Getis-Ord G_i^* provides a method to identify spatial clusters. The effect is similar to conditional random fields/Markov random fields in standard computer vision post-processing methods. However, it is different from them since the spatial clustering depends dynamically on the feature maps and the geospatial location while post-processing methods rely only on the predictions of the models.

Local geospatial patterns. Even though Getis-Ord G_i^* analysis is typically used to detect hotspots over larger regions than we are applying it to, it still characterizes local geospatial patterns in a way that is informative for spatial pooling. Also, since we perform two $\mathcal{G}$-pooling operations sequentially to feature maps of decreasing size, the "receptive field" of our pooling in the input image is actually larger. In particular, the first 4×4 pooling window is slid over a 256×256 feature map, resulting in a feature map of size 64×64. This is input to the next conv layer, after which a second $\mathcal{G}$-pooling is applied, again using a 4×4 sliding window. Tracing this back, this corresponds to a region of size 16×16 which is $1/16$ of the whole image along each dimension.

Limitations. There are some limitations of our investigation. For example, we did not explore the optimal window size for performing the Getis-Ord G_i^* analysis. We also only considered one kind of spatial pattern, clusters. And, there might be better places than pooling to incorporate geospatial knowledge into CNN architectures.

8 Conclusion

In this paper, we investigate how geospatial knowledge can be incorporated into deep learning for geospatial image analysis through modification of the network architecture itself. In particular, we replace standard pooling in CNNs with a novel pooling method motivated by general geographic rules and computed using the Getis-Ord G_i^* statistic. We investigate the impact of our proposed method on semantic segmentation using an evaluation dataset. We realize, though, that ours is just preliminary work into geospatial guided deep learning. In the future, we will explore other ways to encode geographic rules so they can be incorporated into deep learning models.

References

1 ISPRS 2D Semantic Labeling Challenge. `http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html`.

2 N. Audebert, B. Saux, and S. Lefèvre. Beyond RGB: Very High Resolution Urban Remote Sensing with Multimodal Deep Networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2018.

3 V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

4 J. Chen, C. Wang, A. Yue, J. Chen, D. He, and X. Zhang. Knowledge-Guided Golf Course Detection Using a Convolutional Neural Network Fine-Tuned on Temporally Augmented Data. *J. Appl. Remote Sens.*, 2017.

5 L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

6 L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

7 L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *European conference on computer vision (ECCV)*, 2018.

8 X. Chen, S. Xiang, C. Liu, and C. Pan. Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks. *IEEE Geoscience and Remote Sensing Letters*, 2014.

9 Y. Chen, W. Chen, Y. Chen, B. Tsai, Y. Wang, and M. Sun. No More Discrimination: Cross City Adaptation of Road Scene Segmenters. In *International Conference on Computer Vision (ICCV)*, 2017.

10 X. Deng, H. L. Yang, N. Makkar, and D. Lunga. Large Scale Unsupervised Domain Adaptation of Segmentation Networks with Adversarial Learning. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2019.

11 J. Hoffman, E. Tzeng, T. Park, J. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *International Conference on Machine Learning (ICML)*, 2018.

12 J. Hoffman, D. Wang, F. Yu, and T. Darrell. FCNs in the Wild: Pixel-Level Adversarial and Constraint-based Adaptation. *arXiv preprint arXiv:1612.02649*, 2016.

13 G. Huang, Z. Liu, L. Van Der Maaten, and K. Weinberger. Densely Connected Convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

14 A. Karpatne, W. Watkins, J. Read, and V. Kumar. Physics-Guided Neural Networks (PGNN): An Application in Lake Temperature Modeling. *arXiv preprint arXiv:1710.11431*, 2017.

15 J. Kuen, X. Kong, G. Wang, and Y. Tan. DelugeNets: Deep Networks with Efficient and Flexible Cross-Layer Information Inflows. In *International Conference on Computer Vision (ICCV)*, 2017.

16 N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov. Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data. *IEEE Geoscience and Remote Sensing Letters*, 2017.

17 C. Lee, P. Gallagher, and Z. Tu. Generalizing Pooling Functions in Convolutional Neural Nnetworks: Mixed, Gated, and Tree. In *Artificial Intelligence and Statistics*, 2016.

18 Y. Liu, B. Fan, L. Wang, J. Bai, S. Xiang, and C. Pan. Semantic Labeling in Very High Resolution Images via a Self-Cascaded Convolutional Neural Network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2018.

19 J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

20 X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors*, 2017.

21 E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. High-Resolution Aerial Image Labeling with Convolutional Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*, 2017.

22 V. Mnih and G. E. Hinton. Learning to Detect Roads in High-Resolution Aerial Images. In *European Conference on Computer Vision (ECCV)*, 2010.

23 L. Mou, Y. Hua, and X. X. Zhu. A Relation-Augmented Fully Convolutional Network for Semantic Segmentation in Aerial Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

24 J. K. Ord and Arthur Getis. Local Spatial Autocorrelation Statistics: Distributional Issues and an Application. *Geographical Analysis*, 1995.

25 O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

26 F. Saeedan, N. Weber, M. Goesele, and S. Roth. Detail-preserving pooling in deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

27 Z. Shao, W. Zhou, X. Deng, M. Zhang, and Q. Cheng. Multilabel Remote Sensing Image Retrieval Based on Fully Convolutional Network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2020.

28 Jamie Sherrah. Fully Convolutional Networks for Dense Semantic Labelling of High-Resolution Aerial Imagery. *arXiv preprint arXiv:1606.02585*, 2016.

29 K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.

30 J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for Simplicity: The All Convolutional Net. In *International Conference on Learning Representation workshop (ICLR workshop)*, 2015.

31 Y. Tian, X. Deng, Y. Zhu, and S. Newsam. Cross-Time and Orientation-Invariant Overhead Image Geolocalization Using Deep Local Features. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.

32 W. R. Tobler. A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*, 1970.

33 Y. Tsai, W. Hung, S. Schulter, K. Sohn, M. Yang, and M. Chandraker. Learning to Adapt Structured Output Space for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

34 Y. Tsai, K. Sohn, S. Schulter, and M. Chandraker. Domain Adaptation for Structured Output via Discriminative Representations. In *International conference on Computer Vision (ICCV)*, 2019.

35 E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial Discriminative Domain Adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

36 Z. Wei, J. Zhang, L. Liu, F. Zhu, F. Shen, Y. Zhou, S. Liu, Y. Sun, and L. Shao. Building Detail-Sensitive Semantic Segmentation Networks with Polynomial Pooling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

37 J. Yuan. Learning Building Extraction in Aerial Scenes with Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

38 S. Zagoruyko and N. Komodakis. Wide Residual Networks. *arXiv preprint arXiv:1605.07146*, 2016.

39 P. Zhang, X. Niu, Y. Dou, and F. Xia. Airport Detection from Remote Sensing Images using Transferable Convolutional Neural Networks. In *International Joint Conference on Neural Networks (IJCNN)*, 2016.

40 H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid Scene Parsing Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

41 X. X. Zhu, D. Tuia, L. Mou, G. Xia, L. Zhang, F. Xu, and F. Fraundorfer. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geoscience and Remote Sensing Magazine*, 2017.

42 Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro. Improving Semantic Segmentation via Video Propagation and Label Relaxation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Serverless GEO Labels for the Semantic Sensor Web

Anika Graupner
Institute for Geoinformatics, University of Münster, Germany
a_grau05@uni-muenster.de

Daniel Nüst [1]
Institute for Geoinformatics, University of Münster, Germany
http://danielnuest.de/
daniel.nuest@uni-muenster.de

Abstract

With the increasing amount of sensor data available online, it is becoming more difficult for users to identify useful datasets. Semantic Web technologies can improve such discovery via meaningful ontologies, but the decision of whether a dataset is suitable remains with the users. Users can be aided in this process through the GEO label, which provides a visual summary of the standardised metadata. However, the GEO label is not yet available for the Semantic Sensor Web. This work presents novel rules for deriving the information for the GEO label's multiple facets, such as user feedback or quality information, based on the Semantic Sensor Network Ontology and related ontologies. Thereby, this work enhances an existing implementation of the GEO label API to generate labels for resources of the Semantic Sensor Web. Further, the prototype is deployed to serverless cloud infrastructures. We find that serverless GEO label generation is capable of handling two evaluation scenarios for concurrent users and burst generation. Nonetheless, more real-world semantic sensor descriptions, an analysis of requirements for GEO label facets specific to the Semantic Sensor Web, and an integration into large-scale discovery platforms are needed.

2012 ACM Subject Classification Information systems → Question answering

Keywords and phrases GEO label, geospatial metadata, data discovery, Semantic Sensor Web, serverless

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.4

Related Version A preprint is published with DOI 10.20944/preprints202002.0326.v2.

Supplementary Material Software, examples, evaluation results, and deployment instructions for the GEO label API implementation are release `0.3.0` of `https://github.com/nuest/GEO-label-java` [22]. The online demo endpoints are `https://glbservice-nvrpuhxwyq-ew.a.run.app/glbservice/api/v1` for Google Cloud Run and `https://6x843uryh9.execute-api.eu-central-1.amazonaws.com/glbservice/api/v1` for AWSLambda. The code for figures and an interactive app to see plots for all test scenarios is at `https://gitlab.com/nuest/geolabel-ssno-paper` and archived on Zenodo (doi:10.5281/zenodo.3908399). The figures app is online at `https://geolabel-ssno-paper.herokuapp.com/`.

Funding *Daniel Nüst* : *Opening Reproducible Research* (`https://o2r.info`); DFG project no. PE 1632/17-1.

Acknowledgements This work is based on the thesis "Ein Metadatenlabel für das semantische Sensorweb" [11]. Contributions (see CRediT) by AG: data curation, investigation, methodology, software, validation, and writing – review & editing; by DN: conceptualisation, software, supervision, writing – original draft, visualisation. We thank Celeste R. Brennecka from the Scientific Editing Service of the University of Münster for her editorial support and the anonymous reviewers for their very helpful comments. The authors declare no competing or conflict of interests.

[1] Corresponding author.
All URLs in this document were last checked on February 19, 2020.

1 Introduction

The amount of sensor data captured and accessible today is ever increasing, not least because of the numerous sensing devices that are part of the Internet of Things, Smart Cities, and the newest Earth observation satellites. The Group on Earth Observation's (GEO) Global Earth Observation System of Systems (GEOSS) [6] has attempted to make near real-time environmental data and processing available to users in the form of a Spatial Data Infrastructure (SDI) based on OGC Web Services and standards (cf. [14]. However, the complexity of the task and the continued growth of sensor data means that this system is, and probably will remain, an evolving work in progress. To improve user recognition of geospatial datasets, to promote trust in datasets, and to assist users in the discovery of suitable datasets, the GEO label was developed [18]. The original GEO label design, as described by Lush [18], comprises several *facets* to convey the most relevant information to users. The label, as a graphical representation for individual datasets in GEOSS, allows users to compare complex metadata as well as objective and subjective quality information to make an informed decision when selecting a dataset [19]. To integrate the GEO label with geospatial catalogues and applications, the generation of the GEO label was encapsulated in a RESTful Web API, the *GEO label API*[2]. This API creates labels in SVG format [8] and other image and machine-readable formats for provided metadata documents or for links to online documents; SVG is the primary format because of its flexibility for scaling and because it enables interactivity, such as pop-ups or links for parts of the image. To transfer the label's usefulness to standardised sensor data, Nüst et al. [24] extended the GEO label for the OGC Sensor Web Enablement [3] (SWE).

However, the GEO label is not used in production today, and its potential for improving sensor data discovery is subsequently untapped. This can be traced back to three gaps, namely a gap in (1) application of the GEO label to additional concepts and implementations for interoperable data exchange, such as the Semantic Web[3], to grow the supported metadata sources and increase coverage, (2) implementation of ready-to-use and scalable platforms for integrating the label into existing services (APIs, portals) and their requirements, and (3) adoption of the label by operators of online sensor data portals.

Regarding the first gap, we propose to extend the GEO label to include metadata from the Semantic Sensor Web (SSW, [25]), which is an important infrastructure for sensor data complementing OGC SWE. Janowicz et al. [14] describe how SDIs can be enhanced with a transparent mapping to the Semantic Web, and the SSW connects the Semantic Web with the OGC SWE suite of standards. The SSW offers a framework for enabling interoperability and meaningful data integration, processing, and reasoning. In the SSW, the metadata captured by ontologies provide a promising source of information for the GEO label, because the information is meaningful and can be drawn from various linked resources. In turn, the GEO label has the potential to improve data discovery in the vastness of geospatial sensor datasets, whereby characteristics such as the lack of a singular inventory or the dynamic structure of sensor webs represent key challenges [16]. Previous work [1, 15, 5] uses specialised ontologies or queries to answer the discovery challenges in OGC SWE and the SSW, but no existing approaches use a visual badge or label for improved data discovery in the Semantic Web.

[2] `https://geolabel.net/api.html`
[3] `https://en.wikipedia.org/wiki/Semantic_Web`

Regarding the second gap, the generation of labels for data portals must serve two different approaches depending on the platform. Demand is either small, intermittent, and unpredictable, if labels are generated on demand with a discontinuous workload depending on users in interactive sessions, or demand is schedulable in isolated but large bulk events, if labels are generated and stored regularly for all available metadata. To serve both scenarios, we propose to deploy the GEO label API to cloud computing infrastructures.

Finally, regarding the third gap, tackling such organisational or strategic issues is out of scope for this work. Nevertheless, closing the former two gaps indirectly helps stakeholders and operators of public or open infrastructures to adopt the label in practice.

The **main contributions** of this work are (1) creating a mapping between metadata fields of the Semantic Sensor Web and the GEO label facets, (2) implementing a prototype of this mapping which conforms to the GEO label API, and (3) evaluating the prototype in serverless computing infrastructures with respect to intermittent and bulk generation of labels. In the remainder of this work, we first identify suitable sources of information in ontologies of the SSW and related ontologies. Then we evaluate existing GEO label API implementations and different cloud computing providers to identify suitable base software and cloud platforms for a prototypical implementation. Finally, we evaluate the prototype's performance. See the *Supplement* section for information about the software prototype, the test data used, and online deployments of the prototype.

2 GEO label for the Semantic Sensor Web

The SSW's main ontology is the *Semantic Sensor Network Ontology* (SSN, [12]). To create a mapping between the SSN and the GEO label, we first evaluated the modular SSN for suitable fields which can provide meaningful information for the different GEO label facets. This evaluation included SSN's core ontology SOSA (Sensor, Observation, Sample and Actuator) and the SSN's aligned modules, such as the Provenance Interchange Ontology (PROV-O) [17]. Each of the over 50 classes and properties of SSN and SOSA and their aligned modules (100+ classes an properties) was checked one by one against the eight facets of the GEO label. Next, we extended the search to include ontologies often used in conjunction with SSN starting from the SSN specification's examples[4]. From those, we adopted generic properties for names and descriptions, e.g., using the Friend of a Friend ontology (FOAF) [4]. Finally, we looked more broadly for ontologies on topics with a relation to until then not covered facets using the Linked Open Vocabularies catalogue[5]. This search lead eventually led to the usage of the Dataset Usage Vocabulary (DUV) [10] and the Bibliographic Reference Ontology (BiRO) [9].

For example, for the facet *Producer Comments* is set to available if a document contains an `rdfs:comment`, because we can assume that such a comment stems from an entity involved in the creation of the metadata record, for the facet *Compliance with Standards*, the mapping checks if one of the used URIs contains `w3.org` and thereby denotes usage of a vocabulary that underwent a development under the auspices of the Word Wide Web Consortium (W3C), while for the mapping *User Feedback* an observation, `sosa:Observation`, must be connected to a `duv:UserFeedback` based on the `duv:hasUserFeedback` property. However, not all mappings are so open respectively direct or simple and allow different options. For example, for the facet *Producer Profile* an SSNO class such as `sosa:Sensor` can be con-

[4] `https://www.w3.org/TR/vocab-ssn/#examples`
[5] `https://lov.linkeddata.es/dataset/lov/`

nected to a `prov:Agent` using either `prov:wasAttributedTo` or `prov:wasAssociatedWith` and the respective PROV subclasses, and for the facet *Lineage Information* any of the relations `ssn:implements`, `ssn:implementedBy`, or `sosa:usedProcedure` can connect a sensing system with its procedure documentation.

Table 1 summarises the result of the manual process and briefly explains the reasoning behind the chosen mapping. See Section 3.2 for the full details on the mapping and the technical realisation. The table shows the ontologies, classes, and properties we identified as suitable sources for the GEO label's facets. The used ontologies and prefixes are listed in Table 2. Note that we did not add new alignments between the SSN and other ontologies, as that is beyond the scope of this work.

3 Serverless GEO label Generation

3.1 Serverless Computing

Serverless computing allows developers to deploy custom code in a shared infrastructure [2], whereby the application is maintained in a scalable way by a platform provider. The automated scaling enables both handling of large spikes of high demand and reducing costs when there is little or no demand. These properties make serverless computing a good fit for the GEO label generation usage scenarios. The GEO label generation can be deployed to a serverless infrastructure quite easily, i.e., without a complex setup including multiple services or a database, because each generation of a label is a relatively small, stateless, atomic operation. The creation of a label externally only relies on the metadata sources for which a label is requested. However, depending on the usage scenario, requests for labels can be erratic and unpredictable. To demonstrate applicability of the prototype the evaluations were conducted within the free tier of the following service providers: Google Cloud Run[6] (GCR) and Amazon Web Services (AWS) Lambda[7]. A comparison of the costs, while relevant for potential operators, is out of scope for this work.

3.2 GEO label API Implementation

The GEO label API is implemented in two software projects, one in Java and one in PHP[8]. In this work, the Java-based implementation is used because PHP is not supported by the serverless computing providers and the PHP project is no longer maintained. The rendering of the GEO label is based on an SVG template file. Labels are generated using the template file according to XPath expressions [7], which detect the presence of certain elements in a provided XML document. To use XPath, the RDF graph must be serialised in RDF/XML. Both implementations support a bespoke JSON-based configuration file format, which allows one to update the rules for transformations of metadata documents to labels without changes to the source code and to deploy these updates to GEO label API instances without updating the installation. To realise the conceptual mapping described above, we created a new transformation file[9]. The file is activated when the implementation is provided as an RDF/XML document, i.e., if the XPath `boolean(/*[local-name()='RDF'])` testing the document's root element evaluates to true. Of note, the implementation of hoverover and drilldown features lies beyond the scope of the proof-of-concept implementation.

[6] `https://cloud.google.com/run/`
[7] `https://aws.amazon.com/lambda/`
[8] `https://geolabel.net/implementations.html`
[9] See transformation file source at `https://github.com/nuest/GEO-label-java/blob/master/server/src/main/resources/transformations/transformerSSN0.json`.

Table 1 GEO label facets' data sources in the Semantic Sensor Web.

Facet	Availability & hoverover text	XPath[1)]
Producer profile	One SSN object (e.g., `sosa:Sensor`) has a responsible `prov:Person` or `prov:Organization` associated using `prov:wasAttributedTo` or `prov:wasAssociatedWith`; hoverover shows name(s) using `foaf:name`.	`//*[rdf:Description[rdf:about= //prov:wasAssociatedWith/rdf:resource or rdf:about= //prov:wasAttributedTo/rdf:resource ]/rdf:type[rdf:resource= "http://www.w3.org/ns/prov#Organization" or .../prov#Person"]]` (partial expression)
Producer comments	A comment provided *within* a sensor description is coming from the producer; hoverover shows count and excerpt of comments.	`//*[rdfs:comment]`
Lineage information	At least one usage of `sosa:Procedure`, either directly, or for an SSN object via one of `ssn:implements`, `ssn:implementedBy`, or `sosa:usedProcedure`; hoverover shows number and type of a procedures' `ssn:Input` and `ssn:Output`	`//*[ssn:implements or ssn:implementedBy or sosa:usedProcedure or sosa:Procedure or rdf:resource= 'http://www.w3.org/ns/sosa/Procedure']`
Standards compliance	If URIs include `w3.org`, at least one ontology is by the W3C as the authority for the Semantic Web.	`//*[contains(*, 'w3.org')]`
Quality information	An `ssn:System` has (a) the property `ssn-system:qualityOfObservation` or (b) at least one property detailing a sensor's capabilities (e.g., accuracy, battery lifetime) identified via the relations `ssn-system:hasSystemProperty`, `ssn-system:hasOperatingProperty`, or `ssn-system:hasSurvivalProperty`; hoverover shows name(s) of available properties.	`//*[ssn-system:hasSystemProperty or ssn-system:hasOperatingProperty or ssn-system:hasSurvivalProperty or ssn-system:qualityOfObservation]`
User feedback	A SSN object has a property `duv:hasUserFeedback` connecting it with a `duv:UserFeedback` or `duv:RatingFeedback`; hoverover shows counts of feedbacks and an statistics of categorical ratings.	`//*[duv:hasFeedback/duv:UserFeedback[ not(prov:qualifiedAssociation)] or duv:hasFeedback/duv:RatingFeedback ]` (partial expression)
Expert reviews	A user feedback (see above) is qualified with a relation `prov:qualifiedAssociation` describing the author's role with `prov:hadRole` as an expert, a role that is not specified within PROV-O but can be the literal `expert`; for hoverover see *User feedback*	`//*[duv:hasFeedback/duv:UserFeedback/ prov:qualifiedAssociation/ prov:Association/prov:hadRole/prov:Role[ contains(rdf:about, 'expert')] ]` (partial expression)
Citations information	An SSN object, e.g., `sosa:Observation`, has a property `biro:isReferencedBy` pointing to a piece of literature; hoverover shows count of references.	`//*[biro:isReferencedBy]`

1) Wrapping `boolean(..)` statements omitted; partial expressions show only one of multiple allowed statements combined with `or`; the full mapping configuration is available at .

Table 2 Used ontologies and vocabularies with their prefixes, and namespaces.

Ontology/vocabulary	Prefix	Namespace
Resource Description Framework	`rdf`	`http://www.w3.org/1999/02/22-rdf-syntax-ns#`
Resource Description Framework Schema	`rdfs`	`http://www.w3.org/2000/01/rdf-schema#`
Semantic Sensor Network Ontology	`ssn`	`http://www.w3.org/ns/ssn/`
Sensor, Observation, Sampling and Actuator	`sosa`	`http://www.w3.org/ns/sosa/`
System Capabilities Module (of SSN)	`ssn-system`	`https://www.w3.org/ns/ssn/systems/`
Provenance Interchange Ontology	`prov`	`https://www.w3.org/ns/prov#`
Friend of a Friend Ontology	`foaf`	`http://xmlns.com/foaf/0.1/`
Dataset Usage Vocabulary	`duv`	`http://www.w3.org/ns/duv#`
Data Catalog Vocabulary	`dcat`	`https://www.w3.org/ns/dcat#`
Bibliographic Reference Ontology	`biro`	`http://purl.org/spar/biro/`
Web Annotation Ontology	`oa`	`http://www.w3.org/ns/oa#`

Table 1 shows excerpts of the XPaths realising the conceptual mapping. The test data[10] was created based on the example data for the SSN vocabulary[11], which was converted to RDF/XML using two online converters for two varying serialisations into RDF/XML. *MyBluemix RDF Validator and Converter*[12] uses `rdf:resource` attributes to define elements at one level (Listings 1), whereas *Easy RDF Converter*[13] uses the class names as XML elements and nests the objects (2). These examples illustrate the reason for the complexity of the XPaths, which allow both options to serialise triples from an RDF graph in RDF/XML.

In GCR, the API can be deployed in a container, which allows one to run the whole GEO label API with the existing Java Servlet[14]. In AWS Lambda, however, the Java Servlet application cannot be run, so a subset of the GEO label API was implemented with a bespoke request handling class[15]. This handler exposes the existing internal methods for generating SVGs based on URLs to metadata documents provided by the API caller. Then, the API, i.e., the request parameters and allowed HTTP methods, are configured in the Amazon API Gateway. Figure 1 shows a GEO label rendered by the prototype implementation developed as part of this work.

3.3 Performance Evaluation

Two usage scenarios were evaluated with an Apache JMeter[16] scripted test plan[17]. For all API queries, the URL of the example RDF serialisation file `MBC_all_factes_available_ip68smartsensor.rdf` hosted on GitHub is passed via the `GET` request query parameter `metadata` to the SVG-generating endpoint of the respective API deployments. The responses

[10] `https://github.com/nuest/GEO-label-java/tree/master/testdata`
[11] `https://www.w3.org/TR/vocab-ssn/#examples`
[12] `http://rdfvalidator.mybluemix.net/`
[13] `http://www.easyrdf.org/converter`
[14] `https://en.wikipedia.org/wiki/Java_servlet`
[15] See code module `lambda`: `https://github.com/nuest/GEO-label-java/tree/master/lambda`
[16] `http://jmeter.apache.org/`
[17] JMeter test plan file `GEO_Label_API.jmx` and the result files are available online at `https://github.com/nuest/GEO-label-java/tree/master/misc/JMeterTests`.

Listing 1 Observation, converted with MyBluemix RDF Validator and Converter.

```
<rdf:Description
  rdf:about="http://example.org/data/iceCore/12#observation">

  <sosa:hasSimpleResult
    rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">
      42
  </sosa:hasSimpleResult>
  <sosa:observedProperty
    rdf:resource="http://example.org/data/iceCore/12#CO2"/>
  <prov:wasAssociatedWith
    rdf:resource="http://example.org/data/Org/exampleOrg"/>
  <rdf:type
    rdf:resource="http://www.w3.org/ns/prov#Activity"/>
  <rdf:type
    rdf:resource="http://www.w3.org/ns/sosa/Observation"/>
</rdf:Description>

<rdf:Description
  rdf:about="http://example.org/data/Org/exampleOrg">

  <foaf:name>Example Organisation</foaf:name>
  <rdf:type rdf:resource="http://www.w3.org/ns/prov#Organization"/>
  <rdf:type rdf:resource="http://www.w3.org/ns/prov#Agent"/>
</rdf:Description>
```

are deemed successful if the HTTP status code is 200 ("OK") and the content type is `image/svg+xml`. The test plan results for all conducted tests are published in the software repository[18]. An interactive app allows to inspect the plots of the results for all conducted tests, including ones not included in this article[19].

For both serverless computing providers, the default configurations were used for the evaluations. GCR allows users to configure the number of containers, the number of parallel requests handled by one container, and the required minimum response time. The GCR deployment used zone `europe-west1` with 256 Mebibyte working memory and 1 CPU, at a concurrency setting of 80. AWS Lambda starts more instances of a Lambda function as needed, limited by a configurable concurrency parameter (default value: 1000) for the number of running functions in the used region `eu-central-1`. The working memory on AWS is set to 1 Gibibyte with the default values for scaling[20].

Scenario A simulates a geospatial catalogue service with 1000 users whose browsing of the catalogue user interface results in 1 request per second per user. Figures 2 and 3 show the response times during the test execution for GCR and AWS Lambda, respectively[21]. All sent requests have a non-failure status code (`HTTP 200`). The two different colours in the plots denote the requests that take less than ("Success") or longer than ("Failure") 1 second. This threshold is used because interactions below one second were found to not interrupt a user's train of thought and are therefore suitable for interactive use [20]. The mean times to complete the request are 414 seconds for GCR and 943 seconds for AWS Lambda.

[18] JMeter test plan file `GEO_Label_API.jmx` and the result files are available online at `https://github.com/nuest/GEO-label-java/tree/master/misc/JMeterTests`.
[19] `https://geolabel-ssno-paper.herokuapp.com/`
[20] `https://docs.aws.amazon.com/lambda/latest/dg/scaling.html`
[21] Data loading and plot functions are based on code from the R package `loadtest` [21].

Figure 1 GEO label based on SSW sensor metadata, rendered by the GCR deployment, with all eight facets fulfilled; source URL: https://glbservice-nvrpuhxwyq-ew.a.run.app/glbservice/api/v1/svg?metadata=https://raw.githubusercontent.com/nuest/GEO-label-java/master/server/src/test/resources/ssno/ERC_all_factes_available_ip68smartsensor.rdf.

Figure 2 Plots of response time for GCR deployment under scenario A: elapsed time to complete requests (left); histrogram with distribution of response times (right); result data file: `GCR_Scenario_2_V1`.

Listing 2 Observation, converted with Easy RDF Converter.

```
<sosa:Observation
  rdf:about="http://example.org/data/iceCore/12#observation">

  <rdf:type rdf:resource="http://www.w3.org/ns/prov#Activity"/>
  <prov:wasAssociatedWith>
    <prov:Agent
      rdf:about="http://example.org/data/Org/exampleOrg">

      <rdf:type
        rdf:resource="http://www.w3.org/ns/prov#Organization"/>
      <foaf:name>Example Organization</foaf:name>
    </prov:Agent>
  </prov:wasAssociatedWith>
  <sosa:observedProperty>
    <rdf:Description
      rdf:about="http://example.org/data/iceCore/12#CO2">
      <ssn:isPropertyOf
        rdf:resource="http://example.org/data/iceCore/12"/>
    </rdf:Description>
  </sosa:observedProperty>

  <sosa:hasSimpleResult
    rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">
      42

</sosa:Observation>
```

Scenario B tests the batch generation of labels where an operator of a sensor catalogue wants to generate 100 labels at once. Here we measure the overall time for processing all requests, and the operations were repeated 5 times. There is no threshold as in Scenario A. For GCR, this led to failures due to the memory limit; but, the test was completed with a memory of 1 Gibibyte and 2 CPUs per container instance. The resulting data is shown in Figure 4. GCR's need for additional resources can be traced back to an overhead of the full Java Servlet, which the Lambda function handler, which is comparably more minimal, does not suffer from. With the increased resources in GCR, the duration was up to 45 seconds for the first run and decreased though to only about 3 seconds for the fifth repetition. For AWS Lambda, the processing took about 8 seconds on the first run and dropped to around 1 second in the second to fifth repetitions, as shown in Figure 5.

A variant of the batch generation is a test scenario with 1000 parallel requests. This scenario could not be completed by either platform with the maximum available hardware configurations. The error messages (`Connection reset` and `SSL handshake terminated`) hint that the services blocked the large number of parallel requests, such that users would need more powerful (and more costly) deployments. Reducing the number of parallel requests eventually led to successful scenario executions at 600 requests in 51 seconds for GCR and 300 requests in 9 seconds for AWS Lambda (see data files `GCR_Scenario_4_2_V3` and `AWS_Scenario_4_2_V4`).

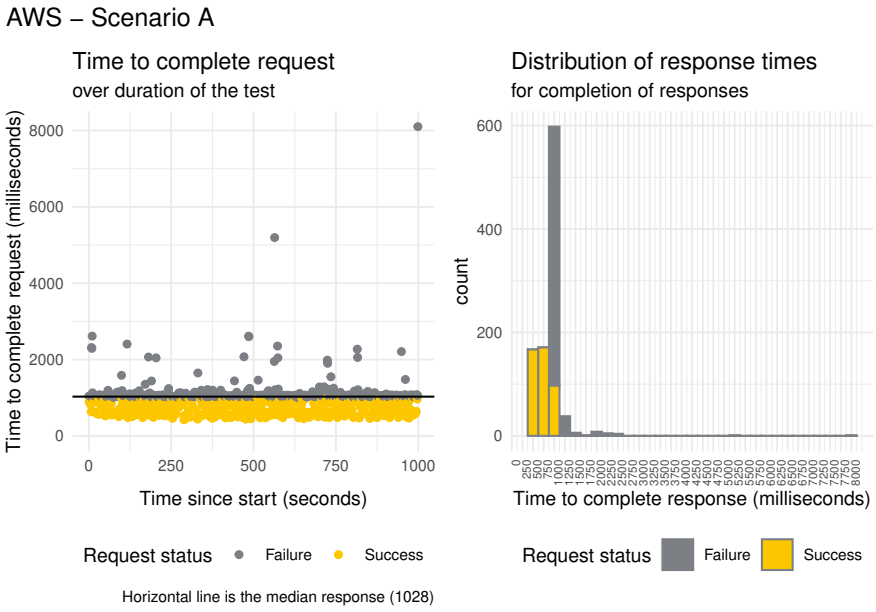


Figure 3 Plots of response time for AWS deployment in scenario A: elapsed time to complete requests (left); histrogram with distribution of response times (right); result data file: `AWS_Scenario_2_V1`.

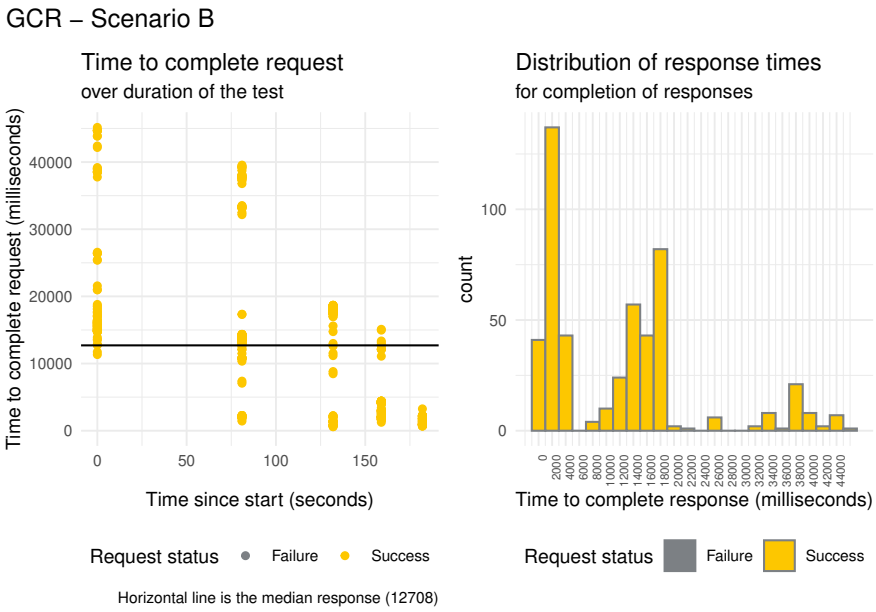


Figure 4 Plots of response time for GCR deployment in scenario B: elapsed time to complete requests (left); histrogram with distribution of response times (right; please note the different bin size compared to other plots); result data file: `GCR_Scenario_3_V2`.

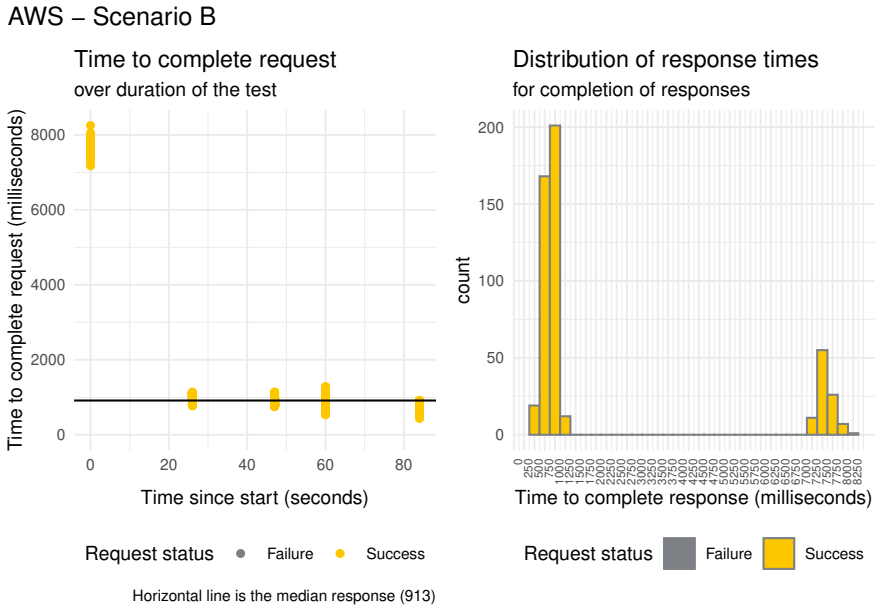


Figure 5 Plots of response time for AWS deployment in scenario A: elapsed time to complete requests (left); histrogram with distribution of response times (right); result data file: `AWS_Scenario_3_V2`.

4 Discussion

The **mapping** of GEO label facets to properties in the Semantic Sensor Web was an iterative process. While we were able to find data sources for all GEO label facets, the mapping is limited by the availability of realistic SSW datasets. First, the variability of real-world data may not be adequately captured. Second, and the nature of the mapping does not capture cases where concepts between the GEO label's facets do not unambiguously match concepts behind SSW elements. Compared to the centrally managed data sources and industry-driven OGC standards of the original GEO label, we find no need to make distinctions between metadata given by providers and by third-parties, e.g., commenting servers. However, such multi-stakeholder perspectives could mitigate shortcomings in the creation process of the GEO label mapping for SSW. More real-world metadata could improve the scope of the facet data sources, e.g., by deriving from common practices if a comment is actually about a relevant part of a sensor's properties, and not about some less relevant part of the RDF graph. The taken iterative, example-based approach could also be contrasted with the initial creation of an ontology for the GEO label facets and then aligning the GEO label ontology with existing (SSW) ontologies. The alignment-based approach could also improve the scalability of the mapping for a larger variety of uses cases and SSW datasets.

Concerning the mapping's **implementation**, we found that a document-based approach using RDF/XML could be built quickly on the existing implementation. However, such an approach does not leverage the power of the Semantic Web, e.g., reasoning on dynamically built graphs and aligned ontologies. Furthermore, for some facets, e.g., producer profile, there were clear complications of relying on serialisations into an RDF/XML document. A different approach based on native Semantic Web technologies, such as SPARQL [13], could help address the limited coverage of the presented mapping, and in addition take

into account relationships between linked resources by, for example, measuring the distance between connected resources in a graph. The GEO label's option to have "half-filled" facets, which denotes availability of information at a higher level, could expose such more complex scenarios. Most critically, the presented approach is limited by the design process starting only from the current GEO label facets. That is why specific discovery challenges of the SSW may not be adequately addressed. While the label itself may be interactive, the majority of information behind the label is seen as rather static. This may partly be attributed to the GEO label's origin in GEO, with more traditional roles of provider and user. The SSW's potentially very dynamic nature, for examples live data streams, and flexible distributed architecture, in which anybody can create and publish new ontologies and datasets, may require additional facets or a more sophisticated presentation of sources and currentness of the data behind a label.

The **evaluation** results of Scenario A show show no discernible cold start effect, as one might have expected, where resources need to be activated for the first request or additional resources are added over time. Only few requests take over 1 second to complete and only relatively few outliers exists on the same order of magnitude. For AWS Lambda, both mean and median of elapsed time to complete a request are close to 1 second. For GCR, the elapsed time is well below 0.5 seconds. These results imply that the serverless label generation is suitable for interactive use, with slight advantages of GCR which has overall shorter durations. A limitation for this scenario is that only the generation of the label is tested, whereas for users additional time would be taken up by the client-side rendering of the images. The effects of dropping durations for batch processing in Scenario B were likely achieved by a combination of autoscaling in the underlying platforms and the built-in caching of the GEO label API Java Servlet. Especially on AWS Lambda, the drop after the first iteration is considerable, even tough no internal caching mechanism exists.

Regarding the platforms we used, the data might further point to an advantage for the reduced implementation of the AWS Lambda functions compared to the full Java Servlet running in containers on GCR, though both showed scaling mechanisms of serverless computing to be effective. The results make clear that specific evaluations for each use case and platform are warranted. More test scenarios could include varying allocated resources at the cloud providers to optimise performance versus costs, touching both on the respective configuration parameters and the client-side implementation. For example, the steep drop in Scenario B on AWS Lambda could be used to warm up a service instance, which may have relatively small resources, with a portion of the data for batch processing, and then following up with several chunks afterwards. In contrast, the shorter average response times of GCR may make it more suitable for a scenario with more constant load even if fewer resources are allocated.

5 Conclusions

In this study, we transferred the goal of the GEO label, which is to improve data discovery by providing a visual overview of available information in machine-readable metadata, to the Semantic Sensor Web. While we were able to find data sources for all GEO label facets using a document-centric approach, the mapping is limited by available datasets and does not leverage the potential of using reasoning in the SSW. Ideally, the creation of a more sustainable mapping and potentially even adaptation of GEO label facets in the future is based on a larger body of public sensor metadata in SSNO format, on a consultation of multiple stakeholders, and on a complementary perspective derived from the SSW's discovery challenges.

We found that the serverless platforms proved suitable for realistic test scenarios, though, naturally, the used free tiers have limits. It became also clear that the different cost models and configurations make serverless solutions difficult to compare. Future evaluations may utilise a strictly cost-based comparison of scenarios with resources tuned to deliver similar performance in the user-facing API.

Finally, the usefulness of the GEO label remains to be demonstrated in broad deployments with many users and extensive user studies. With the practical solutions for label generation introduced in this work, the actual spreading of labels will require leading organisations to add and maintain labels on their widely used geospatial catalogues. In the meantime, a bottom-up approach with client-side label integration [23] could provide the benefits of GEO labels to interested users, and the GEO label can be examined in relation to recent developments on scientific data publication such as the FAIR Guiding Principles [26].

References

1 Grigori Babitski, Simon Bergweiler, Jörg Hoffmann, Daniel Schön, Christoph Stasch, and Alexander C. Walkowski. Ontology-Based Integration of Sensor Web Services in Disaster Management. In Krzysztof Janowicz, Martin Raubal, and Sergei Levashkin, editors, *GeoSpatial Semantics*, Lecture Notes in Computer Science, pages 103–121, Berlin, Heidelberg, 2009. Springer. doi:10.1007/978-3-642-10436-7_7.

2 Ioana Baldini, Paul Castro, Kerry Chang, Perry Cheng, Stephen Fink, Vatche Ishakian, Nick Mitchell, Vinod Muthusamy, Rodric Rabbah, Aleksander Slominski, and Philippe Suter. Serverless Computing: Current Trends and Open Problems. In Sanjay Chaudhary, Gaurav Somani, and Rajkumar Buyya, editors, *Research Advances in Cloud Computing*, pages 1–20. Springer, Singapore, 2017. doi:10.1007/978-981-10-5026-8_1.

3 Mike Botts, George Percivall, Carl Reed, and John Davidson. OGC® sensor web enablement: Overview and high level architecture. In *GeoSensor Networks*, pages 175–190. Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-79996-2_10.

4 Dan Brickley and Libby Miller. FOAF Vocabulary Specification. Technical report, World Wide Web Consortium, January 2014. URL: http://xmlns.com/foaf/spec/.

5 Jean-Paul Calbimonte, Hoyoung Jeung, Oscar Corcho, and Karl Aberer. Semantic sensor data search in a large-scale federated sensor network. In *Proceedings of the 4th international workshop on semantic sensor networks*, volume 839 of *CEUR Workshop Proceedings*, pages 23–38, Bonn, Germany, 2011. URL: http://ceur-ws.org/Vol-839/calbimonte.pdf.

6 Eliot J. Christian. GEOSS Architecture Principles and the GEOSS Clearinghouse. *IEEE Systems Journal*, 2(3):333–337, September 2008. Conference Name: IEEE Systems Journal. doi:10.1109/JSYST.2008.925977.

7 J. Clark and S. DeRose. XML Path Language (XPath), Version 1.0. W3C Recommendation, World Wide Web Consortium, November 1999. URL: http://www.w3.org/TR/xpath.

8 Erik Dahlström, Patrick Dengler, Anthony Grasso, Chris Lilley, Cameron McCormack, Doug Schepers, and Jonathan Watt. Scalable Vector Graphics (SVG) 1.1 (Second Edition). W3C Recommendation, World Wide Web Consortium, 2011. URL: http://www.w3.org/TR/SVG/.

9 Angelo Di Iorio, Andrea Nuzzolese, Silvio Peroni, David Shotton, and Fabio Vitali. Describing bibliographic references in RDF. In Alexander Garc{\'i}a Castro, Christoph Lange, Phillip Lord, and Robert Stevens, editors, *4th Workshop on Semantic Publishing (SePublica 2014)*, volume 1155 of *CEUR Workshop Proceedings*, Anissaras, Greece, May 2014. URL: http://ceur-ws.org/Vol-1155/paper-05.pdf.

10 Bernadette Farias Lóscio, Eric G. Stephan, and Sumit Purohit. Data on the Web Best Practices: Dataset Usage Vocabulary. Technical report, World Wide Web Consortium, 2016. URL: https://www.w3.org/TR/vocab-duv/.

11 Anika Graupner. Ein Metadatenlabel für das semantische Sensorweb. *Institute for Geoinformatics (ifgi) BSc Thesis*, April 2020. doi:10.31237/osf.io/fs48a.

12 Armin Haller, Krzysztof Janowicz, Simon J. D. Cox, Maxime Lefrançois, Kerry Taylor, Danh Le Phuoc, Joshua Lieberman, Raúl García-Castro, Rob Atkinson, and Claus Stadler. The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation. *Semantic Web*, 10(1):9–32, January 2019. doi:10.3233/SW-180320.

13 Steve Harris and Andy Seaborne. SPARQL 1.1 Query Language. W3C Recommendation, World Wide Web Consortium, March 2013. URL: https://www.w3.org/TR/sparql11-query/.

14 Krzysztof Janowicz, Sven Schade, Arne Bröring, Carsten Keßler, Patrick Maué, and Christoph Stasch. Semantic Enablement for Spatial Data Infrastructures. *Transactions in GIS*, 14(2):111–129, 2010. doi:10.1111/j.1467-9671.2010.01186.x.

15 Hoyoung Jeung, Sofiane Sarni, Ioannis Paparrizos, Saket Sathe, Karl Aberer, Nicholas Dawes, Thanasis G. Papaioannou, and Michael Lehning. Effective Metadata Management in Federated Sensor Networks. In *2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pages 107–114, June 2010. doi:10.1109/SUTC.2010.29.

16 Simon Jirka, Arne Bröring, and Christoph Stasch. Discovery Mechanisms for the Sensor Web. *Sensors*, 9(4):2661–2681, April 2009. doi:10.3390/s90402661.

17 Timothy Lebo, Satya Sahoo, and Deborah McGuinness. PROV-O: The PROV Ontology. Technical report, World Wide Web Consortium, April 2013. URL: https://www.w3.org/TR/prov-o/.

18 Victoria Lush. *Visualisation of quality information for geospatial and remote sensing data: providing the GIS community with the decision support tools for geospatial dataset quality evaluation.* PhD thesis, Aston University, 2015. URL: https://research.aston.ac.uk/en/studentTheses/visualisation-of-quality-information-for-geospatial-and-remote-se.

19 Victoria Lush, Lucy Bastin, and Jo Lumsden. Developing a geo label: providing the gis community with quality metadata visualisation tools. *Proceedings of the 21st GIS Research UK (GISRUK 3013), Liverpool, UK*, pages 3–5, 2013. URL: https://www.geos.ed.ac.uk/~gisteac/proceedingsonline/GISRUK2013/gisruk2013_submission_44.pdf.

20 Jakob Nielsen. Response Times: The 3 Important Limits, January 1993. URL: https://www.nngroup.com/articles/response-times-3-important-limits/.

21 Jacqueline Nolis. *loadtest: HTTP load testing directly from R*, 2020. R package version 0.1.2. URL: https://github.com/tmobile/loadtest.

22 Daniel Nüst and Anika Graupner. nuest/GEO-label-java: Release 0.3.0, February 2020. doi:10.5281/zenodo.3673870.

23 Daniel Nüst, Lukas Lohoff, Lasse Einfeldt, Nimrod Gavish, Marlena Götza, Shahzeib Tariq Jaswal, Salman Khalid, Laura Meierkort, Matthias Mohr, Clara Rendel, and Antonia van Eek. Guerrilla Badges for Reproducible Geospatial Data Science. In *AGILE Conference 2019 Short Papers*, Limassol, Cyprus, 2019. AGILE. doi:10.31223/osf.io/xtsqh.

24 Daniel Nüst and Victoria Lush. A GEO label for the Sensor Web. In *AGILE Conference 2015 Short Papers*, Lisbon, Portugal, 2015. AGILE. doi:10.31223/osf.io/ka38z.

25 Amit Sheth, Cory Henson, and Satya S. Sahoo. Semantic Sensor Web. *IEEE Internet Computing*, 12(4):78–83, July 2008 . doi:10.1109/MIC.2008.87.

26 Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018, March 2016. doi:10.1038/sdata.2016.18.

Search Facets and Ranking in Geospatial Dataset Search

Thomas Hervey[1]
Department of Geography, University of California, Santa Barbara, CA, USA
Center for Spatial Studies, University of California, Santa Barbara, CA, USA
http://www.spatial.ucsb.edu
thomasahervey@ucsb.edu

Sara Lafia
Department of Geography, University of California, Santa Barbara, CA, USA
Center for Spatial Studies, University of California, Santa Barbara, CA, USA
http://www.spatial.ucsb.edu
slafia@ucsb.edu

Werner Kuhn
Department of Geography, University of California, Santa Barbara, CA, USA
Center for Spatial Studies, University of California, Santa Barbara, CA, USA
http://www.spatial.ucsb.edu
werner@ucsb.edu

Abstract

This study surveys the state of search on open geospatial data portals. We seek to understand 1) what users are able to control when searching for geospatial data, 2) how these portals process and interpret a user's query, and 3) if and how user query reformulations alter search results. We find that most users initiate a search using a text input and several pre-created facets (such as a filter for tags or format). Some portals supply a map-view of data or topic explorers. To process and interpret queries, most portals use a vertical full-text search engine like Apache Solr to query data from a content-management system like CKAN. When processing queries, most portals initially filter results and then rank the remaining results using a common keyword frequency relevance metric (e.g., TF-IDF). Some portals use query expansion. We identify and discuss several recurring usability constraints across portals. For example, users are typically only given text lists to interact with search results. Furthermore, ranking is rarely extended beyond syntactic comparison of keyword similarity. We discuss several avenues for improving search for geospatial data including alternative interfaces and query processing pipelines.

2012 ACM Subject Classification Information systems → Environment-specific retrieval; Human-centered computing → Interactive systems and tools; Information systems → Retrieval effectiveness

Keywords and phrases search, portal, discovery, GIR, facet, relevance, ranking

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.5

1 Introduction

It is hard to overemphasize the value of open geospatial portals. In less than a decade, a new generation of Digital Earth [10] has unfolded as thousands of municipalities and other data stewards have created online open geospatial portals, turning voluminous isolated geospatial data into provisioned public resources. Every day, these data are used by citizens to learn about their community services and researchers to study their environments [26]. Some open geospatial portals, herein referred to as *portals*, are small, serving specific datasets from niche domains like soil science (e.g., TERENO), while others are broad, serving as aggregation platforms for datasets across many levels of government (e.g., Data.gov).

[1] corresponding author

While portals have been widely adopted, there is a need for more research to evaluate the state of spatially-enabled search across portals that curate scientific, library, and governmental data. Furthermore, there is a need to bring these communities of practice into conversation so that best practices from each can be shared.

In this paper, we review the state of search on open geospatial portals. We focus on how the design of search has developed across communities of practice for the curation of geospatial research data, civic data, and library data. To the best of our knowledge, there has not been a comprehensive examination of portal search functionality across user communities. This is problematic because these portals collect and disseminate geospatial data that is vital to governance and research. The geographic information science community needs to know what search functionality already exists to better inform future developments of geospatial search and geographic information retrieval (GIR).

This work specifically examines what a user can control while searching and how a portal processes and interprets user queries. We focus on search *facets* and *ranking functions*. As described by [16], we use the term facet to broadly mean a search control that allows a user to further specify a query. Facets let users filter out search results, add criteria for including specific search results, sort results, and sometimes navigate results in a specific way. A ranking function orders results by their relevance to a query. For example, a function could use TF-IDF (term frequency-inverse document frequency), a common numerical statistic comparing keyword frequencies between a query and a set of potential search results. By investigating these two components—facets and ranking functions—we can identify gaps in functionality and search effectiveness. Therefore, our research question is *what is the state of faceted search and ranking functions in open geospatial portals?*

To answer this question, we survey several dozen open portals including Data.gov, DataONE, UCSB's Alexandria Digital Library[2], and ArcGIS Hub [3], and examine the front-end and back-end functionality of their search engines. We first survey search facets and record user search controls, interaction modalities, presentation of results, and navigation. Second, we attempt to record how a portal processes and interprets a user query. Specifically, we record how corresponding data are deemed relevant and ranked. This step proves to be a formidable challenge since many portals do not publicly document how they process and reason on queries. Third, we conduct a qualitative sensitivity analysis of search. We construct several search scenarios, execute corresponding queries, and compare the search results both between portals and after reformulating queries.

In Section 2 we discuss previous work to survey portals and measure their functionality. We then describe our procedure for surveying portals in Section 3. Section 4 is a discussion of our results, including a summary of user search facets, ranking functions, alternative query results, and insights on noteworthy portals. We conclude with a discussion of shortcomings in the functionality of current portals and possible avenues for improving portal effectiveness.

2 Background

Portals curate, organize, and disseminate data for public consumption, often through public web applications[36]. In our work, we study portals on the web[4] that leverage an ecosystem of web services for storing, querying, rendering, mapping, and executing functions (like

[2] https://www.alexandria.ucsb.edu
[3] https://hub.arcgis.com/search
[4] https://portal.ogc.org/files/?artifact_id=6669

geocoding) against geospatial data [23]. In many cases, these portals are an extension of enterprise geographic information systems (e.g., ArcGIS Online) or are integral components of governmental Spatial Data Infrastructures (SDIs) (e.g., INSPIRE).

Many portals strive to uphold FAIR and open data[5] principles: effective and efficient data findability, accessibility, interoperability, and reusability [34, 36]. FAIR principles mark a significant turn for data curation, as they make it possible to evaluate data quality (e.g., fitness for use) across application domains [8]. These principles also influence how effectively a user can search for data. Despite the kind of curatorial "best practices" that FAIR principles suggest, user experiences within portals still vary quite a bit. Simple search interactions across different portals show that search differs noticeably. For example, a search experience for scientific research data is very different from a search for library holdings; there is still relatively little agreement on a similar set of "best practices" for spatial search [3]. Each of these communities has developed distinct metadata standards (e.g., Ecological Metadata Language[6] in the bioinformatics community), adopted different data curation platforms (e.g., DataONE[7]), and anticipated different user needs (e.g., search by taxon facets are in the bioinformatics community). Furthermore, it remains unclear how the spatial dimension of search should interact with other search dimensions, like time and theme, to ensure that a user can search for data effectively.

Search on a portal generally proceeds as follows: through a user interface (UI) a user executes a query, the portal processes and interprets the query (often using a service to georeference a query), and then the system returns relevant resources from a back-end content management system (CMS) or a database management system (DBMS) that houses indexed geospatial data and metadata. The user can then explore the results and either download a resource or reformulate their query to change results. On the back-end of a portal, a processing algorithm is used to rank results. This algorithm, which the ranking function is a part of, influences both UI design and how the system interprets a query. For example, a processing pipeline may follow precreated and hand-tuned rules for filtering query results (based on corresponding UI facets a user adjusts on a portal's interface). There may be other processing steps that a user cannot control (such as text normalization).

Modern portals are the primary outlet for search and discovery of geospatial data. Before portals, geospatial data were collected and curated separately by individual organizations. Both governments and GIS companies realized that they needed a better way to make geospatial data discoverable, usable, and interoperable [32, 13]. As web services for sharing data rose, the GIS industry became interested in using them as a medium for bringing both GIS and geospatial data to a larger audience. These services were the first way to get data to the public, but data quality was low and interoperability proved challenging to achieve. The digital earth and SDI movements allowed governments and organizations to centralize and build a technical infrastructure for managing geospatial data [24]. One such example is the U.S. Federal Geographic Data (FGDC) National Spatial Data Infrastructure[8].

The desire to leverage citizen-generated data moved SDIs in a new direction [14, 35]. Portals complemented SDIs, allowing for publication and aggregation of disparate geospatial data [9, 27, 35]. This made portals a popular and accessible form of a distributed GIS [30] yet publishing geospatial content remained technically difficult. Portal architecture began to

[5] We use the definitions for "open knowledge" and "open data" provided by the Open Knowledge Foundation at `https://okfn.org/opendata/`

[6] `https://eml.ecoinformatics.org/`

[7] `https://www.dataone.org/`

[8] `https://www.fgdc.gov/nsdi/nsdi.html`

evolved to adopt search capabilities. Today, portals disseminate and allow users to explore geospatial data. Some are "mashups" built on services such as Google Search or Google Maps for geocoding and OpenStreetMap for visualizing search results [12, 9].

Recent surveys of government-run open data portals across Australia [31] and the U.S. [33] noted that a large portion of portal growth is driven by governments who seek transparency and want to engage citizens in government initiatives. In Australia, for example, dozens of small and large government portals are successful, because they continually publish datasets, refine and clarify open data policies [7], and increase visibility through citizen engagement events like government-sponsored hackathons. Although these surveys are informative, their authors suggest that their survey methods are preliminary. Portal adoption across government, research, and libraries has been rapid in the last few years, so general measures for portal functionality, quality, and effectiveness are still in their infancy. Viewport-based GIR systems have been proposed to support comparison based on the semantic similarity of their features; however, such systems do not yet support realistic information needs [4].

Most current geospatial search challenges and opportunities are described in [29] including novel opportunities like personalized search and interpreting local intent [1], intelligent ranking algorithms based on machine-learned feature combinations [18], and challenges like cataloging cross-disciplinary geospatial search needs or bolstering theory-driven geoparsing methods. Some insights into how portals process and interpret user queries are available from the perspective of portal developers. Search and discovery scenarios in the library community are well illustrated by GeoBlacklight developers [15] and the Alexandria Digital Research Library Project [11]. Advances from the research data community are illustrated by the adoption of standards, such as FAIR data principles [34] and by the examination of challenges in developing domain repositories [25]. Lastly, the adoption of civic data portals across multiple levels of government in the U.S. and E.U. [36] and ArcGIS Online as an open data platform [20] illustrate how user data needs are anticipated and handled.

3 Survey Methods

In this work, we seek to understand three things about portals: 1) what facets a user can control that affect search results, 2) how a portal processes and interprets a user's query (for ranking results), and 3) if and how reformulating a query changes search results. By answering these questions, we gain an understanding of the current capabilities and limitations when a user searches on a portal.

We specifically surveyed 1) search facets, 2) interaction modalities (e.g., maps and text list views), 3) adherence to FAIR principles, and 4) ranking functions (e.g., BM-25, TF-IDF). Note that our methodology uses an individual search and judgement process run solely by the authors. For example, relevance judgements for search results and the adherence to FAIR principles were qualitative judgements. However, the interpretation of FAIR principles remains largely subjective and open to interpretation; to address this, FAIR metrics [34] including rubrics for tools, datasets, and repositories[9] are currently under development. The metrics that are being developed for repositories focus mainly on licensing, protocols, and resource description. Following this, we gave a portal a satisfactory rating for adherence to FAIR principles if its datasets followed at least three of the four main principles–findable, accessible, interoperable, resuable. We gave a portal a higher rating if its datasets followed all four principles, metadata were well documented, and the portal supported to its users,

[9] `https://fairshake.cloud/rubric/`

such as through blog posts on how to use and manipulate dataset metadata. Due to time constraints, we were not able to gather subjects and pool a larger set of judgements. However, in future work we plan to conduct A/B testing between a control and modified search system during which we will gather more judgements from test subjects.

To start, we created a list of sample portals to survey. We hand selected our sample from across three main communities that curate open geospatial portals: 1) civic data portals, 2) scientific research portals, and 3) library portals. These were drawn from four online sources that curate a list of portals, GIS data sources, and GIS learning resources (e.g., Awesome-GIS[10], Awesome-Geospatial[11]). We briefly visited and tested all of the portals on these curated lists and narrowed our list to 35 sample portals. In the remainder of this paper, we will discuss the results from nine unique and diverse portals.

A portal was considered for our list if it: 1) hosts 50 or more open geospatial datasets, 2) has datasets published within the past six months, and 3) provides a way for users to search for datasets. We wanted to achieve broad diversity in our sample. Therefore, we ended up surveying 35 portals that differ in purpose, topic, geographic coverage, or curating body. Two examples of purpose are citizen engagement and academic data reuse. Examples of curating bodies include governments and municipalities, libraries, non-government organizations, and academic institutions. Portals in our list needed to serve georeferenced datasets in formats like .geojson, .shp, .TIFF, or .netCDF (so that can be used in traditional GIS or spatial analysis tasks). For this reason, we intentionally did not survey gazetteers and point of interest (POI) search tools like the World Historic Gazetteer[12], Frankenplace[13], or Yelp[14].

For each portal in our list, we initially took a "follow-your-nose" approach to surveying. This means that when we arrived at the root of a site, we would read the home page and begin exploring by clicking on prominent links. We then reviewed any available documentation for users and developers. Documentation is also useful for understanding how curators articulate the purpose and suggested usage of a portal. When available, we also read open data policies, search and interface user guides, and technology and metadata descriptions.

We then tested a portal's search interface. We first navigated to the root search page (which sometimes was on the portal's home page). Once there, we recorded all the options that a user could control, which included the mode of interaction (e.g., map-based, list-based), navigation (e.g., number of pages, page hierarchy), and search facets (e.g., text search box, filters, map controls, sorting).

Next, we documented portal ranking functions to the extent possible. This was difficult because many systems use proprietary and/or closed-source search engines that do not disclose their ranking functions. Some portals have an application programming interface (API), to bypass the UI and access dataset metadata directly. In some cases, API documentation gave insights into how a query is parameterized and how results are ranked. In other cases, we were able to read documentation from portals that use open-source CMSs, such as CKAN, and a few portals gave us access to internal documentation on their ranking functions.

Our last step was to see how effective and sensitive search is on these portals. The purpose of this step was to 1) try and bolster our understanding of how non-disclosed ranking functions work, and 2) test how sensitive ranking functions are to changes in a user's query.

[10] `https://github.com/sshuair/awesome-gis`
[11] `https://github.com/sacridini/Awesome-Geospatial`
[12] `http://whgazetteer.org/`
[13] `http://www.frankenplace.com/`
[14] `https://www.yelp.com/`

To do this we, selected nine portals from our list on which to execute queries (listed in Table 1). Once we were familiar with their search pages and what specific datasets were available, we developed several search scenarios.

Table 1 Descriptive characteristics of a subset of surveyed portals. For each portal, we recorded the number of public datasets/cataloged items, the temporal range of datasets (starting from either the application time or the creation time), and the coverage (geographic and community focus).

Portal	Datasets	Time	Coverage
DataONE	820k +	1800 - present	global (environmental science)
Data.gov	250k +	mid-1800s - present	U.S. (none, authoritative)
ArcGIS Hub	178k +	1700s - present	global (none, semi-authoritative)
USGS	100k +	2000 - present	U.S. (none, authoritative)
ADRL	33k +	1860 - 2018	California, misc. (library data)
Tereno	1000 +	1995 - present	Germany (environmental science)
INSPIRE	6500 +	1900 - present	Western Europe (none, authoritative)
NASA	6600 +	1587 - present	global (Earth observations)
Heritage Gateway	60+ sources	prehistoric - present	England (structures and landmarks)

These scenarios were modified from three personas of application end users that are described in the GeoBlacklight concept design[15]; for a better understanding of our search scenarios, we refer readers to their descriptions [15]. These personas include a professor of History, a Ph.D. candidate in Environmental Science, and an undergraduate sophomore studying urban planning. Each persona has a motivation, scenario, and expectations of a portal. Although they are exclusively academic, they vary enough to realistically resemble search scenarios from other personas. Based on our interpretation of the persona descriptions, we created a specific search task and respective query to simulate that persona initiating a search. An example of a search scenario is as follows. History professor Brian Diaz needs data about historical and modern churches in Scotland. He does not have a lot of time and he likes using text search, but would also be happy narrowing results using a map. He searches two portals, Heritage Gateway and the INSPIRE Geoportal. He executes and refines the text of several queries without any additional facet adjustments. Example refinements include “churches”, “modern churches”, and “modern churches edinburgh”. When he cannot find relevant results, he tries modifying his queries (using reformulation techniques outlined in Table 2).

After the scenarios were created, we executed an initial query for each, recording the resulting datasets and the number of datasets we considered to be relevant to our search needs. We then iteratively reformulated the query 12 times, and re-recorded the results and portion of the results that we considered to be relevant. We repeated this process for each scenario on each portal with three different initial queries. Table 2 shows the types of query reformulations we used, which were gathered from [19, 21, 22].

4 Results

Some portals are small and have few datasets. For example, TERENO serves soil and geochemical datasets from a few environmental research observatories in Germany. Some portals are curated by small municipalities such as Mono County, California, U.S. Others are

[15] https://geoblacklight.org/documents/GeoBlacklight%20Concept%20Design%20v0.3.3.pdf

Table 2 Types of query reformulations executed during search scenarios. Reformulation types adapted from definitions found in [19, 22, 21].

Query	Reformulation	Type	Purpose
"modern churches"	"churches"	generalization	broaden
"churches"	"modern churches"	specialization	narrow
"modern churches"	"modern temples"	word substitution	change meaning
"modern churches"	"churches modern"	repeat	reformat
"modern churches"	"catholic cemeteries"	new	change meaning
"edinburgh"	"glasglow"	geo-modification	intent modification
"edinburg"	"edinburgh"	geo-correction	correct spelling
"edinburgh"	"edinburgh tx"	geo-disambiguation	placename disambiguation
"churches"	"churches edinburgh"	place insertion	narrow geographically
"churches edinburgh"	"churches"	place deletion	broaden
"food Scotland"	"food Europe"	granularity change	broaden or narrow

large with sophisticated search tools and have many datasets. For example, ArcGIS Hub serves many datasets of widely varying topics and global coverage. We strived for diversity and wanted to ensure that we were not just sampling popular portals. Furthermore, we believed that smaller portals may have more specific user controls since they likely wouldn't have to manage a large amount of diverse datasets. Several smaller portals included those run by The Nature Conservancy[16], Lithuania's federal government[17], and Cyprus's Department of Land and Survey[18].

As mentioned in Section 3, we sampled 35 portals from the lists and will now discuss nine in particular. These nine were chosen because they capture the diversity of all portals sampled. Table 1 includes descriptive characteristics of these nine portals including number of public datasets/cataloged items, the temporal range of datasets (starting from either the application time or the creation time), the geographic coverage, and the focus community/theme. Table 3 describes search facets and our understanding of the ranking functions on these portals. We show that the nine portals mostly show results in list form. first and all include lists. Some show results in map form first. We believe that all portals could improve in their employment of the FAIR principles. DataONE employed FAIR principles the best because they document the ways in which they promote FAIR use such as through webinars. The following subsections describe search facets, ranking functions, and results from query reformulations in more detail.

4.1 Front-End: User Search Controls

On almost all sample portals surveyed, users have the same core set of facets when searching. First, users are given an omnibox for entering free text. A user enters a text query using keywords or natural language. Second, users are given at least two pre-configured facets to refine their text search. Facets are typically located in a sidebar and are check box, radio button, or range slider toggles. A common selection facet is *tag*, which lets users select a descriptor tag that has been associated with a dataset. Approximately half of the portals surveyed have an advanced search feature with an extended interface. Typically, this lets

[16] http://maps.tnc.org/gis_data.html
[17] https://www.geoportal.lt/geoportal/web/en/
[18] https://eservices.dls.moi.gov.cy/#/national/geoportalmapviewer

Table 3 Search characteristics of a subset of surveyed portals. For each portal, we recorded the ease of navigation, the interaction modalities (e.g., list-based, map-based, visualization-based), the types of search facets (e.g., filters, result sorts), ranking functions, and degree of employment of FAIR principles.

Portal	Facets	Modality	FAIR princples	Ranking Function
DataONE	*filters:* [data attribute; annotation; data files; member node; creator; year; identifier; taxon; location], *sort by:* [most recent; identifier; title; author]	map + list	very good	- BM-25 - query expansion
Data.gov	*filters:* [topics; topic categories; dataset type; tags; format; organization type; organizations; publishers; bureaus; location], *sort by:* [relevance; time/date; popular; date added]	list	good	- TF-IDF
ArcGIS Hub	*filters:* [capabilities; source; content; type; tags], *sort by:* [relevance; most recent; trending; name]	list	good	- BM-25 - Query expansion
USGS	*filters:* [map controls; file format; extent; topic sub-category]	map	satisfactory	- TF-IDF
ADRL	*filters:* [search by all fields, title, subject, or accession number; format; collection; contributor; topic; place; genre; date; academic department; library location; rights], *sort by:* [relevance, year created, creator]	list	satisfactory	- TF-IDF
TERENO	*filters:* [“what?” by topic, keywords, sensor type, parameter; “where?” by metadata fields, catalog, regions, map extent; “when?” by date range]	map + list	satisfactory	- TF-IDF
INSPIRE	*text search:* Select country then search by dataset title, *filters:* [country; spatial scope; theme]	list	satisfactory	- TF-IDF
NASA	*map options:* [region, time, hand-drawn region], *filters:* [features; keywords; platforms; instruments; organizations; projects; processing levels; granule data format], *sort by:* [relevance; usage; end date]	list + map	satisfactory	- TF-IDF
Heritage Gateway	*filters:* [“where?” by search geocoder; “what?” by thesaurus of building, object, or evidence type; “who?” by associated person, architect; “when?” by date range, period; “resource?” by parent organization]	list or map	satisfactory	- TF-IDF

users specify additional filters based on less popular metadata. Third, once a user executes a query and the results are presented as a list, users are given a option for sorting the results. For example, the user can sort by relevance (discussed in Section 4.2), by the date that datasets were created/modified, or alphabetically by dataset title. Note that we did not extensively survey individual results or additional result pages, only the first page result list after a query was executed.

UI complexity and navigation varied substantially. Approximately half of the portals have an omnibox for text search or icons for pre-created search topics on their home page. The other half of portals lead users to search through a button or link with a label like *“find data”*. Approximately one fourth of the portals surveyed initially present results as a map or a map with a list. Users can then navigate the results geographically and further refine by clicking on a specific result, or a region that was labeled with the number of results located in that region.

4.2 Back-End: Query Processing and Interpretation

As previously mentioned, it is difficult to determine exactly how a portal processes and interprets a query (and determine which potential search results are relevant) without knowing their ranking algorithm. Fortunately, many portals are built using an open source CMS that use open-source search engines. Many portals in our survey, especially government portals,

used one of three CMSs for all or part of their back-end processing: CKAN, Socrata, or ArcGIS Hub (or ArcGIS OpenData). Previous surveys [31] suggest that investing in open data portals is typically expensive and labor intensive, so it's reasonable to assume that such systems are appealing for hosting and/or serving data. Other geographic data CMSs, such as GeoBlacklight or Samvera, are only popular within specific communities.

Facets are almost always used to forumate a query before execution. However, typically portal UIs include facets on search result pages so that users can refine their queries. Calculating relevance is at the heart of a ranking function. Typically, a relevance score for a potential search result is a composite value calculated by combining one or more weighted criteria. Most portals appear to use the TF-IDF algorithm for scoring potential search results. This ranking algorithm works by comparing keyword frequency between a query and one or more text attributes (such as title, abstract, and tags) of indexed datasets. Several portals, such as ArcGIS Hub, use hand-tuned boosting in their ranking functions to increase the importance of certain criteria (like keyword frequency in a dataset's tags). At least a quarter of the portals surveyed use a CMS (such as CKAN) that leverages Apache Lucene or Apache Solr as their search engine with no noticeable customization beyond the default search ranking function (i.e., using the bag-of-words model and TF-IDF). For example, searching on Data.gov, on the back-end, a user's query is most likely tokenized, sanitized, normalized, and converted compare with potential search results (which also represented as bags of words). ArcGIS Hub uses Elasticsearch[19] as a search engine and the BM-25 ranking algorithm. For the portals where we were able to see the private ranking function including ArcGIS Hub and Heritage Gateway, keyword frequency match is the most important weight for raking potential search results. We found that recency and popularity (e.g., download frequency) are occasionally used in calculating the score (upward of 25 percent of a score). DataONE and ArcGIS Hub, use query expansion to include results with relevant taxa and synonyms. For example, searching "robbery" on an ArcGIS Hub site will give similar but more granular thematic results such as "crime."

Open geospatial portals are unique from other open data portals due to their handling of space and space's relation to theme and time. This is a difficult task and the subfield of GIR is dedicated to effectively serving relevant geospatial information. A technique at the heart of GIR is georeferencing a textual query. This means disambiguating and resolving geographic references, often toponyms, properly interpreting spatial relations, and inferring the geographic intent of the query. In the most basic application, this means interpreting <*theme*><*spatial relation*><*location*> such as "churches in Poznań Poland" [28, 29]. We did not find any portal that interprets a query this way, although we believe that they do exist. At best, portals attempt to properly interpret the intent of thematic terms through techniques like query expansion. A spatial relation such as "near" is usually ignored or assumed to mean containment. Surprisingly, location appears to be ignored during query processing or specified separately in the UI the omnibox input. For example, several portals (DataONE, Heritage Gateway) let users specify a location by typing in toponyms in a separate text box, which is then geocoded and matched with pre-created regions.

4.3 Effects on Results from Query Reformulations

As stated in Section 3, we ran three search scenarios on our nine focus portals. Overall, the results were mostly what we expected. Most portals surveyed did not indicate a ranking function that leverages techniques beyond the bag-of-words model and TF-IDF

[19] `https://www.elastic.co/`

statistic. We determined this because repeat, reformulations had no effect but generalizations, specializations and word substitutions did. DataONE is an exception because the query "coral disease" yielded more results and more relevant results than "disease coral". In most cases, generalization resulted in more results and and about half the time more relevant results. All portals sampled were sensitive to generalization, specialization, and word substitution reformulations. Once again, this makes sense because most portals are keyword sensitive, so including or removing keywords tended to make a large difference.

In all of the portals surveyed, making geo-modifications, geo-corrections, or geo-disambiguations during a query reformulation did not have noticeable effects on query results. For example, a geo-disambiguation from "edinburgh" to "edinburgh tx" usually reduced the number of results likely because more keywords were included, not because a geography was disambiguated and constrained. A geo-correction from "edinburg" to "edinburgh" increased results. Changing spatial granularity didn't appear to be any different from word substitution except in a few portals like ArcGIS Hub. With access to documentation on Hub's query processing, we know that Hub leverages a process for comparing locations with different spatial granularities. For example, results for "scotland" were more frequent than results for "edinburgh" and not simply because of increased keyword frequency.

4.4 Noteworthy Examples

Search functionality on several portals was unique enough to merit distinction. These portals used novel techniques to make searching easier or more specific. The first portal is Heritage Gateway, which is a historic building and landmark portal for England. On their portal, users interact with search almost entirely through a map. Once users execute a query, results are displayed as symbolized points on a map. Users can then click on individual features to read more about them in a pop-up window or download them. The portal's advanced search lets users refine a text query with *where*, *what*, *when*, and *who* filter criteria. Figure 1 shows a portion of the XML schema for query processing with specific ways users can set criteria (e.g., specifying *where* using a reference system like *gridref*, *osgridref*, *latitude*, *longitude*, etc).

A second notable portal is DataONE. DataONE's UI is shown in in Figure 2. The search interface is primarily map-based with a sidebar for entering text queries and numerous filters. As users pan with the map, search results and facets automatically update to reflect only what datasets are visible. DataONE uniquely disseminates data via RSS, GeoRSS, and other data casting feeds. In 2020, DataONE may start a collaboration with RDMLA[20] for guiding data management and curation best practices that has the potential to support curation efforts in other scientific repositories and libraries.

4.5 Suggested Improvements

As the results show, most portals treat search similarly. On the front end, most portals implement an omnibox for text input and facets to balance user control with effort. On the back end, most portals use a bag-of-words model to represent queries and potential search results, and match them based on keyword frequency. Generic portals also appear to be similarly designed. However, we argue that the uniqueness of geospatial content merits more sophisticated search facets and ranking functions than those that are currently used.

We recommend that most portals transition to primarily map or other visualization modalities (like topic explorers), instead of lists and text boxes. For example, interfaces could orient around self-organizing topic maps [20] or geographic maps. Most interfaces that we

[20] `https://rdmla.github.io/`

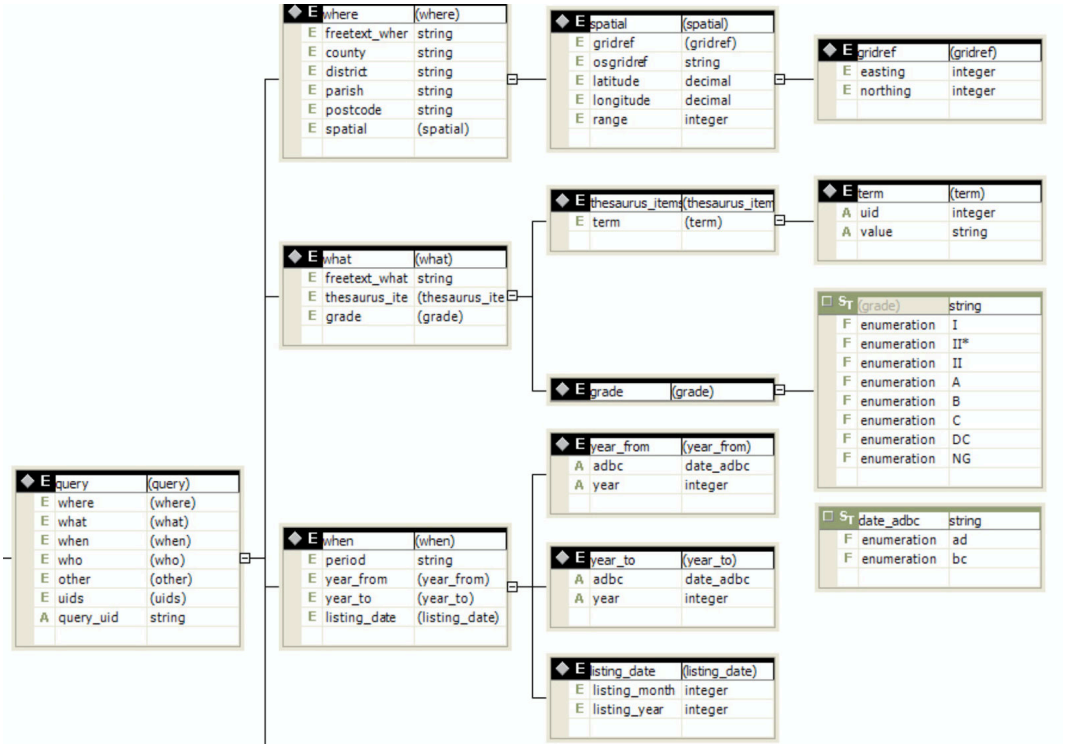


Figure 1 A portion of an XML schema for user query criteria on the Heritage Gateway portal. A text query can be refined by any number of top level criteria on the left including where, what, when, who, other, uids, or query-uid. Each criteria can be further specified such as year-from under the when criteria which is a custom date, or grade which is one of nine predetermined building grades such as I,II,III, DC, or NG.

surveyed currently offer minimal help tools. Therefore, offering users comprehensive guides or wizards for navigating search interfaces could simplify the search process. One idea would be to display a dataset in a map-based interface and then, using a wizard, ask users what aspects they would like to change in order to find other datasets. Regarding search facets, we believe that most portals include too many search filters instead of dedicating resources to improving natural language query interpretation. Effectively balancing functionality and usability is difficult, and unfortunately most web search interfaces fail at this [2].

In terms of developing explicitly spatial ranking and relevance metrics, we saw few examples of portals that did this. One suggestion is to build upon the multidimensional ranking scheme proposed by Sharma and Beard [5] that use space, time, and theme as dimensions. The spatial component of a result would be weighed based on topological relations; the temporal component would be weighed based on Allen intervals; and multiple thematic components would be selected by user in the form of "glyphs". Any score boosting for a dimension should be based on the portal's needs. This solution brings the three key dimensions of spatial information [6] to bear in developing ranking and relevance metrics for spatial data.

There is a clear need for finer grained spatial and thematic processing and interpretation. Few portals compare individual data values to a text query. Those that do only do so when a user specifies advanced search features. However, these features are typically relegated to comparing dataset metadata to a query, not individual data values within a dataset. In other

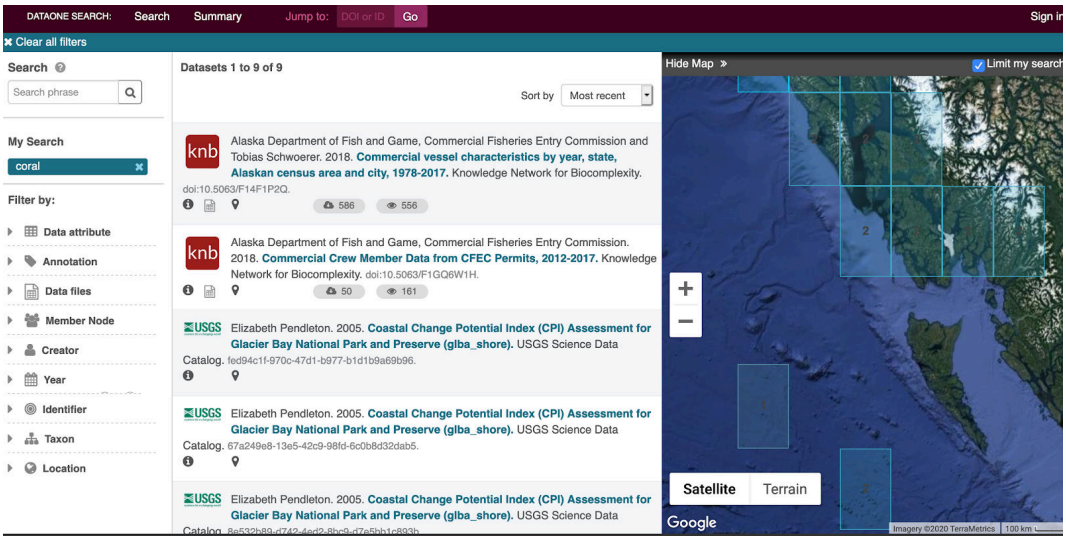


Figure 2 The search interface on DataONE. After executing a text query, users can filter results using criteria including data attribute, year, and taxon. The list of results changes when filters are applied or when a user pans the map and/or selects a gridded region of interest.

words, systems should extract numeric values and ordinal values (like "most" or "nearest") from a query and compare them with potential search results using at a minimum hand-tuned rules. These improvements parallel a Semantic Web goal of returning specific data points for a query, not just datasets [17].

5 Conclusion

In this work we have taken a critical look at the current state of search on open geospatial portals. We surveyed the front end of systems and focused on search facets, a type of control users have while searching. We then surveyed how the back ends of systems process and interpret queries, and how they rank relevant results. To corroborate our understanding about the back-end of these systems and test how effective searching is, we executed several search scenarios. In these scenarios we iteratively reformulated queries against nine specific portals. We found that most portals leverage an omnibox for raw text search and filters to refine them. We also found that most portals use a syntactic-based keyword frequency model for representing queries and potential search results (found in most basic search architectures). As expected, after most query reformulations, changes in results were simple and aligned with what we would expect from this model. We then described distinctive characteristics of nine unique portals and further detailed two notable portals and why they stood out as models for geospatial search.

Open geospatial data portals, which are growing in popularity as resources for accessing geospatial data, have an opportunity to be forefront models of advanced GIR and geospatial computing. However, based on the current state of search facets and ranking, there are several substantial improvements needed to make portals easier to use, easier to navigate, and adhere better to FAIR principles. Optimally, in addition to search, portals would more effectively enable serendipitous discovery.

There are several notable limitations to this work. First, we were not able to quantitatively assess the effectiveness of each portal surveyed. In future work, we plan to create effectiveness criteria based on explicit relevance feedback. Also, since many portals use proprietary search

engines, we were not able to explicitly see how their ranking functions work. However, the intention of this work is to survey, frame, and motivate a quantitative analysis of user search behavior. In future work, we plan to use query logs from the ArcGIS Hub platform to model search behavior. Through those efforts, we hope to see if and how search success and abandonment patterns relate to the limitations of the portals surveyed herein.

References

1. Lars Backstrom, Jon Kleinberg, Ravi Kumar, and Jasmine Novak. Spatial variation in search engine queries. In *Proceeding of the 17th International Conference on World Wide Web 2008, WWW'08*, pages 357–366, 2008. doi:10.1145/1367497.1367546.
2. Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.
3. Andrea Ballatore, Werner Kuhn, Mary Hegarty, and Ed Parsons. Special issue introduction: Spatial approaches to information search. *Spatial Cognition and Computation*, 16(4):245–254, 2016. doi:10.1080/13875868.2016.1243693.
4. Andrea Ballatore, David C Wilson, and Michela Bertolotto. A holistic semantic similarity measure for viewports in interactive maps. In *International Symposium on Web and Wireless Geographical Information Systems*, pages 151–166. Springer, 2012.
5. Kate Beard and Vyjayanti Sharma. Multidimensional ranking for data in digital spatial libraries. *International Journal on Digital Libraries*, 1(2):153–160, 1997. doi:10.1007/s007990050011.
6. Brian J.L. Berry. Approaches to Regional Analysis: A Synthesis. *Annals of the Association of American Geographers*, 54(1):2–11, 1964. doi:10.1111/j.1467-8306.1964.tb00469.x.
7. John Carlo Bertot, Ursula Gorham, Paul T. Jaeger, Lindsay C. Sarin, and Heeyoon Choi. Big data, open government and e-government: Issues, policies and recommendations. *Information Polity*, 19(1-2):5–16, 2014. doi:10.3233/IP-140328.
8. Bradley Wade Bishop and Carolyn Hank. Measuring fair principles to inform fitness for use. *International Journal of Digital Curation*, 13(1):35–46, 2018. doi:10.2218/ijdc.v13i1.630.
9. Christopher Bone, Alan Ager, Ken Bunzel, and Lauren Tierney. A geospatial search engine for discovering multi-format geospatial data across the web. *International Journal of Digital Earth*, 9(1):47–62, 2016. doi:10.1080/17538947.2014.966164.
10. Max Craglia, Michael F Goodchild, Alessandro Annoni, Gilberto Camara, Michael Gould, Werner Kuhn, David Mark, Ian Masser, David Maguire, Steve Liang, and Ed Parsons. A position paper from the Vespucci Initiative for the Advancement of Geographic Information Science. *International Journal of Spatial Data Infrastructures Research*, 3:146–167, 2008. doi:10.2902/1725-0463.2008.03.art9.
11. James Frew, Michael Freeston, Nathan Freitas, Linda Hill, Greg Janee, Kevin Lovette, Robert Nideffer, Terence Smith, and Qi Zheng. The Alexandria digital library architecture. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1513:61–73, 1998. doi:10.1007/pl00021470.
12. Tamar Ganor. An Integrated Spatial Search Engine for Maps and Aerial Photographs on a Google Maps API Platform. *Journal of Map and Geography Libraries*, 13(2):175–197, 2017. doi:10.1080/15420353.2016.1277574.
13. Christian Philipp Geiger and Jörn Von Lucke. Open Government and (Linked) (Open) (Government) (Data). *JeDEM - eJournal of eDemocracy and Open Government*, 4(2):265–278, 2012. doi:10.29379/jedem.v4i2.143.
14. Michael F. Goodchild, Pinde Fu, and Paul Rich. Sharing geographic information: An assessment of the geospatial one-stop. *Annals of the Association of American Geographers*, 97(2):250–266, 2007. doi:10.1111/j.1467-8306.2007.00534.x.
15. Darren Hardy and Kim Durante. A Metadata Schema for Geospatial Resource Discovery Use Cases. *Code4Lib Journal*, 25:1–1, 2014. URL: http://journal.code4lib.org/articles/9710.
16. Marti Hearst. User interfaces for search. *Modern Information Retrieval*, pages 21–55, 2011.

17 Krzysztof Janowicz, Frank van Harmelen, James A Hendler, and Pascal Hitzler. Why the Data Train Needs Semantic Rails. *AI Magazine*, 36(May):5–14, 2015. doi:10.1609/aimag.v36i1.2560.

18 Yongyao Jiang, Yun Li, Chaowei Yang, Fei Hu, Edward M. Armstrong, Thomas Huang, David Moroni, Lewis J. McGibbney, and Christopher J. Finch. Towards intelligent geospatial data discovery: a machine learning framework for search ranking. *International Journal of Digital Earth*, 11(9):956–971, 2018. doi:10.1080/17538947.2017.1371255.

19 Rosie Jones, Wei Vivian Zhang, Benjamin Rey, Pradhuman Jhala, and Eugene Stipp. Geographic intention and modification in web search. *International Journal of Geographical Information Science*, 22(3):229–246, 2008. doi:10.1080/13658810701626186.

20 Sara Lafia, Andrew Turner, and Werner Kuhn. Improving discovery of open civic data. *Leibniz International Proceedings in Informatics, LIPIcs*, 114(9):1–9, 2018. doi:10.4230/LIPIcs.GIScience.2018.9.

21 Tessa Lau and Eric Horvitz. Patterns of search: analyzing and modeling web query refinement. In *UM99 user modeling*, pages 119–128. Springer, 1999.

22 Chang Liu, Jacek Gwizdka, Jingjing Liu, Tao Xu, and Nicholas J. Belkin. Analysis and evaluation of query reformulations in different task types. *Proceedings of the ASIST Annual Meeting*, 47, 2010. doi:10.1002/meet.14504701214.

23 David J. Maguire and Paul A. Longley. The emergence of geoportals and their role in spatial data infrastructures. *Computers, Environment and Urban Systems*, 29(1 SPEC.ISS.):3–14, 2005. doi:10.1016/j.compenvurbsys.2004.05.012.

24 Ian Masser. *GIS worlds: creating spatial data infrastructures*, volume 338. Esri Press Redlands, CA, 2005.

25 Matthew S Mayernik. Research data and metadata curation as institutional issues. *Journal of the Association for Information Science and Technology*, 67(4):973–993, 2016. doi:10.1002/asi.23425.

26 Karen Okamoto. What is being done with open government data? An exploratory analysis of public uses of New York City open data. *Webology*, 13(1):1–12, 2016.

27 Ricardo Oliveira and Rafael Moreno. Harvesting, integrating and distributing large open geospatial datasets using free and open-source software. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 41(July):939–940, 2016. doi:10.5194/isprsarchives-XLI-B7-939-2016.

28 José M. Perea-Ortega, Miguel A. García-Cumbreras, and L. Alfonso Ureña-López. Evaluating different query reformulation techniques for the geographical information retrieval task considering geospatial entities as textual terms, 2012.

29 Ross S. Purves, Paul Clough, Christopher B. Jones, Mark H. Hall, and Vanessa Murdock. Geographic information retrieval: Progress and challenges in spatial search of text. *Foundations and Trends® in Information Retrieval*, 12(2-3):164–318, 2018. doi:10.1561/1500000034.

30 Michael G. Tait. Implementing geoportals: Applications of distributed GIS. *Computers, Environment and Urban Systems*, 29(1 SPEC.ISS.):33–47, 2005. doi:10.1016/j.compenvurbsys.2004.05.011.

31 Akemi Takeoka and Christopher G Reddick. A longitudinal cross-sector analysis of open data portal service capability : The case of Australian local governments. *Government information quarterly*, 34:231–243, 2017. doi:10.1016/j.giq.2017.02.004.

32 W Tang and J Selwood. Spatial portals: Adding value to spatial data infrastructures. In *ISPRS Workshop on Service and Application of Spatial Data Infrastructure*, pages 14–16, 2005.

33 Jeffrey Thorsby, Genie N.L. Stowers, Kristen Wolslegel, and Ellie Tumbuan. Understanding the content and features of open data portals in American cities. *Government Information Quarterly*, 34(1):53–61, 2017. doi:10.1016/j.giq.2016.07.001.

34 Mark D. Wilkinson, Susanna Assunta Sansone, Erik Schultes, Peter Doorn, Luiz Olavo Bonino Da Silva Santos, and Michel Dumontier. Comment: A design framework and exemplar metrics for FAIRness. *Scientific Data*, 5:1–4, 2018. doi:10.1038/sdata.2018.118.

35 Phil Yang, John Evans, Marge Cola, Steve Marley, Nadine Alameh, and Myra Bambacus. The emerging concepts and applications of the spatial web portal. *Photogrammetric Engineering and Remote Sensing*, 73(6):691–698, 2007. doi:10.14358/PERS.73.6.691.

36 Anneke Zuiderwijk and Marijn Janssen. Open data policies, their implementation and impact: A framework for comparison. *Government Information Quarterly*, 31(1):17–29, 2014. doi:10.1016/j.giq.2013.04.003.

How Do People Describe Locations During a Natural Disaster: An Analysis of Tweets from Hurricane Harvey

Yingjie Hu
GeoAI Lab, Department of Geography, University at Buffalo, NY, USA
http://www.acsu.buffalo.edu/~yhu42/
yhu42@buffalo.edu

Jimin Wang
GeoAI Lab, Department of Geography, University at Buffalo, NY, USA
https://geoai.geog.buffalo.edu/people/
jiminwan@buffalo.edu

Abstract

Social media platforms, such as Twitter, have been increasingly used by people during natural disasters to share information and request for help. Hurricane Harvey was a category 4 hurricane that devastated Houston, Texas, USA in August 2017 and caused catastrophic flooding in the Houston metropolitan area. Hurricane Harvey also witnessed the widespread use of social media by the general public in response to this major disaster, and geographic locations are key information pieces described in many of the social media messages. A geoparsing system, or a geoparser, can be utilized to automatically extract and locate the described locations, which can help first responders reach the people in need. While a number of geoparsers have already been developed, it is unclear how effective they are in recognizing and geo-locating the locations described by people during natural disasters. To fill this gap, this work seeks to understand how people describe locations during a natural disaster by analyzing a sample of tweets posted during Hurricane Harvey. We then identify the limitations of existing geoparsers in processing these tweets, and discuss possible approaches to overcoming these limitations.

2012 ACM Subject Classification Information systems → Content analysis and feature selection; Information systems → Retrieval effectiveness

Keywords and phrases Geoparsing, geographic informational retrieval, social media, tweet analysis, disaster response

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.6

Supplementary Material An annotated dataset, a full list of terms, and the constructed regular expression are available at: https://github.com/geoai-lab/HowDoPeopleDescribeLocations.

1 Introduction

Hurricane Harvey was a Category 4 tropical storm which started on August 17, 2017 and ended on September 2, 2017 and made a landfall on Texas and Louisiana, USA. It dropped more than 1,300 mm of rain over the Houston metropolitan area and caused catastrophic flooding [44]. During the hurricane and the subsequent flooding, social media platforms, such as Twitter, were used by many residents in the city of Houston and the surrounding areas to share disaster-related information and send help requests.

The use of social media during natural disasters is not new. An early work by Longueville et al. [6] used Twitter to analyze a forest fire in the South of France back in July 2009. In the following years, many studies were conducted based on the social media data collected from disasters to understand the emergency situations on the ground and the reactions of the

general public. Examples include the 2010 Pakistan flood [29], the 2011 earthquake on the East Coast of the US [4], Hurricane Sandy in 2012 [27], the 2014 wildfire of California [42], Hurricane Joaquin in 2015 [41], and Hurricane Irma in 2017 [43]. Social media data, such as tweets, provide near real-time information about what is happening in the disaster-affected area, and are suitable for applications in disaster response and situational awareness [25]. Twitter, in particular, allows researchers to retrieve about 1% of the total number of public tweets for free via its API, and this ability enables various tweet-based disaster studies.

While social media has already been used in disasters and emergency situations, Hurricane Harvey was probably the first major disaster in which the use of social media was comparable or even surpassed the use of some traditional communication methods during a disaster. The National Public Radio (NPR) of the US published an article with the headline "Facebook, Twitter Replace 911 Calls For Stranded In Houston" [35], which described how social media platforms were widely used by Houston residents to request for help when 911 could not be reached. The fact that the storm took out over a dozen emergency call centers and that there were too many 911 calls during and after the hurricane were among the reasons responsible for the failure of the 911 system. Another article published in The Wall Street Journal was titled "Hurricane Harvey Victims Turn to Social Media for Assistance", which described similar stories in which people turned to social media for help after their 911 calls failed [34]. In addition, Hurricane Harvey was called by The Time Magazine as "The U.S.'s First Social Media Storm" [33]. Besides news articles, a survey was conducted by researchers [28] after Hurricane Harvey, which filtered through 2,082 people in Houston and the surrounding communities, and focused on 195 Twitter users. They found that about one-third of their respondents indicated that they used social media to request for help because they were unable to connect to 911.

With the ubiquity of smart mobile devices and the popularity of social media, it seems to be a natural choice for people to turn to Twitter, Facebook, or other social media platforms when their 911 calls fail. People are already familiar with the basic use of these social media platforms (e.g., how to create a post and how to upload a photo), and they can stay connected with their friends and family members online, follow the latest information from public figures (e.g., the Twitter account of the mayor of the affected city), authoritative agencies (e.g., FEMA), and voluntary organizations, and can "@" related people and organizations to send targeted messages. Indeed, a survey by Pourebrahima et al. [30] based on Hurricane Sandy in 2012 revealed that Twitter users received emergency information faster and from more sources than non-Twitter users. The survey by Mihunov et al. [28] found that about 76% of their respondents considered Twitter as "very useful" or "extremely useful" for seeking help during Hurricane Harvey, and roughly three quarters of their respondents indicated that Twitter and other social media were easy to use. Their survey also revealed some challenges in the use of Twitter during a natural disaster, such as not knowing whether volunteers received their requests or when they would send help. However, these situations could change in future disasters, as volunteers and relief organizations learn to collect the requests from social media. In addition to Twitter, other social media platforms were also used by people to seek help [22]. For example, an online group named "Hurricane Harvey 2017 - Together We Will Make It" was created on Facebook to enable victims to post messages about their situations during the flooding [35].

One major challenge in handling the help requests that people sent on social media platforms is to efficiently process the huge number of posts. As described by a disaster responding consultant during Hurricane Harvey [35], "It is literally trying to drink from a firehose". Disaster responders simply do not have the bandwidth and time to manually

monitor the huge number of posts on social media and identify actionable information. In fact, there exist multiple challenges in effectively using the information from social media platforms, including verifying the veracity of the posted information, understanding the purpose of the posts (e.g., whether a post is about requesting rescue, reporting disaster situation, calling for donation, or praying for the affected people), and extracting critical information pieces (e.g., the locations of the people who need help). Much research has already been devoted to identifying true information from false information [13, 38], classifying the purposes of social media posts [15, 3], and extracting information from tweets [16, 32].

This paper focuses on the specific challenge of extracting locations from the tweets posted during a natural disaster. As a first step, we focus on understanding how people describe locations during a disaster by analyzing a sample of tweets randomly selected from over 7 million tweets posted during Hurricane Harvey. The contribution of this paper is twofold:

- We conduct an analysis on a sample of 1,000 randomly selected tweets to understand and categorize the ways people describe locations during a natural disaster.
- We identify the limitations of existing tools in extracting locations from these tweets and discuss possible approaches to overcoming these limitations.

The remainder of this paper is organized as follows. Section 2 reviews related work in geoparsing and tweet analysis in the context of disasters. Section 3 describes the dataset from Hurricane Harvey. In Section 4, we analyze and classify location descriptions in the selected tweets. Section 5 reports the experiment results of using existing tools for processing the tweets. Finally, Section 6 summarizes this work and discusses future directions.

2 Related work

Locations in tweets can be extracted through *geoparsing*, a process of recognizing and geolocating place names (or toponyms) from texts [8, 12, 40]. Geoparsing is often studied within the topic of geographic information retrieval (GIR) [17, 31]. A software tool developed for geoparsing is called a *geoparser*, which typically functions in two consecutive steps: *toponym recognition* and *toponym resolution*. The first step recognizes toponyms from texts, and the second step resolves any place name ambiguity and assigns suitable geographic coordinates. Figure 1 illustrates these two steps. It is worth noting that geoparsing can be applied to other types of texts in addition to social media messages, such as Web pages, news articles, organization documents, and others.

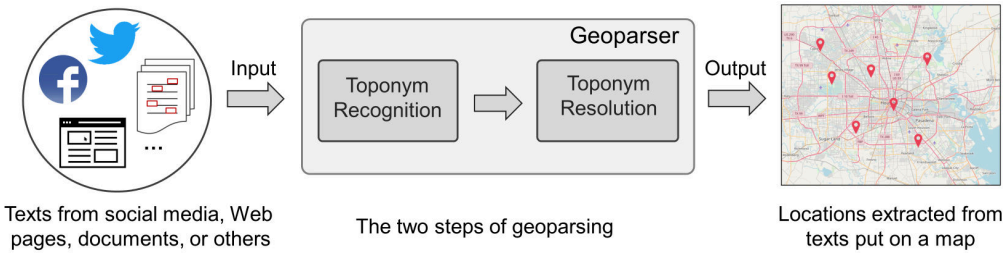


Figure 1 The typical process of geoparsing text to extract locations.

A number of geoparsers have already been developed by researchers. GeoTxt is an online geoparser developed by Karimzadeh et al. [19, 20], which uses the Stanford Named Entity Recognition (NER) tool and several other NER tools for toponym recognition and

employs the GeoNames gazetteer[1] for toponym resolution. TopoCluster, developed by Delozier et al. [7], is a geoparser that uses the Stanford NER for toponym recognition and leverages a technique based on the geographic word profiles for toponym resolution. The Edinburgh Geoparser, developed by the Language Technology Group at Edinburgh University [1], uses their own natural language processing (NLP) tool, called LT-TTT2, for toponym recognition, and a gazetteer (e.g., GeoNames) and pre-defined heuristics for toponym resolution. Cartographic Location And Vicinity INdexer (CLAVIN)[2] is a geoparser developed by Berico Technologies that employs the NER tool from the Apache OpenNLP library or the Stanford NER for toponym recognition, and utilizes a gazetteer and heuristics for toponym resolution. CamCoder is a toponym resolution model developed by Gritta et al. [11], which integrates a convolutional neural network and geographic vector representations. Gritta et al. further converted CamCoder into a geoparser by employing the spaCy NER tool for toponym recognition.

Twitter data were used in many previous studies on situational awareness and disaster response. Imran et al. [15] and Yu et al. [43] developed machine learning and text mining systems for automatically classifying tweets into topics, e.g., *caution and advice* and *casualty and damage*. Huang and Xiao [14] classified tweets into different disaster phases, such as preparedness, response, impact and recovery. Kryvasheyeu et al. [21] and Li et al. [23] used tweets for assessing the damages of disasters. Existing studies, however, often used only the geotagged locations of tweets [5, 42] or the locations in the profiles of Twitter users [45, 46], rather than the locations described in tweet content. Many geotagged locations were collected by the GPS receivers in smart mobile devices, and therefore are generally more accurate than the locations geoparsed from the content of tweets. This can be a reason that motivated researchers to use the geotagged locations of tweets. Meanwhile, geotagged locations reflect only the current locations of Twitter users, which may not be the same as the locations described in the content of tweets. In addition, only about 1% tweets were geotagged [36], and the number of geotagged tweets further decreased with Twitter's removal of precise geotagging in June 2019. By contrast, researchers found that over 10% tweets contain some location references in their content [25]. For the locations in the profiles of Twitter users, they may reflect neither the current locations of the users nor the locations described by the users, since the profile locations can be their birthplaces, work places, marriage places, or even imaginary places, and are not always updated.

Some research examined location extraction from the content of tweets. GeoTxt is a geoparser originally developed for processing tweets [20]; however, their testing experiments were based on a tweet corpus, GeoCopora [39], whose toponyms are mostly country names and major city names, rather than fine-grained place names in a disaster affected area (although GeoCopora does contain some fine-grained locations, such as school names). Gelernter and Balaji [9] geoparsed locations in the tweets from the 2011 earthquake in Christchurch, New Zealand, and Wang et al. [41] extracted locations from tweets for monitoring the flood during Hurricane Joaquin in 2015. However, both work focused on using a mixture of NLP techniques and packages (e.g., abbreviation expansion, spell correction, and NER tools) for location extraction, rather than a more detailed analysis on the characteristics of the location descriptions. This paper aims to fill such a gap by examining how people describe locations in tweets during a natural disaster, with the ultimate goal of helping design more effective geoparsers for assisting disaster response.

[1] https://www.geonames.org/
[2] https://clavin.bericotechnologies.com/Berico_CLAVIN.pdf

3 Dataset

The dataset used in this work is a set of 7,041,866 tweets collected during Hurricane Harvey and the subsequent flooding from August 18, 2017 to September 22, 2017. This dataset was prepared by the University of North Texas Libraries, and the tweets were retrieved based on a set of hashtags and keywords, such as "#HurricaneHarvey", "#HoustonFlood", and "Hurricane Harvey". The entire dataset is available from the library repository of North Texas University (NTU)[3], and it is in the public domain.

Among the over seven million tweets in the entire dataset, only 7,540 are geotagged with longitude and latitude coordinates. These geotagged tweets are distributed not only within the Houston area but also throughout the world, with most of the tweets located inside the United States. Figure 2(a) shows the locations of the geotagged tweets in the Houston area, and the locations of all the geotagged tweets are visualized in the overview map in the lower-left corner. The low percentage of geotagged tweets (about 0.1%) in this

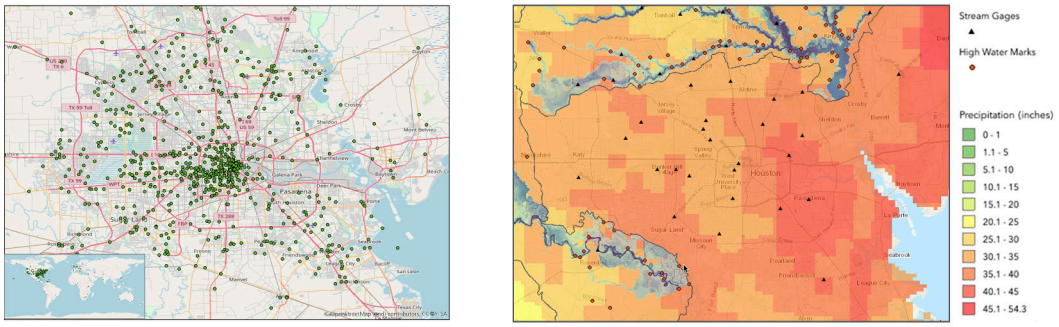


Figure 2 A comparison of the locations of geotagged tweets and the precipitation during Hurricane Harvey: (a) locations of the geotagged tweets; (b) precipitation in the Houston area from the USGS.

dataset and the fact that the geotagged tweets are distributed throughout the world can be attributed to the data collection process: the data were collected using a list of keywords and hashtags rather than focusing on a particular geographic area. We compare the locations of the geotagged tweets with the precipitation map[4] from the US Geological Survey (USGS) (Figure 2(b)). No clear relationship can be visually identified between the locations of the geotagged tweets and the severity of the precipitation in different areas. For example, the northwestern region received relatively less precipitation than the southeastern region, but there were more geotagged tweets in the former region.

In this work, we are particularly interested in the locations described in the content of tweets. While both the news and literature told us that people used Twitter and other social media platforms to request for help and share information, we still do not know how specifically people describe locations in social media messages during this natural disaster. Manually analyzing the 7,041,866 tweets is practically impossible. Thus, we use a simple regular expression to narrow down the target tweets to be analyzed. The regular expression contains about 70 location-related terms that are frequently observed in place names and location descriptions, such as "street", "avenue", "park", "square", "bridge", "rd", and "ave". A full list of these terms and the constructed regular expression can be accessed at: `https://github.com/geoai-lab/HowDoPeopleDescribeLocations`. Running this regular

[3] `https://digital.library.unt.edu/ark:/67531/metadc993940/`
[4] `https://webapps.usgs.gov/harvey/`

expression against the 7 million tweets returns 15,834 tweets. A quick examination of these 15,834 tweets shows that many of them contain detailed location descriptions, such as house number addresses or school names. For curiosity, we also run the same regular expression against the 7,540 geotagged tweets. Only 203 tweets are returned. This result suggests that there are many tweets that contain location descriptions but are not geotagged. Thus, we will miss important information if we focus on geotagged locations only.

We randomly select 1,000 tweets from the 15,834 records returned by the regular expression. This selection is performed as follows: we first remove retweets to avoid duplication; we then index the remaining tweets, and generate 1,000 non-repeating random integers that are used as the indexes to retrieve the corresponding tweets. As we read through some of these tweets, we see vivid images of people seeking help and sharing information during Hurricane Harvey. Some examples are provided as below:

- *"12 Y/O BOY NEEDs RESCUED! 8100 Cypresswood Dr Spring TX 77379 They are trapped on second story! #houstonflood"*
- *"80 people stranded in a church!! 5547 Cavalcade St, Houston, TX 77026 #harveyrescue #hurricaneharvey"*
- *"Rescue needed: 2907 Trinity Drive, Pearland, Tx. Need boat rescue 3 people, 2 elderly one is 90 not steady in her feet & cant swim. #Harvey"*
- *"Community is responding at shelters in College Park High School and Magnolia High School #TheWoodlands #Harvey. . . "*
- *"#Houston #HoustonFlood the intersection of I-45 & N. Main Street"*

While the above tweets certainly do not represent all of those posted during Hurricane Harvey, they demonstrate the urgency of some requests. Effectively and efficiently extracting locations from these tweets can help responders and volunteers to reach the people at risk more quickly and can even save lives. In addition, these examples also show that some people were requesting help for others. Thus, even if their tweets were geotagged, it is necessary to focus on the locations described in the content rather than the geotagged locations.

4 Understanding the locations described in Harvey tweets

In this section, we examine and understand the ways people describe locations based on the 1,000 tweets. To do so, we carefully read through each of the tweets, identify and annotate the locations described in their content, and classify the location descriptions. It is worth noting that we focus on the descriptions that refer to specific geographic locations rather than general *locative expressions* [24], such as "this corner" or "that building". The data annotation is done in the following steps. First, the second author reads each tweet and annotates the location descriptions identified; second, the first author goes through the entire dataset, checking each location annotation and discussing with the second author to resolve any annotation difference; a preliminary list of location categories is also identified in this step; third, the first author goes through the entire dataset again, refines the list of categories, and classifies the location descriptions; fourth, the second author performs another round of checking to examine the classified location descriptions. The locations are annotated using the IOB model widely adopted in the CoNLL shared tasks [37]. In the process of annotating the data, we also find that some of the initial 1,000 tweets do not contain specific locations (e.g., a tweet may say: "My side street is now a rushing tributary"). We replace those tweets with others randomly selected from the rest of the data, so that each of the 1,000 tweets contains at least one specific location description. The annotated dataset is available at: `https://github.com/geoai-lab/HowDoPeopleDescribeLocations`.

Ten categories of location descriptions are identified based on the 1,000 Hurricane Harvey tweets (Table 1). The number of tweets in each category is also summarized in Table 1 in the column *Count*. It is worth noting that a tweet may contain more than one type of location descriptions, and therefore can be counted toward more than one category.

Table 1 Ten categories of location descriptions identified from the 1,000 Harvey tweets.

Category	***Examples***	***Count***
C1: House number addresses	- *"Papa stranded in home. Water rising above waist. HELP **8111 Woodlyn Rd, 77028** #houstonflood"* - *"#HurricaneHarvey family needs rescuing at **11800 Grant Rd. Apt. 1009. Cypress, Texas 77429**"*	257
C2: Street names	- *"#Harvey LIVE from San Antonio, TX. Fatal car accident at **Ingram Rd.**, Strong winds."* - *"**Allen Parkway**, **Memorial**, **Waugh overpass**, Spotts park and Buffalo Bayou park completely under water"*	571
C3: Highways	- *"9:00AM update video from Hogan St over White Oak Bayou, **I-10**, **I-45**: water down about 4' since last night. . . "* - *"Left Corpus bout to be in San Angelo #HurricaneHarvey Y'all be safe Avoided **highway 37** Took the back road"*	68
C4: Exits of highways	- *"Need trailers/trucks to move dogs from Park Location: Whites Park Pavillion off **I-10 exit 61** Anahuac TX"* - *"**TX 249 Northbound** at Chasewood Dr. **Louetta Rd. Exit**. #houstonflood"*	8
C5: Intersections of roads (rivers)	- *"Guys, this is **I-45** at **Main Street** in Houston. Crazy. #hurricane #harvey. . . "* - *"Major flooding at **Clay Rd** & **Queenston** in west Houston. Lots of rescues going on for ppl trapped..."*	109
C6: Natural features	- *"**Buffalo Bayou** holding steady at 10,000 cfs at the gage near Terry Hershey Park"* - *"Frontage Rd at the river #hurricaneHarvey #hurricaneharvey @ **San Jacinto River**"*	77
C7: Other human-made features	- *"Houston's **Buffalo Bayou Park** - always among the first to flood. #Harvey"* - *"If you need a place to escape #HurricaneHarvey, The **Willie De Leon Civic Center**: 300 E. Main St in Uvalde is open as a shelter"*	219
C8: Local organizations	- *"#Harvey does anyone know about the flooding conditions around **Cypress Ridge High School**?! #HurricaneHarvey"* - *"Cleaning supply drive is underway. 9-11 am today at **Preston Hollow Presbyterian Church**"*	60
C9: Admin units	- *"#HurricaneHarvey INTENSE eye wall of category 4 Hurricane Harvey from **Rockport, TX**"* - *"Pictures of downed trees and damaged apartment building on Airline Road in **Corpus Christi**."*	644
C10: Multiple areas	- *"#HurricaneHarvey Anyone doing high water rescues in the **Pasadena/Deer Park** area? My daughter has been stranded in a parking lot all night"* - *"FYI to any of you in **NW Houston/Lakewood Forest**, Projections are showing Cypress Creek overflowing at Grant Rd"*	6

For category *C1*, we are surprised to see many tweets using the very standard U.S. postal office address format, with a house number, street name, city name, state name, and postal code. Those house number addresses, once effectively extracted from the text, can be located

via a typical geocoder (although today's geocoders and geoparsers are developed as separate tools). Some addresses only contain a house number and a street name. Those addresses can be located by narrowing down to the area that is affected by the disaster, e.g., Houston or Texas in the case of Hurricane Harvey.

Categories *C2* and *C3* cover location descriptions about roads and highways. These two categories could be merged into one. We separate them because our experiments later find that existing NER tools have difficulty in recognizing the US highway names, such as *I-45* and *Hwy 90*. Yet, those highway names are common in many geographic areas of the US and in the daily conversations of people. Thus, we believe that this category is worth to be highlighted from the perspective of developing better geoparsers.

Category *C4* covers highway exits. People can use an exit to provide a more precise location related to a highway. They may use the exit number, e.g., *"exit 61"*, or the street name of an exit, e.g., *"Louetta Rd. Exit"*. This may be related to the US culture since road signs on the US highways often provide both the exit numbers and the corresponding street names. One may also use two exits in one tweet to describe a segment of a highway, such as in *"My uncle is stuck in his truck on I-45 between Cypress Hill & Huffmeister exits"*.

Category *C5* covers location descriptions related to road (or river) intersections. We identify five ways used by people in tweets to describe road intersections: (1) Road A and Road B, (2) Road A & Road B, (3) Road A at Road B, (4) Road A @ Road B, (5) Road A / Road B. Besides, people often use abbreviations when describing intersections, e.g., they may write *"Mary Bates and Concho St"* instead of *"Mary Bates Blvd and Concho St"*. The intersections of two rivers, or a road and a river, are described in similar ways, such as in *"White Oak Bayou at Houston Avenue 1:00 pm Saturday #Houston"*. A tweet may contain more than one intersection, such as in *"Streets Flooded: Almeda Genoa Rd. from Windmill Lakes Blvd. to Rowlett Rd."* which uses two intersections to describe a road segment.

Categories *C6*, *C7*, and *C8* cover location descriptions related to natural features, other human-made features (excluding streets and highways), and local organizations. These location descriptions are generally in the form of place names, such as the name of a bayou, a church, a school, or a park. We find that many tweets also provide the exact address in addition to a place name, such as the second example of *C7*.

Category *C9* covers location descriptions related to towns, cities, and states. Examples include *Houston*, *Katy*, *Rockport*, *Corpus Christi*, *Texas*, and *TX*. This type of locations has limited value from a disaster response perspective, due to their coarse geospatial resolutions.

Category *C10* covers locations related to multiple areas. We find that people use this way to describe a geographic region that typically involves two smaller neighborhoods, towns, or cities, such as *"Pasadena"* and *"Deer Park"* in the first example.

In summary, we have identified ten categories of location descriptions based on the 1,000 tweets from Hurricane Harvey. Overall, people seem to describe their locations precisely by providing the exact house number addresses, road intersections, exit numbers of highways, or adding detailed address information to place names. One reason may be that people, when under emergency situations, may choose to describe locations in precise ways in order to be understood by others such as first responders and volunteers. While these categories are identified based on the 1,000 tweets from a particular disaster, they seem to be general and are likely to be used by people in future disasters in the US. Understanding these location descriptions is fundamental for designing effective geoparsers to support disaster response.

5 Extracting locations from Harvey tweets using existing tools

With the 1,000 Harvey tweets annotated, we examine the performance of existing tools on extracting locations from these tweets. While this seems to be a straightforward task, there are limitations in existing geoparsers that prevent their direct application. First, none of the five geoparsers that we discussed previously, namely GeoTxt, TopoCluster, CLAVIN, the Edinburgh Geoparser, and CamCoder, have the capability of geocoding house number addresses which are an important type of location descriptions (the category of *C1*). Second, none of the five geoparsers have the capability of geo-locating local street names and highway names (the categories of *C2* and *C3*) at a sub-city level (largely due to their use of the GeoNames gazetteer which focuses on the names of cities and upper-level administrative units), let alone road intersections and highway exits (the categories of *C4* and *C5*). It is worth noting that these limitations do not suggest that existing geoparsers are not well designed; instead, they suggest that there is a gap between the demand of processing disaster-related tweets focusing on a local area and the expected application of the existing geoparsers for extracting city- and upper-level toponyms throughout the world (the category of *C9*). Such an application fits well with one of the important objectives of GIR research, namely to geographically index documents such as Web pages [2]. Although we cannot directly apply existing geoparsers to the Harvey tweets, we can examine their components on toponym recognition and resolution respectively.

5.1 Toponym recognition

Existing geoparsers typically use off-the-shelf NER tools for the step of toponym recognition rather than designing their own models. A rationale of doing so is that toponym recognition, to some extent, can be considered as a subtask of named entity recognition. Indeed, many NER tools can recognize multiple types of entities from text, such as persons, companies, locations, genes, music albums, and others. Thus, one can use an NER tool for toponym recognition by keeping only *locations* in the output, and save the effort of developing a model from scratch. How would the NER tools used in existing geoparsers perform on the Hurricane Harvey tweets? In the following, we conduct experiments to answer this question.

The NER tools to be tested in our experiments are the Stanford NER and the spaCy NER, both of which are used in existing geoparsers. Particularly, the Stanford NER has been used in GeoTxt, TopoCluster, and CLAVIN, and the spaCy NER has been used in CamCoder. The Stanford NER has both a default version, which is sensitive to upper and lower letter cases, and a caseless version. Considering that the content of tweets may not have regular capitalization as in well-formatted text, we test both the default case-sensitive Stanford NER and the caseless version. With the typically used 3-class model, both case-sensitive and caseless Stanford NER have three classes in their output: *Person*, *Organization*, and *Location*. Given the names of the three classes, one might choose to keep *Location* only in the output. However, doing so will miss schools and churches described in the tweets, which are often used as shelters during a disaster, because the Stanford NER considers schools and churches as *Organization*. An alternative choice is to keep both *Location* and *Organization* in the output. However, such a design choice will include false positives. For example, in the sentence "The Red Cross has provided recovery assistance to more than 46,000 households affected by Hurricane Harvey", *"Red Cross"* will be included in the output since it is an *Organization*; this adds a false positive into the toponym recognition result. The spaCy NER has a similar issue, whose output includes multiple classes related to geography. These classes are *Facility* (e.g., buildings, airports, and highways), *Organization* (e.g., companies,

agencies, and institutions), *GPE* (Geo-Political Entity; e.g., countries, cities, and states), and *Location* (e.g., non-GPE locations, mountain ranges, and bodies of water). Again, one might choose to keep *Location* only given the names of these classes, and a direct consequence is that the spaCy NER will only recognize natural features, such as rivers and mountains, and will miss all other valid toponyms. On the other hand, keeping all the classes can introduce false positives into the output of the spaCy NER. In this work, we test these different design choices for the Stanford NER and the spaCy NER. Specifically, we examine the performances of the Stanford NER when only *Location* is kept in the output (we call it *"narrow"* version) and when both *Organization* and *Location* are kept (*"broad"* version). For the spaCy NER, we examine its performances when only *Location* is kept (*"narrow"*) and when all location-related entities are kept (*"broad"*). In total, we test six versions of the NER tools: the default Stanford NER (narrow and broad), the caseless Stanford NER (narrow and broad), and the spaCy NER (narrow and broad).

In the first set of experiments, we evaluate the performances of these NER tools on recognizing all locations regardless of their categories from the 1,000 Hurricane Harvey tweets. The metrics used are *precision*, *recall*, and *F-score* (Equations 1-3).

$$Precision = \frac{tp}{tp + fp} \tag{1}$$

$$Recall = \frac{tp}{tp + fn} \tag{2}$$

$$F\text{-}score = 2 \cdot \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

Precision measures the percentage of correctly recognized locations (true positives or *tp*) among all the locations that are recognized by the model (both *tp* and false positives (*fp*)). *Recall* measures the percentage of correctly recognized locations among all the annotated locations which include *tp* and false negatives (*fn*). *F-score* is the harmonic mean of the precision and the recall. F-score is high only when both precision and recall are fairly high, and is low if either of the two is low.

The performances of the six versions of NER tools are reported in Table 2. Overall, the

Table 2 Performances of the NER tools on the 1,000 Hurricane Harvey tweets.

NER tool	***Precision***	***Recall***	***F-score***
Stanford default (*Narrow*)	**0.829**	0.400	0.540
Stanford default (*Broad*)	0.733	**0.441**	**0.551**
Stanford caseless (*Narrow*)	0.804	0.321	0.458
Stanford caseless (*Broad*)	0.723	0.337	0.460
spaCy NER (*Narrow*)	0.575	0.024	0.046
spaCy NER (*Broad*)	0.463	0.305	0.367

performances of all four versions of the Stanford NER dominate the spaCy NER. This result suggests the effectiveness of this classic and open source NER tool developed by the Stanford Natural Language Processing Group [26]. The default Stanford NER with a *narrow* output

(i.e., keeping *Location* only) achieves the highest precision, while the default Stanford NER with a *broad* output (i.e., keeping both *Location* and *Organization*) achieves the highest recall and F-score. This result can be explained by the capability of the *broad* Stanford NER in recognizing schools, churches, and other organizations that are also locations in these Hurricane Harvey tweets. The lower precision of the *broad* Stanford NER compared with the *narrow* Stanford NER is explained by the included false positives of the *broad* version. Another interesting observation from the result is that the default Stanford NER overall performs better than the caseless Stanford NER. Since tweets are user-generated content that may not follow the regular upper and lower cases, we may be tempted to use the caseless version of the Stanford NER. While there do exist tweets with ill-capitalized words, we find that a large percentage of the analyzed tweets (over 85%) still use correct capitalization. Thus, using a caseless version of the Stanford NER, which completely ignores letter cases in the text, will miss the information contained in the correct capitalization used by many tweets. On the other hand, if one expects that most capitalization in the text is incorrect or the text is not capitalized at all, then the caseless version is likely to be a better choice.

In the second set of experiments, we evaluate the performances of the NER tools on the different categories of location descriptions reported in Table 1. Here, we cannot use the same *Precision*, *Recall*, and *F-score* as the evaluation metrics. This is because these NER tools do not differentiate the ten categories of locations (e.g., the Stanford NER considers all of the entities as *Location* or *Organization*, while the spaCy NER does not differentiate streets, highways, and other human-made features). Thus, we use the metric of *Accuracy* that has been used in previous studies, such as [10, 18, 12, 40]. It is calculated using the equation below:

$$Accuracy_c = \frac{|Recognized \cap Annotated_c|}{|Annotated_c|} \tag{4}$$

where $Accuracy_c$ represents the *Accuracy* of a model on the location category c; *Recognized* represents the set of all locations recognized by the model; and $Annotated_c$ represents the set of annotated locations in the category c.

In addition, an NER tool cannot recognize a location that consists of multiple entities. For example, a house number address like *"5547 Cavalcade St, Houston, TX 77026"* (category *C1*) consists of a door number, a street name, a city name, a state name, and a zip code, which are typically recognized as separate entities by an NER. Similar situation applies to road intersections (category *C5*) and multiple areas (category *C10*). These three categories are thus not included in the experiments. The performances of the NER tools on the other seven location categories are shown in Figure 3.

A number of observations can be obtained from the result. First, all six versions of the NER tools fail on the category *C4: Exits of highways*. This suggests a major limitation of using these off-the-shelf NER tools for toponym recognition: they will miss all the rescue requests whose locations are in the form of highway exits. Second, the broad version of the default Stanford NER has the highest *accuracy* across different categories of location descriptions. However, the broad version likely sacrifices *precision* for *recall* (which cannot be directly measured for each individual category), given its lower overall *precision* compared with the narrow version reported in Table 2. As can be seen in Figure 3, the broad version of the Stanford NER shows a major gain in recognizing organizations (*C8*), since it includes entities in the type of *Organization* in the output. While the broad version also recognizes more locations in other categories, this is often because those locations are considered as *Organization* by the Stanford NER in general. For example, centers, such as *"Walnut Hill Rec Center"* and *"Delco Center"*, in our category *C7* are considered as *Organization*

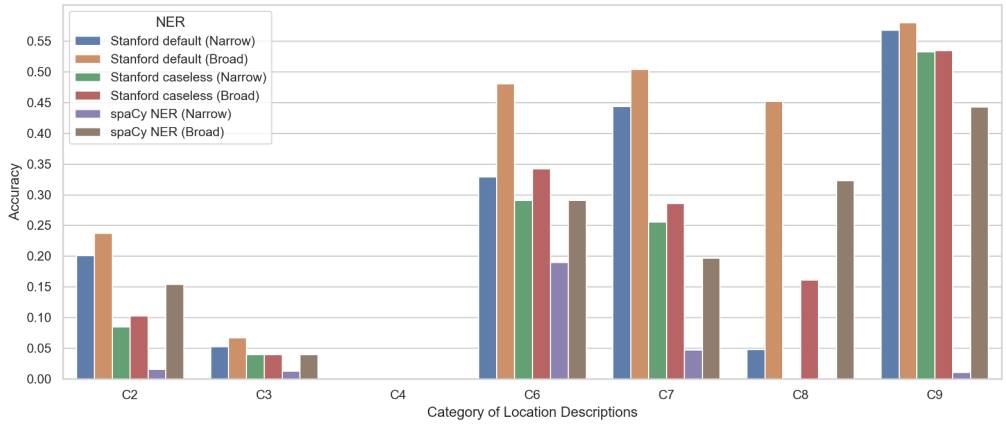


Figure 3 Performances of the NER tools on the different categories of location descriptions.

by the Stanford NER. Third, five out of the six NER tools recognize fair percentages of administrative unit names (the category of *C9*), such as *"Houston"* and *"Texas"*. The only exception is the narrow version of the spaCy NER, since it only recognizes the names of natural geographic features. Despite the fair performances of the NER tools, this category of locations has limited value for disaster rescue purposes. Fourth, the performances of the NER tools on street names (*C2*) and highway names (*C2*) are low, but these location description are usually critical for locating the people who need help. A more detailed examination of the result shows that these NER tools often miss the street names that contain numbers, such as *26th St* and *31st Ave*. Similarly, they miss the highway names, such as *I-10* and *Hwy 90*, in which numbers are used even more frequently than in street names. Finally, these NER tools have only low to fair performances on natural features (e.g., rivers and bayous; *C6*) and other human-made features (e.g., parks; *C7*).

In sum, the experiment results suggest that existing NER tools have limited performance in recognizing locations, especially sub-city level locations, from disaster-related tweets. They do not have the capability of recognizing location descriptions that consist of multiple entities, such as house number addresses, road intersections, and multiple areas, and largely fail on highways, highway exits, and the street names that contain numbers. As a result, there is a need for developing more effective toponym recognition models that can recognize these location descriptions from tweets.

5.2 Toponym resolution

The toponym resolution components of existing geoparsers use a variety of strategies to resolve ambiguity and geo-locate place names. These strategies include heuristics based on the population of cities (e.g., a toponym is resolved to the place with the highest population), the co-occurrences of related place names (e.g., the names of higher administrative units), and others [1, 20]. There are also methods that create a grid tessellation covering the surface of the Earth and calculate the probability of a place name to be located in each grid [7, 11]. However, existing toponym resolution components focus more on the task of disambiguating and geo-locating place names at a world scale, such as understanding which *"Washington"* the place name is referring to, given the many places named *"Washington"* in the world.

By contrast, the task of resolving locations described in disaster related tweets has different characteristics. First, these locations are generally at sub-city level, such as roads and house number addresses. Unlike cities, these fine-grained locations are often not associated with

populations. This makes it difficult to apply existing toponym resolution heuristics based on population. Second, given these location descriptions are about a disaster-affected local area, the task of toponym disambiguation becomes easier. While there can still be roads having the same name within the same city, the number of places that share the same name decreases largely (e.g., there is no need to disambiguate over 80 different *"Washington"*s when we focus on a local area). Third, point-based location representations typically returned by existing geoparsers become insufficient. We may need lines or polygons, in addition to points, to provide more accurate representation for the described locations.

Given that existing toponym resolution strategies are not applicable to the task of resolving location descriptions in disaster-related tweets, we discuss what are needed if we are going to develop a toponym resolution model for handling this task. First, it is necessary to have a local gazetteer that focuses on the disaster affected area and has detailed geometric representation (i.e., points, lines, and polygons) of the geographic features. Compared with the typically used GeoNames gazetteer, a local gazetteer serves two roles: (1) it reduces place name ambiguity by limiting place names to the disaster-affected area; and (2) it provides detailed spatial footprints for representing fine-grained locations. Such a local gazetteer could be constructed by conflating OpenStreetMap data, the GeoNames data within the local region, and authoritative geospatial data from mapping agencies. Second, we need a geocoder embedded in the toponym resolution model to handle house number addresses. Successfully embedding such a geocoder also requires the local gazetteer to contain house number data along with the roads and streets. Third, additional natural language processing methods are necessary to identify the spatial relations among the multiple locations described in the same tweet. This is especially important for location descriptions in Categories *C4*, *C5*, and *C10* when we need to locate the intersection of two roads (or a road and a river), the exit of a highway, or a combination of two regions. In addition, the NLP methods can help the toponym resolution model determine which geometric representation to use. Consider two possible tweets *"Both Allen Parkway and Memorial Dr are flooded"* and *"Flooding at the intersection of Allen Parkway and Memorial Dr"*. While the same roads are described in these two tweets, the ideal geometric representation for them should be different.

6 Conclusions and future work

Hurricane Harvey is a major natural disaster that devastated the Houston metropolitan area in 2017. Hurricane Harvey also witnessed the wide use of social media, such as Twitter, by the disaster-affected people to seek help and share information. Given the increasing popularity of social media among the general public, they are likely to be used in future disasters. One challenge in using social media messages for supporting disaster response is automatically and accurately extracting locations from these messages. In this work, we examine a sample of tweets sent out during Hurricane Harvey in order to understand how people describe locations in the context of a natural disaster. We identify ten categories of location descriptions, ranging from house number addresses and highway exits to human-made features and multiple regions. We find that under emergency situations people tend to describe their locations precisely by providing exact house numbers or clear road intersection information. We further conduct experiments to measure the performances of existing tools for geoparsing these Harvey tweets. Limitations of these tools are identified, and we discuss possible approaches to developing more effective models. In addition to social media messages, other approaches, such as *what3words* (`what3words.com`), could also be promoted to help people communicate their locations in emergency situations. *What3words* could be especially useful in geographic areas that lack standard addresses; meanwhile, it will also require people to have some familiarity with the system and install the relevant app.

A number of research topics can be pursued in the near future. First, while we have gained some understanding on how people describe locations during a natural disaster, it is limited to English language and within the culture of the United States. People speaking other languages or in other countries and cultures are likely to describe locations in different ways that need further investigation. Second, we can move forward and experiment possible approaches to developing models for recognizing and geo-locating the location descriptions in tweets posted during disasters. Examples include toponym recognition models that can correctly recognize highways and streets whose names contain numbers, and toponym resolution models that can correctly interpret the spatial relations of the multiple locations described in the same tweet. Finally, location extraction is only one part (although an important part) of the whole pipeline for deriving useful information from social media messages. Future research can integrate location extraction with other methods, such as those for verifying information veracity and classifying message purposes, to help disaster responders and volunteer organizations make more effective use of social media and reach the people in need.

References

1 Beatrice Alex, Kate Byrne, Claire Grover, and Richard Tobin. Adapting the Edinburgh geoparser for historical georeferencing. *International Journal of Humanities and Arts Computing*, 9(1):15–35, 2015.

2 Einat Amitay, Nadav Har'El, Ron Sivan, and Aya Soffer. Web-a-where: geotagging web content. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 273–280, New York, NY, USA, 2004. ACM.

3 Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. Tweedr: Mining twitter to inform disaster response. In *ISCRAM*, 2014.

4 Andrew Crooks, Arie Croitoru, Anthony Stefanidis, and Jacek Radzikowski. # earthquake: Twitter as a distributed sensor system. *Transactions in GIS*, 17(1):124–147, 2013.

5 Joao Porto De Albuquerque, Benjamin Herfort, Alexander Brenning, and Alexander Zipf. A geographic approach for combining social media and authoritative data towards identifying useful information for disaster management. *International Journal of Geographical Information Science*, 29(4):667–689, 2015.

6 Bertrand De Longueville, Robin S Smith, and Gianluca Luraschi. "OMG, from here, I can see the flames!" a use case of mining location based social networks to acquire spatio-temporal data on forest fires. In *Proceedings of the 2009 International Workshop on Location Based Social Networks*, pages 73–80, 2009.

7 Grant DeLozier, Jason Baldridge, and Loretta London. Gazetteer-independent toponym resolution using geographic word profiles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2382–2388, Palo Alto, CA, USA, 2015. AAAI Press.

8 Nuno Freire, José Borbinha, Pável Calado, and Bruno Martins. A metadata geoparsing system for place name recognition and resolution in metadata records. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 339–348, New York, NY, USA, 2011. ACM.

9 Judith Gelernter and Shilpa Balaji. An algorithm for local geoparsing of microtext. *GeoInformatica*, 17(4):635–667, 2013.

10 Judith Gelernter and Nikolai Mushegian. Geo-parsing messages from microtext. *Transactions in GIS*, 15(6):753–773, 2011.

11 Milan Gritta, Mohammad Taher Pilehvar, and Nigel Collier. Which melbourne? augmenting geocoding with maps. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1285–1296, 2018.

12 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.

13 Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22nd international conference on World Wide Web*, pages 729–736, 2013.

14 Qunying Huang and Yu Xiao. Geographic situational awareness: mining tweets for disaster preparedness, emergency response, impact, and recovery. *ISPRS International Journal of Geo-Information*, 4(3):1549–1568, 2015.

15 Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. AIDR: Artificial intelligence for disaster response. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 159–162, 2014.

16 Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, and Patrick Meier. Extracting information nuggets from disaster-related messages in social media. In *ISCRAM*, 2013.

17 Christopher B. Jones and Ross S. Purves. Geographical information retrieval. *International Journal of Geographical Information Science*, 22(3):219–228, 2008.

18 Morteza Karimzadeh. Performance evaluation measures for toponym resolution. In *Proceedings of the 10th Workshop on Geographic Information Retrieval*, page 8, New York, NY, USA, 2016. ACM.

19 Morteza Karimzadeh, Wenyi Huang, Siddhartha Banerjee, Jan Oliver Wallgrün, Frank Hardisty, Scott Pezanowski, Prasenjit Mitra, and Alan M MacEachren. Geotxt: a web api to leverage place references in text. In *Proceedings of the 7th workshop on geographic information retrieval*, pages 72–73, New York, NY, USA, 2013. ACM.

20 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.

21 Yury Kryvasheyeu, Haohui Chen, Nick Obradovich, Esteban Moro, Pascal Van Hentenryck, James Fowler, and Manuel Cebrian. Rapid assessment of disaster damage using social media activity. *Science advances*, 2(3):e1500779, 2016.

22 Jing Li, Keri K Stephens, Yaguang Zhu, and Dhiraj Murthy. Using social media to call for help in Hurricane Harvey: Bonding emotion, culture, and community relationships. *International Journal of Disaster Risk Reduction*, 38:101212, 2019.

23 Zhenlong Li, Cuizhen Wang, Christopher T Emrich, and Diansheng Guo. A novel approach to leveraging social media for rapid flood mapping: a case study of the 2015 south carolina floods. *Cartography and Geographic Information Science*, 45(2):97–110, 2018.

24 Fei Liu, Maria Vasardani, and Timothy Baldwin. Automatic identification of locative expressions from social media text: A comparative analysis. In *Proceedings of the 4th International Workshop on Location and the Web*, pages 9–16, 2014.

25 Alan M MacEachren, Anuj Jaiswal, Anthony C Robinson, Scott Pezanowski, Alexander Savelyev, Prasenjit Mitra, Xiao Zhang, and Justine Blanford. Senseplace2: Geotwitter analytics support for situational awareness. In *Visual analytics science and technology (VAST), 2011 IEEE conference on*, pages 181–190. IEEE, 2011.

26 Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

27 Stuart E Middleton, Lee Middleton, and Stefano Modafferi. Real-time crisis mapping of natural disasters using social media. *IEEE Intelligent Systems*, 29(2):9–17, 2013.

28 Volodymyr V Mihunov, Nina SN Lam, Lei Zou, Zheye Wang, and Kejin Wang. Use of twitter in disaster rescue: lessons learned from Hurricane Harvey. *International Journal of Digital Earth*, pages 1–13, 2020.

29 Dhiraj Murthy and Scott A Longwell. Twitter and disasters: The uses of twitter during the 2010 pakistan floods. *Information, Communication & Society*, 16(6):837–855, 2013.

30 Nastaran Pourebrahim, Selima Sultana, John Edwards, Amanda Gochanour, and Somya Mohanty. Understanding communication dynamics on twitter during natural disasters: A case study of hurricane sandy. *International journal of disaster risk reduction*, 37:101176, 2019.

31 Ross S Purves, Paul Clough, Christopher B Jones, Mark H Hall, Vanessa Murdock, et al. Geographic information retrieval: Progress and challenges in spatial search of text. *Foundations and Trends® in Information Retrieval*, 12(2-3):164–318, 2018.

32 J Rexiline Ragini, PM Rubesh Anand, and Vidhyacharan Bhaskar. Big data analytics for disaster response and recovery through sentiment analysis. *International Journal of Information Management*, 42:13–24, 2018.

33 Maya Rhodan. Hurricane Harvey: The U.S.'s first social media storm. *Time Magazine*, 2017. URL: `https://time.com/4921961/hurricane-harvey-twitter-facebook-social-/`.

34 Deepa Seetharaman and Georgia Wells. Hurricane Harvey victims turn to social media for assistance. *The Wall Street Journal*, 2017. URL: `https://www.wsj.com/articles/hurricane-harvey-victims-turn-to-social-media-for-assistance-1503999001`.

35 Lauren Silverman. Facebook, twitter replace 911 calls for stranded in houston. *National Public Radio*, 2017. URL: `https://www.npr.org/sections/alltechconsidered/2017/08/28/546831780/texas-police-and-residents-turn-to-social-media-to-communicate-amid-harvey`.

36 Luke Sloan, Jeffrey Morgan, William Housley, Matthew Williams, Adam Edwards, Pete Burnap, and Omer Rana. Knowing the tweeters: Deriving sociologically relevant demographics from twitter. *Sociological research online*, 18(3):1–11, 2013.

37 Erik F Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics, 2003.

38 Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

39 Jan Oliver Wallgrün, Morteza Karimzadeh, Alan M. MacEachren, and Scott Pezanowski. Geocorpora: building a corpus to test and train microblog geoparsers. *International Journal of Geographical Information Science*, 32(1):1–29, 2018.

40 Jimin Wang and Yingjie Hu. Enhancing spatial and textual analysis with eupeg: An extensible and unified platform for evaluating geoparsers. *Transactions in GIS*, 23(6):1393–1419, 2019.

41 Ruo-Qian Wang, Huina Mao, Yuan Wang, Chris Rae, and Wesley Shaw. Hyper-resolution monitoring of urban flooding with social media and crowdsourcing data. *Computers & Geosciences*, 111:139–147, 2018.

42 Zheye Wang, Xinyue Ye, and Ming-Hsiang Tsou. Spatial, temporal, and content analysis of twitter for wildfire hazards. *Natural Hazards*, 83(1):523–540, 2016.

43 Manzhu Yu, Qunying Huang, Han Qin, Chris Scheele, and Chaowei Yang. Deep learning for real-time social media text classification for situation awareness–using Hurricanes Sandy, Harvey, and Irma as case studies. *International Journal of Digital Earth*, pages 1–18, 2019.

44 Wei Zhang, Gabriele Villarini, Gabriel A Vecchi, and James A Smith. Urbanization exacerbated the rainfall and flooding caused by Hurricane Harvey in Houston. *Nature*, 563(7731):384–388, 2018.

45 Lei Zou, Nina SN Lam, Heng Cai, and Yi Qiang. Mining twitter data for improved understanding of disaster resilience. *Annals of the American Association of Geographers*, 108(5):1422–1441, 2018.

46 Lei Zou, Nina SN Lam, Shayan Shams, Heng Cai, Michelle A Meyer, Seungwon Yang, Kisung Lee, Seung-Jong Park, and Margaret A Reams. Social and geographical disparities in twitter use during Hurricane Harvey. *International Journal of Digital Earth*, 12(11):1300–1318, 2019.

Introducing Diversion Graph for Real-Time Spatial Data Analysis with Location Based Social Networks

Sameera Kannangara
School of Computing and Information Systems, The University of Melbourne, Australia
kannangarad@student.unimelb.edu.au

Hairuo Xie
School of Computing and Information Systems, The University of Melbourne, Australia
xieh@unimelb.edu.au

Egemen Tanin
School of Computing and Information Systems, The University of Melbourne, Australia
etanin@unimelb.edu.au

Aaron Harwood
School of Computing and Information Systems, The University of Melbourne, Australia
aharwood@unimelb.edu.au

Shanika Karunasekera
School of Computing and Information Systems, The University of Melbourne, Australia
karus@unimelb.edu.au

Abstract

Neighbourhood graphs are useful for inferring the travel network between locations posted in the Location Based Social Networks (LBSNs). Existing neighbourhood graphs, such as the Stepping Stone Graph lack the ability to process a high volume of LBSN data in real time. We propose a neighbourhood graph named Diversion Graph, which uses an efficient edge filtering method from the Delaunay triangulation mechanism for fast processing of LBSN data. This mechanism enables Diversion Graph to achieve a similar accuracy level as Stepping Stone Graph for inferring travel networks, but with a reduction of the execution time of over 90%. Using LBSN data collected from Twitter and Flickr, we show that Diversion Graph is suitable for travel network processing in real time.

2012 ACM Subject Classification Information systems → Geographic information systems

Keywords and phrases moving objects, shortest path, graphs

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.7

Funding This research is funded in part by the Defence Science and Technology Group, Edinburgh, South Australia, under contract MyIP:6104.

1 Introduction

Location Based Social Networks (LBSNs) contain a large volume of location information posted by the users. The location data collected from LBSN can be further processed to understand various aspects of users' lives [19, 20]. LBSN data can be processed to infer the travel network between the posted locations [8]. Inferring a travel network is to find a set of edges between the posted locations or a subset of the locations so that a path can be found between any pair of the locations in the network. Processing LBSN data for such purposes is difficult due to the scale of the data to be processed. We are interested in efficient methods for inferring travel networks with LBSN data.

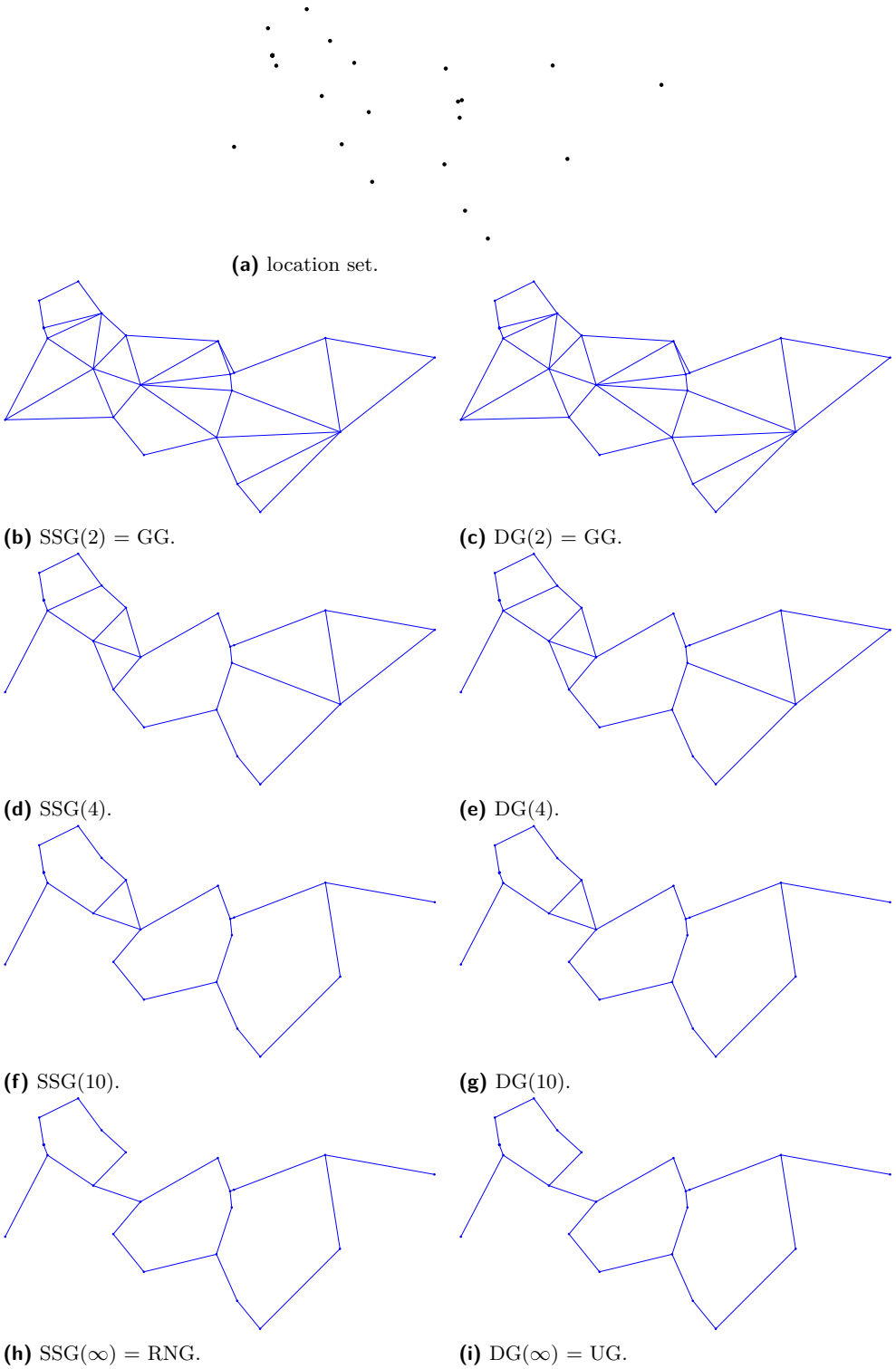
Location data collected from LBSNs is usually in the form of GPS points. Distance-based connected neighbourhood graphs have been used for inferring the relationship between a set of distinct GPS points [5]. Neighbourhood graphs are also called proximity graphs, where edges between the points are built based on certain spatial relationship between the points. Delaunay Triangulation (DT) is a well-known distance-based connected neighbourhood graph. Gabriel Graph (GG) [7], Relative Neighbourhood Graph (RNG) [17] and Urquhart Graph (UG) [18] are extended from DT for movement network analysis. For example, Figure 1 (a) represents locations collected from Twitter relating to a state election. Figures 1 (c,h,i) represent GG, RNG and UG skeletons, which are the geometric realization of neighbourhood graphs and show the geometric shape of the point set.

Unlike the aforementioned graphs, there is a type of graphs called variable graphs, which can generate a spectrum of possible skeletons based on different values of given parameters. Therefore, the variable graphs are making them more versatile than other types of graphs. In the rest of the paper, we specify the parameters used by variable graphs in the name of the graphs. The Shortest Path Graph ($SPG(t)$) [6] and the Stepping Stone Graph ($SSG(d)$) [8] are two commonly used variable graphs, built on the idea that the shortest path through the inferred edges can be aligned with the shortest path through the imprecise region represented by the point set. While $SPG(t)$ considers the shortest path over all points when inferring edges, $SSG(d)$ only considers the shortest path that goes through points within the relative neighbourhood between two points. Due to this difference, the travel networks created with $SSG(d)$ are more similar to real world travel networks. Both $SPG(t)$ and $SSG(d)$ can generate various graph skeletons based on a single parameter. Figure 1 (b,d,f,h) represent different $SSG(d)$ skeletons of the same point set based on different parameter settings.

Although existing neighbourhood graphs can process LBSN data with a few hundred locations, they are not suitable for large datasets due to the long running times. With the widespread use of GPS-enabled mobile phones, the size of LBSN datasets tends to be significantly large. Therefore we need to investigate efficient methods to infer travel networks based on the location data collected from LBSNs.

In this paper, we propose a new type of variable graph, which we refer to as the Diversion Graph ($DG(d)$). The skeletons inferred by $DG(d)$ are likely to be close to human perception of the corresponding point set. We show in our experiments that $DG(d)$ is easier and faster to build than $SSG(d)$, and gives similar results in processing certain spatial queries as $SSG(d)$. Similar to $SSG(d)$, $DG(d)$ is defined on top of DT and uses Diversion Neighbourhood ($DN(d)$) to cull edges from DT. Instead of checking all the points that lie in $DN(d)$ between two points, $DG(d)$ only considers points in the neighbouring Delaunay triangles of the edge that is considered for culling. This approach is suitable for inferring travel networks with LBSN data as we assume that the social network data gives us partial data per individual user in terms of its path but with a good picture of where people could be in an event in a city. As explained with the definitions of the $DG(d)$ (Section 3.1), for all endpoint pairs the value of d indicates the preference of inferring a longer alternative path with less distance between all point pairs on the path compared to the direct distance between the endpoint pair. This is useful when we have a very dense point set to cull some connections. Similar to both $SPG(t)$ and $SSG(d)$, as d increases, the number of edges in $DG(d)$ monotonically decreases and therefore the path length between any two non-adjacent points in the skeleton monotonically increases. It is also important to note that GG is a special case of $DG(d)$ when $d = 2$.

We use publicly available LBSN data to evaluate the performance of $DG(d)$ and $SSG(d)$ for inferring travel networks. We show that $DG(d)$ performs as well as $SSG(d)$ in terms of the quality of the inferred network but $DG(d)$ achieves significantly faster execution times



(a) location set.

(b) SSG(2) = GG.

(c) DG(2) = GG.

(d) SSG(4).

(e) DG(4).

(f) SSG(10).

(g) DG(10).

(h) SSG(∞) = RNG.

(i) DG(∞) = UG.

Figure 1 SSG(d) and DG(d) skeletons created on a subset of locations from Twitter data set. Note that two skeletons in a row are created using two algorithms, but exhibit the same graph structure.

than $SSG(d)$. In fact, $DG(d)$ takes less than 10% of the time required to infer a movement network than $SSG(d)$. $DG(d)$ contains a few more edges (less than 2% of the total number of edges [1]) compared to $SSG(d)$. However, compared to the time advantage of $DG(d)$, the negative impact of the additional edges in negligible. We observe that the resulting $DG(d)$ and $SSG(d)$ are very similar in terms of their shape and topology.

2 Related Work

In this section, we present the related work in two categories. The first category, LBSN data processing, presents systems and techniques used to process LBSN data. The second category, neighbourhood graphs, provides an overview of neighbourhood graphs related to the proposed Diversion Graph.

2.1 LBSN Data Processing

MacEachren et al. develop an LBSN analysis system SensePlace2 which is used to query and visualize social media data over an interactive map interface [11]. Chae et al. develop another systems for analysing public behaviour using LBSN data [4]. Geospatial heatmaps are used in both systems to provide a summarized view of the spatial distribution of LBSN posts. Many LBSN-based analytics systems support real time processing of LBSN data. For example, RAPID is a real-time analytics platform for interactive data mining [10]. It streams social media data and processes it to generate real time results. There are many types of analytics that can be performed by the systems like RAPID. For example, the detection of the most popular path followed by the users and the extraction of movement corridors [8]. When performing such analytics at real time it is important to have a neighbourhood graph like the proposed Diversion Graph that generates high quality results while minimizing execution time.

2.2 Neighbourhood Graphs

Neighbourhood graphs infer edges between points based on a neighbourhood defined on the points [3]. On two dimensional space neighbourhood represents an area, which can be defined per point, per point pair or per all points in the sample. Neighbourhood graphs that infer edges based on the emptiness (absence of other points within a region) of the neighbourhood surrounding the endpoint pair of the edge are referred as Empty Region Graphs (ERG) [3].

Gabriel Graph (GG) [7] is a static ERG, first proposed as a tool for geographic variation analysis. GG uses the closed circle (a circle where inferring edge becomes a diameter) as the empty region for inferring edges. Relative Neighbourhood Graph (RNG) [17] is another ERG, which uses open lune as the empty regions of inferring its edges. RNG can infer a structure close to human perception of a point set [17]. Both GG and RNG are useful for analysing the shape of a point set.

Urquhart Graph (UG) [18] was first proposed for fast construction of RNG. It was later proved that the UG is not always similar to RNG [16], but UG only differs from RNG by 2% maximally. Therefore UG can be seen as a faster method to approximate RNG [1]. We are combining the thought process behind UG creation and the Diversion neighbourhood of the $SSG(D)$ to create $DG(d)$.

Delaunay Triangulation (DT) is a Triangulated Irregular Network (TIN) with many benefits. It serves as a planar graph which has similar properties as the complete graph. For this reason, it can be used as the starting graph for inferring many other planar graphs. Due

to having a low spanning ratio and faster inferring it is used in many travel network analysis problems. Note that $MST \subseteq RNG \subseteq UG \subseteq GG \subseteq DT$. As later shown in properties section, $DG(2) = GG$ and $DG(\infty) = UG$.

Mark de Berg et al. proposed Shortest Path Graph ($SPG(t)$) as a base skeleton for delineating mechanism to identify boundary and cavities within an imprecise region [6]. They show that $SPG(2)$ is better for delineating imprecise regions compared to both Kernel Density Estimation (KDE) and GG [6]. $SPG(t)$'s global evaluation criteria is highlighted as the main reason for its success. However, the quality of results generated using $SPG(t)$ heavily depends on certain parameter settings.

$SSG(d)$ follows the general intuition used in proposing $SPG(t)$, which is to roughly align paths in the graph with paths in the imprecise region. Rather than using global criteria as used in $SPG(t)$, $SSG(d)$ uses local criteria making it faster than $SPG(t)$ and more effective for movement analysis [8].

When the value of the parameter in SPG and SSG approaches infinity, $SPG(t)$ converges to MST and $SSG(d)$ converges to RNG, and our proposed $DG(d)$ converges to UG (Theorem 6) which is a close approximation of RNG. Since UG is a close approximation of RNG, structures generated using $DG(d)$ are closely related to the human perception of the point set. It should be noted that $DG(d)$ may contain some additional edges compared to the $SSG(d)$ with the same d value. However, due to the relaxed nature of evaluation criteria for $DG(d)$, inferring the graph takes much less time compared to inferring $SSG(d)$ or $SPG(t)$.

3 Diversion Graph

Given a set of points, a Diversion Graph ($DG(d)$) connects the points in a traversable manner.

3.1 Definitions

We construct $DG(d)$ in the form of an undirected graph $G(V, E)$ where $V \subseteq \mathbb{R}^2$ represents a given point set and E represents the inferred edges between the points. An edge between two endpoints $p, q \in V$ is represented as $pq \in E$. Length l_{pq} represents the Euclidean distance between the two points.

As $DG(d)$ is defined using Delaunay Triangulation (DT), let us briefly iterate a useful property of DT. DT is a triangulation of a point set, in which each triangle's circumcircle does not contain any other points other than triangle's vertices. Also, DT is the dual of Voronoi diagram. Our proposed graph $DG(d)$ is evaluated by removing some edges from the DT.

▶ **Definition 1** (Diversion Graph). *For $V \subseteq \mathbb{R}^2$, the Diversion Graph of V at $d \in \mathbb{R} : d \geq 2$, denoted $DG(V, d)$ or simply $DG(d)$, is an undirected graph containing a subset of $DT(V)$ such that for each edge $pq \in DT(V)$:*

$$pq \text{ is not an edge of } DG(V, d) \text{ iff } l^d_{pz} + l^d_{zq} \leq l^d_{pq},$$

where z is the other point in a Delaunay triangle where pq is an edge.

By this definition in common terms, $DG(d)$ is the graph created by removing edges pq from DT if and only if $l^d_{pz} + l^d_{zq} \leq l^d_{pq}$ where z is a vertex of a Delaunay triangle where pq is an edge. Therefore inherently $DG(d)$ is a subgraph of DT. We explore the properties of $DG(d)$ in the next section.

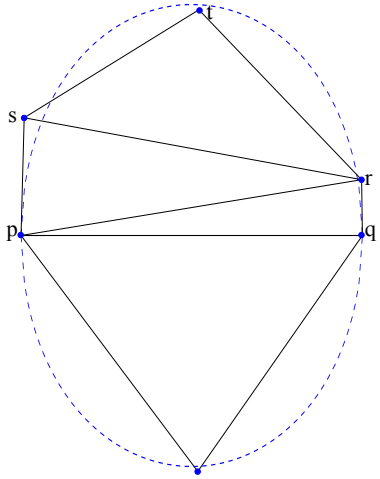


Figure 2 Counter example to show $DG(d)$ does not always equal to $SSG(d)$. Dashed line shows diversion neighbourhood at $d = 4$ ($DN(4)$). Points p and q have equal y coordinates. Both points r and s reside out side of $DN(4)$. Point t lies inside shown $DN(4)$.

3.2 Diversion Graph Properties

Since $DG(d)$ is created by removing edges from DT, we can state the following theorem.

▶ **Theorem 2.** *For* $2 \leq d, DG(d) \subseteq DT$.

Consider the case $DG(2)$ against GG. By comparing definitions of the two graphs we can state the following theorem.

▶ **Theorem 3.** $DG(2) \equiv GG$.

Proof. Consider the definition of GG from [12]. The vertices $p, q \in V$ are least squares adjacent forming the edge $pq \in GG$ iff

$$l_{pq}^2 < l_{pz}^2 + l_{zq}^2 \forall z \in V \setminus \{p, q\}.$$

Furthermore, in the same paper [12] it is proven that the GG can be extracted from DT by evaluating the above inequality on each triangle, for each edge. Now let us look at $DG(2)$ definition. It is extracted from DT by removing edges pq iff $l_{pz}^2 + l_{zq}^2 \leq l_{pq}^2$, for each z which are other points of the Delaunay triangles where pq is an edge. Since both GG and $DG(2)$ are evaluated from DT using the same inequality, they are equivalent. ◀

Since equation used in $DG(d)$ definition is same as the diversion neighbourhood definition [8], one may think $DG(d)$ and $SSG(d)$ are the same thing. As shown in the following theorem, $SSG(d) \subseteq DG(d)$.

▶ **Theorem 4.** *For* $2 \leq d, SSG(d) \subseteq DG(d)$.

Proof. Consider the five points p, q, r, s, t in Figure 2. Their Delaunay triangulation is shown in solid straight lines. Points p and q have equal y coordinates. The dashed line indicates diversion neighbourhood at $d = 4$ ($DN(4)$) between p and q. Both points r and s reside outside of $DN(4)$. Point t lies inside shown $DN(4)$. Let us consider inclusion of edge pq in $DG(4)$ and $SSG(4)$. Since t is within the $DN(4)$ of pq, pq will not be an edge of $SSG(4)$. However, since $DG(4)$ considers only points in neighbouring triangles of pq, it only considers point r when considering the inclusion of pq. Since r is outside the $DN(4)$ of pq, pq becomes an edge of $DG(4)$. Therefore, for $2 < d, SSG(d) \subseteq DG(d)$. ◀

In fact, $DG(d)$ can contain more edges than $SSG(d)$. Therefore we can state the following corollary.

▶ **Corollary 5.** *For $2 < d, DG(d)$ is not always equal to $SSG(d)$.*

Let us consider behaviour of $DG(d)$ when $d \to \infty$.

▶ **Theorem 6.** *As $d \to \infty, DG(d) \to UG$.*

Proof. When $d \to \infty$, by $DG(d)$ definition for an edge pq to be removed from DT, both other edges of the neighbouring Delaunay triangles only needs to be shorter than pq. In other words, if pq is the longest edge in a Delaunay triangle it will be removed from $DG(d)$ when $d \to \infty$. UG is created from DT by removing the longest edge of each Delaunay triangle. Therefore, as $d \to \infty, DG(d) \to UG$. ◀

▶ **Theorem 7.** *For $2 \leq d, DG(d)$ is planar and connected.*

Proof. Since for $2 \leq d, DG(d) \subseteq DT$, $DG(d)$ is planar. Inequality used in $DG(d)$ is the same as the diversion neighbourhood of $SSG(d)$. In [8] it is shown that diversion neighbourhood does not get bigger than open lune neighbourhood. Open lune neighbourhood is proven as the tight neighbourhood that ensures a connected edge embedding in empty region graphs in [3]. Therefore $DG(d)$ is connected. Combining these arguments we can say for $2 \leq d, DG(d)$ is planar and connected. ◀

Next we consider the relationship between two $DG(d)$s as d increases.

▶ **Theorem 8.** *For $2 \leq d \leq d', DG(d') \subseteq DG(d)$*

Proof. Define the *edge weight* of pq with respect to d' as $l_{pq}^{d'}$, for some $d' \geq 2$. Assume that for all $z \in \Lambda(pq) \setminus \{p, q\}$, $l_{pz}^{d'} + l_{zq}^{d'} > l_{pq}^{d'}$, where $\Lambda(pq)$ is the set of points in neighbouring Delaunay triangles of pq. In this case, pq is an edge in $DG(d')$. Now we show that for $d \leq d'$, pq is also an edge in $DG(d)$. Let us write $d = d'\,\epsilon$ where $\frac{2}{d'} \leq \epsilon \leq 1$. Then we need to show that:

$$
\begin{aligned}
l_{pz}^{d'\,\epsilon} + l_{zq}^{d'\,\epsilon} &> l_{pq}^{d'\,\epsilon} \qquad (1)\\
\frac{l_{pz}^{d'\,\epsilon} + l_{zq}^{d'\,\epsilon}}{l_{pq}^{d'\,\epsilon}} &> 1\\
\left(\frac{l_{pz}}{l_{pq}}\right)^{d'\,\epsilon} + \left(\frac{l_{zq}}{l_{pq}}\right)^{d'\,\epsilon} &> 1\\
\left(\left(\frac{l_{pz}}{l_{pq}}\right)^{d'\,\epsilon} + \left(\frac{l_{zq}}{l_{pq}}\right)^{d'\,\epsilon}\right)^{\frac{1}{\epsilon}} &> 1
\end{aligned}
$$

Since the function $x \mapsto x^{\beta}$ is subadditive for $\beta \geq 1$ then:

$$
\left(\left(\frac{l_{pz}}{l_{pq}}\right)^{d'\,\epsilon} + \left(\frac{l_{zq}}{l_{pq}}\right)^{d'\,\epsilon}\right)^{\frac{1}{\epsilon}} \geq \left(\frac{l_{pz}}{l_{pq}}\right)^{d'} + \left(\frac{l_{zq}}{l_{pq}}\right)^{d'}.
$$

We know the right hand side is greater than 1 due to our initial assumption and therefore Eq. 1 is true. Therefore pq is also an edge in $DG(d)$ and this completes the proof. ◀

Algorithm 1 Create $DG(d)$.

Input: V - Filtered locality set
d - Configuration parameter
Output: $DG(d)$

```
1  DT ← create Delaunay Triangulation of V
2  initialize DG(d) to empty set
3  foreach (Edge pq : pq ∈ DT) do
4      foreach (Point z : z ∈ Λ(pq) \ {p, q}) do
5          if (l^d_pq < l^d_pz + l^d_zq) then
6              Add pq to DG(d)
7          end
8      end
9  end
10 return DG(d)
```

3.3 Algorithms

In this section, we present an efficient algorithm to compute $DG(d)$. Since $DG(d)$ is defined based on DT we can use DT as the starting graph to compute $DG(d)$. There are two approaches we can use to compute $DG(d)$ using DT. One approach is to process DT as triangles and check each edge of the triangle for removal from DT. The second approach is to process DT as a set of edges and evaluate each edge against the points in the neighbouring Delaunay triangles to check whether they belong in $DG(d)$. The approach we are presenting in this paper is the second approach which evaluates edges to check their membership of $DG(d)$, as it can be easily compared with the d-spectrum algorithm of the $SSG(d)$.

We propose a simple and effective algorithm to calculate $DG(d)$ (Algorithm 1). In the algorithm, $\Lambda(pq)$ is the set of points in neighbouring Delaunay triangles of pq. Each other point in the $\Lambda(pq) \setminus \{p, q\}$ are evaluated against pq to see whether pq is an edge of $DG(d)$. For simplicity, the condition in line 5 in Algorithm 1 is directly derived from the definition of $DG(d)$. However, it can be further improved by checking whether other edges connected with $\Lambda(pq)$ are longer than pq. The algorithm is readily parallelizable as there is no race condition between separate edge evaluations.

Let us consider the time complexity of the proposed algorithm for calculating $DG(d)$. In line 3, as DT has $\mathcal{O}(n)$ edges, the code between line 4 and line 8 runs $\mathcal{O}(n)$ times. As each edge pq has at most two neighbouring triangles, the code between line 5 and line 7 runs at most two times per edge. Line 5 is assumed to run in $\mathcal{O}(1)$ time. Therefore, the whole algorithm runs in $\mathcal{O}(n)$ time.

3.3.1 Improving Running Time of $SSG(d)$

Introduction of $DG(d)$ allows us to efficiently calculate $SSG(d)$ for a specific $2 \leq d$ value without calculating d-Spectrum [8]. Since $DG(d)$ is a super graph of $SSG(d)$ for $2 \leq d$, once $DG(d)$ is calculated we can use it to evaluate those edge using the triangle sweeping method presented in Algorithm 1 of [8]. As later shown in the experiments $DG(d)$ only contains a very small number of additional edges compared to $SSG(d)$. Therefore this is a very efficient method of computing $SSG(d)$.

However, it should be noted that computing $SSG(d)$ from $DG(d)$ may be slower for varying skeleton generation compared to using d-Spectrum. Since d-Spectrum pre-compute the minimum d-value necessary for an DT edge to be in the $SSG(d)$, varying skeleton generation takes less time. But for generating $SSG(d)$ for a specific $2 \leq d$ using $DG(d)$ is faster than creating d-Spectrum.

3.4 Applications

The proposed $DG(d)$ can be used in many applications detailed as follows.

3.4.1 Nearest Neighbour Queries

The $DG(d)$ graph structure can be used to search for the path to the nearest interesting locations from a given location. For example, this kind of query can be used to find the nearest exit gate in a park. We can use breadth first search starting from the query location and traverse the graph until a required interesting point is found.

Similarly, we can perform breadth first search on $DG(d)$ for finding k-nearest neighbours. Instead of stopping breadth first search when the first interesting point is found, it can be continued until k interesting points are found. As for the edge weights, we can use weights calculated in the section 3.4.4 according to the usage of edges. This will make sure that the most popular path to the nearest neighbour will be found. This approach can be extended to solve reverse nearest neighbour queries and group nearest neighbour queries as suggested in [9].

3.4.2 Refinement of Movement Corridors

Once $DG(d)$ is created using posted localities in LBSN data, user trajectories can be used to refine the created travel network. The approach for refining the travel network is as follows. For each consecutive location pair in user trajectories, the shortest path is determined using $DG(d)$. For each $DG(d)$ edge, the number of trajectories passed through that edge is recorded as a usage count (Definition 9). We can then represent movement corridors in the travel network based on the edges where the usage counts are higher than a given threshold.

▶ **Definition 9** (Usage count). *Assuming a path is a sequence of edges traversed by a trajectory trace, for all $pq \in E$, Usage Count of pq (denoted $UC(pq)$), is defined as the trajectory count,*

$$UC(pq) = |\{path : pq \in path\}|$$

One of the problems of using $DG(d)$, is that it does not consider the existence of obstacles. As trajectories do not appear on obstacles such as rivers, incorporating trajectory information into $DG(d)$ allows filtering edges not used by the trajectories. By filtering edges not used by trajectories, we are able to eliminate edges that do not represent user movement information. In summary, refined movement corridors calculated using $DG(d)$ is an edge subset of $DG(d)$ which are used by the trajectories for movement.

3.4.3 Inferring Road Network

The aforementioned approach for refining movement corridors can be used to infer the road network in an area where we do not have prior knowledge about the road networks. Ideally for this purpose, we need GPS locations published on the road network. The easiest way to obtain such information is to collect LBSN post published while travelling in vehicles. Using the GPS data in LBSN posts and associating the GPS points with trajectories, we can infer the road network in an area.

3.4.4 Most Popular Paths

After calculating usage counts of $DG(d)$ edges, the counts can be used to find the most popular path between locations. To find the most popular path, the edge weight in $DG(d)$ should reflect the popularity of the edges. We can use a shortest path algorithm to calculate the paths. The edge weights are defined in Definition 10. To ensure edges with more usage have lower weights, we divide the length of the edge by the usage count of that edge. For an edge with no usage, the edge length multiplied by a fixed value is used as the edge weight. After calculating edge weights in this manner, the Dijkstra's shortest path algorithm can be used to find the most popular path between two locations.

▶ **Definition 10** (Edge Weight). *For all $pq \in E$, weight of pq is defined as,*

$$weight(pq) = \begin{cases} l_{pq}/UC(pq), & \text{if } 0 < UC(pq). \\ l_{pq} \times C : 1 < C, & \text{if } UC(pq) = 0. \end{cases}$$

3.4.5 Other Applications

$DG(d)$ can be used in other applications such as tour recommendation, trajectory clustering and group movement detections [9]. For all these applications we need to process user trajectories after creating the initial graph structure to incorporate additional movement related information to the created graph skeleton.

4 Experiments

4.1 Data Sets

We conducted experiments on two real world LBSN datasets and one synthetic data set. The first real data set consists of geo-located posts collected from Twitter. It is collected from 06th March to 23rd April 2012, within a bounding box over Australia and New Zealand. It contains 724651 LBSN posts authored by 36639 users. For our analysis, an LBSN post is defined as a tuple containing four elements - userID, voluntarily generated textual content, timestamp and the location where the post was authored.

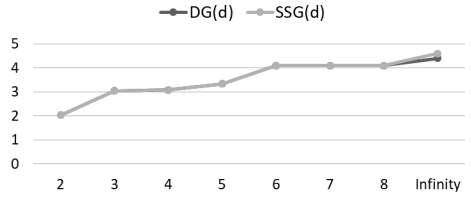
We used Yahoo! Flickr Creative Commons 100M (YFCC100M) dataset [15] as the second real dataset. It contains metadata such as user information, timestamp and location of 100 million photos and videos shared on Flickr. Only the entries with point geo-locations were used for our experiments. For our experiment, we use the localities around the Thames river in London from the YFCC100M data set.

The synthetic data set for our experiments was generated using SMARTS simulator [13]. We simulated vehicle movement in the Melbourne central business district (CBD) and collected GPS locations of the vehicles every 0.5 seconds. The data set used for our experiment contains 100000 GPS points.

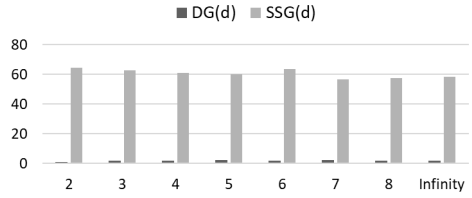
4.2 Implementation

To visualize the inferred neighbourhood skeletons, a visualization tool was implemented utilizing GeoTools[1] Java libraries. All the skeleton visualizations presented in this paper are generated using this tool. Both $DG(d)$ and $SSG(d)$ algorithms are implemented using Java 8. The datasets are stored in a MongoDB database.

[1] `http://www.geotools.org/`



(a) Spanning ratio with varying configuration parameter d.



(b) The time to generate the graphs in mili seconds with varying d.

Figure 3 Graphs depicting different properties between $DG(d)$ and $SSG(d)$.

To infer $DG(d)$, firstly, DT is created using SweepHull [14] algorithm. Then, $DG(d)$ is calculated using Algorithm 1. $SSG(d)$ is extracted from the planar d-Spectrum created using DT. We used numerical analysis to calculate d-value of an edge. More specifically, a Java method was implemented to perform Secant method[2] to approximate the d-Value. The same DT was reused for construction of $DG(d)$ and $SSG(d)$ with different d values. We selected 3 as the configuration value for d to compare resulting graphs generated using $DG(d)$ and $SSG(d)$ based on preliminary tests.

4.3 Results

4.3.1 Event Analysis

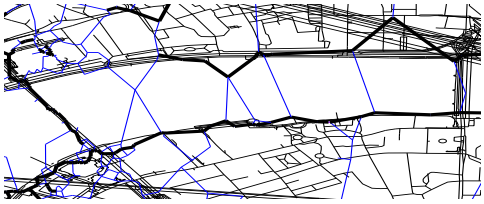
In this experiment, we use a Twitter dataset relating to Queensland state election 2012 [3]. All users participating in the event are there for a common reason and exhibit a similar movement pattern. We implement an LBSN post filtering technique used in [8], to filter posts relating to the election. For temporal bound of the dataset, we took the time period between 23rd and 26th of March 2012. As for the spatial bound, we considered a bounding box over the Queensland state. We consider all users who have posted with "#qldvote" hashtag within spatial and temporal bounds of the event. The data set contains, 1270 unique points after filtering the original Twitter data.

We generate skeletons using $DG(d)$ and $SSG(d)$ with different settings of d. The spanning ratio [2] of graphs are calculated with varying configuration parameters. Spanning ratio of a graph indicates the maximum ratio between the shortest path distance over the graph and direct distance between any point pair. Therefore, graphs with low spanning ratios are preferred to represent movement networks [2].

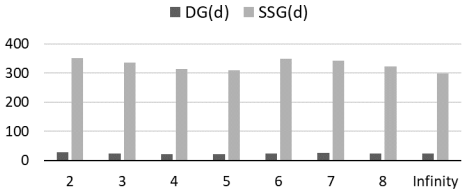
Figure 3 (a) shows the variation of spanning ratio as configuration parameter varies to demonstrate how the shortest path distances between locality pairs change. Both $DG(d)$ and $SSG(d)$ have a low and stable spanning ratio, making them suitable for movement analysis. Furthermore, both $DG(d)$ and $SSG(d)$ have the same spanning ratio when d is less or equal to 8.

The time taken to calculate skeletons of $DG(d)$ and $SSG(d)$ are shown in Figure 3 (b). Execution time for $DG(d)$ calculation is around 95% less compared to $SSG(D)$ for all configuration values. The relaxed criteria for culling edges in $DG(d)$ algorithm gives it a significant advantage in computation time.

[2] https://en.wikipedia.org/wiki/Secant_method
[3] https://en.wikipedia.org/wiki/Queensland_state_election,_2012

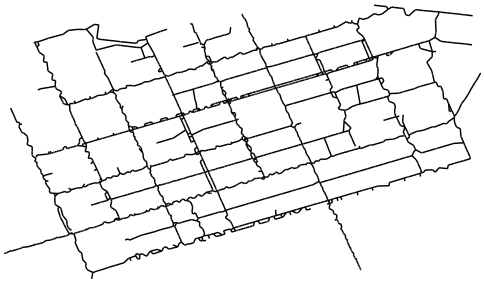


(a) Thick lines indicate refined movement corridors extracted using $DG(3)$ and $SSG(3)$.

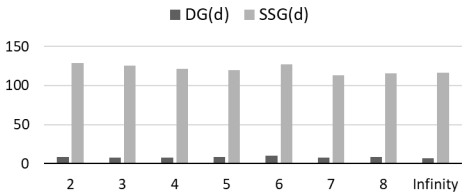


(b) Execution time in mili seconds to generate each graph with varying d.

Figure 4 Results of movement corridor refinement.



(a) Road network extracted using $DG(3)$ and $SSG(3)$.



(b) Execution time in mili seconds to generate each graph with varying d.

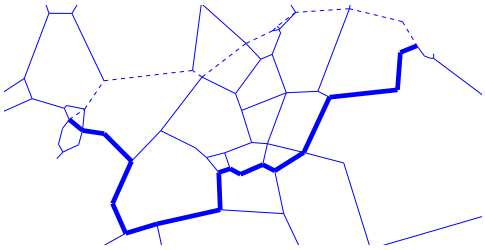
Figure 5 Results of road network extraction.

4.3.2 Movement Corridors Refinement

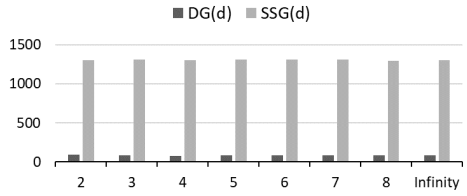
The refined movement corridors refer to the edges of the graph that are used by the users on the move. These edges are selected by aligning user trajectories along the graph edges using shortest path calculation. We analyse the refined movement corridors relating to the trajectories filtered from the YFCC100M dataset, which is collected from around the Thames river in London. We take locations posted over one month. The total number of locations is 6318. To represent the travel networks, $DG(3)$ and $SSG(3)$ are used. This data set is selected because it has higher randomness in tourist movement compared to the Twitter data set. After that trajectories are aligned to both graph skeletons, and all the edges with usage counts of more than 5 are filtered as refined movement corridors. That is, if an edge is used by 5 or more trajectories, the edge is selected as a refined movement corridor. Both graphs resulted in the same refined movement corridor structure (Figure 4 (a)). Edges created across the river are filtered out as there cannot be any movement on those edges. Figure 4 (b) shows the execution times taken to calculate the graph structures. We can see that the time for creating $DG(d)$ is only 8% of the time taken to create $SSG(d)$.

4.3.3 Road Network Inference

Refined movement corridor extraction can be used to infer the road network of an area. Using our synthetic dataset we infer the road network of the Melbourne CBD. In order to simulate the sparseness of data points in LBSN data, we filtered out some of the points in the original synthetic dataset. The filtering process first sorts all GPS points based on the timestamp and then takes one point for every n points from the sorted set. There are 2763 locations collected for this experiment.



(a) Most popular path extracted using $DG(3)$ and $SSG(3)$ is shown in thick lines.



(b) Execution time in mili seconds to generate each graph with varying d.

Figure 6 Results of most popular path extraction.

In Figure 5 (a), we show the road networks inferred using $DG(3)$ and $SSG(3)$. The two road networks almost totally overlap with each other. It should be noted that as the filtering parameter n grows, the data set used to infer road network become more sparse, degrading the result road network. Results heavily degrade when n reaches around 120. Figure 5 (b) shows the execution times taken to calculate the graph structures. We can see that $DG(d)$ creation takes 7% of the time that is used for creating $SSG(d)$.

4.3.4 Most Popular Path Finding

By calculating edge weights to reflect the popularity of edges we can use the resulting graph to calculate the most popular paths. We used the Twitter data set to run experiments on extracting the most popular paths. This data set was used because it contains users with regular movement patterns. We executed the experiment in Melbourne city area where we found 28431 locations. Figure 6 (a) shows a most popular path found between two locations. $DG(3)$ and $SSG(3)$ produce the same path. The dashed lines indicate the shortest path between the two locations while the thick lines indicate the most popular path. When comparing this result with a map there are roads along the most popular path detected while there are building on top of the shortest path. Overall 78% of the edges selected for the most popular path were sitting on the road network of the Melbourne city. Figure 6 (b) shows the executions times taken by $DG(d)$ and $SSG(d)$ to create graphs. Due to the size of the dataset $SSG(d)$ has resulted in running times longer than one second. However, $DG(d)$ has generated the graph in 7% of the time taken by the $SSG(d)$, making it suitable for processing large data sets in real time.

5 Discussion and Future Works

From our experiments, it is clear that given a point set $DG(d)$ creation takes less time compared to $SSG(d)$ creation. Also, $DG(d)$ shows the similar effectiveness as the $SSG(d)$ in solving various queries. The low execution time of DG(d) is due to several reasons. Firstly, for any edge, $DG(d)$ creation algorithm (Algorithm 1) only processes two triangles. However, for $SSG(d)$, it can be empirically shown that per edge at least three triangles are processed. This effect should give a $2:3$ advantage to $DG(d)$ creation compared to $SSG(d)$. However, our result shows that the ratio of the execution times are $1:10$ between $DG(d)$ and $SSG(d)$. Reason for this massive difference is due to the numerical analysis method used to evaluate the d-value of an edge for $SSG(d)$. For $DG(d)$, only a simple inequality is evaluated based on Definition 3.1, per processing triangle. For d-spectrum calculation method of $SSG(d)$, the numerical analysis method runs to determine the least empty diversion neighbourhood

around an edge. In our implementation, the analysis method used by $SSG(d)$ is the Secant method , which may need to run hundreds of iterations when processing one edge. Therefore we experience this massive time difference between $DG(d)$ and $SSG(d)$. Due to this reason, it is advantageous to use $DG(d)$ in applications where very little processing time is available to generate results. This also highlights the need for looking into faster ways to locate the d-value for $SSG(d)$, as future work.

In our experiments, we have used data sets with few thousands of locations. As the dataset size increases, one may think we can apply existing data processing techniques applicable on dense GPS data. Even though the LBSN datasets are large, the locations in the datasets are spread across large areas , resulting in a low density of data points. Therefore, existing techniques for processing high-density GPS data may not be suitable for processing LBSN datasets.

The future work on $DG(d)$ can include the autonomous detection of configuration value d and the analysis of the relationship between $DG(d)$ and β-Skeletons. Determining the bounds of the spanning ratio for $DG(d)$ is another promising future direction. Investigating how $DG(d)$ can be used in more application scenarios is also an interesting research topic. Incorporating geographical feature when solving queries with $DG(d)$ is also an interesting research topic. With the introduction of $DG(d)$ as a super graph of $SSG(d)$ it is necessary to investigate how much faster $SSG(d)$ can be generated using $DG(d)$. Since $DG(d)$ definition is distance based, the concept of $DG(d)$ can be extended to higher dimensions and with different distance measurements.

6 Conclusion

We presented the Diversion Graph ($DG(d)$), a connected graph that varies depending on a single parameter d. We analysed how $DG(d)$ relates to some well-known graph structures, and we presented how $DG(d)$ can be used to improve running time of the state-of-the-art graph, the Stepping Stone Graph ($SSG(d)$). We have empirically shown that $DG(d)$ is both efficient and effective to analyse LBSN data due to its distance based local evaluation criteria.

References

1 D. V. Andrade and L. H. de Figueiredo. Good approximations for the relative neighbourhood graph. In *Proceedings of the 13th Canadian Conference on Computational Geometry, University of Waterloo, Ontario, Canada, August 13-15, 2001*, pages 25–28, 2001. URL: `http://www.cccg.ca/proceedings/2001/lhf-96805.ps.gz`.

2 P. Bose, L. Devroye, W. S. Evans, and D. G. Kirkpatrick. On the spanning ratio of gabriel graphs and beta-skeletons. *SIAM J. Discrete Math.*, 20(2):412–427, 2006. `doi:10.1137/S0895480197318088`.

3 J. Cardinal, S. Collette, and S. Langerman. Empty region graphs. *Computational geometry*, 42(3):183–195, 2009. `doi:10.1016/j.comgeo.2008.09.003`.

4 J. Chae, D. Thom, Y. Jang, S. Kim, T. Ertl, and D. S. Ebert. Public behavior response analysis in disaster events utilizing visual analytics of microblog data. *Computers & Graphics*, 38:51–60, 2014. `doi:10.1016/j.cag.2013.10.008`.

5 J. Cortés, S. Martínez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8):1289–1298, 2006. `doi:10.1109/TAC.2006.878713`.

6 M. de Berg, W. Meulemans, and B. Speckmann. Delineating imprecise regions via shortest-path graphs. In I. F. Cruz, D. Agrawal, C. S. Jensen, E. Ofek, and E. Tanin, editors, *19th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems,*

ACM-GIS 2011, November 1-4, 2011, Chicago, IL, USA, Proceedings, pages 271–280. ACM, 2011. doi:10.1145/2093973.2094010.

7 K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Biology*, 18(3):259–278, 1969.

8 S. Kannangara, E. Tanin, A. Harwood, and S. Karunasekera. Stepping stone graph for public movement analysis. In F. Banaei-Kashani, E. G. Hoel, R. H. Güting, R. Tamassia, and L. Xiong, editors, *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2018, Seattle, WA, USA, November 06-09, 2018*, pages 149–158. ACM, 2018. doi:10.1145/3274895.3274913.

9 S. Kannangara, E. Tanin, A. Harwood, and S. Karunasekera. Stepping stone graph: A graph for finding movement corridors using sparse trajectories. *ACM Trans. Spatial Algorithms and Systems*, 5(4):23:1–23:24, 2019. doi:10.1145/3324883.

10 K. H. Lim, S. Jayasekara, S. Karunasekera, A. Harwood, L. Falzon, J. Dunn, and G. Burgess. RAPID: real-time analytics platform for interactive data mining. In U. Brefeld, E. Curry, E. Daly, B. MacNamee, A. Marascu, F. Pinelli, M. Berlingerio, and N. Hurley, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018, Dublin, Ireland, September 10-14, 2018, Proceedings, Part III*, volume 11053 of *Lecture Notes in Computer Science*, pages 649–653. Springer, 2018. doi:10.1007/978-3-030-10997-4_44.

11 A. M. MacEachren, A. R. Jaiswal, A. C. Robinson, S. Pezanowski, A. Savelyev, P. Mitra, X. Zhang, and J. I. Blanford. Senseplace2: Geotwitter analytics support for situational awareness. In *2011 IEEE Conference on Visual Analytics Science and Technology, VAST 2011, Providence, Rhode Island, USA, October 23-28, 2011*, pages 181–190. IEEE Computer Society, 2011. doi:10.1109/VAST.2011.6102456.

12 D. W. Matula and R. R. Sokal. Properties of gabriel graphs relevant to geographic variation research and the clustering of points in the plane. *Geographical analysis*, 12(3):205–222, 1980.

13 K. Ramamohanarao, H.Xie, L. Kulik, S. Karunasekera, E. Tanin, R. Zhang, and E. B. Khunayn. Smarts: Scalable microscopic adaptive road traffic simulator. *ACM TIST*, 8(2):26:1–26:22, 2017. doi:10.1145/2898363.

14 D. Sinclair. S-hull: a fast radial sweep-hull routine for delaunay triangulation. *CoRR*, abs/1604.01428, 2016. arXiv:1604.01428.

15 B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li. YFCC100M: the new data in multimedia research. *Commun. ACM*, 59(2):64–73, 2016. doi:10.1145/2812802.

16 G. T. Toussaint. Comment: Algorithms for computing relative neighbourhood graph. *Electronics Letters*, 16(22):860–860, October 1980.

17 G. T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern recognition*, 12(4):261–268, 1980.

18 R. B. Urquhart. Algorithms for computation of relative neighbourhood graph. *Electronics Letters*, 16(14):556–557, July 1980.

19 X. Wei and X. Angela Yao. A conceptual framework for representation of location-based social media activities (short paper). In S. Winter, A. Griffin, and M. Sester, editors, *10th International Conference on Geographic Information Science, GIScience 2018, August 28-31, 2018, Melbourne, Australia*, volume 114 of *LIPIcs*, pages 62:1–62:7. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.GISCIENCE.2018.62.

20 J. Xie, T. Yang, and G. Li. Extracting geospatial information from social media data for hazard mitigation, typhoon hato as case study (short paper). In S. Winter, A. Griffin, and M. Sester, editors, *10th International Conference on Geographic Information Science, GIScience 2018, August 28-31, 2018, Melbourne, Australia*, volume 114 of *LIPIcs*, pages 65:1–65:6. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.GISCIENCE.2018.65.

Not Arbitrary, Systematic! Average-Based Route Selection for Navigation Experiments

Bartosz Mazurkiewicz
TU Wien, Austria
bartosz.mazurkiewicz@geo.tuwien.ac.at

Markus Kattenbeck
TU Wien, Austria
markus.kattenbeck@geo.tuwien.ac.at

Peter Kiefer
ETH Zurich, Switzerland
pekiefer@ethz.ch

Ioannis Giannopoulos
TU Wien, Austria
igiannopoulos@geo.tuwien.ac.at

Abstract

While studies on human wayfinding have seen increasing interest, the criteria for the choice of the routes used in these studies have usually not received particular attention. This paper presents a methodological framework which aims at filling this gap. Based on a thorough literature review on route choice criteria, we present an approach that supports wayfinding researchers in finding a route whose characteristics are as similar as possible to the population of all considered routes with a predefined length in a particular area. We provide evidence for the viability of our approach by means of both, synthetic and real-world data. The proposed method allows wayfinding researchers to justify their route choice decisions, and it enhances replicability of studies on human wayfinding. Furthermore, it allows to find similar routes in different geographical areas.

2012 ACM Subject Classification Information systems → Geographic information systems; Information systems → Location based services; Information systems → Decision support systems; General and reference → Empirical studies

Keywords and phrases Route Selection, Route Features, Human Wayfinding, Navigation, Experiments, Replicability

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.8

1 Introduction

Selecting a route for a human wayfinding study in a systematic manner is a non-trivial task. Despite its potential impact on the results, reasonable justifications for routes based on their features are often neglected. In this paper, we propose, implement and evaluate a methodological framework which enables researchers to choose a route for human wayfinding experiments in a given area according to predefined, weighted criteria. The determined routes are – with respect to these criteria – representative for a (weighted) average route for the chosen area. Using this framework will, therefore, lead, among others, to an increased comparability and replicability of in-situ wayfinding studies.

Starting with the replication crisis in psychology [34], reproducibility and replicability have both seen increased interest in all subfields of geographical sciences in recent years (see e.g., [32, 35, 20, 24]). At the same time, studies which aim to understand human wayfinding and/or how interactive assistance can be provided to wayfinders have gained momentum [21, 12]. These research efforts will likely be continued in the future, as there is neither a

general agreement on algorithms nor route descriptions or an anywhere close to definite understanding of the interplay between spatial cognition and the assistance provided by mobile navigation companions. While there has been some recent progress in terms of reproducibility (i.e., software and data are made available to the scientific community and rerunning the analysis using the software and data yields the published results, see [6]), e.g., through initiatives like the AGILE initiative on reproducible publications (see also [32]), increasing the level of replicability may be much harder to achieve. In particular, up until now, the replicability of wayfinding studies often suffers from the possibility to choose a route in a systematic manner: the decisions which led to the choice of a particular route are often not made explicit, leading to the impression that routes are often chosen in an ad-hoc manner (see Section 2). As a result, oftentimes information other than length, number of decision points, and a rough classification of the urban environment (e.g., European) is not given. As a consequence, the impact of differences in route properties cannot be assessed in an appropriate manner if researchers fail to replicate the results.

2 Related Work

This section provides a thorough overview of route features in studies involving wayfinding tasks. It provides the basis for the set of features, we use in our methodological framework. (see Section 3). In order to gain an insight into common practice among researchers to justify their route choices and the route characteristics they pay attention to, we have systematically screened six major venues (conferences and journals) in the broader area of geographic information science and related fields since 2010.

While our search is not exhaustive by any means, the number of papers screened is still suitable to provide a reasonably grounded insight into the state-of-the-art. In identifying relevant papers, we focused exclusively on studies involving either wayfinding tasks by participants or studies, in which routes were presented to users, e.g., on maps. This implies, that we deliberately excluded all studies involving route retrieval from memory without performing an actual wayfinding task or which involved human wayfinding without predefined routes.

Overall, 32 papers were found which present studies on wayfinding/navigation in both, virtual and real-world environments. Each of the relevant articles/papers found was checked for the rationale researchers have given for the chosen route and which route characteristics they have mentioned explicitly.

Table 1 reveals several important insights regarding common practices among researchers: The three most often named aspects are: the length of a route (mentioned by 16 publications), the type (e.g., a residential area) of environment a study was conducted in (15), and the name of the city/town of a study (11). While these criteria are the most frequent ones, it is important to note that only half of the papers mention route length and type of environment whereas the name of the city/town is stated only by one third of the papers explicitly. In addition to basic route data and information about the local environment of different granularity, a variety of features mentioned by researchers deal with decision points (DPs). We consider each intersection on a route as a decision point, which is neither the start nor the end point of the route. While authors describe at least the overall number of DPs and the proportion of those DPs which require a turn, the layout of the DPs is given rather rarely. Several other aspects related to route instructions, visibility of environmental cues and – in case two or more routes are compared – how routes relate to one another are mentioned occasionally.

Table 1 Overview of route features named (multiple features per paper possible) in human wayfinding studies in major research outlets since 2010. Relevant papers for the AGILE conference: [1, 14, 23]; for the GIScience conference: [38, 29, 19]; for the COSIT conference: [40, 46, 47, 18, 22, 11, 3, 2]; for the IJGIS: [26]; for the LBS Journal: [13, 37, 39]) and for the SCC Journal: [33, 45, 49, 27, 36, 17, 43, 25, 48, 42, 16, 7, 31, 44].

	Feature	AGILE	COSIT	GIScience	IJGIS	LBS	SCC	Freq (N=32)
Basic Route Data	length	3	4	1	1	2	5	16
	walking duration	0	2	2	0	0	0	4
	name of city	1	4	0	0	2	4	11
	size of area	0	0	0	0	0	1	1
Local Environment	uniformity of env.	0	1	0	0	0	0	1
	type of env. (e.g., residential)	1	2	1	0	2	9	15
	terrain (e.g., flat)	0	1	0	0	0	0	1
	complexity of env. (e.g., narrow streets)	0	4	0	0	1	3	8
	type of walkways (e.g., sidewalk)	0	0	1	0	1	0	2
Decision Point / Intersection	#DP	1	1	1	0	0	3	6
	#DP with turn	2	1	0	0	2	3	8
	#type of turn (l,r, non-turn)	1	0	0	1	0	0	2
	Inclusion of diff. actions at DP	1	0	1	0	0	0	2
	DP layout (e.g., 3-way, 4-way) described	1	1	0	0	0	1	3
	variety of DP layouts mentioned	0	0	2	0	2	0	4
	DP density	0	0	0	0	1	0	1
	Distance between DP	0	1	0	0	0	0	1
Route Instruction Features	inclusion of landmarks	0	0	1	0	1	4	6
	inclusion of street names	0	0	1	0	0	0	1
	Destination (landmark)	1	0	0	0	0	0	1
View / Visibility related	views offered (e.g., open vista)	0	0	1	0	0	0	1
	visibility of dest. from start (or vice versa)	0	1	0	0	0	1	2
	long-distance vistas	0	1	0	0	0	1	2
	visibility of street names	0	0	0	0	0	1	1
Relation to other Routes	equal length	0	0	0	0	0	1	1
	equal starting and end points	0	0	0	0	0	1	1
Number of distinct criteria		9	13	10	2	9	14	26

Taken together, this overview of common practices provides evidence for a lack of proper justification of route choices and only very basic features of routes being made explicit. In particular, half of the publications do not even mention basic properties, such as route length, and even environmental and decision point-related aspects are insufficiently described. This is, from our perspective, a clear barrier to any attempts to the replicability of these research results.

3 Route Selection Criteria

It is obvious that route selection is deeply intertwined with a study's research question. The literature review above has revealed, however, that this selection is often insufficiently justified. Moreover, even basic route properties are often not made explicit. This may be a hint to the practice to use ad-hoc choices for routes, a decision which may result in a considerable bias stemming from route choice. Even for those studies, which want to assess the impact of a given route, it would be desirable to be able to quantify the degree as to which a chosen route represents a special case given a set of criteria researchers want to take into account. The possibility to select routes for human wayfinding studies in a systematic and reproducible manner is, therefore, highly desirable. In order to achieve scientifically valid results, researchers interested in conducting (not only replicating) human wayfinding studies must base their research on a route, which is selected in a systematic and reproducible manner. For many of these studies, it is desirable not to use a route which would represent a special case given the researcher's requirements about routes. In human wayfinding studies in real-world, the population of routes to select from encompasses millions of possible routes

of a given number of decision points for any area of non-trivial size. Given these figures, selecting a route based on the average of all routes fulfilling the researchers' requirements seems reasonable for those studies which do not use a route as an independent variable. Outliers are expected to have only a small effect as the population is vast, and the number of criteria to be taken into account is large. Therefore, the best possible route to be chosen would be a route, which meets the average for all criteria a researcher wants to take into account as close as possible. We refer to such a route using the expression *average route* because it is average-based. As mentioned above, even those studies in which route is an independent variable, knowing the deviation from the average route in an area may provide researchers with valuable information to interpret their results.

In order to make research more comparable and to provide other stakeholders (scientists, urban planners, politicians etc.) with assistance to choose one route for their needs, we present an approach which finds a route which is as close as possible to a theoretically existing average route in a given area. The idea is that a route selected in such a systematic manner should provide more transferable results as it reflects the characteristics of the specified area.

Based on the set of criteria currently used by researchers (see Section 2) our framework takes the following criteria into account. We base the decision made for in-/exclusion on both, prior research practice and the widespread availability of data:

Pre-emptive criteria

Researchers must select, first and foremost, an area in which they want to conduct their study in. In accordance with the widespread report of this criterion, we use the number of decision points (DPs) as a criterion researchers must specify. If researchers wish to do so, they can additionally provide a minimum and maximum route length.

Used criteria

According to the literature reviewed, researchers consider criteria related to DPs as important. Therefore, our framework considers the *average number of options a DP* offers and the *number of n-way intersections* on a route – both of which are derived from the intersection framework [10]. The same framework [10] provides information about the *regularity of a DP* (the sum of angles branches need to be rotated in order to create a regular intersection, see [10, p. 3:4] for further details). As a fourth DP-related aspect, we consider the *number of right, left and non-turns* at DPs on a route. We calculate these properties according to the point orientation algorithm [4]. In order to count non-turns and avoid false negatives we use a 10 degree threshold, i.e., a 20 degree cone, to identify continuations. Undoubtedly, landmarks play an important role in human navigation. However, we lack sources of salience values for arbitrary regions. Consequently, we use points of interest (POI) as a proxy (see e.g., [9] or [41] for publications with a similar approach). As there is no commonly agreed definition of POI available, we extract POIs from OpenStreetMap data based on tag `amenity=*`. Our methodological framework, however, is open to other definitions researchers may want to employ. We take two POI-related criteria into account: *the average number of POIs at a DP* and the *uniqueness of a POI category at a DP*. The average number of POIs on a route is given by the amount of POIs within a given radius from any DP divided by the number of DPs. The uniqueness of a POI, according to Rousell and Zipf [41], is defined as $\frac{1}{j}$ where j is the number of POIs of the same type (e.g. restaurant) in the considered set. Finally, two environmental features are considered: *Slope* (shares of route with negative, positive and zero slope sourced from a digital elevation model[1]) is taken into account as a proxy for criterion terrain, whereas land cover data (Urban Atlas[2]) reflect the *type of environment*.

[1] `https://www.wien.gv.at/ma41datenviewer/public/`, last access June 5th, 2020
[2] `https://land.copernicus.eu/local/urban-atlas/urban-atlas-2012`, last access March 20th, 2020

This list can be extended if more data is available or of particular interest for a navigation study to be conducted. In short, we are aiming to get as close as possible to the average route based on user-defined weights for route features in a given area.

4 Methodology

Given a certain area in a built environment, we aim at ranking all possible routes with a given number of DPs. This ranking is based on the average of all routes in this area according to a set of given criteria (see Section 3). The closer a route is to the average values, the higher this route will be ranked. In the following we provide a step-by-step description of the required computation steps. At the end of this section information about software and hardware used is provided. Our street network data are based on OpenStreetMap. The computations are based on a graph created out of nodes representing intersections and edges representing the street segments. For a detailed description of all data sources see Section 3.

Step 1: Extracting all potential routes. We represent *all* potential routes[3] in the given area with their criteria as a decision matrix $\boldsymbol{X'}$. As these criteria are measured on different scales, a z-score standardization is applied in order to normalize the values, i.e., a z-score of $z = 0$ represents the average. Since we are interested only in the deviation from the average, $\boldsymbol{X'}$ contains only absolute values of z-scores.

$$X' = \begin{bmatrix} x'_{11} & x'_{12} & \dots & x'_{1m} \\ x'_{21} & x'_{22} & \dots & x'_{2m} \\ \dots & \dots & \dots & \dots \\ x'_{n1} & x'_{n2} & \dots & x'_{nm} \end{bmatrix} \tag{1}$$

where n denotes the number of routes and m the number of criteria. In order to retrieve all possible routes of a certain number of DPs without loops, a street network graph was utilized. This can be approached as a subgraph isomorphism problem, which is NP-Complete. Although street networks can be modeled as planar graphs (for simplification reasons) in reality they are not [5]. Thus, the subgraph isomorphism problem on non planar graphs grows in general, exponentially. However, there are algorithms with acceptable practical execution time[8].

Step 2: Best possible solution. Based on the z-scores for all criteria, we retrieve the *best possible solution* A^+ (see Eq. 2): This is an artificial (and unlikely to exist) route which comprises the minima of all z-scores, i.e., it is as close to the average of all criteria one can get.

$$A^+ = (y_1^+, y_2^+, \dots, y_m^+) \qquad where \quad y_j^+ = \min_{i=1,2,\dots,n} x'_{ij} \tag{2}$$

The best possible solution contains the minimum for each criterion. A value of 0 means that this value reflects the global mean perfectly. Negative values are not possible due to the performed standardization step.

[3] It is important to note that users of the proposed method are free to take any type of routes into account,i.e. routes w/o loops, shortest path between two distinct points, round tours etc.

Step 3: Weighted similarity. There are several spatial as well as spatio-temporal similarity measures available for a variety of problems [15, 30]. We identified the cosine similarity and the weighted euclidean distance as the most promising ones for our approach. The cosine similarity measure, which is widely used for multidimensional data, had to be discarded after encountering counter-intuitive results during testing. The explanation for this discrepancy between intuition and hard numbers is that cosine similarity measures only the angle between two normalized vectors, and therefore ignores the magnitude of difference between them.

As described earlier (see Section 3), researchers can specify weights for each criterion according to their research interest (i.e., the higher a weight, the more important an average value of a characteristic is to a researcher). These weights are used during the distance calculation between a route and the best possible solution. Each route is compared to the best possible solution (equation 2) by means of the n-dimensional weighted euclidean distance: In Equation 3, x'_j represents the j-th criterion of a route and w_j is the weight for this high-level criterion.

$$dist = \sqrt{\sum_{j=1}^{m} w_j (x'_j - y_j^+)^2} \tag{3}$$

A high-level criterion is, for example, the regularity of a decision point which can be represented by the sum of angles needed to obtain a regular intersection [10]. It is, however, not reasonable to build averages across different n-way intersections. Therefore, the sum of angles is computed for each n-way intersection (called subdimension) separately. For example: If seven is the largest number of branches for all intersections in the area-of-interest, the sum of angles is calculated for 3- to 7-way intersections separately. In this particular example each subdimension would have a weight of $w_j/5$, where w_j is the weight assigned to criterion *decision point regularity*. The sum of the weight vector is 1.

Step 4: Ranking of results. Finally, all routes are ranked according to their distance (equation 3) to the best possible solution (equation 2). The smaller the distance, the closer a route is to the average in the area of interest, given the user defined weights for the applied criteria.

Implementation. This paragraph specifies the software and hardware used to implement our approach. In order to find all possible routes without loops (step 1) SageMath 9.0 with its SubgraphSearch function[4] was used, whereas steps 2-4 were implemented in Python 3.6. Two features from the real world example (see section 5.2), namely, the *average number of POIs per DP* and *type of environment* were calculated in a PostGIS (v 2.4) database. All analyses run on an AMD Ryzen Threadripper 1950X 16-Core Processor, 3400 Mhz, with 64 GB RAM.

5 Evaluation

As a proof of concept, we first evaluate our approach on synthetic data (subsection 5.1). Using synthetic data enables us to use predefined values for all criteria and, thereby, formulate the expected results. We then continue with a real-world example in Vienna, Austria (see subsection 5.2).

[4] `http://sage-doc.sis.uta.fi/reference/graphs/sage/graphs/generic_graph_pyx.html#sage.graphs.generic_graph_pyx.SubgraphSearch`, last access June 5th, 2020

5.1 Synthetic data

We use 100 x 100 regular grid graphs as synthetic data. The graph used has 10 000 nodes and 19 800 edges. All edges have the same length and characteristics (which is a difference to the real-world data, see Section 5.2). We distinguish between type I and type II nodes. While type I nodes have 3 POIs all of which have unique categories, type II nodes have 6 POIs which show an average uniqueness of their categories of 1/3. Therefore, routes will have different average POI numbers due to different proportions of type I and II nodes in a route. They have different characteristics regarding POIs in order to be able to observe changes in results. It is important to note that the order of magnitude of these differences does not matter as long as it is unequal to 0. The 4 corners of the grid have only 2 edges and are considered as "2-way intersections": Taking them into account is reasonable to show that our approach takes the global distribution (frequency) of n-way intersections into account. All nodes along the border of the graph, with exception of the 4 corners points just mentioned, have 3 edges. All other nodes have 4 edges, i.e., they are regular 4-way intersections.

For all evaluations on synthetic data we set the number of decision points to $k = 7$. This number was chosen due to computation time limitations, which is reasonable based on the fact that the route recommendation algorithm is NP-Complete (due to the subgraph search problem). Based on all these routes the best possible route was calculated (see equation 2) as target route. In total, 55 396 400 possible routes without loops (represented as subgraphs) having 7 DPs plus 1 starting and 1 end point were found in this synthetic graph. These routes do not have to be a shortest path between two points. Routes have, in general, the same characteristics (e.g., slope) but they vary considering with respect to the type of actions taken at decision points (i.e., turning right or left and continuations).

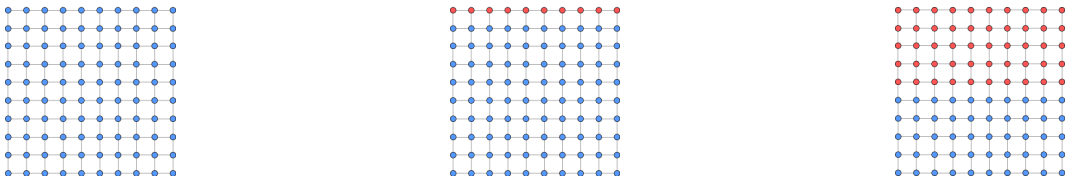


Figure 1 Schematic Representation of Synthetic Data (data used were 10 times bigger, but with the same ratios of type I (blue) and type II (red) nodes): (a) Scenario 1: Regular grid network with only nodes of type I; (b) Scenario 2: Regular grid network with ratio 9:1 of type I to type II nodes; (c) Scenario 3: Regular grid network with equal shares of type I to type II nodes.

We evaluate our approach with respect to synthetic data based on three scenarios, which differ in the proportion of type I and II nodes (see Figure 1). Each of the scenarios share three high-level criteria, namely the number of 2-, 3-, 4-way intersections, the sum of angles needed to obtain regular 3- and 4-way intersections [10] and the frequency of right and left turns and non-turns at decision points.

Scenario 1

In this scenario the whole 100 x 100 regular grid network consists of type I nodes, only (see Figure 1a). 97% of all possible routes contain 4-way intersections only and all of these are regular 4-way intersections. The average number of right-, left- and non-turns is 2.18, 2.18 and 2.64, respectively. Based on these figures we expect the route with the least distance to

the average route to have 4-way intersections only, 2 right, 2 left and 3 non-turns at DPs. As all intersections are regular, the sum of angles equals zero and, therefore, is omitted in the results for synthetic data.

Table 2 presents the results. Due to the synthetic dataset, we observe many routes having equal scores. Therefore, the table reports the first 10 groups of routes, where each group represents a unique combination of the two high-level criteria (number of n-way intersections and frequency of right and left turns and non-turns). The results meet our expectations, as rank 1 group contains routes which comprise 4-way intersections only and show 2 right and left turns and 3 non-turns at DPs. While lower ranks in the table show the same distribution of intersections, rank 1 routes are the ones with the least euclidean distance to the best possible route. It is important to note that the euclidean distance reflects different degrees of deviations from the best possible route: The worst group, which is not shown in Table 2 consists of routes which have one 2-way and six 3-way intersections, 1 left or right and 6 non-turns. Similarly (also not presented in Table 2 for space reasons), routes with the same distribution (0,0,7) but with no left/right turns and 7 non-turns got a lower rank than routes with n-way distribution 0, 1, 6 and a more balanced distribution of actions at decision points.

Table 2 Results for scenario 1 where all nodes are of type I. Only the first 10 highest ranked groups of routes are shown in the table, some of which share a rank.

Rank	# Routes	# Intersections			# Turns		
		2-way	3-way	4-way	left	straight	right
1	7 635 056	0	0	7	2	3	2
2	6 341 188	0	0	7	2	2	3
	6 341 188				3	2	2
3	3 931 208	0	0	7	1	3	3
	3 931 208				3	3	1
4	3 808 196	0	0	7	1	4	2
	3 808 196				2	4	1
5	3 869 072	0	0	7	3	1	3
6	1 458 408	0	0	7	1	2	4
	1 458 408				4	2	1

Scenario 2

In scenario 2 the grid network now contains type I and type II nodes at a ratio of 9:1 (see Figure 1b). This induces variance in the data by including points-of-interest (POIs) as an additional high-level criterion, which comprises the number of POIs and the average uniqueness of a POI at a DP. Again, all high-level criteria are equally weighted. As no changes to the layout of the graph were applied, we expect routes with exclusively 4-way intersections to be higher ranked than those including also other types of intersections. In contrast to scenario 1, however, routes can now have a different number of type I and II nodes: As the average number of POIs per DP in all routes is 3.26 and the average uniqueness of POIs per DP equals 0.94, we expect routes with six type I nodes and one type II node to be higher ranked than other combinations of those types[5].

[5] This assumption is also backed up by the average number of type II nodes in a route which equals 0.61.

Table 3 Results for scenario 2. Only the first 11 highest ranked groups of routes are shown in the table, some of which share a rank. POI subdimensions are rounded to 2 decimals.

Rank	# Routes	# Intersections			# Turns			# Type II Nodes	Avg. # of POIs	Avg. Uniq. of POIs
		2-way	3-way	4-way	left	straight	right			
1	31 024	0	0	7	2	3	2	1	3.43	0.90
2	6 914 264	0	0	7	2	3	2	0	3	1
3	23 934	0	0	7	2	2	3	1	3.43	0.90
	23 934				3	2	2	1	3.43	0.90
4	5 743 264	0	0	7	2	2	3	0	3	1
	5 743 264				3	2	2	0	3	1
5	38 668	0	0	7	2	3	2	2	3.86	0.81
6	32 526	0	0	7	2	2	3	2	3.86	0.81
	32 526	0	0	7	3	2	2	2	3.86	0.81
7	14 160	0	0	7	3	3	1	1	3.43	0.90
	14 160				1	3	1	1	3.43	0.90

The results presented in Table 3 meet our assumptions. The highest ranked group represents routes which have only 4-way intersections, a balanced (close to global average) frequency going right, left or straight ahead at a decision point throughout the route, one type II node and the closest possible values to the global average regarding POI subdimensions.

Scenario 3

In scenario 3 we increase the variance in the data by changing the proportion of type I to type II nodes to 1:1, while keeping the graph layout unchanged (see Figure 1c). This means, scenario 3 simulates an area in which two 2 subareas are clearly different but have an equal share. The same high-level criteria as in scenario 2 are applied. As the frequency of n-way intersections and direction changes remain unchanged, we still expect routes with 4-way intersections only and a balanced frequency of right-, left- and non-turns at a decision point to be higher ranked. Given the 1:1 ratio of node types and the odd number of decision points (7), we expect routes with either three type I and four type II or four type I and three type II nodes to be ranked highest. These two combinations of type I and type II nodes are equally close to the global average for both POI subdimensions (avg. number POI: 4.5, avg. uniqueness POI: 0.66). The results for the third scenario are presented in table 4. In-line with our expectations, the highest ranked group has only 4-way intersections, a balanced (close to global average) frequency of (non-)turns and a balanced ratio between type I and type II nodes and, therefore, close to global average values for both POI subdimensions.

Taken together, the results of these three scenarios provide evidence that our approach yields reasonable results based on the controlled conditions of synthetic data. We will now continue with real-world data and the full set of criteria mentioned before (see Section 3).

5.2 Real World Example

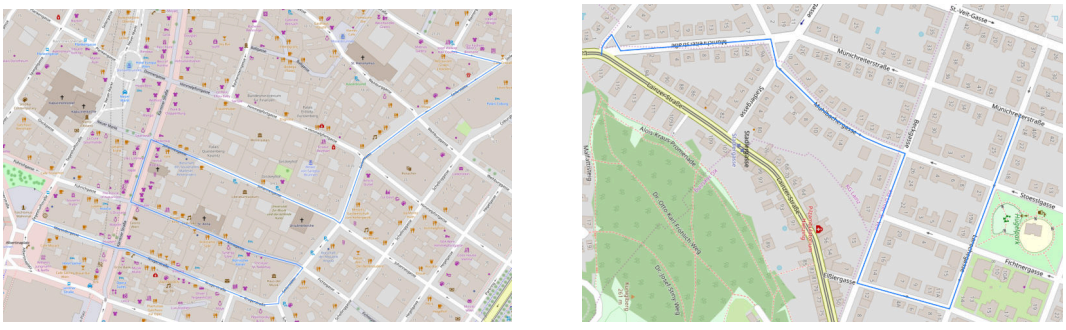
We have chosen two different areas in Vienna, Austria. Both regions significantly differ with respect to their degree of sealed soil, where Region 1 (located in the city center) shows a high degree and Region 2 (residential area) a low-medium degree of soil sealing (according to Urban Atlas 2012). We specified both pre-emptive criteria (see Section 3) and set the number of DPs to $k = 10$, and route length to a range between 1 000 m and 1 500 m. The length of possible routes in terms of both, the number of DPs and the distance, was chosen

Table 4 Results for scenario 3. Only the first 8 highest ranked groups of routes are shown in the table, some of which share a rank. POI subdimensions are rounded to 2 decimals.

Rank	# Routes	# Intersections			# Turns			# Type II Nodes	Avg. # of POIs	Avg. Uniq. of POIs
		2-way	3-way	4-way	left	straight	right			
1	37 092	0	0	7	2	3	2	3	4.29	0.71
	37 092				2	3	2	4	4.71	0.62
2	38 668	0	0	7	2	3	2	2	3.86	0.81
	38 668				2	3	2	5	5.14	0.52
3	28 310	0	0	7	2	2	3	3	4.29	0.71
	28 310				3	2	2	3	4.29	0.71
	28 310				2	2	3	4	4.71	0.62
	28 310				3	2	2	4	4.71	0.62

based on computation time (see Section 5.1). The underlying graph for Region 1 has 1 196 nodes, 1 740 edges and 4 290 636 possible routes of 12 points length (10 decision points plus start and end point which are not considered to be DPs). Of these routes, 62 294 have a length between 1 000 m and 1 500 m. The underlying graph for Region 2 has 498 nodes, 744 edges and 2 276 070 possible routes of 12 points length and 834 114 of these have a length between 1 000 m and 1 500 m. The observed difference in the number of considered routes is likely a result of the fact that the average segment length between two subsequent DPs in the city center area (Region 1, 2.25 km^2 area) is less than in case of the residential area (Region 2, 2.84 km^2 area).

For each region the closest to average route was calculated regarding the following 6 high-level and equally weighted criteria: *cardinality of decision points* (the number of n-way intersections on a route and the derived average options per DP), *frequency of right/left and non-turns*, *terrain* (proportion of negative, positive and zero slope), *POIs* (average number within a 10 meter radius and average uniqueness of category per DP), *regularity of DPs* and *type of environment* (land cover data). Figure 2 shows the routes for both regions which are closest to the best possible solution. Considering the above-mentioned criteria, routes from A to B achieve the same score as those from B to A. They only differ symmetrically in *slope* and *frequency of right/left and non-turns*. This symmetry causes an equal distance to the best possible route. Non symmetrical attributes like directed viewsheds would lead to a difference in score between route A to B and route B to A.



(a) **(b)**

Figure 2 The closest to average routes for Region 1 (a) and Region 2 (b) considering all criteria mentioned above (see Sec 3), which were equally weighted.

Table 5 Comparison between highest ranked routes and the best possible solution. Land cover classes are Urban Atlas classes: A (11100), B (11210), C (11220), D (11230), E (12100), F (12220), G (12230), H (14100) and I (14200). Land cover values do not sum up to 1 due to rounding. If there are two numbers for a feature this is due to having 2 winners for a region. Why the number of turns of best possible routes do not sum up to 10 is explained in the discussion.

Name	Avg. Options	# Intersec.				# Turns			Slope			Avg. # of POIs	Avg. Uniq. of POIs	Regularity				Land Cover %								
		3	4	5	6	l	s	r	neg	none	pos			3	4	5	6	A	B	C	D	E	F	G	H	I
Win. Reg 1	3.7	3	7	0	0	4/4	2	4/4	.05/0	.95	.05/0	.2	.2	55.68	16.31	NaN	NaN	.57	0	0	0	.09	.34	0	0	0
Best Reg 1	3.7	3	7	0	0	4	3	4	.03	.94	.03	.3	.17	51.97	17.24	83.2	69.34	.54	0	0	0	.09	.36	0	.01	0
Win. Reg 2	3.9	3	5	2	0	3/4	3	4/3	.03/.06	.91	.06/.03	0	0	57.75	19.86	72.80	NaN	0	.09	.48	.07	0	.36	0	0	0
Best Reg 2	3.9	3	5	2	0	3	3	3	.08	.84	.08	0	0	58.98	18.30	72.59	148.46	0	.06	.44	.07	0	.36	.01	.03	.01

Table 5 presents numerical results by providing figures for both, the highest ranked routes (will be referred to as *winners*) and the best possible solution, i.e., a hypothetical route which shows closest to average values for all criteria (will be referred to as *best*). Two aspects are important to be kept in mind: 1) The best possible solution does not need to be an actually existing route (see Sec 6); 2) there are two winners per region as each route can be traversed in both directions.

For both regions, the distribution of scores (i.e., the euclidean distance to the best possible solution) is similar (see discussion for an explanation of the maxima). The quantiles for the score in Region 1 are 0%: 0.2250, 25%: 0.5894, 50%: 0.7290, 75%: 0.8569 and 100%: 32.3001. The score quantiles in Region 2 are 0%: 0.1738, 25%: 0.5198, 50%: 0.6468, 75%: 0.8875, 100%: 5.2246. Regarding the *cardinality of DPs*, both winners in each region show a perfect match with best, respectively. With respect to *slope*, winners 1 are closer to best 1 than winners 2 are to best 2. It is vice versa regarding *POIs*, in which case winners 2 match best 2 perfectly (generally speaking, Region 2 is an area which is poor in POIs), whereas winners 1 have, on average, slightly less POIs at a DP than the best possible solution, but their uniqueness is higher. Looking at the *regularity of DPs* both routes reflect global averages very well if and only if they have this kind of n-way intersection[6]. Regarding *land cover* the differences between winners and best in both regions are minimal[7]. Regarding *frequency of right/left and non-turns* winners in Region 1 show one continuation less than the winner, whereas winners in Region 2 show either one left or right more than the best possible route. In both cases, the frequency of the best possible route is impossible to achieve (see Sec 6 below). Taken together, both winners in each region come close to the best possible route – which is hypothetical in this case and very unlikely to exist in general but reflects global averages as good as possible.

6 Discussion

In this work we propose and evaluate a systematic approach for the selection of pedestrian routes in a street network, with a focus on wayfinding experiments. As described in the related work section, a proper selection of street routes is crucial for several types of empirical studies. Such a systematic approach can help select a route based on a multitude of criteria and, furthermore, reduce the time necessary for manual selection. Moreover, the proposed approach can be seen as a step towards replicability of research, allowing to select a similar route at a completely different geographic location by exchanging the best possible solution

[6] NaN in a route are not contributing to the euclidean distance.
[7] If land cover does not sum up to 1 this is due to rounding.

with the target route of another location. The proposed approach was evaluated utilizing synthetic data serving as a ground truth. The results of this evaluation confirmed the validity and applicability of our approach. We performed a proof of concept evaluation using real data, once taken from the city center and once from a residential area in Vienna, Austria.

Two aspects of the results achieved for the real-world data need to be discussed in more detail: Firstly, the difference in distance between the the upper quartile and the maximum is very large for Region 1. However, the two routes (out of 62 294) having scores above 32 both have a 6-way intersection – a feature which is very uncommon for Region 1. Obviously, Region 2 has no large outliers as the maximum euclidean distance is far less than for Region 1. For both regions, however, the distances up to the upper quartile are numerically small; it is, therefore, a matter of future research whether these differences are meaningful for wayfinding research and with respect to which criteria this might be the case (see Section 7). Secondly, the fact that the best possible solutions do not match the predefined number of DPs by one needs in-depth discussion. All best possible solutions are calculated based on a z-score, which depends on the population mean and standard deviation. Due to the size of the population of possible routes, it is very unlikely that mean and standard deviations both are integers. The number of right, left and non-turns on an actual route (which is the third factor needed to calculate a z-score), however, must be integers. The figures need to be rounded (i.e., either floored or ceiled depending on the decimal digits), accordingly. In addition to that, the means of right and left turns must be symmetric. Hence, the best possible solution as a hypothetical route can show this anomaly of more/less (± 1) DPs than actually requested, whereas all actual routes in the population always have the predefined number of decision points (and turns). It is important to note that, although slope is a symmetric feature as well, its value can be decimal. Moreover, all other criteria are invariant to the direction of travel on a route. To conclude, our framework supports systematic and deterministic route selection for experiments considering weighted features provided by the researcher. Furthermore, exchanging the best possible solution with another target route (using this route as the average one) allows to find a similar route in a different place of the world.

The criteria utilized in this work served as an example and can be easily extended or even replaced by others. Of course, the more criteria used, the longer the route in terms of DPs, or the larger the search area, the more computation time will be required. In most cases, however, finding a reasonable route at the city level should be sufficient and this should be possible in less than one day of computing time as our results were. Our methodological framework allows to extend the list of criteria taken into account. Several aspects come to mind: the segment length and orientation might be worthwhile to be taken into account; if doing so, the number of POIs per segment of a given length may be worthwhile to take into consideration in order to study on-route landmarks (see [28]). Traffic data, flow of humans in an area and noise (e.g., stemming from factories) may have an impact on in-situ studies and might be considered, although it might be very difficult to obtain this type of data on a large-scale basis. While DPs per se have been extensively considered already, the order of turns (e.g., llrrslr) and the sequence of intersection types might be included (see e.g., [12]). One particularly important environmental feature, which is also missing due to unavailability of large-scale data, is the architectural style/diversity of buildings in a given area.

Computation time and difficulty of validating the results obtained from real data are the main limitations of this work. Concerning computation time, although this approach cannot be utilized for real-time purposes, most of the relevant cases for wayfinding will not be affected by that. Nevertheless, reducing computation time based on existing sub-graph

search algorithms is already feasible (see Section 4), although this is out of the scope of our work. Results for real data are difficult, if not impossible, to validate. Synthetic data approaches for validation like the one presented above, however, ensure the validity of the results at least for the cases covered.

7 Conclusion and Outlook

The proposed approach can be considered as a valuable methodological framework, which can help to make informed decisions concerning route selections. As a consequence, this framework can partially support the design of experiments and enhance replicability.

The results of the presented approach strongly rely on the availability of appropriate data sources. The availability of pre-computed data, such as DP type and regularity [10] are crucial for lowering the required computational costs. As a consequence, we will follow the path of open data and pre-compute several features that might be relevant for route selection. Furthermore, we plan to provide an API[8] that will ease the access to our framework and allow to compute a winner route with minimal effort.

Although for most cases only the best result (i.e., the winner route) is relevant, there might be cases were the comparison between routes is of interest. Therefore, it is reasonable to study whether the Euclidean distance is actually justifiable by means of empirical results: The distance metric chosen should reflect empirical results, i.e., if participants are subject to routes which differ more, less comparable results should occur and vice versa. We are going to conduct within-group design wayfinding studies on this problem.

References

1 Vanessa Joy A. Anacta, Jia Wang, and Angela Schwering. Routes to remember: Comparing verbal instructions and sketch maps. In Joaquín Huerta, Sven Schade, and Carlos Granell, editors, *Connecting a Digital Europe Through Location and Place - International AGILE'2014 Conference, Castellon, Spain, 13-16 June, 2014*, Lecture Notes in Geoinformation and Cartography, pages 311–322. Springer, 2014. doi:10.1007/978-3-319-03611-3_18.

2 Crystal J. Bae and Daniel R. Montello. Dyadic route planning and navigation in collaborative wayfinding. In Sabine Timpf, Christoph Schlieder, Markus Kattenbeck, Bernd Ludwig, and Kathleen Stewart, editors, *14th International Conference on Spatial Information Theory, COSIT 2019, September 9-13, 2019, Regensburg, Germany*, volume 142 of *LIPIcs*, pages 24:1–24:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.COSIT.2019.24.

3 Christina Bauer and Bernd Ludwig. Schematic maps and indoor wayfinding. In Sabine Timpf, Christoph Schlieder, Markus Kattenbeck, Bernd Ludwig, and Kathleen Stewart, editors, *14th International Conference on Spatial Information Theory, COSIT 2019, September 9-13, 2019, Regensburg, Germany*, volume 142 of *LIPIcs*, pages 23:1–23:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.COSIT.2019.23.

4 Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition, 2008.

5 Geoff Boeing. Planarity and street network representation in urban form analysis. *Environment and Planning B: Urban Analytics and City Science*, 2018. doi:10.1177/2399808318802941.

6 Kenneth Bollen, John T. Cacioppo, Robert M. Kaplan, Jon A. Krosnick, and James L. Olds. Social, behavioral, and economic sciences perspectives on robust and reliable science. Report

[8] Check https://geoinfo.geo.tuwien.ac.at/index.php/resources/ for updates

of the Subcommittee on Replicability in Science Advisory Committee to the National Science Foundation Directorate for Social, Behavioral, and Economic Sciences, 2015. last access on Mar 5th, 2020. URL: https://www.nsf.gov/sbe/AC_Materials/SBE_Robust_and_Reliable_Research_Report.pdf.

7 Tad T. Brunyé, Shaina B. Martis, Breanne Hawes, and Holly A. Taylor. Risk-taking during wayfinding is modulated by external stressors and personality traits. *Spatial Cognition & Computation*, 19(4):283–308, 2019.

8 Vincenzo Carletti, Pasquale Foggia, Alessia Saggese, and Mario Vento. Challenging the time complexity of exact subgraph isomorphism for huge and dense graphs with vf3. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:804–818, 2018.

9 Matt Duckham, Stephan Winter, and Michelle Robinson. Including landmarks in routing instructions. *Journal of Location Based Services*, 4(1):28–52, 2010.

10 Paolo Fogliaroni, Dominik Bucher, Nikola Jankovic, and Ioannis Giannopoulos. Intersections of Our World. In Stephan Winter, Amy Griffin, and Monika Sester, editors, *10th International Conference on Geographic Information Science (GIScience 2018)*, pages 3:1–3:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

11 Ioannis Giannopoulos, David Jonietz, Martin Raubal, Georgios Sarlas, and Lisa Stähli. Timing of pedestrian navigation instructions. In Eliseo Clementini, Maureen Donnelly, May Yuan, Christian Kray, Paolo Fogliaroni, and Andrea Ballatore, editors, *13th International Conference on Spatial Information Theory, COSIT 2017, September 4-8, 2017, L'Aquila, Italy*, volume 86 of *LIPIcs*, pages 16:1–16:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.COSIT.2017.16.

12 Ioannis Giannopoulos, Peter Kiefer, and Martin Raubal. GazeNav: Gaze-Based Pedestrian Navigation. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices & Services*, MobileHCI '15, pages 337–346. ACM, 2015.

13 Charalampos Gkonos, Ioannis Giannopoulos, and Martin Raubal. Maps, vibration or gaze? Comparison of novel navigation assistance in indoor and outdoor environments. *Journal of Location Based Services*, 11(1):29–49, 2017.

14 Jana Götze and Johan Boye. "turn left" versus "walk towards the café": When relative directions work better than landmarks. In Fernando Bação, Maribel Yasmina Santos, and Marco Painho, editors, *AGILE 2015 - Geographic Information Science as an Enabler of Smarter Cities and Communities, Lisboa, Portugal, 9-12 June 2015*, Lecture Notes in Geoinformation and Cartography, pages 253–267. Springer, 2015. doi:10.1007/978-3-319-16787-9_15.

15 Jiawei Han, Micheline Kamber, and Jian Pei. 2 - getting to know your data. In Jiawei Han, Micheline Kamber, and Jian Pei, editors, *Data Mining (Third Edition)*, The Morgan Kaufmann Series in Data Management Systems, pages 39–82. Morgan Kaufmann, Boston, third edition edition, 2012.

16 Gengen He, Toru Ishikawa, and Makoto Takemiya. Collaborative navigation in an unfamiliar environment with people having different spatial aptitudes. *Spatial Cognition & Computation*, 15(4):285–307, 2015.

17 Toru Ishikawa and Uiko Nakamura. Landmark selection in the environment: Relationships with object characteristics and sense of direction. *Spatial Cognition & Computation*, 12(1):1–22, 2012.

18 Markus Kattenbeck. How subdimensions of salience influence each other. comparing models based on empirical data. In Eliseo Clementini, Maureen Donnelly, May Yuan, Christian Kray, Paolo Fogliaroni, and Andrea Ballatore, editors, *13th International Conference on Spatial Information Theory, COSIT 2017, September 4-8, 2017, L'Aquila, Italy*, volume 86 of *LIPIcs*, pages 10:1–10:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.COSIT.2017.10.

19 Markus Kattenbeck, Eva Nuhn, and Sabine Timpf. Is salience robust? A heterogeneity analysis of survey ratings. In *10th International Conference on Geographic Information Science, GIScience 2018, August 28-31, 2018, Melbourne, Australia*, volume 114 of *LIPIcs*,

pages 7:1–7:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.GISCIENCE.2018.7.

20 Peter Kedron, Amy E. Frazier, Andrew B. Trgovac, Trisalyn Nelson, and A. Stewart Fotheringham. Reproducibility and replicability in geographical analysis. *Geographical Analysis*, NA(NA):NA, 2020. doi:10.1111/gean.12221.

21 Peter Kiefer, Ioannis Giannopoulos, and Martin Raubal. Where am I? Investigating map matching during self-localization with mobile eye tracking in an urban environment. *Transactions in GIS*, 18(5):660–686, 2014.

22 Vasiliki Kondyli, Carl P. L. Schultz, and Mehul Bhatt. Evidence-based parametric design: Computationally generated spatial morphologies satisfying behavioural-based design constraints. In Eliseo Clementini, Maureen Donnelly, May Yuan, Christian Kray, Paolo Fogliaroni, and Andrea Ballatore, editors, *13th International Conference on Spatial Information Theory, COSIT 2017, September 4-8, 2017, L'Aquila, Italy*, volume 86 of *LIPIcs*, pages 11:1–11:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.COSIT.2017.11.

23 Markus Konkol, Christian Kray, and Morin Ostkamp. Follow the signs - countering disengagement from the real world during city exploration. In Arnold K. Bregt, Tapani Sarjakoski, Ron van Lammeren, and Frans Rip, editors, *Societal Geo-innovation - Selected Papers of the 20th AGILE Conference on Geographic Information Science, Wageningen, The Netherlands, 9-12 May 2017*, Lecture Notes in Geoinformation and Cartography, pages 93–109, 2017. doi:10.1007/978-3-319-56759-4_6.

24 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, 2019.

25 Hengshan Li and Nicholas A. Giudice. Assessment of between-floor structural and topological properties on cognitive map development in multilevel built environments. *Spatial Cognition & Computation*, 18(3):138–172, 2018.

26 Hua Liao, Weihua Dong, Haosheng Huang, Georg Gartner, and Huiping Liu. Inferring user tasks in pedestrian navigation from eye movement data in real-world environments. *International Journal of Geographical Information Science*, 33(4):739–763, 2019.

27 Lynn S. Liben, Lauren J. Myers, and Adam E. Christensen. Identifying locations and directions on field and representational mapping tasks: Predictors of success. *Spatial Cognition & Computation*, 10(2-3):105–134, 2010.

28 K L Lovelace, M Hegarty, and D R Montello. Elements of good route directions in familiar and unfamiliar environments. In Freksa, C. and Mark, D. M., editor, *Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science*, Lecture Notes in Computer Science, pages 65–82, 1999.

29 William A. Mackaness, Phil J. Bartie, and Candela Sanchez-Rodilla Espeso. Understanding information requirements in "text only" pedestrian wayfinding systems. In *Geographic Information Science - 8th International Conference, GIScience 2014, Vienna, Austria, September 24-26, 2014. Proceedings*, volume 8728 of *Lecture Notes in Computer Science*, pages 235–252. Springer, 2014. doi:10.1007/978-3-319-11593-1_16.

30 N. Magdy, M. A. Sakr, T. Mostafa, and K. El-Bahnasy. Review on trajectory similarity measures. In *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 613–619, December 2015.

31 Stefan Münzer and Christoph Stahl. Learning routes from visualizations for indoor wayfinding: Presentation modes and individual differences. *Spatial Cognition & Computation*, 11(4):281–312, 2011.

32 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.

33 Christina Ohm, Manuel Müller, and Bernd Ludwig. Evaluating indoor pedestrian navigation interfaces using mobile eye tracking. *Spatial Cognition & Computation*, 17(1-2):89–120, 2017.

34 Open Science Collaboration. Reproducibility Project: Psychology, 2015.

35 Frank O. Ostermann and Carlos Granell. Advancing science with vgi: Reproducibility and replicability of recent studies using vgi. *Transactions in GIS*, 21(2):224–237, 2017.

36 Marianna Pagkratidou, Alexia Galati, and Marios Avraamides. Do environmental characteristics predict spatial memory about unfamiliar environments? *Spatial Cognition & Computation*, 20(1):1–32, 2020.

37 Martin Perebner, Haosheng Huang, and Georg Gartner. Applying user-centred design for smartwatch-based pedestrian navigation system. *Journal of Location Based Services*, 13(3):213–237, 2019.

38 Karl Rehrl, Elisabeth Häusler, and Sven Leitinger. Comparing the effectiveness of gps-enhanced voice guidance for pedestrians with metric- and landmark-based instruction sets. In *Geographic Information Science, 6th International Conference, GIScience 2010, Zurich, Switzerland, September 14-17, 2010. Proceedings*, volume 6292 of *Lecture Notes in Computer Science*, pages 189–203. Springer, 2010. doi:10.1007/978-3-642-15300-6_14.

39 Karl Rehrl, Elisabeth Häusler, Sven Leitinger, and Daniel Bell. Pedestrian navigation with augmented reality, voice and digital map: final results from an in situ field study assessing performance and user experience. *Journal of Location Based Services*, 8(2):75–96, 2014.

40 Karl Rehrl, Sven Leitinger, Georg Gartner, and Felix Ortag. An analysis of direction and motion concepts in verbal descriptions of route choices. In Kathleen Stewart Hornsby, Christophe Claramunt, Michel Denis, and Gérard Ligozat, editors, *Spatial Information Theory, 9th International Conference, COSIT 2009, Aber Wrac'h, France, September 21-25, 2009, Proceedings*, volume 5756 of *Lecture Notes in Computer Science*, pages 471–488. Springer, 2009. doi:10.1007/978-3-642-03832-7_29.

41 Adam Rousell and Alexander Zipf. Towards a landmark-based pedestrian navigation service using osm data. *ISPRS International Journal of Geo-Information*, 6(3):64, 2017.

42 Wiebke Schick, Marc Halfmann, Gregor Hardiess, Friedrich Hamm, and Hanspeter A. Mallot. Language cues in the formation of hierarchical representations of space. *Spatial Cognition & Computation*, 19(3):252–281, 2019.

43 Helmut Schrom-Feiertag, Volker Settgast, and Stefan Seer. Evaluation of indoor guidance systems using eye tracking in an immersive virtual environment. *Spatial Cognition & Computation*, 17(1-2):163–183, 2017.

44 S. Schwarzkopf, S. J. Büchner, C. Hölscher, and L. Konieczny. Perspective tracking in the real world: Gaze angle analysis in a collaborative wayfinding task. *Spatial Cognition & Computation*, 17(1-2):143–162, 2017.

45 Angela Schwering, Jakub Krukar, Rui Li, Vanessa Joy Anacta, and Stefan Fuest. Wayfinding through orientation. *Spatial Cognition & Computation*, 17(4):273–303, 2017.

46 Makoto Takemiya and Toru Ishikawa. I can tell by the way you use your walk: Real-time classification of wayfinding performance. In Max J. Egenhofer, Nicholas A. Giudice, Reinhard Moratz, and Michael F. Worboys, editors, *Spatial Information Theory - 10th International Conference, COSIT 2011, Belfast, ME, USA, September 12-16, 2011. Proceedings*, volume 6899 of *Lecture Notes in Computer Science*, pages 90–109. Springer, 2011. doi:10.1007/978-3-642-23196-4_6.

47 Makoto Takemiya and Toru Ishikawa. Strategy-based dynamic real-time route prediction. In Thora Tenbrink, John G. Stell, Antony Galton, and Zena Wood, editors, *Spatial Information Theory - 11th International Conference, COSIT 2013, Scarborough, UK, September 2-6, 2013. Proceedings*, volume 8116 of *Lecture Notes in Computer Science*, pages 149–168. Springer, 2013. doi:10.1007/978-3-319-01790-7_9.

48 Lin Wang, Weimin Mou, and Xianghong Sun. Development of landmark knowledge at decision points. *Spatial Cognition & Computation*, 14(1):1–17, 2014.

49 Flora Wenczel, Lisa Hepperle, and Rul von Stülpnagel. Gaze behavior during incidental and intentional navigation in an outdoor environment. *Spatial Cognition & Computation*, 17(1-2):121–142, 2017.

Traffic Congestion Aware Route Assignment

Sadegh Motallebi
The University of Melbourne, Australia
s.motallebi@student.unimelb.edu.au

Hairuo Xie
The University of Melbourne, Australia
xieh@unimelb.edu.au

Egemen Tanin
The University of Melbourne, Australia
etanin@unimelb.edu.au

Kotagiri Ramamohanarao
The University of Melbourne, Australia
kotagiri@unimelb.edu.au

Abstract

Traffic congestion emerges when traffic load exceeds the available capacity of roads. It is challenging to prevent traffic congestion in current transportation systems where vehicles tend to follow the shortest/fastest path to their destinations without considering the potential congestions caused by the concentration of vehicles. With connected autonomous vehicles, the new generation of traffic management systems can optimize traffic by coordinating the routes of all vehicles. As the connected autonomous vehicles can adhere to the routes assigned to them, the traffic management system can predict the change of traffic flow with a high level of accuracy. Based on the accurate traffic prediction and traffic congestion models, routes can be allocated in such a way that helps mitigating traffic congestions effectively. In this regard, we propose a new route assignment algorithm for the era of connected autonomous vehicles. Results show that our algorithm outperforms several baseline methods for traffic congestion mitigation.

2012 ACM Subject Classification Information systems → Geographic information systems

Keywords and phrases Road Network, Traffic Congestion, Route Assignment, Shortest Path

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.9

1 Introduction

Traffic congestion has significant negative impact on the economy and public health in many countries. For example, road users in the United States wasted at least 6.9 billion hours and 3.1 billion gallons of fuel in a recent year due to traffic congestions [19]. Traffic congestion generally appears when traffic demand for certain roads exceeds the available capacity of the roads. During a traffic congestion, the speed of vehicles reduces, leading to longer travel times. Statistics show that traffic congestions affect the central area of a city more than the surrounding suburbs [23].

Navigating vehicles with the optimized routes can reduce traffic congestion significantly [11, 15, 3]. However, existing approaches are focused on vehicle-level route optimizations where individual vehicle routes are optimized independent to each other. The next generation of vehicles, *connected autonomous vehicles (CAVs)*, can drive with the minimal need for human driver's intervention. Based on our traffic management vision [17], such vehicles bring a valuable opportunity to build a coordinated *traffic management system (TMS)* that can optimize traffic at the network-level for all vehicles. As CAVs are highly coordinated with TMS and rarely deviate from their given routes, TMS can optimize traffic by coordinating the routes of all vehicles.

A TMS that performs network-level route optimization with CAVs can manage traffic congestions effectively as the system can predict the future traffic congestions based on the demand and capacity of roads. For example, let us assume that a TMS can predict the traffic conditions in the central area of a city as shown in Figure 1, which illustrates the general behavior of traffic congestion around the area when the majority of vehicles are heading towards the center. Figure 1(a) shows how an increase of traffic demand results in the increase of congestion levels. Figure 1(b) shows how a traffic congestion on a grid road network propagates to a large area during a certain period of time. Given the traffic congestion prediction, the TMS can prevent the predicted traffic congestions by suggesting alternative routes to CAVs where possible. In this regard, the TMS has a crucial role in shaping the traffic such that vehicles can reach their destinations faster.

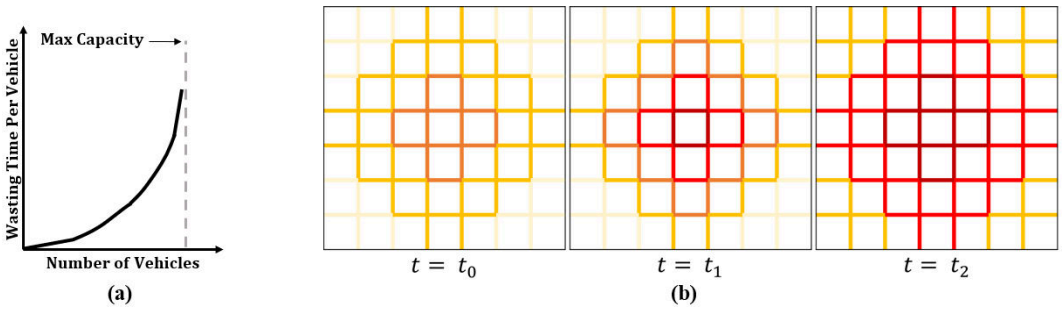


Figure 1 The change of traffic congestion with traffic demand and time: (a) Average waiting time per vehicle vs. the number of waiting vehicles [20]; (b) Traffic congestion propagation during a period when vehicles are heading toward the center of a city [25, 12]. Road links with darker colour have a higher level of congestions.

The simplest way to assign routes is by utilizing the shortest (fastest) path algorithms [5, 3]. However, this approach ignores the impact of routes future traffic conditions. Consequently, traffic congestions can form on the road segments that are shared by a large number of shortest paths. On the other hand, some algorithms assume that a route can affect the travel time of other vehicles [11, 15]. This study follows the same assumption. We want to assign routes to vehicles effectively to optimize traffic fluency at the network-level. Previously, we proposed a centralized routing algorithm for the aforementioned TMS [15]. Our algorithm reduces congestion by minimizing intersections between routes. In this work, we propose a route assignment algorithm, *Traffic Congestion Aware Route Assignment Algorithm (TCARA)*, to mitigate traffic congestions in the central area of a city. To help vehicles avoid future traffic congestions, the proposed algorithm uses certain predictive traffic congestion models that can estimate the effect of existing routes on the traffic in the future. As traffic optimization problems are NP-hard [10] and a TMS needs to respond to navigation requests in a short time, our method uses certain traffic heuristics to accelerate the route allocation process. We should note that traffic congestion can also happen because of unexpected issues like accidents. In such cases, a TMS can resolve the congestion reactively by rerouting vehicles. We left such cases for future work. Our algorithm differs from other algorithms substantially by proposing a predictive queue-based congestion model. Based on the model and certain aggregated traffic information, TCARA optimizes traffic in real time without predicting the detailed movement of all individual vehicles, which can result in huge savings in computation cost and storage cost. This allows TCARA to assign routes efficiently and enables it to outperform state-of-the-art algorithms significantly.

The main contributions of our work are summarized as follows.

- We propose a predictive congestion model for route assignment on roads in which the dynamic behavior of road links is considered.
- We propose a streaming route assignment algorithm based on the predictive congestion model.
- We evaluate our algorithm with a prototype traffic management system based on traffic simulation.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 defines the research problem. Section 4 presents the proposed algorithm. Section 5 reports experimental results. Section 6 concludes the paper.

2 Related Work

In this section, we first elaborate on route assignment optimization and the state-of-the-art algorithms in this area. Then, we review the existing traffic congestion models.

2.1 Route Assignment Approaches

Route assignment optimization is to find the optimal routes for a given set of trip queries. In this regard, there are two general approaches: user optimum [24] and system optimum [1]. The user-optimum approach, aims to reach an equilibrium state in which no vehicle can find a faster route than the assigned route. On the other hand, the system-optimum approach aims to minimize the total travel time for all vehicles. So, in the user-optimum approach, vehicles with the same source and destination get routes with the same travel time, while in the system-optimum approach, vehicles with the same source and destination might get routes with different travel times.

Route assignment can be static or dynamic. A static traffic assignment is applicable when the traffic condition is almost stable and route assignment does not lead to the change of traffic conditions [14]. When traffic condition is not stable, such as when the flow of vehicles changes quickly like in rush hours, route assignment needs to be dynamic which means the routes need to be assigned based on the changing traffic conditions [2, 6]. This study is about a dynamic route assignment algorithm that follows the system-optimum approach. Existing algorithms in this area are mainly focused on diversifying traffic on alternative routes to decrease traffic congestion. To achieve this goal, Nguyen et al. [16] propose a modified version of A* algorithm which suggests alternative routes to vehicles with the same source and destination. They propose a heuristic function that adds randomness into the computations of paths. Jeong et al. [11] propose a Self-Adaptive Interactive Navigation Tool (SAINT) which computes a set of shortest paths for a given source and destination and selects the path that leads to the minimum increase of congestion level. Vehicles with the same source and destination are likely to get different routes from SAINT. Zhang et al. [27] propose an algorithm, DIFTOS, which suggests the shortest path to vehicles initially and reroutes vehicles based on traffic congestion prediction. As traffic conditions may change and DIFTOS needs to maintain traffic load for roads at different times, it costs more time and space compared to other methods. Our previous work addresses a key problem that causes congestion, which is the intersection of routes at road junctions [15]. We proposed an algorithm, named MIRA, in which routes are less likely to intersect at junctions compared to suggesting shortest paths. To assign a route, MIRA divides the road network into blocks and maintains a heat map showing the average travel times for roads. MIRA also maintains a

reservation graph showing the impact of allocated routes at each road link on the routes. By having the data structures, it suggests routes that detour the congested blocks and road links. We show that the detouring policy leads to a significant reduction of travel time. Among the described methods, we consider SAINT and MIRA as two baseline methods. It is worth mentioning that there are iterative dynamic route assignment algorithms [21, 22]. However, as their time complexity is significantly high, we do not consider them in this study.

2.2 Traffic Congestion Models

Traffic congestion occurs when the traffic load of a road exceeds the available capacity of the road, leading to the increase of travel time due to the decrease of vehicle speed [13]. According to the literature, congestion on roads leads to the queueing of vehicles. So, the queue length is a good indicator to quantify congestion levels because a longer queue length generally indicates a longer travel time on roads [8, 26, 7]. There are also studies that model traffic congestion based on historical data [12, 25, 4]. However, the historical data might not always be available. Moreover, such models cannot model traffic congestions that are not captured well in the historical data. In this study, we use a traffic congestion model based on queue length.

The queue length is normally measured by the number of vehicles with very low speed on a road link. It has been used as a measure of congestion named *pressure* [7]. This simple but effective measure reveals the congestion level. Pressure-based models are mainly used for finding the optimal schedule of traffic lights [7, 26, 9]. We utilize this model in our route assignment algorithms to predict traffic congestion. For a given road network $G(V, E)$, any edge $e \in E$ may have a queue of vehicles waiting at the end vertex (intersection). Whenever a vehicle stops at an intersection, it adds to the corresponding queue, and after finishing the edge (i.e., passing the intersection), it leaves the queue. An example scenario with traffic queues at several intersections is illustrated in Figure 2.

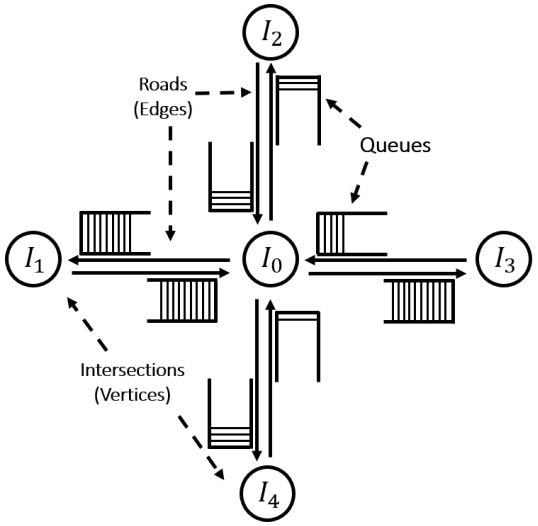


Figure 2 Road network queue model: each edge has a queue containing the vehicles waiting at its end vertex. More vehicles in a queue indicates higher congestion levels.

It is observed that the congestion level increases linearly with an increase of traffic demand before the traffic demand reaches a certain threshold, after which the congestion increases non-linearly. Although the basic version of pressure addresses the linear relationship between

traffic load and traffic congestion, it cannot follow the nonlinear behavior of congestion on the links. Gregoire et al. [7] propose an enhanced version of pressure. The proposed pressure function (i.e., $C(Q_e)$ defined in Equation 1) models the relationship between the queue length and the traffic congestion of a road link based on certain key characteristics of road links [7]. Although the pressure function is complex, it has only one variable input, which is the queue length. In Equation 1, Q_e and C_e are the queue length at edge e and the maximum capacity of edge e, respectively. C_∞ and m are two constant parameters. The first parameter, C_∞, determines the behavior of edge e for light traffic, and the second parameter, m, is used for tuning the transition point from linear behavior to nonlinear behavior of edge e. In Section 5, the model with different values for C_∞ and m is analyzed. The model computes the current congestion value based on the existing vehicles on the roads. The value of the computed pressure varies between zero (when no vehicle waits on a road, i.e., empty queue) and one (when the road is full). This pressure can be considered as a real-time congestion model as it is based on the real-time queue lengths. To model future traffic congestions, the traffic congestion model needs to be updated such that Q_e is based on the number of vehicles that are going to wait on the roads. To assign routes, we utilize the updated version of this model in our algorithm (Section 4).

$$C(Q_e) = min(1, \frac{\frac{Q_e}{C_\infty} + (2 - \frac{C_e}{C_\infty})(\frac{Q_e}{C_e})^m}{1 + (\frac{Q_e}{C_e})^{m-1}}) \tag{1}$$

3 Problem Definition

▶ **Definition 1** (Delay Function). *A delay function $\epsilon(r_i, r_j)$ models the effect of one vehicle with route r_j on a vehicle with route r_i.*

The delay function gives an extra delay that the vehicle with r_i experiences because of the existence of the vehicle with r_j. Apparently, when $i = j$, the outcome of the epsilon function is zero as no vehicle has an impact on itself.

▶ **Definition 2** (Delayed Travel Time). *A delayed travel time $DTT(R'|R)$ is the total travel time of vehicles with all the routes in R' when there are existing vehicles with all the routes in R.*

Based on the definition, $DTT(r|\emptyset)$ is the shortest possible travel time of a vehicle with route r, which can be achieved when there is no existing vehicle on the road network. Equation 2 models travel time of a new vehicle with route r when there are already n vehicles with assigned routes on the network. The set R contains all the routes of the n existing vehicles. Each of the existing vehicles can affect the travel time of the new vehicle.

$$DTT(\{r\}|R) = DTT(\{r\}|\emptyset) + \sum_{j=1}^{n} \epsilon(r|r_j) \tag{2}$$

In this study, we assume trip queries arrive at a TMS in a streaming fashion. The streaming route assignment problem is defined in Equation 3 for a given trip query (i.e., a pair of source and destination locations).

$$r* = \underset{r \in \mathcal{R}_{candidate}}{\arg\min} \; DTT(\{r\}|\emptyset) + \sum_{j=1}^{n} \epsilon(r|r_j) \tag{3}$$

Problem Statement: Given a trip query from a user and a set of n existing vehicles, find the optimum route r^* among the set of candidate routes $R_{candidate}$ such that the travel time of the user is minimized (Equation 3).

4 Traffic Congestion Aware Route Assignment Algorithm (TCARA)

We propose an algorithm, *Traffic Congestion Aware Route Assignment Algorithm (TCARA)*, for optimizing route allocation based on the navigation requests from CAVs. TCARA is based on the A* algorithm that finds route in a weighted graph. The weights are computed based on the aforementioned congestion model (Section 2.2). As the congestion model uses the predicted queue length to estimate future congestion levels, it is important to get an accurate prediction of the queue length. For this purpose, we define **Allocated Capacity (AC)** based on the existing routes.

Allocated capacity shows the impact of a vehicle on the queue length at specific road links. When a vehicle is currently on a road link, we define the allocated capacity of the vehicle at the edge as 1. When the vehicle leaves the link, the AC of this vehicle at the edge is 0. The AC values of the vehicle at the links on the remaining of route are higher than 0 but less than 1. The AC values decrease gradually for the links farther away from the current link of the vehicle, indicating the diminishing impact from the vehicle on the traffic conditions that are further away into the future. We define AC based on the average travel times (showing traffic condition of roads) in Equation 4. In the equation, AC_{e_i} and TT_{e_i} represent the allocated capacity of the i^{th} edge e_i in a given route $r =< e_1, ..., e_n >$ and the travel time of e_i, respectively. Here, n is the number of road links between the vehicle's current position and the destination. The travel time at road links is updated frequently by a TMS. Whenever a vehicle leaves a road link, ACs for the rest of the route are recomputed. The aggregated value of the ACs at an edge is used as the predicted queue length Q_e for the edge in the congestion model (Section 2.2).By assigning the ACs we would be able to quantify the influence of all the route allocations on the traffic of a specific edge.

$$AC_{e_i} = 1 - \frac{\sum_{j=1}^{i-1} TT_{e_j}}{\sum_{j=1}^{n} TTe_j} \tag{4}$$

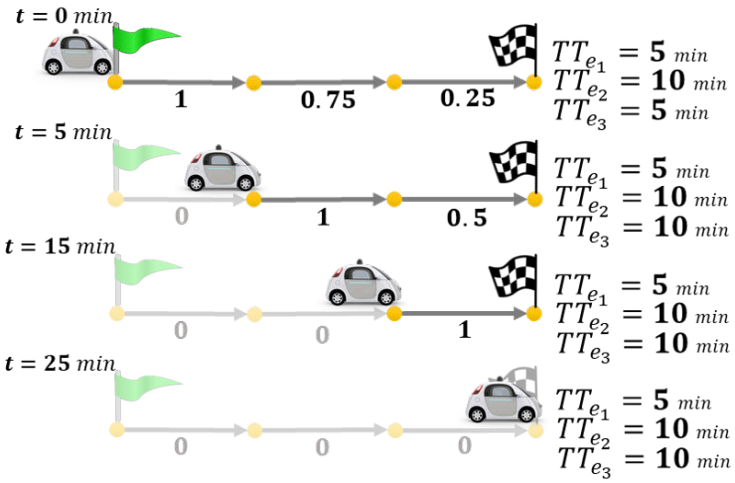


Figure 3 Updating allocated capacity for a given route over time. At $t = 5$ when the vehicle leaves e_1, the travel time at e_3 increases in 10 min.

Figure 3 shows how AC is computed for a vehicle during its trip. Let us assume a three-link route is assigned for a given source and destination. As shown in the figure, the AC values at the start of the trip are $1 - \frac{0}{5+10+5} = 1$, $1 - \frac{5}{5+10+5} = 0.75$, and $1 - \frac{5+10}{5+10+5} = 0.25$

for e_1, e_2, and e_3, respectively. Whenever the vehicle leaves a road segment, the AC values for the rest of the route are recomputed. Let us assume the travel time of e_3 increases to 10 minutes at the 5^{th} minute. The new travel time will be used when updating AC values after the time point. As the vehicle leaves the first link, AC_{e_1} becomes 0. The second and third links get $1 - \frac{0}{10+10} = 1$ and $1 - \frac{10}{10+10} = 0.5$, respectively. The same procedure runs for the updates at the $15th$ minute. At the end of the trip, all AC values become 0.

TCARA needs to maintain the aggregated AC values at road links. The AC values are used to capture the pressure at the road links based on the congestion model. The traffic management system (TMS) is responsible for keeping the AC values updated based on the received location updates from the vehicles. Algorithm 1 defines TCARA with details. This algorithm is based on the A* algorithm in which the congestion model is utilized as a heuristic function. TCARA computes the congestion values (pressure values at edges) during its route search. As vehicles are connected to the TMS, the real-time traffic conditions are available. For a given pair of source and destination, it computes a route with the minimum value of congestion. Once a new route is computed by TCARA, the TMS updates aggregated AC values at the edges on the route. Whenever a vehicle leaves a road link, the aggregated AC values need to be updated by the TMS as well. Although the route of the vehicle remains unchanged when the vehicle leaves an edge, the AC values at the edges in the rest of the vehicle's path get updated, which can affect the creation of new routes for other vehicles in the future.

The time complexity of TCARA is the same as Dijkstra's algorithm, $O(|V|log|V| + |E|)$. TCARA needs storage in order of $O(|V| + |E|)$ same as Dijkstra's algorithm. Also, it needs $O(n|E|)$ for storing the AC values of existing routes (n is the number of vehicles). The cost of updating the AC values with a route is $O(|E|)$.

5 Experiments

We evaluate the proposed algorithm TCARA. We focus on the traffic scenarios in cities and assume that there is no street blockage due to accidents or traffic light failures.

5.1 Baseline Approaches

We compare TCARA against several baseline methods, *First-In-First-Assigned Fastest (FIFA-Fastest)*, SAINT [11], and MIRA [15]. FIFA-Fastest uses Dijkstra's algorithm to compute routes with the minimum travel times. Although FIFA-Fastest is a simple algorithm, it is utilized in well-known navigation tools currently. However, the algorithm does not consider future traffic conditions as its computation is based on the current travel time at road links. SAINT is the second traffic assignment baseline method, as described in section 2.1. The third baseline method is MIRA, as described in section 2.1 as well. MIRA is a state-of-the-art route assignment algorithm. We also include an algorithm, *Time-wise Fastest Route Assignment (TFRA)*, as the fourth baseline method. Similar to TCARA, TFRA assigns routes based on traffic congestion prediction. Both algorithms use the same congestion model as shown in Equation 1. They differ in the computation of travel cost at the edges. Given a source and a destination, TFRA searches for a routes based on Dijkstra's algorithm. When the search expands to an edge, TFRA estimates the time point at which a vehicle with the new route arrives at the edge. Then, TFRA estimates the number of existing vehicles that would be at the edge at that time. The estimated value is used as the queue length (Q_e) in the congestion model. The pressure value compared with the model is then used as the weight (travel cost) of the edge.

Algorithm 1 Traffic Congestion Aware Route Assignment.

Input: Road network graph $G(V, E)$ where any edge $e_{m,n}$ has a weight $w(e_{m,n})$ that equals to the aggregated AC values at the edge, source s, destination d
Output: Route r from s to d

// Vertices in Q are always sorted based on the travel cost between s and the vertices.
$Q \leftarrow$ Empty-Priority-Queue()
for $m \in V$ **do**
    $cost_m \leftarrow \infty$; $m.previous \leftarrow NIL$; $m.time = 0$; $Q.insert(m)$
end for
$cost_s \leftarrow 0$
while Q is not empty **do**
    $m \leftarrow$ vertex in Q with the lowest cost to s
    remove m from Q
    if $m = d$ **then**
        break;
    end if
    for $n \in$ End points of the edges starting from m **do**
        $C_{m,n} \leftarrow C(w_{m,n})$ // Pressure value of $e_{m,n}$ based on Equation 1, where the value of Q_e is $w(e_{m,n})$
        if $cost_n > cost_m + C_{m,n}$ **then**
            $cost_n \leftarrow cost_m + C_{m,n}$
            $n.previous \leftarrow m$
        end if
    end for
end while
$m \leftarrow d$; $L \leftarrow$ Empty-Linked-List() ; $L.append(m)$
while $m \neq s$ **do**
    $m \leftarrow m.previous$
    $L.append(m)$
end while
Reverse L // The first item will be source after reverse
Return L

5.2 Experiment Environment

We create an experiment environment using a traffic simulator, SMARTS [18], which can perform real-time microscopic simulation for vehicles on road networks. Kotagiri et al. [18] show that SMARTS can preform realistic simulations. Moreover, SMARTS simulates adaptive traffic lights as in the real world, where traffic lights tune their light cycle based on incoming traffic flows. In our experiments, SMARTS generates trip queries and sends them to a route allocator, which computes routes and sends them to SMARTS. The routes are assigned to CAVs in SMARTS. Whenever a CAV leaves a road link, SMARTS gives this information immediately to the route allocator for updating the weights in TFRA/TCARA. SMARTS also sends updates of travel time to the route allocator periodically. The travel time of a road link is the average travel time of CAVs finished the link since the last report. If no CAV has traveled during a report time interval, we compute the average travel time as the road link length over the speed limit of the link. The average travel time is used for computing weights in TFRA/TCARA.

5.3 Performance Metrics

As route assignment algorithms aim to minimize travel time, we define two metrics based on the travel time of CAVs. The performance metric, *Travel Time Ratio at Individual level (TTRI)*, measures average travel time for individuals. It is defined in Equation 5, where $TT(v_i)$, $BTT(v_i)$, and $|\mathcal{V}|$ represent the actual travel time of vehicle v_i, the best travel time for vehicle v_i, and the number of all vehicles. The actual travel time is the travel time achieved by following the computed route, while the best travel time is computed based on the shortest path (in terms of travel time) assuming vehicles always travel at the free flow speed. We should note that it is not suitable to evaluate the algorithms using the optimum total travel time, which is the minimum travel time of all trip queries. This is because obtaining the optimum total travel time implies that trip queries need to be available at first, while this is not true in the streaming route assignment. Lower TTRI values are better. The best value is one when the actual travel time of vehicles is equal to their best theoretical travel times. We also measure *gridlock threshold*, which is the maximum number of vehicles that can finish their routes with no gridlock. An increase in traffic load and congestion can lead to a gridlock where no vehicle can move further. The TTRI metric has no meaningful values in a gridlock situation. So, all experiment results are limited to gridlock thresholds.

$$TTRI = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \frac{TT(v_i)}{BTT(v_i)} \tag{5}$$

5.4 Experimental Settings

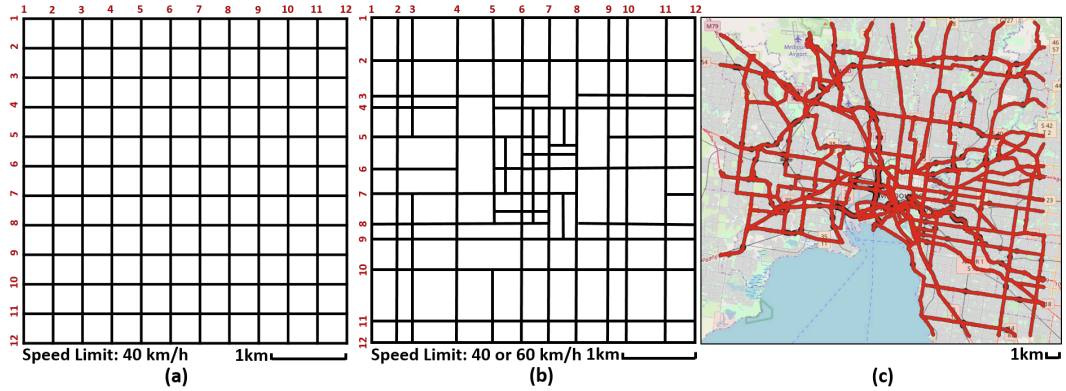


Figure 4 (a) The Manhattan-grid road network used in the experiments where all intersections are signalized (area statistics $4.4km \times 4.4km$) (b) The semi-real road network used in the experiments where all intersections are signalized (area statistics $4.4km \times 4.4km$) (c) The real road network (Metropolitan of Melbourne, Australia) used in the experiments (area statistics $30km \times 30km$) where most of intersections are signalized.

We investigate the impact of the number of vehicles and the impact of the spatial distribution of source and destination on all algorithms.

Number of Vehicles is an essential indicator of traffic conditions on the roads. Having more vehicles on the roads can increase congestion, and its impact can be examined by measuring travel times. A better algorithm can manage more traffic load with a higher gridlock threshold. When testing the effect of this parameter, we start from a certain value and increase the value gradually until all algorithms reach gridlock.

We consider two distributions for source and destination locations: uniform and Gaussian. The uniform distribution means the locations are uniformly distributed around the city, while Gaussian distribution means the locations are more likely to be around the city center. We define four source-destination distribution scenarios: 1) Uniform-Uniform (representing off-peak hours), 2) Uniform-Gaussian (representing morning peak hours), 3) Gaussian-Uniform (representing afternoon peak hours), and 4) Gaussian-Gaussian (representing an extreme case of congestion at the city center). The default value for this parameter is Gaussian-Gaussian.

We run two experiment sets. In each experiment set, we vary one parameter while keeping the other parameter at its default value. The first experiment set evaluates the effect of the number of vehicles, and the second experiment set evaluates the impact of the spatial distribution of source and destination. Both sets of experiments are conducted with three road networks as described below.

5.4.1 Manhattan-Grid Road Network

The Manhattan-grid network represents an urban area in which roads are organized as a grid, which can be seen in some urban areas like Manhattan in New York. It is a 12 by 12 network (Figure 4(a)). All intersections are signalized. A road link between two consecutive intersections is two-way and 400 meters long. Road links have the same maximum allowable speed, which is 40 km/h. These settings represent a structured city with the same block sizes and similar traffic rules. The default number of vehicles is 6000 (as this is the gridlock threshold for TFRA and SAINT algorithms on this network). The default spatial distribution is Gaussian-Gaussian.

5.4.2 Semi-real Road Network

We also experiment with a semi-real road network (Figure 4(b)). Compared to the previous network, the semi-real network has more intersections and road links. All intersections have traffic lights. This road network represents many real road networks of cities with a dense central part as a Central Business District area. The maximum speed allowed for each road link is uniformly random set as 40 km/h or 60km/h. The default value of vehicles is 6000. The default spatial distribution is Gaussian-Gaussian.

5.4.3 Real Road Network

The real road network covers a $30km \times 30km$ area in Melbourne, shown in Figure 4(c). The center of the road network is the CBD of Melbourne. The network is extracted from OpenStreetMap. We preprocessed the map and removed the intermediate nodes that are not real intersections. The default number of vehicles is 40000. The default spatial distribution of source and destination is Gaussian-Gaussian.

5.4.4 Parameter Tuning for TFRA and TCARA

The congestion models of TFRA and TCARA are based on the normalized pressure model expressed in Equation 1. The model has two parameters m and C_∞ to fit the behavior of road links. Figure 5 shows different outputs of the congestion model for four combinations of m and C_∞. For a road link, when C_∞ equals to the capacity of road link, C, the output is a straight line, shown in green in Figure 5. The bigger value of C_∞ results in more bending of the trend line, shown in blue. The curve needs to be tuned for each road network. We define the best parameter values for a given road network as the values that lead to the maximum

traffic fluency in terms of TTRI. It is worth mentioning that $C_\infty \geq C$ [7]. The effect of different m values is shown in the figure in blue and orange lines. Also, the figure depicts the output curves for two road links with different capacities in blue and red. Gregoire et al. [7] set $m = 4$ and C_∞ to the largest capacity of road links in the network. As all roads have almost the same length in the Manhattan-grid network, the model becomes linear which does not correspond to the non-linear behavior of roads mentioned in Section 4. To get larger values, we consider $C_\infty = \alpha C_{max}$. Based on our tests the best parameter values of TFRA and TCARA for the Manhattan-grid network are $m = 4$ and $\alpha = 11$, which result in the minimum value of TTRI. By doing the same procedure for the semi-real and real networks, the result shows that $m = 4$ and $\alpha = 1$ are the best values. The parameter values are also suggested in the original study [7].

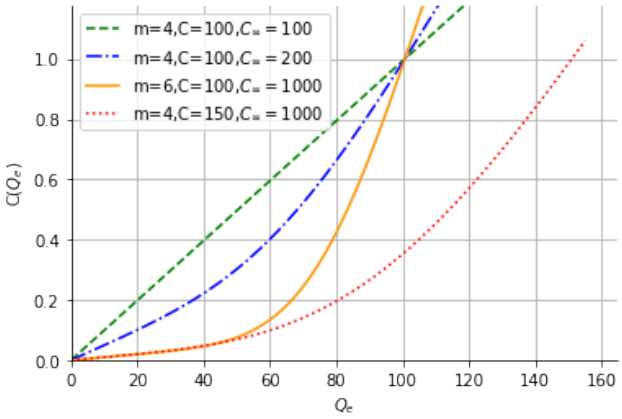


Figure 5 The capacity aware pressure function with different parameters.

5.5 Results

5.5.1 Manhattan-grid Road Network

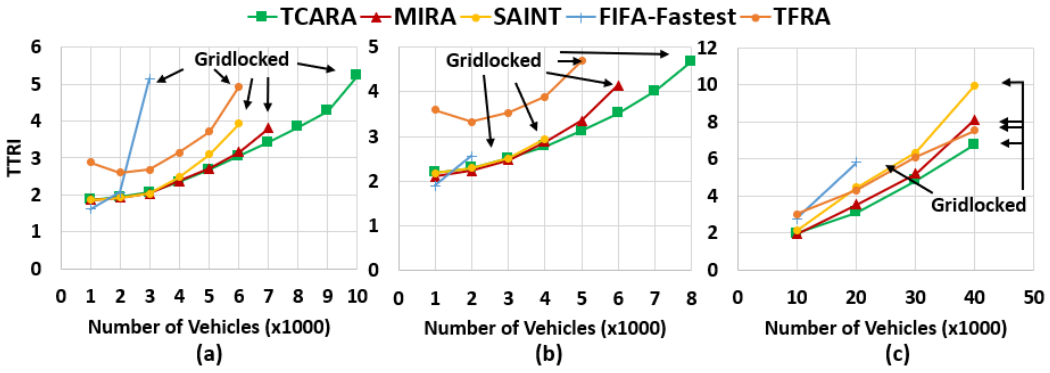


Figure 6 TTRI for all algorithms under different traffic loads (number of vehicles) for (a)Manhattan-grid network, (b)Semi-real network, and (c) real network.

TCARA outperforms all other algorithms except for the light traffic condition, as we expected in terms of TTRI (Figure 6(a)). As TCARA tries to avoid existing traffic when computing routes, it suggests longer routes than FIFA-Fastest routes when the traffic load

is low. However, under normal traffic load, TCARA outperforms other algorithms. The advantage of TCARA is significant when traffic load is high, which is generally accompanied by a high level of traffic congestions. As the congestion model helps TCARA to predict traffic congestion, TCARA can avoid congestion or reduce the propagation of congestion significantly. We summarize the result of the experiment in terms of the applicability of algorithms for different traffic loads in Table 1. The table expresses that for light traffic load, $n < 2k$, suggesting the fastest routes to vehicles is the best strategy, as there is no congestion and the impact of routes on each other is negligible. For the low traffic loads, $2k \leq n < 4k$, TCARA outperforms other algorithms slightly. Among the three candidates, SAINT is the worst choice as it has the biggest time complexity. For the high traffic load, $4k \leq n < 6k$, SAINT cannot avoid gridlocks, and TCARA outperforms MIRA slightly. For the intensive traffic load, $6k \leq n \leq 10k$, TCARA is the only algorithm that can manage traffic effectively. The result shows that TCARA can increase the gridlock threshold by 42% for the same road network compared with the second-best algorithm MIRA. The baseline algorithm, TFRA, does not outperform others except FIFA-Fastest. Its gridlock threshold is the same as SAINT in the Manhattan-grid network.

Table 1 Candidate algorithms for different traffic loads.

# vehicles	Candidate Algorithms	Description
$n < 2k$	FIFA-Fastest	The fastest and most effective
$2k \leq n < 4k$	TCARA, MIRA, SAINT	TCARA performs slightly better
$4k \leq n < 6k$	TCARA, MIRA	TCARA performs slightly better
$6k \leq n \leq 10k$	TCARA	The only workable algorithm for $7k \leq n$

5.5.2 Semi-real Road Network

The result of the semi-real road network (Figure 6(b)) indicates that TCARA outperforms all algorithms except for light traffic loads in terms of TTRI. The result shows that FIFA-fastest is less effective for the same traffic load compared with the previous experiment, but still is the best solution for light traffic with $1k$ vehicles. The figure shows that TCARA increases the gridlock threshold from the second-best approach, MIRA, by 33%. Comparing the result of the Manhattan-grid and semi-real network, we can see that a more complex road network topology and a larger variation in speed limits affect the maximum gridlock threshold for all algorithms significantly. The maximum gridlock threshold decreases by 20% when the network changes from the Manhattan-grid network to the semi-real network.

5.5.3 Real Road Network

TCARA outperforms all other algorithms under all traffic loads in terms of TTRI for the real road network (Figure 6(c)). The figure shows that for $10k$ vehicles TCARA, MIRA, and SAINT have no significant difference. FIFA-Fastest performs ineffectively and reaches a gridlock situation at $20k$. The other algorithms face gridlock at $40k$. TCARA outperforms MIRA and SAINT by 17% and 32% in terms of TTRI, respectively. By comparing the results with different maps, we can conclude that the topology of road networks plays a crucial rule in traffic optimization. Moreover, accurate traffic congestion prediction, as achieved with TCARA, can help decrease traffic congestion considerably.

5.5.4 Source and Destination Distribution

Figure 7 shows clearly that the distribution of trips affects traffic flow considerably. In the off-peak (Uniform-Uniform) situation, traffic is distributed uniformly and there is no heavy congestion. So, the performance of different algorithms is very close to each other. The figure shows that TCARA is stable for all distributions in all networks. It outperforms all algorithms in most scenarios as it benefits from a predictive congestion model that helps it to suggest routes with sufficient detours. Also, TCARA is stable in all situations while others are sensitive to the road network structure, the traffic distribution, or both. The results show that the baseline algorithm TFRA works well in off-peak hours. Moreover, by comparing the results, we can conclude that TFRA works with the real network better than with other networks. It can be because the travel time estimation becomes more accurate when the network structure becomes denser at the center. The FIFA-Fastest algorithm faces gridlock in all scenarios except for the uniform-uniform (off-peak) scenario. From the result, we can conclude that the algorithms following the system optimum approach (i.e., all algorithms except FIFA-Fastest) manage traffic significantly better compared with the current navigation systems that optimize routes independently based on current traffic conditions.

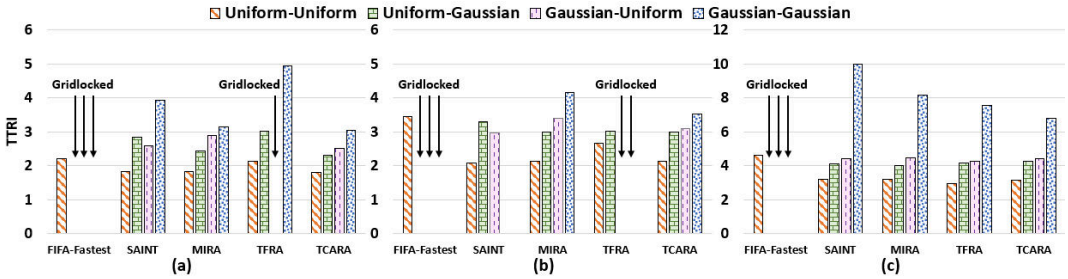


Figure 7 TTRI for different spatial distribution of trips and road networks: (a) Manhattan-grid network with 6000 vehicles (b) Semi-real network with 6000 vehicles (c) Real network with 40000 vehicles.

5.6 Time Complexity

In this experiment, we compare the computation time of the algorithms based on synthetic grid networks with 1000 to 10000 vertices. Figure 8 shows that FIFA-Fastest is the fastest algorithm, and SAINT is the slowest algorithm. Although the time complexity of TFRA, TCARA, MIRA, and FIFA-Fastest are the same (i.e., $O(|V|log|V|+|E|)$), FIFA-Fastest runs faster than others as it has the smallest overhead (i.e., the cost for computing edge weights). The result shows that TCARA is fast enough for practical use as it can compute a route in less than 100 milliseconds.

6 Conclusions and Future Work

In this study, we proposed a route assignment algorithm TCARA. We showed that how a predictive congestion model can help reduce traffic congestion significantly. We evaluated TCARA under different traffic loads, with various road networks, and different spatial distribution of source and destination. We showed that TCARA suggests faster routes compared with the state-of-the-art algorithms. TCARA is tailored for the era of CAVs, where all the vehicles are coordinated by a central traffic management system. A possible

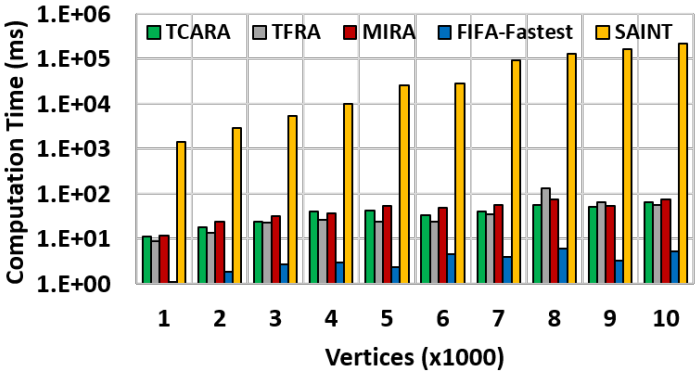


Figure 8 Computation time achieved with all algorithms for different road network sizes. The number of vertices varies from 1000 to 10000.

direction of future work is to incorporate traffic lights directly in our model. In this regard, considering the light cycles as a parameter to enhance the traffic congestion model and investigating the models for a road network that has a mix of signalized and unsignalized intersections are the next steps to extend our algorithm. Another possible direction is to extend the algorithm for situations when vehicles are not fully autonomous, and the drivers can decide about their routes which adds unpredictability to the problem. Also, such real-time network-level traffic optimization can be utilized in the solutions for transport applications like for transport-as-a-service when there is no personal vehicle and all vehicles are CAVs. So, a central system navigates all CAVs, while the system receives trip queries in a streaming fashion.

References

1 Martin Beckmann, Charles B McGuire, and Christopher B Winsten. Studies in the economics of transportation. https://trid.trb.org/view/91120, 1956.

2 Yi-Chang Chiu, Jon Bottom, Michael Mahut, Alex Paz, Ramachandran Balakrishna, Travis Waller, and Jim Hicks. Dynamic traffic assignment: A primer. *Transportation Research Circular*, 2011.

3 Ugur Demiryurek, Farnoush Banaei-Kashani, and Cyrus Shahabi. A case for time-dependent shortest path computation in spatial networks. In *SIGSPATIAL*, pages 474–477. ACM, 2010.

4 Xiaolei Di, Yu Xiao, Chao Zhu, Yang Deng, Qinpei Zhao, and Weixiong Rao. Traffic congestion prediction by spatiotemporal propagation patterns. In *IEEE MDM*, pages 298–303, June 2019.

5 Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

6 Terry L Friesz and D Bernstein. Analytical dynamic traffic assignment models. In *Handbook of transport modelling*, pages 181–195. Elsevier, 2000.

7 Jean Gregoire, Xiangjun Qian, Emilio Frazzoli, Arnaud de La Fortelle, and Tichakorn Wongpiromsarn. Capacity-aware backpressure traffic signal control. *IEEE TCNS*, 2(2):164–173, June 2015.

8 Randolph W Hall. Transportation queueing. In *Handbook of Transportation Science*, pages 113–153. Springer, 2003.

9 Hsu-Chieh Hu and Stephen F. Smith. Softpressure: A schedule-driven backpressure algorithm for coping with network congestion. In *IJCAI*, pages 4324–4330, 2017.

10 Olaf Jahn, Rolf H Möhring, Andreas S Schulz, and Nicolás E Stier-Moses. System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations research*, 53(4):600–616, 2005.

11 Jaehoon Jeong, Hohyeon Jeong, Eunseok Lee, Tae Oh, and David Du. SAINT: Self-adaptive interactive navigation tool for cloud-based vehicular traffic optimization. *IEEE TVT*, 65(6):4053–4067, 2016.

12 Yuxuan Liang, Zhongyuan Jiang, and Yu Zheng. Inferring traffic cascading patterns. In *SIGSPATIAL*, pages 2:1–2:10. ACM, 2017.

13 Tim Lomax, Shawn Turner, Gordon Shunk, Herbert S. Levinson, Richard H. Pratt, Paul N. Bay, and G. Bruce Douglas. *Quantifying Congestion, Volume 1: Final Report*. National Academy Press, Washington, D.C., 1997. URL: `https://trid.trb.org/view/475257`.

14 Marin Lujak, Stefano Giordani, and Sascha Ossowski. Route guidance: Bridging system and user optimization in traffic assignment. *Neurocomputing*, 151:449–460, 2015.

15 Sadegh Motallebi, Hairuo Xie, Egemen Tanin, Jianzhong Qi, and Kotagiri Ramamohanarao. Streaming route assignment for connected autonomous vehicles (systems paper). In *SIGSPATIAL*, page 408–411. ACM, 2019.

16 Uyen TV Nguyen, Shanika Karunasekera, Lars Kulik, Egemen Tanin, Rui Zhang, Haolan Zhang, Hairuo Xie, and Kotagiri Ramamohanarao. A randomized path routing algorithm for decentralized route allocation in transportation networks. In *SIGSPATIAL*, pages 15–20. ACM, 2015.

17 Kotagiri Ramamohanarao, Jianzhong Qi, Egemen Tanin, and Sadegh Motallebi. From how to where: Traffic optimization in the era of automated vehicles. In *SIGSPATIAL*, pages 10:1–10:4. ACM, 2017.

18 Kotagiri Ramamohanarao, Hairuo Xie, Lars Kulik, Shanika Karunasekera, Egemen Tanin, Rui Zhang, and Eman Bin Khunayn. SMARTS: Scalable microscopic adaptive road traffic simulator. *ACM TIST*, 8(2):26:1–26:22, 2016.

19 David Schrank, Bill Eisele, Tim Lomax, and Jim Bak. 2015 urban mobility scorecard. *Technical Report, Texas A&M Transportation Institute*, 2015.

20 Cambridge Systematics. Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation. Technical report, United States. Federal Highway Administration, 2005.

21 WY Szeto and Hong K Lo. Dynamic traffic assignment: properties and extensions. *Transportmetrica*, 2(1):31–52, 2006.

22 Nicholas B Taylor. The contram dynamic traffic assignment model. *Networks and Spatial Economics*, 3(3):297–322, 2003.

23 Marion Terrill. *Stuck in traffic? Road congestion in Sydney and Melbourne*, 2017. https://grattan.edu.au/report/stuck-in-traffic.

24 John Glen Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1(3):325–362, 1952.

25 Haoyi Xiong, Amin Vahedian, Xun Zhou, Yanhua Li, and Jun Luo. Predicting traffic congestion propagation patterns: A propagation graph approach. In *IWCTS*, pages 60–69. ACM, 2018.

26 Ali A Zaidi, Balázs Kulcsár, and Henk Wymeersch. Back-pressure traffic signal control with fixed and adaptive routing for urban vehicular networks. *IEEE TITS*, 17(8):2134–2143, 2016.

27 Weidong Zhang, Nyothiri Aung, Sahraoui Dhelim, and Yibo Ai. DIFTOS: A distributed infrastructure-free traffic optimization system based on vehicular ad hoc networks for urban environments. *Sensors*, 18(8), 2018.

Estimating Hourly Population Distribution Patterns at High Spatiotemporal Resolution in Urban Areas Using Geo-Tagged Tweets and Dasymetric Mapping

Jaehee Park
Department of Geography, San Diego State University, CA, USA
jpark1200@sdsu.edu

Hao Zhang
HDMA center, San Diego State University, CA, USA
zhanghaoshogo@gmail.com

Su Yeon Han
Department of Geography, San Diego State University, CA, USA
shunny1004@gmail.com

Atsushi Nara
Department of Geography, San Diego State University, CA, USA
anara@sdsu.edu

Ming-Hsiang Tsou[1]
Department of Geography, San Diego State University, CA, USA
mtsou@sdsu.edu

Abstract

This paper introduces a spatiotemporal analysis framework for estimating hourly changing population distribution patterns in urban areas using geo-tagged tweets (the messages containing users' geospatial locations), land use data, and dasymetric maps. We collected geo-tagged social media (tweets) within the County of San Diego during one year (2015) by using Twitter's Streaming Application Programming Interfaces (APIs). A semi-manual Twitter content verification procedure for data cleaning was applied first to separate tweets created by humans from non-human users (bots). The next step was to calculate the number of unique Twitter users every hour within census blocks. The final step was to estimate the actual population by transforming the numbers of unique Twitter users in each census block into estimated population densities with spatial and temporal factors using dasymetric maps. The temporal factor was estimated based on hourly changes of Twitter messages within San Diego County, CA. The spatial factor was estimated by using the dasymetric method with land use maps and 2010 census data. Comparing to census data, our methods can provide better estimated population in airports, shopping malls, sports stadiums, zoo and parks, and business areas during the day time.

2012 ACM Subject Classification Human-centered computing → Social media

Keywords and phrases Population Estimation, Twitter, Social Media, Dasymetric Map, Spatiotemporal

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.10

Funding This material is partially based upon work supported by the National Science Foundation under Grant No. 1416509 and Grant No. 1634641.Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

[1] Corresponding author

1 Introduction

The widespread use of social media and mobile phone data provides a great research opportunity for researchers to map and analyze dynamic human behaviors, communications, and movements [27, 8, 24, 25]. People use smartphones, mobile devices, and personal computers, leaving their digital footprints on the Internet. These human-made digital records provide a foundation for human dynamics research. Human dynamics is a new transdisciplinary research field attracting scientists and researchers from different domains, including complex systems [3], video analysis [6, 28], spatial diffusion of events [18], human mobility and network [14, 15], public health [22] and geography [13, 26]. One key research topic of human dynamics is to estimate the dynamic change of population distribution in urban areas. Although the census provides the detailed population statistics covering age, sex, and race, it does not reflect the dynamic change of population since census population is based on the location of residence. Therefore, estimating the dynamic change of population is crucial for evacuation planning, disaster management, epidemic management, event planning, and urban planning. For example, dynamic population estimation at finer scales can be useful for a stage-based evacuation planning during emergency situation[23]. Conventionally, the change of population distribution is estimated from the census survey by using data sampling and forecasting techniques. Recently, scientists have started using satellite images [5], mobile phone data [4, 8], or vehicle probe data [16] to estimate the dynamic change of population distribution at small area level. One example is to use mobile phone-based call detail records (CDR) to detect spatial and temporal differences in everyday activities among multiple cities [1]. Another example is to estimate seasonal, weekly, and daily changes in population distribution over multiple timescales with aggregated and anonymized mobile phone data [8].

In Geographic Information Systems (GIS) and cartographic research, dasymetric mapping methods have been applied to estimate population density using census data and ancillary data sources [29, 12, 17]. In the previous studies, the authors have identified that it is a challenging problem to integrate vector-based census tracks and raster-based land cover data and satellite images for dasymetric mapping. To improve the traditional problems of binary value in categorical data and areal weighting, [21] introduced an intelligent dasymetric mapping technique (IDM) with a data-driven methodology to calculate the ratio of class densities. Similar to the IDM method, this study utilizes social media data (geo-tagged data), other GIS data sources (land use and census data), and dasymetric mapping techniques to estimate the hourly change of population distribution. There are several advantages of using social media for population estimation[19]. The real-time updates of social media messages can better reflect dynamic changes of population than remote sensing imageries, which are often more expensive in cost and time to collect and process data [9]. Alternatively, mobile phone data, such as CDR, are also very expensive and inaccessible. Another drawback of CDR is that it is not possible to identify the content of communications in each phone call. In contrast, social media data are easy-to-collect, free (using public access methods), content-rich, and updated in real-time [25, 18].

In this study, we estimate hourly population distribution patterns at a high spatiotemporal resolution in urban areas using geo-tagged tweets and dasymetric mapping. The remainder of this paper follows the process as shown in Figure 1.

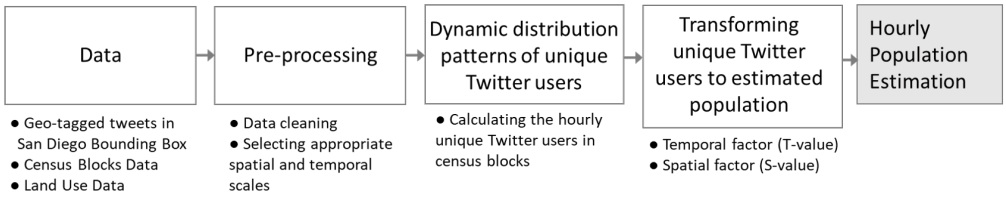


Figure 1 Overview of the process.

2 Data and pre-processing

2.1 Data collection

This study utilized public Twitter Application Programming Interfaces (APIs) to collect geo-tagged Twitter messages (tweets) through customized Python programs. The geo-tagged tweets were downloaded via the Twitter Streaming APIs and stored in a NoSQL database (MongoDB). We collected geo-tagged tweets within the bounding box of San Diego County for one year (from 2015/1/1 to 2015/12/31). There are total 7,884,806 geotagged tweets. Among the collected data, 2,601,560 (33.2%) tweets do not contain the exact coordinates and 2,355,945 (30.1%) were created outside the San Diego County. This study only utilized the remaining 2,927,301 (37.7%) geo-tagged tweets within San Diego County for population estimation. We noticed that the number of monthly geo-tagged tweets in San Diego County in 2015 fluctuated. The months of March and April 2015 have the biggest number of geo-tagged tweets. A similar trend reported by other researchers, such as Business Insider [11] suspecting that the causes might be due to Twitter's systematic updates. Figure 2 illustrates the spatial distribution of geotagged tweets from 12am to 1am in downtown, San Diego during weekdays in July 2015 (over one month).

To apply dasymetric mapping based on different types of land use, the 2017 parcel land use data was downloaded from the San Diego Association of Governments (SANDAG) website (`http://www.sandag.org`). The census blocks and their population estimates in San Diego County were obtained from the 2010 Decennial Census data.

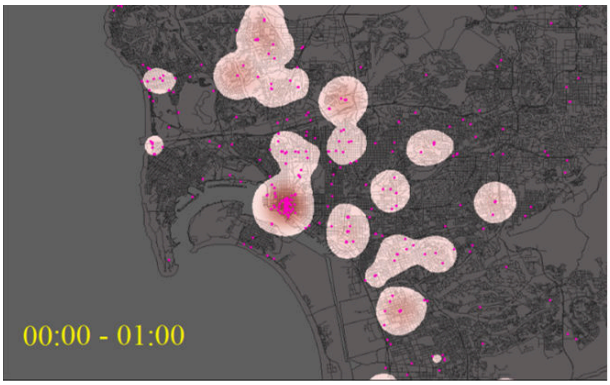


Figure 2 The distribution of geo-tagged Twitter messages (tweets as red dots) in San Diego downtown from 12am to 1am during weekdays in July of 2015 (26 days combined).

2.2 Data cleaning

Previous research has identified some major types of data noises in Twitter data, including spams, bots, and cyborgs [30, 7]. Spams and bot messages are created for reaching more users and increasing the financial gain for spammers. Since spam and bots messages can not represent the actual locations of human beings, we removed all the identifiable spams and bots based on the source field in Twitter metadata and some general bot detection rules (for example, removing tweets from TweetMyJOBS and others based on a black list of the source field). The major portion of the noise (spams and bots) in San Diego dataset includes job posting (9.07% of the total geo-tagged tweets, such as TweetMyJOBS), advertisements (1.60%, such as dlvr.it), and earthquake (1.06%) in San Diego County. The earthquake event-related tweets are geo-tagged in the localities of the earthquakes. In this study, 13.01% of geo-tagged tweets were identified as noises and removed. After removing these spams and bot posts, 2,546,385 tweets were used for calculating the unique Twitter users in each census block within one hour by filtering multiple messages posted by a single user for weekdays and weekends.

2.3 Selecting appropriate spatial and temporal scales for population estimation

For spatial units, the U.S. Census block was selected to estimate the distribution of the population. A census block is the smallest geographic unit defined by the U.S. Census Bureau for demographic analysis and therefore, it can be aggregated to census tract or other spatial units for the purpose of analysis. For example, census blocks can be aggregated to traffic analysis zones(TAZ), which is a special area formalized by local transportation officials for analyzing traffic-related data and evacuation planning. Researchers can utilize TAZ to create disaster evacuation plans and emergency response procedures. We selected one hour as our temporal resolution for estimating population density in San Diego County to meet the need for evacuation planning. In Figure 3, during weekdays, the unique Twitter user activities of posting Twitter messages decrease from midnight to 4 am. From 4 am to noon, the user activity starts to climb up. We assume that relatively a large number of Twitter activities around noon are due to tweets related to lunch time activities posted by residents and visitors. The peak of the tweeting activities comes at around 8 pm when people are getting dinner or enjoying leisure time with friends or family members. We also noticed that tweeting activities show different patterns between weekdays (Monday to Friday) and weekends (Saturday and

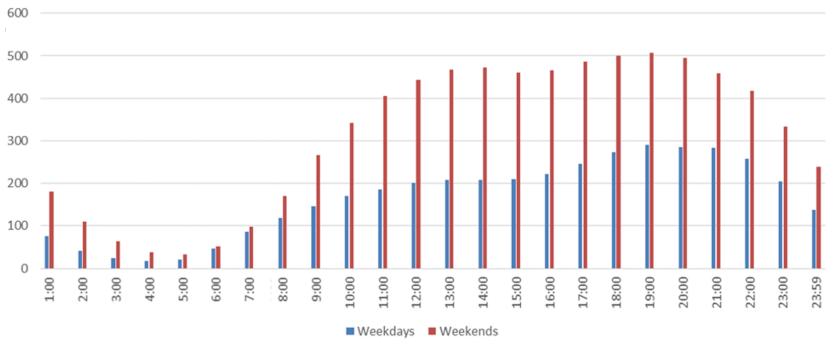


Figure 3 Comparison of hourly average numbers of unique Twitter users in San Diego County on weekdays (Monday to Friday) and weekends (Saturday to Sunday) in 2015.

Sunday). In general, the tweeting activities are more active during the weekends comparing to weekdays. Despite the similar pattern found on the weekdays where people tweeted most around 8 pm, the tweeting rate is high at around 2 pm during weekends. Therefore, we distinguish weekdays from weekends for the hourly population density estimation.

3 Methodology

3.1 Dynamic distribution patterns of unique Twitter users

3.1.1 Calculating the hourly unique Twitter users in census blocks

Within each geographical unit of census blocks, we estimate the population during a specific hourly time slot by calculating the frequency of the unique user IDs. Since one Twitter user can post several tweets within an hour from the same region (a census block), we counted one unique user ID once within an area for one hour rather than the total number of tweets. Figure 4(a) and (b) represent the distribution of unique Twitter users from 6 am – 6:59 am (a) and from 8 pm – 8:59 pm (b) respectively during weekdays in 2015 in San Diego County. The unique Twitter user density was calculated by using the total unique Twitter users within one census block during the specific hour, divided by the area of the census block. Figure 4(c) displays the 2010 population census data to visually compare its geographical distribution to that of unique Twitter users. In these maps, we selected the quantile classification method at

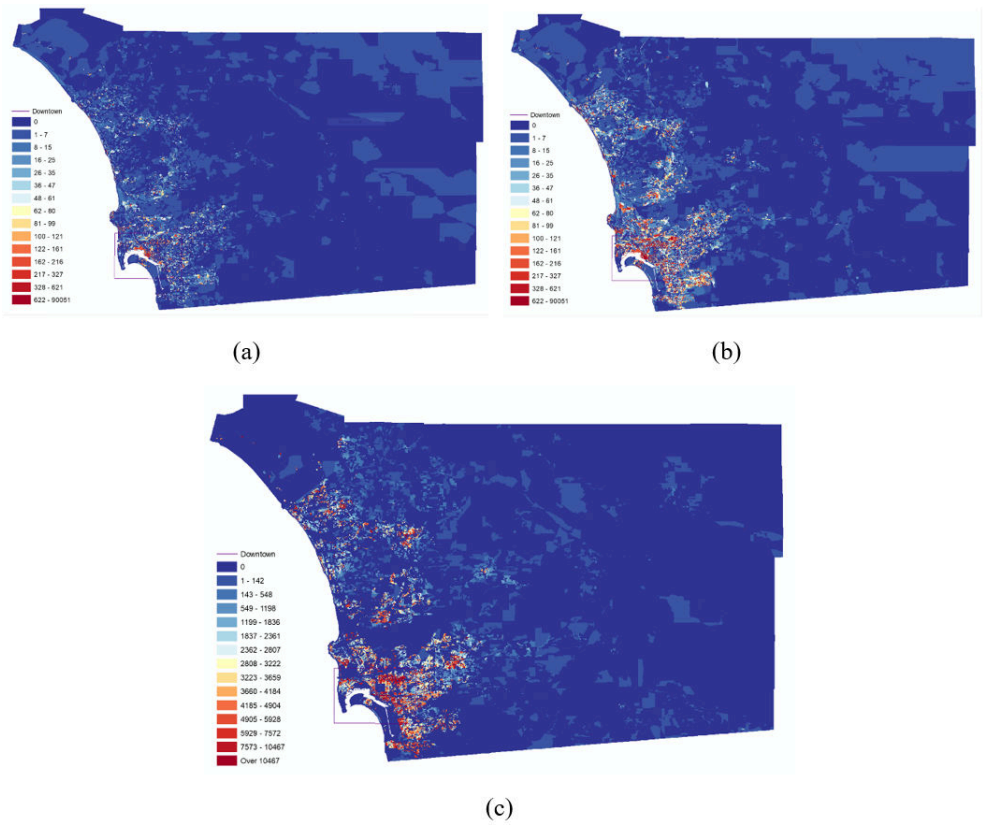


Figure 4 Spatial distribution patterns of unique Twitter users using census blocks in San Diego County from 6am – 6:59am (a) and from 8pm – 8:59pm (b) with 2015 geo-tagged tweets for weekdays. The (c) map displays the population density using 2010 census data.

8pm as the classification framework (applied to other time slots) in order to compare their spatial patterns. Figure 4(a) and (b) show an increase in unique Twitter users from 6 am to 8 pm in Western urbanized areas. The geographical distribution of unique Twitter users from 8 pm – 8:59 pm (Figure 4(b)), when has the highest average number of unique Twitter users in 2015 in San Diego County, is similar to that based on the 2010 census data (Figure 4(c)).

Maps in Figure 5 are enlarged views of Figure 4 exhibiting San Diego City downtown areas. Figure 5(a) and (b) highlight the increase of the number of unique Twitter users in areas shopping malls in Fashion Valley and Mission Valley, Balboa Park and San Diego Zoo, and the downtown Gaslamp area. The dynamic changes in these areas are reflecting the real world activities in San Diego downtown area. By comparing the 8 pm map (b) with the 2010 census block population map (c), we found that the large number of unique Twitter users in areas where there is no population in the census data. These areas are governmental and commercial lands including the (A) San Diego international airport, (B) the downtown Gaslamp quarter area, (C) Balboa Park and San Diego Zoo, (D) shopping malls in Fashion Valley and Mission Valley, and (E) Qualcomm stadium. Since the census population is considered as nighttime population estimated from residential addresses, this example shows the capability of utilizing social media data to estimate daytime population distribution at a finer spatio-temporal scale.

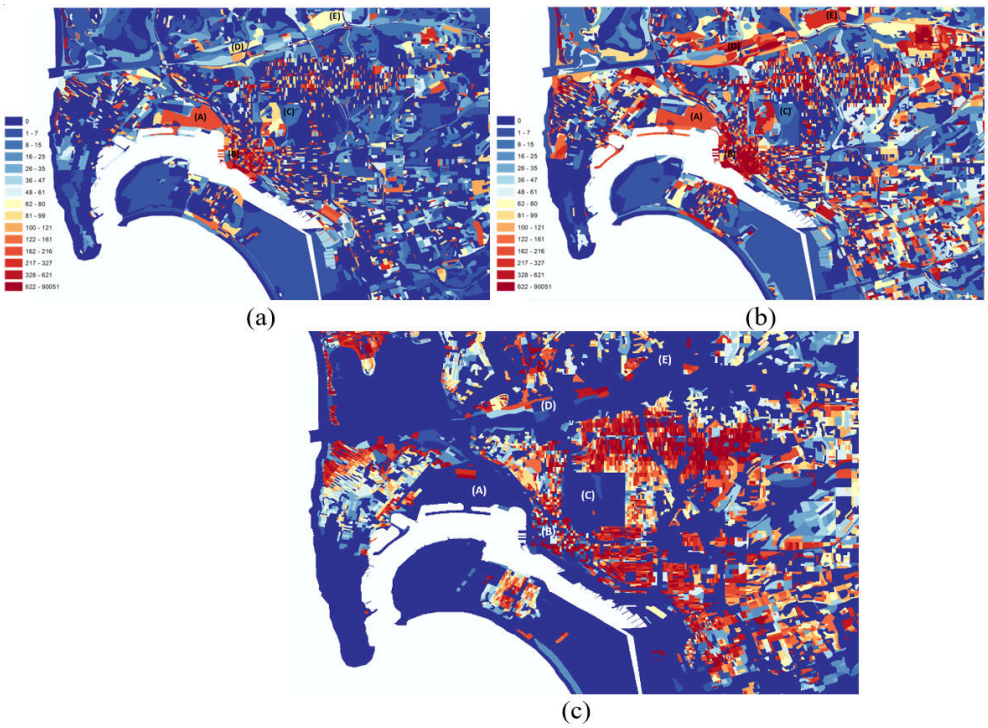


Figure 5 The spatial distribution of unique Twitter users in census blocks of San Diego downtown areas from 6 am to 6:59 am (a) and from 8 pm to 8:59 pm (b) with 2015 geo-tagged tweets for weekdays. The (c) map displays the 2010 census data in San Diego downtown areas.

3.1.2 Comparing the population change patterns of unique Twitter users between weekdays and weekends

With the hourly unique Twitter users density maps being produced (Figure 4 and Figure 5) based on weekdays and weekends, some human movement patterns can be detected and further analyzed. One of the advantages of visualizing dynamic Twitter user population patterns is that their dynamic changes can reflect the real-world situation with a high spatial resolution (census blocks) and a high temporal resolution (hourly). The following example introduces a case study in the Qualcomm Stadium with a comparison between weekdays and weekends (Figure 6). The Qualcomm Stadium is a multi-purpose stadium located in San Diego City, CA. The Qualcomm Stadium events data is archived through their official website in the events calendar. During the weekdays, the stadium usually hosts one to three events per day from 15:00 to 20:30. The events held on weekends usually started from 10:30 and ended at 17:30. The population density of unique Twitter users in Qualcomm Stadium during the weekdays shows the highest peak of Twitter user activities at 6pm. The high peaks of weekend's activities are from 1pm to 5pm. These patterns match the real-world situations since most football game events are happening between 1pm to 5pm on weekends. Figure 7 illustrates the comparison of the unique Twitter user density patterns in the Qualcomm's census blocks between weekday (a) and weekends (b) from 12pm to 12:59pm with its surrounding area. Qualcomm Stadium has a higher density of population at 12pm during weekends (comparing to weekdays).

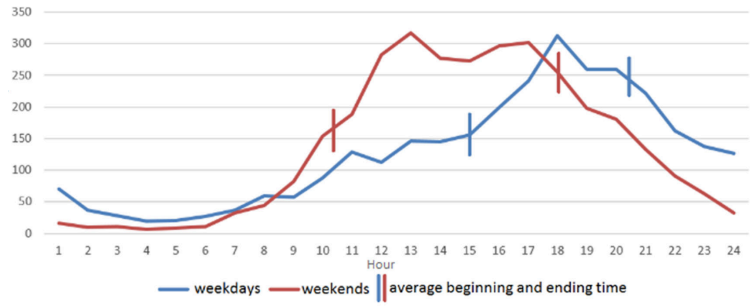


Figure 6 Comparing weekdays (blue) and weekends (red) hourly unique Twitter user density in the Qualcomm Stadium census block using 2015 geo-tagged tweets.

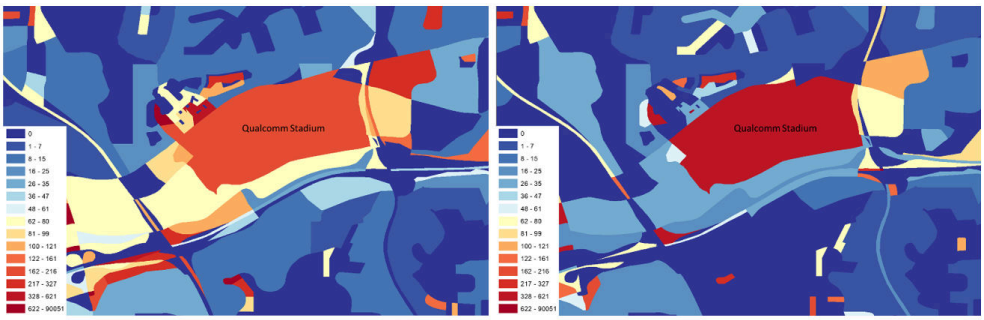


Figure 7 Hourly Unique Twitter User Density from 12pm to 12:59 pm at the Qualcomm Stadium census block for Weekdays (a) and Weekends (b) in 2015.

3.1.3 Comparing unique Twitter population with census data

Comparing the weekdays and weekends unique Twitter user density map in Census Block polygon with census population can reveal the fact whether Twitter population can be used to represent the human mobility and real human population during different period of time in a day. The Census population represents the population distribution during the nighttime since it collects the number of people living in their household.

The Table 1(a) presents the $Z_{hx \cap pop}$ values in San Diego County area which compares the similarity of census block with unique Twitter user in different time slot from H1 to H24. Each Z value represents for the sum of absolute difference (SAD value) of two sets of data within range 0 to 1 based on formula 1.

$$Z_{hx \cap pop} = \sum \left| \frac{P_{A \cap hx}}{P_{hx_{\max}}} - \frac{P_{A \cap pop}}{P_{pop_{\max}}} \right| \tag{1}$$

Where:
$Z_{hx \cap pop}$ = the sum of the absolute difference of number of population between time slot hx and census population pop;
$P_{A \cap hx}$ = the value of unique Twitter population in time slot hx in Polygon P_A;
$P_{hx_{\max}}$ = the maximum value of unique Twitter population in time slot hx.

Note that *sd* refers to San Diego, *cb* refers to census block polygon, *wd* refers weekdays, and *we* refers to weekends. Thus, the intersection between H1 (0:00 to 0:59) and $Z_{sd\_cb\_wd}$ stands for the SAD Value of comparing the unique Twitter user density map with census block population density in the scale of San Diego County during weekdays. Based on the results showed in the table for census block polygon, the H5 (4:00 to 4:49) in weekdays and H6 (5:00 to 5:59) in weekends are the two time slot where the unique Twitter user is the closest to the census block population. The census block population records the number of human population in the residential area in detail. Meanwhile, 4:00 to 5:59 is usually the time when people get up during the morning time. Thus, it is possible to reflect the human residential area by using Twitter data.

Table 1(b) presents the $Z_{hx \cap pop}$ values in San Diego downtown area by comparing the census block population with unique Twitter user in downtown area, San Diego. Note that *dt* refers to downtown area of San Diego, H5 (4:00 to 4:49) for both weekdays and weekends is the time slot where the unique Twitter user is the closest to the census block population. On the other side, from the perspective of dissimilarity, H24 (23:00 to 23:49) and H1 (0:00 to 0:59) have the most dissimilar unique Twitter user distribution comparing to the census block population.

3.2 Transforming unique Twitter users to estimated population with spatial and temporal variation factors

The previous sections illustrate how to calculate the dynamic changes of unique Twitter users in high spatial and temporal resolution units. The next step is to create a dynamic population model to transform the numbers of unique Twitter users into estimated population. We proposed a simplified population estimation model using census blocks, land use data, and dasymetric mapping methods like the following:

$$\widehat{D}_{hx \cap a} = UserNumber_{hx \cap A} * (T_{hx}) * (S_{hx \cap A}) \tag{2}$$

Table 1 The sum of absolute difference between the number of hourly unique twitter data(from 0:00 to 23:59) with census block population during weekdays and weekends in (a) San Diego County and (b) San Diego Downtown.

		(a) San Diego County		(b) San Diego Downtown	
Time Slot	Description	Weekdays $Z_{sd\_cb\_wd}$	Weekends $Z_{sd\_cb\_we}$	Weekdays $Z_{dt\_cb\_wd}$	Weekends $Z_{dt\_cb\_we}$
H1	00:00 to 00:59	402.1	412.3	131.3	120.0
H2	01:00 to 01:59	399.5	403.9	126.4	116.0
H3	02:00 to 02:59	430.7	408.9)	121.3	1116.1
H4	03:00 to 03:59	377.6	402.6	109.0	113.1
H5	04:00 to 04:59	366.7	367.9	97.5	98.3
H6	05:00 to 05:59	367.9	367.0	97.8	98.6
H7	06:00 to 06:59	381.1	377.4	101.9	102.4
H8	07:00 to 07:59	387.4	386.5	104.4	106.4
H9	08:00 to 08:59	391.7	381.8	106.5	105.3
H10	09:00 to 09:59	390.8	388.8	106.5	108.0
H11	10:00 to 10:59	391.6	388.9	107.2	108.3
H12	11:00 to 11:59	391.7	397.5	107.3	111.6
H13	12:00 to 12:59	393.1	396.1	108.2	111.0
H14	13:00 to 13:59	394.2	399.9	108.5	112.9
H15	14:00 to 14:59	392.1	398.4	108.0	112.1
H16	15:00 to 15:59	392.1	396.3	108.0	111.3
H17	16:00 to 16:59	398.4	398.1	110.9	112.4
H18	17:00 to 17:59	411.2	392.7	116.7	110.3
H19	18:00 to 18:59	387.9	396.4	107.2	111.6
H20	19:00 to 19:59	390.7	392.3	108.0	110.1
H21	20:00 to 20:59	405.4	394.6	114.1	110.6
H22	21:00 to 21:59	428.3	397.5	123.4	111.8
H23	22:00 to 22:59	441.2	392.1	129.2	110.0
H24	23:00 to 23:59	428.0	409.3	132.8	118.6

3.2.1 Temporal variation factor (t-value)

The temporal variation factor (T-value) is defined as a value of factor multiples with the frequency number of hourly average Twitter user in each census block or land use polygon. A temporal factor was based on hourly frequency changes of unique Twitter users within the County of San Diego. Figure 8 illustrated the creation of temporal variation factor (T-value). First of all, we calculate the total number of unique Twitter users in the whole San Diego County at each hour (from 0am, 1am, 2am ...). Then we select the highest number (at 18:00-18:59 or H19, 75690) as the base number (T-value = 1). Each T-value is calculated using the base number (75690) divided by the total unique Twitter user numbers in each time slot. For example, the T-value at 4am will be 75690 / 5481 = 13.81.

Figure 9 shows the original unique Twitter user density map (a) and the estimated population density map (b) with temporal variation factor (T-value = 3.82) from 0:00 to 0:59 in San Diego downtown for Weekdays in 2015. As Figure 9(b) shows, estimated population with temporal variation factor at H1(0:00-0:59) is the result of the population in every census block increased by T-value times. Given that people less likely tweet during nighttime, temporal variation factor tends to be exaggerated during those hours.

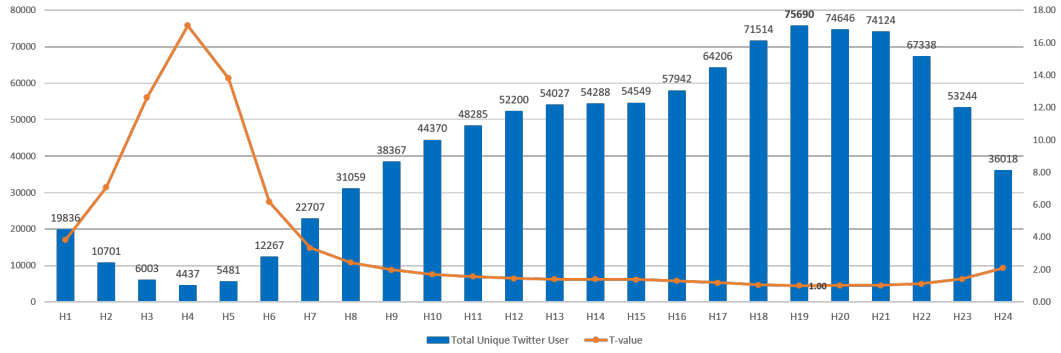


Figure 8 The total unique Twitter user numbers in each time slot and their T-values.

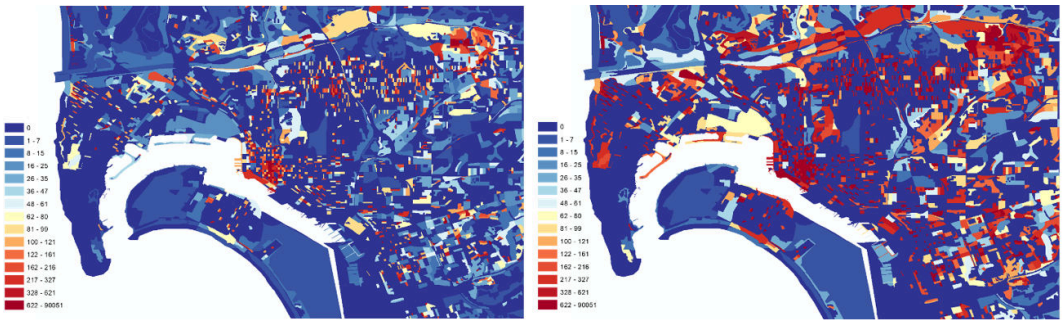


Figure 9 The original unique Twitter user density map (a) and the population density estimation (b) with temporal variation factor (T-value = 3.82) from 0:00 to 0:59 in San Diego County during weekdays in 2015.

3.2.2 Spatial change factor using dasymetric mapping method (s-value)

We utilized dasymetric mapping technique to redistribute the unique Twitter user population based on the ratio of average census population and the average hourly unique Twitter user population in each type of land use categories. Various human activities happen at a certain time in a certain land use type. For example, people would shop at shopping malls during its open hours, meaning the population in commercial land use type during daytime. Therefore, the goal is to refine the population density maps by taking different types of land use data (residential areas, commercial areas, etc.) and census data into consideration.

The census block boundaries (43,326 polygons in San Diego County) were overlaid with the 2016 parcel land use data (189,635 polygons) which created a union map with 740,843 polygons. The parcel land use data contains 10 types of land use which include unzoned, single- family, minor multiple, restricted multiple, multiple residential, restricted commercial, commercial, industrial, agricultural, and special. We downgraded the 10 types of land cover into 6 categories which are unzoned, residential, commercial, industrial, agricultural, and special. The road section were added into the parcel shapefile by extracting the road polygons from SANDAG's land use shapefile which shares the same dimension with parcel data. The new land use map ended up with 7 types of land use in total (see Table 2). Both census population and unique Twitter user population are re-distributed from the larger census

block polygon to the finer polygons (subareas) in the overlaid map. The following formula (3) were applied to calculate the number of census population with certain land use type (a) as:

$$\widehat{SCP}_a = CP_{\mathrm{A}} \left(\frac{SA_{\mathrm{A}(a)}}{A_{\mathrm{A}}} \right) \tag{3}$$

Where:
$\widehat{SCP}_a$= the estimated count of census population in subarea of land use a;
CP_{A} = the count of census population in census block A;
$SA_{\mathrm{A}(a)}$ = the area of subarea a under census block A;
A_{A} = the area of census block A;
a = the land use type;
A = census block ID.

The method of calculating unique Twitter population (formula 3) is similar to the way of re-distributing census population, while adding the temporal variation variable (T-value) into consideration. The count of unique Twitter population in census block A during time slot hx, $TP_{hx \cap A}$ is acquired by multiplying average unique Twitter user with T-Value as:

$$TP_{hx \cap A} = tp_{hx \cap A} \left(T_{hx} \right) \tag{4}$$

Where:
$tp_{hx \cap A}$ = the count of original Twitter population in census block A during time slot hx;
T_{hx} = T-Value for certain time slot hx.

The estimated count of unique Twitter population in each subarea is then calculated based on the ratio of the size of subarea and area of census block A.

$$\widehat{STP}_{hx \cap a} = TP_{hx \cap A} \left(\frac{SA_{\mathrm{A}(a)}}{A_{\mathrm{A}}} \right) \tag{5}$$

Where:
$\widehat{STP}_{hx \cap a}$ = the estimated count of unique Twitter population during time slot hx in subarea of land use A;
$TP_{hx \cap A}$ = the count of unique Twitter population in census block A during time slot hx.

The estimated population density $\widehat{D}_{hx \cap a}$ aims to estimate the hourly human population based on the ratio of the sum of census population in land use Type a and the sum of hourly unique Twitter user population in land use Type a. The ratio (R_{A}) is defined as:

$$R_{\mathrm{A}} = \frac{\sum \widehat{SCP}_a}{\sum \widehat{STP}_{hx \cap a}} \tag{6}$$

$$\widehat{D}_{hx \cap a} = R_{\mathrm{A}} \left(\frac{\widehat{STP}_{hx \cap a}}{SA_{\mathrm{A}(a)}} \right) \tag{7}$$

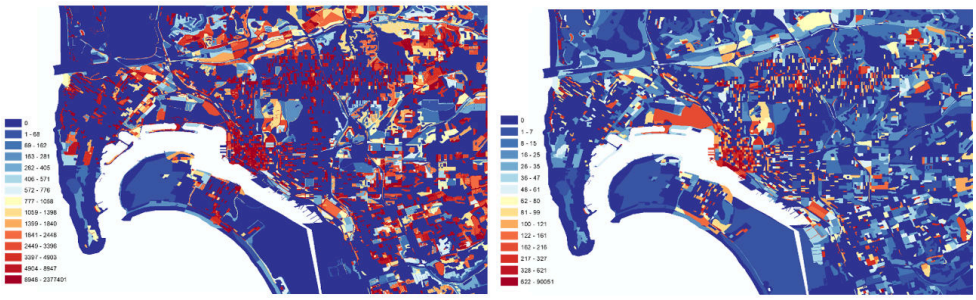
While the estimated population density $\widehat{D}_{hx \cap a}$ for certain land use type is the estimated count of unique Twitter population with R_A and divided by the size of the corresponding subarea as formula (7). Table 2(a) shows the area of 7 land use types in square kilometer, the total number of estimated unique Twitter population during H7 (6:00 to 6:59) and H21 (20:00 to 20:59) after applying T-value, and the estimated census population based on different types of land use. Table 2(b) shows the ratio (R_{A}) which was calculated based on the division of cenpop ($\sum \widehat{SCP}_a$) with twepop_h7 ($\sum \widehat{STP}_{h7 \cap a}$)) and twepop_h21 ($\sum \widehat{STP}_{h21 \cap a}$).

Table 2 (a) The area of seven types of land use, the total number of estimated unique Twitter user population during 6:00 to 6:59 (twepop_h7) and 20:00 to 20:59 (twepop_h21), and the total number of estimated census population (cenpop) based on land use. (b) The Ratio for estimating the h7 (6:00 to 6:59) and h21 (20:00 to 20:59) real population and its corresponding land use type.

		(a)				(b)	
LC	Land Use	Area(km^2)	twepop_h7	twepop_h21	cenpop	ratio_h7	ratio_h21
0	Unzoned	6437.72	65.61	46.37	99553.74	1517.25	2147.05
1	Residential	1626.62	96.56	109.85	1128499.76	11686.73	10272.84
2	Commercial	394.29	42.07	49.32	112381.49	2671.32	2278.48
3	Industrial	322.65	21.50	18.15	24372.34	1133.52	1342.97
4	Agricultural	1704.09	2.97	2.73	15602.81	5252.44	5708.23
5	Special	291.88	8.43	7.13	24342.04	2889.01	3413.78
6	Road	285.68	52.55	55.82	392448.08	7467.56	7030.45

4 Results

Figure 10 and Figure 11 show the preliminary result of applying dasymetric mapping equations (4) and (5) to adjust and re-distribute hourly unique Twitter user population into estimated population density. The purpose of the comparison between maps is not to examine the difference in numbers in each census. Instead, the focus is to visually compare the relative distribution of areas with high and low frequency between the two maps.



(a) (b)

Figure 10 (a) Population density estimation with spatial variation factor and the dasymetric mapping method from 6:00 to 6:59 in San Diego downtown areas during Weekdays in 2015; (b) the original hourly unique Twitter user density from 6:00 to 6:59 in San Diego downtown areas during Weekdays in 2015.

Table 2(b) shows that the value of residential area is higher than the values of the rest 6 types of land use types due to the influence brought by census block data. Therefore, when the estimated population is calculated by reflecting temporal variation factors and spatial change factor, more population can be redistributed to the residential area. Based on the side by side comparison of estimated population density and the original unique Twitter user population, the estimated population is transformed by the landuse types and more population is redistributed on the residential area than the rest 6 types of land use due to the influence brought by overlaying landuse with census data. Since census data represents the count of population at home, the dasymetric mapping methods could improve the estimations using Twitter density maps and to adjust the shortage of the people who may not tweet much when they are sleeping or at home. Figure 10(a) shows more population in residential area instead of the original situation where downtown areas have higher density population.

In Figure 11, the maps show the comparison of the estimated population density map (a) and the original unique Twitter user population density map (b) from 20:00 to 20:59 during weekdays in San Diego downtown areas, with the 2010 population density based on 2010 census data (c). The result shows that dasymetric mapping technique could provide a balanced population estimation comparing to the hourly unique Twitter user density and the census (night-time only) population. Comparing to the Twitter density map in the same time slot (from 20:00 to 20:59), high population density areas, such as Balboa Park and San Diego Zoo, shopping malls, and San Diego International Airport, are better estimated by using dasymetric maps with Twitter user population density data and landuse data.

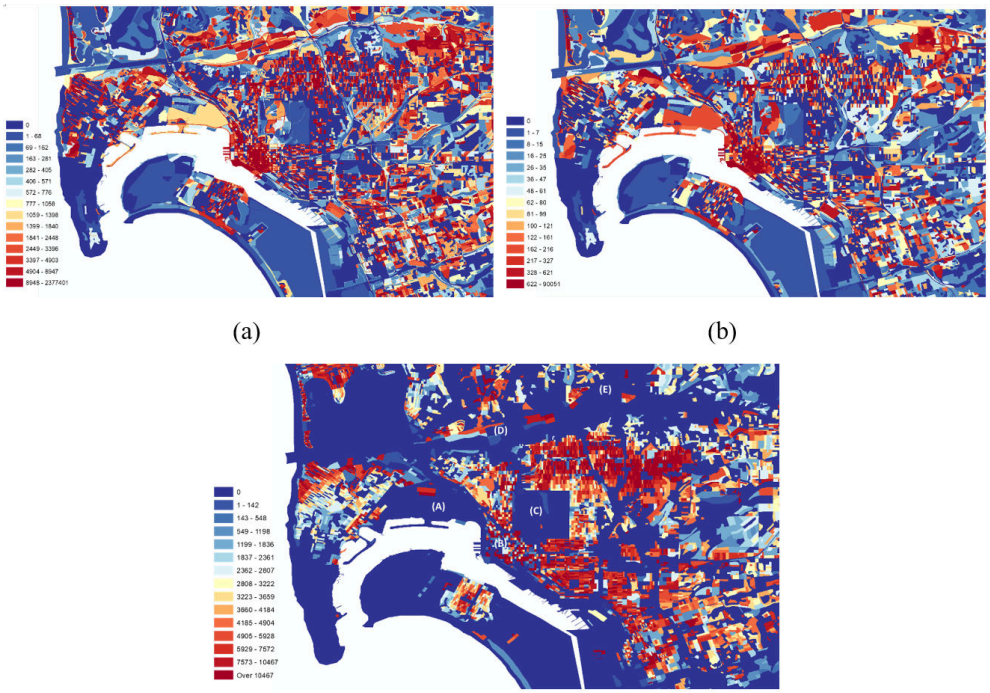


Figure 11 (a) Population density estimation map with the spatial variation factor and dasymetric mapping method from 20:00 to 20:59 in San Diego downtown areas during Weekdays in 2015.; (b) the original hourly unique Twitter user density map (middle) from 20:00 to 20:59 in San Diego downtown areas during Weekdays in 2015; (c) the 2010 census block population density map using census data.

5 Limitations and future study

There are several research limitations in our study as the following:

(a) Geo-tagged Twitter users can not represent the total population. In general, social media users are younger comparing to the general population, and more users live in urban areas than rural areas [10].

(b) It is very difficult to validate our dynamic population model because there is no similar data existed in San Diego County. We can only estimate the night time population to compare to the actual 2010 census data. However, these data are not created originally for displaying the dynamic hourly population density and may not be suitable for the validation purpose.

(c) Spatial and temporal factors in population estimation are usually correlated and should be considered together [2]. Our simplified model does not consider the autocorrelation between the spatial and temporal factors.

(d) This study only utilizes one single social media data (Twitter) among many kinds of them. For sustainability, we should consider combining other social media, such as Instagram, Facebook check-in, Foursquare, and other possible digital footprints to enhance our population model. However, different types of social media platforms and digital footprints may have different types of spatiotemporal patterns, which will be another challenge research question.

(e) The public Streaming APIs provided by Twitter is not very stable. We found that unequal number of tweets collect in different months and days, which may create some biases in our estimation of population density. For example, the Twitter use activities during March and April may more influence to the final population estimation result.

To improve and refine our future study of population density models, we are planning to use more complicated dasymetric mapping methods similar to intelligent dasymetric mapping technique (IDM)[21] to calculate the probability of population distribution in a more detailed land use category and census blocks using other spatial statistic methods, such as Weighted Linear Combination (WLC). We recognized that validation is a key challenge to evaluate our dynamic population estimation model. While collecting dynamic population from real world in a large area is extremely difficult, it might be possible to partially compare the estimate during a certain temporal duration with existing data. For example, Census American Community Survey (ACS) provides a daytime population estimate [20]. Therefore, we can measure the goodness of fit between the estimates from the model and ACS during daytime (e.g., 9 am to 3 pm, a core work hour). However, it is necessary to carefully consider the validation process since social media data are drawn from potentially biased population and the data may include not only local residents but also visitors whereas ACS data account for residents and workers. Taking visitors in San Diego into consideration can be helpful for revealing the real pattern of human dynamic. Therefore, further social media data filtering procedures should be applied to identify local residents for validation. While data at finer spatial and temporal scales can provide better understanding of human movement, it can raise privacy concerns. Population estimation needs to find balance between privacy and accuracy. Within the context of social media studies, fine scale results are not the most appropriate because they can reveal users' location. Results should be aggregated to the point at which they show significance without jeopardizing users' privacy. Therefore, researchers should ask how fine does the data need to be to protect users' privacy while also providing meaningful results. Methods such as data anonymization and using aggregated data to mitigate privacy risks can be considered.

The finalized framework, with frontend web design and backend database, can be applied with real-time data as well in the future by upgrading the current 1 hour temporal resolution to 10 minutes or even higher scale. To summarize, although the Twitter data cannot perfectly represent the entire population, this study has revealed the potential research framework using social media data and dasymetric maps to calculate the dynamic change of population distribution patterns. Our proposed methods can provide a better estimation of hourly population patterns in airports, sports stadiums, shopping malls, downtown areas, parks and other tourist locations comparing to traditional census data or ACS data.

The combination of multiple social media data, mobile phone records, and other digital footprints created by human beings will be a great source to study human dynamics and help us to understand different types of human behaviors, movements, and activities in high spatial

and temporal resolution. This integration of utilizing multiple sources of information would be able to increase the demographic comprehensiveness of this research. This information can facilitate the improvement of our transportation systems, emergency evacuation procedures, and urban planning in the future.

References

1 Rein Ahas, Anto Aasa, Y Yuan, Martin Raubal, Zbigniew Smoreda, Yu Liu, Cezary Ziemlicki, Margus Tiru, and Matthew Zook. Everyday space–time geographies: using mobile phone-based sensor data to monitor urban activity in harbin, paris, and tallinn. *International Journal of Geographical Information Science*, 29(11):2017–2039, 2015.

2 Li An, Ming-Hsiang Tsou, Stephen ES Crook, Yongwan Chun, Brian Spitzberg, J Mark Gawron, and Dipak K Gupta. Space–time analysis: Concepts, quantitative methods, and future directions. *Annals of the Association of American Geographers*, 105(5):891–914, 2015.

3 Albert-Laszlo Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211, 2005.

4 Linus Bengtsson, Xin Lu, Anna Thorson, Richard Garfield, and Johan Von Schreeb. Improved response to disasters and outbreaks by tracking population movements with mobile phone network data: a post-earthquake geospatial study in haiti. *PLoS medicine*, 8(8), 2011.

5 Budhendra Bhaduri, Edward Bright, Phillip Coleman, and Marie L Urban. Landscan usa: a high-resolution geospatial and temporal modeling approach for population distribution and dynamics. *GeoJournal*, 69(1-2):103–117, 2007.

6 Christoph Bregler. Learning and recognizing human dynamics in video sequences. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 568–574. IEEE, 1997.

7 Zi Chu, Indra Widjaja, and Haining Wang. Detecting social spam campaigns on twitter. In *International Conference on Applied Cryptography and Network Security*, pages 455–472. Springer, 2012.

8 Pierre Deville, Catherine Linard, Samuel Martin, Marius Gilbert, Forrest R Stevens, Andrea E Gaughan, Vincent D Blondel, and Andrew J Tatem. Dynamic population mapping using mobile phone data. *Proceedings of the National Academy of Sciences*, 111(45):15888–15893, 2014.

9 Pinliang Dong, Sathya Ramesh, and Anjeev Nepali. Evaluation of small-area population estimation using lidar, landsat tm and parcel data. *International Journal of Remote Sensing*, 31(21):5571–5586, 2010.

10 Maeve Duggan, Nicole B Ellison, Cliff Lampe, Amanda Lenhart, and Mary Madden. Social media update 2014. *Pew research center*, 19, 2015.

11 Jim Edwards. Leaked twitter api data shows the number of tweets is in serious decline. *Business Insider*, February 2016. URL: `http://www.businessinsider.com/tweets-on-twitter-is-in-serious-decline-2016-2`.

12 Cory L Eicher and Cynthia A Brewer. Dasymetric mapping and areal interpolation: Implementation and evaluation. *Cartography and Geographic Information Science*, 28(2):125–138, 2001.

13 Su Yeon Han, Ming-Hsiang Tsou, and Keith C Clarke. Do global cities enable global views? using twitter to quantify the level of geographical awareness of us cities. *PloS one*, 10(7), 2015.

14 Su Yeon Han, Ming-Hsiang Tsou, and Keith C Clarke. Revisiting the death of geography in the era of big data: the friction of distance in cyberspace and real space. *International Journal of Digital Earth*, 11(5):451–469, 2018.

15 Su Yeon Han, Ming-Hsiang Tsou, Elijah Knaap, Sergio Rey, and Guofeng Cao. How do cities flow in an emergency? tracing human mobility patterns during a natural disaster with big data and geospatial data science. *Urban Science*, 3(2):51, 2019.

16 Yusuke Hara and Masao Kuwahara. Traffic monitoring immediately after a major natural disaster as revealed by probe data–a case in ishinomaki after the great east japan earthquake. *Transportation research part A: policy and practice*, 75:1–15, 2015.

17 James B Holt, CP Lo, and Thomas W Hodler. Dasymetric estimation of population density and areal interpolation of census data. *Cartography and Geographic Information Science*, 31(2):103–121, 2004.

18 Elias Issa, Ming-Hsiang Tsou, Atsushi Nara, and Brian Spitzberg. Understanding the spatio-temporal characteristics of twitter data with geotagged and non-geotagged content: two case studies with the topic of flu and ted (movie). *Annals of GIS*, 23(3):219–235, 2017.

19 Bin Jiang, Ding Ma, Junjun Yin, and Mats Sandberg. Spatial distribution of city tweets and their densities. *Geographical Analysis*, 48(3):337–351, 2016.

20 Brian McKenzie, William Koerber, Alison Fields, Megan Benetsky, and Melanie Rapino. Commuter-adjusted population estimates: Acs 2006-10. *Washington, DC: Journey to Work and Migration Statistics Branch, US Census Bureau*, 2010.

21 Jeremy Mennis and Torrin Hultgren. Intelligent dasymetric mapping and its application to areal interpolation. *Cartography and Geographic Information Science*, 33(3):179–194, 2006.

22 Anna C Nagel, Ming-Hsiang Tsou, Brian H Spitzberg, Li An, J Mark Gawron, Dipak K Gupta, Jiue-An Yang, Su Han, K Michael Peddecord, Suzanne Lindsay, et al. The complex relationship of realspace events and messages in cyberspace: case study of influenza and pertussis using tweets. *Journal of medical Internet research*, 15(10):e237, 2013.

23 Atsushi Nara, Xianfeng Yang, Sahar Ghanipoor Machiani, and Ming-Hsiang Tsou. An integrated evacuation decision support system framework with social perception analysis and dynamic population estimation. *International journal of disaster risk reduction*, 25:190–201, 2017.

24 Tao Pei, Stanislav Sobolevsky, Carlo Ratti, Shih-Lung Shaw, Ting Li, and Chenghu Zhou. A new insight into land use classification based on aggregated mobile phone data. *International Journal of Geographical Information Science*, 28(9):1988–2007, 2014.

25 Ming-Hsiang Tsou. Research challenges and opportunities in mapping social media and big data. *Cartography and Geographic Information Science*, 42(sup1):70–74, 2015.

26 Ming-Hsiang Tsou, Ick-Hoi Kim, Sarah Wandersee, Daniel Lusher, Li An, Brian Spitzberg, Dipak Gupta, Jean Mark Gawron, Jennifer Smith, Jiue-An Yang, et al. Mapping ideas from cyberspace to realspace: visualizing the spatial context of keywords from web page search results. *International Journal of Digital Earth*, 7(4):316–335, 2014.

27 Ming-Hsiang Tsou and Michael Leitner. Visualization of social media: seeing a mirage or a message?, 2013.

28 Jessica JunLin Wang and Sameer Singh. Video analysis of human dynamics—a survey. *Real-time imaging*, 9(5):321–346, 2003.

29 John K Wright. A method of mapping densities of population: With cape cod as an example. *Geographical Review*, 26(1):103–110, 1936.

30 Sarita Yardi, Daniel Romero, Grant Schoenebeck, et al. Detecting spam in a twitter network. *First Monday*, 15(1), 2010.

Multiple Resource Network Voronoi Diagram

Ahmad Qutbuddin
Department of Computer and Electrical Engineering and Computer Science,
Florida Atlantic University, Boca Raton, FL, USA
aqutbuddin2017@fau.edu

KwangSoo Yang
Department of Computer and Electrical Engineering and Computer Science,
Florida Atlantic University, Boca Raton, FL, USA
http://faculty.eng.fau.edu/yangk/home/
yangk@fau.edu

Abstract

Given a spatial network and a set of service center nodes from k different resource types, a Multiple Resource-Network Voronoi Diagram (MRNVD) partitions the spatial network into a set of Service Areas that can minimize the total cycle distances of graph-nodes to allotted k service center nodes with different resource types. The MRNVD problem is important for critical societal applications such as assigning essential survival supplies (e.g., food, water, gas, and medical assistance) to residents impacted by man-made or natural disasters. The MRNVD problem is NP-hard; it is computationally challenging due to the large size of the transportation network. Previous work is limited to a single or two different types of service centers, but cannot be generalized to deal with k different resource types. We propose a novel approach for MRNVD that can efficiently identify the best routes to obtain the k different resources. Experiments and a case study using real-world datasets demonstrate that the proposed approach creates MRNVD and significantly reduces the computational cost.

2012 ACM Subject Classification Information systems → Geographic information systems

Keywords and phrases Network Voronoi Diagram, Resource Allocation, Route Optimization

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.11

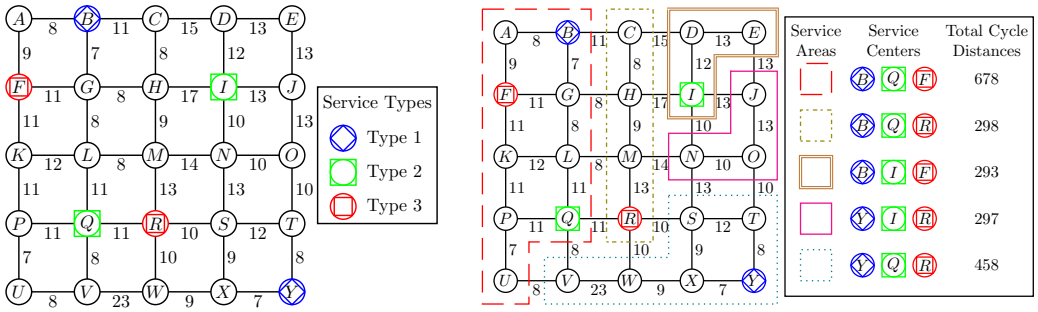
Supplementary Material The source code is available on our research group website `http://faculty.eng.fau.edu/yangk/home/NSF_Career_Projects.html` [3].

Funding This work is supported by the National Science Foundation CAREER under Grant No. 1844565.

Acknowledgements We would like to thank the US National Science Foundation under Grant No. 1844565. We are particularly thankful to GIScience reviewers for their helpful comments.

1 Introduction

Given a spatial network and a set of service center nodes from k different resource types (e.g. gas stations, grocery stores, shelters, hospitals, etc), a Multiple Resource-Network Voronoi Diagram (MRNVD) partitions the spatial graph into a set of Service Areas (SA) that can minimize the total cycle distances of graph-nodes to allotted k service center nodes with different resource types. Figure 1a shows an example input of MRNVD consisting of a graph with 25 graph-nodes ($i.e., A, B, \ldots, Y$) and service center nodes with three types (i.e., $Type1(B, Y)$, $Type2(I, Q)$, and $Type3(F, R)$). Figure 1b shows an example output of MRNVD where the graph is partitioned such that every graph-node is allotted to three service centers with different types. The objective is to minimize the total cycle distances of graph-nodes to allotted k service center nodes with different types. The MRNVD problem is NP-hard (a proof is provided in Section 2.1). Intuitively, the problem is computationally challenging because of the large size of the transportation network.



(a) Input with three types of service centers. **(b)** Output (Polygons show Service Areas).

Figure 1 Example of Input and Output of MRNVD (Best in Colors).

1.1 Application Domain

The MRNVD problem is important for critical societal applications such as assigning essential resources (e.g., food, water, gas, and medical assistance) to residents impacted by man-made or natural disasters. The objective of MRNVD is to minimize the total cycle distances such that residents can quickly visit their allotted service centers and back to their original location. MRNVD can help us to identify the most efficient route to visit all required service centers. In addition, the simple format of information is vital to communicate effectively during an emergency. MRNVD provides compact and simple representation of Service Areas (SA) that can mitigate panic and chaos and allow for efficient delivery of critical information to the public. Examples of such situations are provided in Table 1.

Table 1 Applications of MRNVD.

Applications	Benefit of MRNVD Service Areas
Emergency Resource Allocation	Develop an emergency plan to help citizens to minimize their travel times to obtain all required resources.
Store Choices	Provide an efficient route to save time and gas while shopping.
Tourist Site Selection	Recommend a tourist route that can visit attractions with different types.

2 Problem Definition

In our formulation of the MRNVD problem, a transportation network is represented and analyzed as an undirected graph composed of nodes and edges. Every node represents a spatial location in geographic space (e.g., road intersections), which can be used as a proxy for locations of residents. Every edge between two nodes represents a road segment and has a travel distance. Every service center has a resource type (e.g., water, food, gas, medicine, etc.). The $MRNVD(N, E, S, D)$ problem is defined as follows:

Intput: A transportation network G with

- a set of graph-nodes N and a set of edges E,
- a set of service center locations with k different resource types $S \subset N$, and
- a set of nonnegative real distances of edges $D : E \rightarrow R_0^+$

Output: A Multiple Resource Network Voronoi Diagram (MRNVD)

Objective:

- Min-sum: Minimize the total cycle distances of graph-nodes to their allotted k service center nodes with different types of resources.

Constraints:

- Service Area (SA) allotment must be k service center nodes with different types of resources.

▶ **Definition 1 (Cycle Distance).** *Given a starting point and a set of k different service centers, the cycle distance is the distance of the shortest route that visits k service centers and returns to the starting point.*

2.1 Problem Hardness

The NP-hardness of MRNVD follows from a well-known result about the NP-hardness of the traveling salesman problem.

▶ **Theorem 1.** *The MRNVD problem is NP-hard.*

Proof. The NP-hardness of MRNVD can be proved by reduction from a well known NP-hardness problem, the traveling salesman problem (TSP) [19]. Given a starting point o and a set of service centers S, TSP finds the shortest cycle distance of o. Let $A = (o, S)$ be an instance of TSP, where o is the starting point and S is a set of service centers. Let $B = (O, S)$ be an instance of the MRNVD problem, where O is a set of staring points and S is a set of service centers. Assume that every service center has a different type. Let $O = \{o\}$. Then the instance of TSP is a special case of MRNVD, where O is a set with a single element (i.e., o). Since A is constructed from B in polynomial-bounded time, the proof is complete. ◀

2.2 Our Contribution

In this paper, we propose a novel algorithm for creating MRNVD based on two Distance bounded Pruning (DP) methods. Our approach has three key components: 1) Straight-Distance bounded Pruning (SDP), 2), Triangular-Distance bounded Pruning (TDP) and 3) 2-opt cycle route computation. In addition, we design a baseline algorithm to evaluate the performance of the proposed approach. Specifically, our contribution is as follows:

- We introduce a new Network Voronoi Diagram, namely Multiple Resource Network Voronoi Diagram (MRNVD).
- We prove that the MRNVD problem is NP-hard.
- We design a baseline algorithm that can produce the optimal solution of MRNVD.
- We propose the Distance bounded Pruning (DP) algorithm based on three key ideas: 1) Straight-Distance bounded Pruning (SDP), 2), Triangular-Distance bounded Pruning (TDP) and 3) 2-opt cycle route computation.
- Our experimental results and a case study using real-world datasets demonstrate that our proposed algorithm outperforms the baseline algorithm and significantly reduces the computational cost to create a MRNVD.

2.3 Related Work

Network Voronoi Diagram (NVD) is extensively used to identify the nearest service center [7, 16, 15, 22]. However, the application of NVD is limited to a single type of resource [17]. Consider the example of the resident who is looking for gas, water, and medicine at the same time. NVD cannot minimize the travel time to visit three service centers for each resource. Recently two-site network Voronoi diagrams were proposed to identify the best route for two different resources [5, 6]. The general idea is to find the minimum triangle-perimeter to

partition the spatial network to a set of Service Areas. However, two-site network Voronoi diagrams cannot be generalized into MRNVD due to the hardness of the cycle distance computation. The Voronoi based k nearest neighbor search for spatial network databases was proposed to identify k different nearest service centers [13]. However, the Voronoi k Nearest Neighbor cannot produce the minimum cycle distance because it considers only the distance of the graph-node to service center nodes. There are slightly different approaches for partitioning urban areas into functional or service regions. The multiplicatively weighted order-k Minkowski-metric Voronoi diagrams were utilized to develop a map-based emergency support system [14]. The partitioning method based on street intersections and barriers was developed to support mobility infrastructure planning and optimization in an urban environment [10]. In this work, we propose a novel approach for creating MRNVD that can minimize the total cycle distances of graph-nodes to their allotted k service center nodes with different types.

2.4 Scope and outline

The rest of the paper is organized as follows: Section 3 explains the baseline and proposed pruning approaches for the MRNVD problem. We provide correctness proofs of the proposed approaches in Section 4. In Section 5, we give a cost model of our proposed approaches. Section 6 presents the experimental observations and results. A real world example is given in Section 7 as a case study. Finally, Section 8 concludes the paper.

3 Proposed Approach for MRNVD

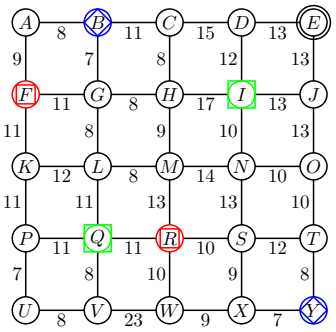
In this section, we first describe the baseline approach that creates the optimal MRNVD, and then we introduce the Distance bounded Pruning (DP) approach that can reduce the computational cost by using three key components: 1) Straight-Distance bounded Pruning (SDP), 2) Triangular-Distance bounded Pruning (TDP), and 3) 2-opt cycle route computation.

3.1 Baseline approach

The baseline approach starts by generating all possible combinations of k different service center nodes and identifies the shortest cycle distances of graph-nodes to their allotted service centers. The key component of the baseline approach is to utilize the dynamic programming technique to find the shortest cycle distance of a graph-node to its allotted service centers [2, 12].

Consider the example input of MRNVD in Figure 2a (reproduced from Figure 1a). Let us identify the cycle distance for node E. First, the baseline approach generates all combinations for three types of service centers. In this example, the service centers are of three types: Type1, Type2, and Type3, and each type has two service centers (i.e., $Type1(B, Y)$, $Type2(I, Q)$, and $Type3(F, R)$). The number of combinations is $2^3 = 8$ and these combinations are $\{B, F, I\}$, $\{B, I, R\}$, $\{B, F, Q\}$, $\{B, Q, R\}$, $\{F, I, Y\}$, $\{I, R, Y\}$, $\{F, Q, Y\}$, and $\{Q, R, Y\}$. Second, it computes the cycle distance for each combination using the Held–Karp algorithm (see Figure 2b) [12, 21]. Since combination $\{B, F, I\}$ can produce the shortest cycle for node E (i.e., $E \to I \to F \to B \to E$ (117)), the baseline approach assigns service center nodes $\{B, F, I\}$ to node E.

The baseline approach examines all graph-nodes (i.e., nodes $A-Y$) and computes the cycle distance for each graph-node. It creates the optimal solution for MRNVD (see Lemma 1). However, since the computational cost is exponential, it is challenging to find the optimal solution for large-size transportation network (see Section 5.1) [21].



(a) Input Example.

Combinations	Shortest Cycle	Cycle Distance
$\{B, F, I\}$	$E \xrightarrow{25} I \xrightarrow{36} F \xrightarrow{17} B \xrightarrow{39} E$	117
$\{F, I, Y\}$	$E \xrightarrow{25} I \xrightarrow{36} F \xrightarrow{66} Y \xrightarrow{44} E$	171
$\{B, I, R\}$	$E \xrightarrow{25} I \xrightarrow{33} R \xrightarrow{36} B \xrightarrow{39} E$	133
$\{I, R, Y\}$	$E \xrightarrow{25} I \xrightarrow{33} R \xrightarrow{26} Y \xrightarrow{44} E$	128
$\{B, F, Q\}$	$E \xrightarrow{39} B \xrightarrow{17} F \xrightarrow{30} Q \xrightarrow{63} E$	149
$\{F, Q, Y\}$	$E \xrightarrow{44} Y \xrightarrow{37} Q \xrightarrow{30} F \xrightarrow{55} E$	166
$\{B, Q, R\}$	$E \xrightarrow{39} B \xrightarrow{26} Q \xrightarrow{11} R \xrightarrow{58} E$	134
$\{Q, R, Y\}$	$E \xrightarrow{44} Y \xrightarrow{26} R \xrightarrow{11} Q \xrightarrow{63} E$	144

(b) Cycle Distance Computation for node E.

Figure 2 Example of service center allotment for node E using the cycle distance computation.

3.2 Proposed Approaches

In this section, we describe two novel pruning methods (i.e., Straight-Distance bounded Pruning (SDP) and (b) Triangular-Distance bounded Pruning (TDP)) to reduce the search space for the MRNVD problem. In addition, we introduce the 2-opt cycle route computation method to minimize the computational cost for the cycle distance.

3.2.1 Straight-Distance bounded Pruning (SDP)

The main performance bottleneck of the baseline approach is to compute the cycle distance for each combination of service centers. In this subsection, we introduce the Straight-Distance bounded Pruning (SDP) method that can reduce the search space for the combinations using a Set Window (SW).

▶ **Definition 2 (Set Window).** *Given a set of service center nodes S, a node n, and a distance bound d, a Set Window (SW) is defined as a set of service centers $SW \subset S$ that are within distance of d from node n.*

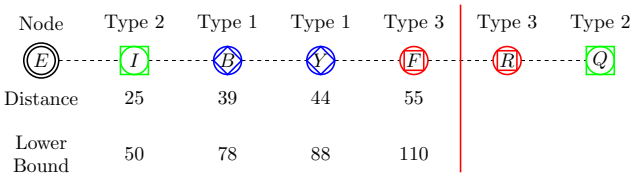
The core idea of SDP is to find the lower and upper bounds of the cycle distance based on a Set Window (SW) and rule out non-optimal combinations when the center nodes in these combinations violate the bound constraints. This approach can reduce the number of computations for the cycle distance without losing the optimality of the solution (see Lemma 3 and 4). The SDP method proceeds in three steps. First, it constructs the initial Set Window (SW). Next, it incrementally increases the size of SW until it meets the lower and upper bounds. Finally it finds the minimum cycle distance in SW.

▶ **Definition 3 (Initial Set Window).** *Given a set of service center nodes S and a node n, the Initial Set Window is defined as the minimum set of closest service centers to n that contain all types of service centers.*

SDP starts by creating an initial SW for each graph-node. Given a node $n \in N$, SDP constructs an ordered-list of service centers based on the distance from node n. Then, it identifies the minimum-sized SW that includes all types of service centers. We set the minimum-sized SW as the initial SW because it contains k different service centers and creates the cycle distances that are feasible but may not be optimal (Lemma 2).

The SDP method incrementally increases the size of SW for node n and updates the lower and upper bounds of the optimal cycle distance of node n. Given a Set Window (SW), the lower-bound of the cycle distance is obtained by doubling the distance of n to the farthest

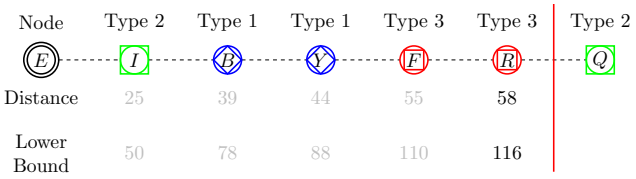
service center node in SW (see Lemma 3). The upper-bound of the cycle distance is the minimum cycle distance among all combinations in SW (see Lemma 4). If the lower-bound is greater than the upper-bound, SDP stops increasing the size of SW and finds the optimal cycle distance in the current SW.



Combinations	Shortest Cycle	Cycle Distance
$\{B, F, I\}$	$E \to I \to F \to B \to E$	117
$\{F, I, Y\}$	$E \to I \to F \to Y \to E$	171

Figure 3 Initial Set Window (SW) for node E and lower and upper bounds.

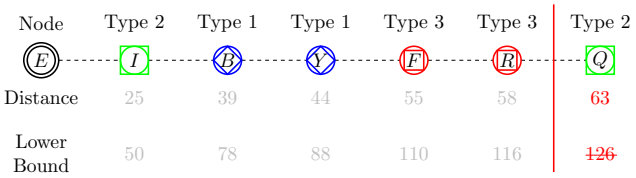
Consider again node E in Figure 2. Figure 3 shows the example of the initial SW for node E. Given a node E, all service centers are ordered by the distance from node E. The vertical bar splits the ordered-list into the left and right parts; The left part becomes the initial SW (i.e., $\{I, B, Y, F\}$) whose size is minimal and includes all types of service centers. Then SDP computes the initial lower and upper bounds of the cycle distance of node E. The lower-bound is the double of the distance from node E to the farthest node in SW (see Lemma 3). Since node F is the farthest service center node from E in SW, the lower-bound becomes 110. Next, SDP generates all possible combinations of the three different service types (i.e., $\{B, F, I\}$ and $\{F, I, Y\}$) and identifies the minimum cycle distance among these combinations. Since the minimum cycle distance is 117 in the initial SW, the upper-bound becomes 117 (see Lemma 4). Since the upper-bound is greater than the lower-bound, SDP can increase the size of SW.



Combinations	Shortest Cycle	Cycle Distance
$\{B, F, I\}$	$E \to I \to F \to B \to E$	117
$\{F, I, Y\}$	$E \to I \to F \to Y \to E$	171
$\{B, I, R\}$	$E \to I \to R \to B \to E$	133
$\{I, R, Y\}$	$E \to I \to R \to Y \to E$	128

Figure 4 SDP: Iteration 1: lower and upper bounds in SW.

After the construction of the initial SW, SDP incrementally increases the size of SW by one and updates the lower and upper bounds until SW violates the bound constraints. Figure 4 shows SW whose size is increased by one (i.e., $\{I, B, Y, F, R\}$). The new combinations generated by SW are $\{B, I, R\}$ and $\{I, R, Y\}$. The upper-bound is the same as the previous upper-bound (i.e., 117), but the lower-bound is updated to 116. Since the upper-bound is greater than the lower-bound, SDP continues to increase the size of SW.



Combinations	Shortest Cycle	Cycle Distance
$\{B, F, I\}$	$E \to I \to F \to B \to E$	117
$\{F, I, Y\}$	$E \to I \to F \to Y \to E$	171
$\{B, I, R\}$	$E \to I \to R \to B \to E$	133
$\{I, R, Y\}$	$E \to I \to R \to Y \to E$	128

Figure 5 SDP: Iteration 2: lower and upper bounds in SW.

Figure 5 shows that SDP adds node Q to increase the size of SW by one. The lower-bound is updated to 126. Since the lower-bound is greater than the upper-bound (i.e., 117), SW violates the bound constraints. Therefore, SDP assigns $\{B, F, I\}$ to node E and stop the search immediately. The SDP method can be summarized as follows. 1) construct the initial SW, 2) incrementally increase the size of SW and update lower and upper bounds, 3) stop when SW violates the bound constraints and return the optimal cycle distance.

3.2.2 Triangle-Distance bounded Pruning Approach

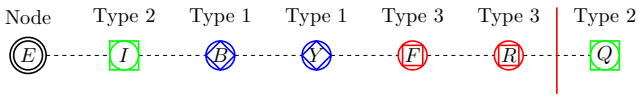
In this subsection, we introduce the Triangle-Distance bounded Pruning (TDP) method that can prune the search space for combinations using the max-min triangle-distance.

▶ **Definition 4 (Triangle-Distance(n, s_1, s_2)).** *Given a starting node n and two service center nodes s_1 and s_2, the triangle-distance is defined as the cycle distance of $n \to s_1 \to s_2 \to n$.*

▶ **Definition 5 (Min Triangle-Distance(n, s_1, t)).** *Given a starting node n, a service center node s_1, and a type of service centers t, the min triangle-distance is defined as the minimum Triangle-Distance(n, s_1, s_2), where type(s_2) $= t$.*

▶ **Definition 6 (Max-Min Triangle-Distance(n, s_1)).** *Given a starting node n, a service center node s_1, and a set of types of service centers T, the max-min triangle-distance is defined as the maximum of min triangle-distance(n, s_1, $t \in T$).*

The core idea of TDP is that when Max-Min Triangle-Distance (n, s_1) is greater than the upper-bound of the cycle distance, the algorithm will not compute the cycle distance of the combinations that includes node s_1 (see Lemma 5). We refer to s_1 as the anchor-node.



Combinations	Shortest Cycle	Cycle Distance
$\{B, F, I\}$	$E \to I \to F \to B \to E$	117
$\{F, I, Y\}$	$E \to I \to F \to Y \to E$	171
$\{B, I, R\}$		
$\{I, R, Y\}$		

Figure 6 Node E Set Window (SW) and all combinations for TDP.

Given a Set Window (SW), TDP starts by constructing the triangle-distance table for anchor-nodes (i.e., service center nodes) and computes the max-min triangle-distance for each anchor-node. Consider the Set Window (SW) in Figure 6 (reproduced from Figure 4). First, TDP groups a set of service center nodes based on types and constructs the triangle-distance table for anchor-nodes (i.e., nodes I, B, Y, F, and R) (see Figure 7). In this example, the group of type 1 is $\{B, Y\}$, the group of type 2 is $\{I\}$, and the group of type 3 is $\{F, R\}$. Next, TDP computes the min triangle-distance for each type of service centers. For instance, the min triangle-distance with anchor-node I and Type 1 becomes 96. Then, it defines the max-min triangle-distance for each anchor-node.

Note that the upper-bound of the cycle distance in SW is 117 (see Figure 6). Since the max-min triangle-distances of nodes Y and R are greater than the upper-bound of the cycle distance (i.e., 128), nodes Y and R cannot be a part of the shortest cycle. Therefore, TDP can rule out the computations of the cycle distances for combinations $\{F, I, Y\}$ $\{B, I, R\}$ and $\{I, R, Y\}$ because these combinations cannot produce the optimal cycle distance. The TDP method can be summarized as follows. 1) group a set of service centers based on

Anchor Nodes	Type 1		Type 2	Type 3		Min Triangle-Distance			Max-Min Triangle-Distance
	B	Y	I	F	R	Type 1	Type 2	Type 3	
I	96	107	50	116	116	96	50	116	116
B	78	148	96	111	133	78	96	111	111
Y	148	88	107	165	128	88	107	128	128
F	111	165	116	110	153	111	116	110	116
R	133	128	116	153	116	128	116	116	128

Figure 7 Triangle-Distance Table for node E (Highlighted values violate the triangle-distance bound).

types, 2) compute the min-triangle-distance for each anchor-node and each type, 3) compute the max-min triangle-distance for each anchor-node, and 4) rule out the combinations that violate the triangle-distance bound.

3.3 2-opt Cycle Route Computation

Although SDP and TDP rule out the computations of the non-optimal cycle distance, the proposed DP algorithm may be inapplicable for sizable road networks because the computational cost of the optimal cycle distance is exponential in terms of the number of service types (see Section 5.2) [2, 12]. Thus we propose a more scalable algorithm using the 2-opt method [4, 8]. The 2-opt method is a heuristic that repeatedly applies 2-opt swaps to minimize the cycle distance. Our proposed approach uses the nearest neighbor heuristic to construct the initial solution and applies the 2-opt method to find the near-optimal cycle distance [1]. The novel component of our approach is to utilize the Tabu-search method that can easily transform 2-opt swaps to 4-opt or more swaps. A Tabu-search uses a Tabu-list in order to escape from local minima and search neighboring solutions until a certain stopping criterion is satisfied. The algorithm convergence of Distance bounded Pruning (DP) with Tabu-search follows from a well-known result about the convergence of the Convergence Tabu Search (CTS) [11].

Given a solution s, let $N(s)$ be the set of neighborhood solutions of s. Let $G_N = (V, E)$ be a graph induced by $N(s)$, where V is a set of solutions and E represents the neighborhood relationship between two solutions. The CTS algorithm converges and terminates after exploring all solutions S if the following two conditions hold [11]:

1. The neighborhood relation is symmetric, i.e. $x \in N(y) \Leftrightarrow y \in N(x)$ for all $x, y \in S$
2. Given a graph G_N, there exists a path between every pair of solutions $x, y \in S$.

Since the 2-opt method satisfies the two conditions, DP with Tabu-search converges and terminates (see Lemma 6). In addition, it can significantly reduce the computational cost of the DP algorithm (see Section 5.3).

Algorithm 1 presents the pseudo-code for Distance bounded Pruning (DP). DP computes the distance matrix for graph-nodes in G (Line 1). For each graph-node n, DP computes the cycle distance of n (Line 2-8). First, it constructs the initial Set Window (SW) and compute the lower and upper bounds of the optimal cycle distance (Line 3-4). Next, it incrementally increases the size of SW and updates the lower and upper bounds of the optimal cycle distance (Line 6-7). SDP and TDP are used to rule out the non-optimal combinations for the cycle distance computation. When SW violates the bound constraints, DP finds the optimal cycle distance and assigns service center nodes to n (Line 9). This process continues until all graph-nodes are allotted (Line 2). Finally, MRNVD is returned (Line 11).

Algorithm 1 Distance bounded Pruning algorithm (Pseudo-code).

Inputs:
- A transportation network $G(N, E)$ with graph-nodes N and edges E.
- A set of service center locations with k different resource types $S \subset N$.
- Every edge has a distance $d(e)$

Output: Multiple Resource Network Voronoi Diagram
Steps:
1: Compute the distance matrix for graph-nodes in G.
2: **for** graph-node $n \in N$ in $G(N, E)$ **do**
3: Construct the initial Set Window (SW) for n.
4: Compute the initial lower and upper bounds of the cycle distance.
5: **while** the bound constraints are not violated **do**
6: Increase the size of SW by one.
7: Update the lower and upper bounds and prune search space using SDP and TDP.
8: **end while**
9: Identify the cycle distance of n and allot service centers nodes to n.
10: **end for**
11: return MRNVD (i.e., allotment of graph-nodes to their service centers).

4 Analysis of the MRNVD proposed approaches

In this section, we prove that the proposed DP approaches are correct, i.e., the DP algorithm creates a MRNVD.

▶ **Lemma 1.** *The baseline approach to the MRNVD problem creates the optimal solution.*

Proof. The baseline approach considers all combinations of service centers for the cycle distance. For each combination, it utilizes the dynamic programming method to compute the cycle distance [12]. The optimal structure of the cycle distance is that every sub-path of the minimum cycle is itself a path with the minimum distance. Therefore, the output of the baseline approach is optimal. ◀

▶ **Lemma 2.** *The initial Set Window (SW) should be the minimum-sized SW that contains k different service centers.*

Proof. Assume that the initial SW has less than k different service centers. Then the initial SW cannot produce the feasible solution for the allotment. This contradicts the original assumption. ◀

▶ **Lemma 3.** *The lower-bound of the cycle distance is obtained by doubling the distance of n to the farthest service center node in the Set Window (SW).*

Proof. The lower-bound of the cycle distance can be proven by the mathematical induction method. Let n be the starting point, let S_{sw} be a set of service center nodes in SW, and let s_0 be the farthest service center node from n. We begin with the initial SW. The initial SW is the minimum node set that includes all different types of service centers. Therefore, the feasible solution of the minimum cycle should include the farthest service center node in the initial SW. Let the shortest distance of n to s_0 be $cost(n, s_0)$. Assume that we add one service center $s \in S$ to cycle $n \to s_0 \to n$. After the addition of the service center $s \in S_{sw}$, the shortest distance of the cycle monotonically increases according to the triangle inequality theorem. Therefore, $2 \cdot cost(n, s_0)$ becomes the lower-bound of the cycle distance for the initial SW. Next, we increase the size of SW by one. Then the new added node becomes the farthest service center node from n. Let s_1 be the farthest service center node from n. SW should include s_1 to compute the cycle distance. If not, we do not need to increase the size of SW. According to the triangle inequality theorem, $2 \cdot cost(n, s_1)$ becomes the lower-bound of the cycle distance for SW. Therefore, we complete the proof by induction. ◀

▶ **Lemma 4.** *The upper-bound of the cycle distance is the minimum cycle distance in SW.*

Proof. Let S be a set of service center nodes and S_{sw} be a set of service center nodes. Since $S_{sw} \subset S$, the minimum cycle distance in S_{sw} is greater than or equal to the optimal cycle distance in S. Therefore, the upper-bound of the cycle distance is the minimum cycle distance in SW. ◀

▶ **Lemma 5.** *If Max-Min Triangle-Distance(n, s_1) is greater than the upper-bound of the cycle distance, then anchor-node s_1 cannot be a part of the optimal cycle.*

Proof. Let n be the starting point, let s_1 be the anchor point, and let T be a set of types of service centers. *Min Triangle-Distance*(n, s_1, t) becomes a lower-bound of the cycle distance for every type $t \in T$. Therefore, $\max_{t \in T}$ *Min-Triangle-Distance*(n, s_1, t) becomes a lower-bound of the cycle distance that include anchor-node s_1. Since the lower-bound cannot be greater than the upper-bound, anchor-node s_1 cannot be a part of the optimal cycle. ◀

▶ **Lemma 6.** *The 2-opt method with Tabu-search converges and terminates.*

Proof. The 2-opt method has the symmetric neighborhood relation. Moreover, every solution has a path to other solutions by swapping two nodes. Since the 2-opt method satisfies the two conditions of CTS, the proof is complete. ◀

5 Algebraic Cost Model of Pruning Algorithms

The goal of this section is to present cost models for our proposed approaches. Let n be the number of graph-nodes, let k be the number of types in service centers, let c be the maximum number of service centers for a service type.

5.1 Baseline Approach

The baseline approach starts by generating all possible combinations of k different service centers. This takes $O(c^k)$. Given a combination, the cost of computation for the cycle distance is $2^k \cdot k^2$ [1]. Since the number of combinations is bounded by $O(c^k)$, the minimum cycle distance for a graph-node can be obtained by the cost of $O(c^k \cdot 2^k \cdot k^2)$. The number of graph-nodes is n. Therefore, the baseline approach takes $O(c^k \cdot 2^k \cdot k^2 \cdot n)$.

5.2 Distance Bounded Pruning (DP) Approach

The Distance bounded Pruning (DP) approach starts by ordering service centers based on the distance. This takes $O(c \cdot k \cdot \log(c \cdot k))$. Next, The Straight-Distance bounded Pruning (SDP) method creates an initial Set Window (SW) and incrementally increase the size of SW. The size of SW is bounded by $O(c \cdot k)$. During this incremental process, the cost for computing the lower-bounds is $O(c \cdot k)$ and the cost for computing the upper-bound is $O(c^k \cdot 2^k \cdot k^2)$. The Triangle-Distance bounded Pruning (TDP) method creates the Triangle-Distance Tables to compute the max-min triangle-distances. This takes $O(c^2 \cdot k^2)$. Thus, the total cost of the allotment for each node is $O(c \cdot k + c^2 \cdot k^2 + c^k \cdot 2^k \cdot k^2) = O(c^k \cdot 2^k \cdot k^2)$. Since the number of graph-nodes is n, DP takes $O(c^k \cdot 2^k \cdot k^2 \cdot n)$. In worst case, the cost model of DP is the same as the cost model of the baseline approach. However, DP can rule out the non-optimal combinations and significantly reduce the number of computations of the cycle distances using SDP and TDP.

5.3 DP with 2-opt cycle route computation

DP with 2-opt cycle route computation (DP 2-opt) uses the Tabu-search method and reduces the computational cost of the cycle distances. The Tabu-search requires multiple iterations to find the near-optimal solution [9, 20]. At each iteration, it swaps two pairs of nodes and temporally locks these nodes for the iteration. Each swap takes $O(k^2)$. The number of swaps is bounded by $O(k)$. Thus, the cost of each iteration takes $O(k^3)$. Assume that the number of iteration is bounded by $O(i)$. Then, the cost of computations of the cycle distance is $O(k^3 \cdot i)$. Therefore, the cost model for DP 2-opt is $O(c^k \cdot k^3 \cdot i \cdot n)$. Since the number of iterations (i.e., i) until convergence is often small, the cost model in practice is considered to be $O(c^k \cdot k^3 \cdot n)$.

6 Experimental Evaluation

We conducted experiments to evaluate performance of Baseline and Distance bounded Pruning (DP) approaches. The overall goal was to show the performance improvements to create a MRNVD that can be obtained by the DP approach. We wanted to answer four questions: (1) What is the effect of the number of service types? (2) What is the effect of the number of service centers? (3) What is the effect of the size of the network (i.e., number of graph-nodes)? (4) Is DP algorithm correct, and is the solution quality preserved?

6.1 Experiment Layout

Figure 8 shows our experimental setup. We chose five different municipal areas in the U.S. from OpenStreetMap [18]. We used the locations of service centers in these areas and created a Multiple Resource Network Voronoi Diagram (MRNVD). We tested three approaches: (1) Baseline approach (BL), (2) Distance bounded Pruning approach (DP), and (3) DP with 2-opt heuristic for cycle distance calculation approach (DP 2-opt).

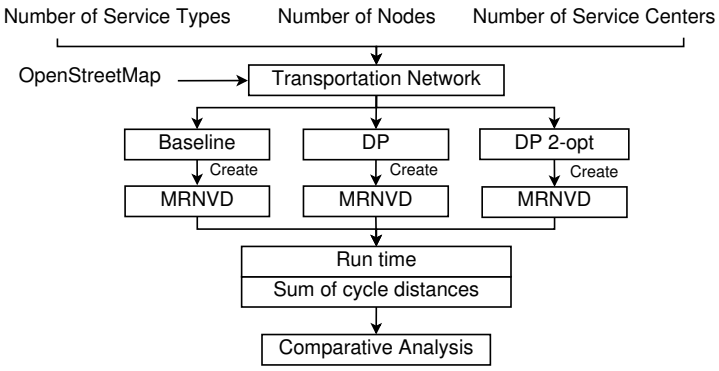


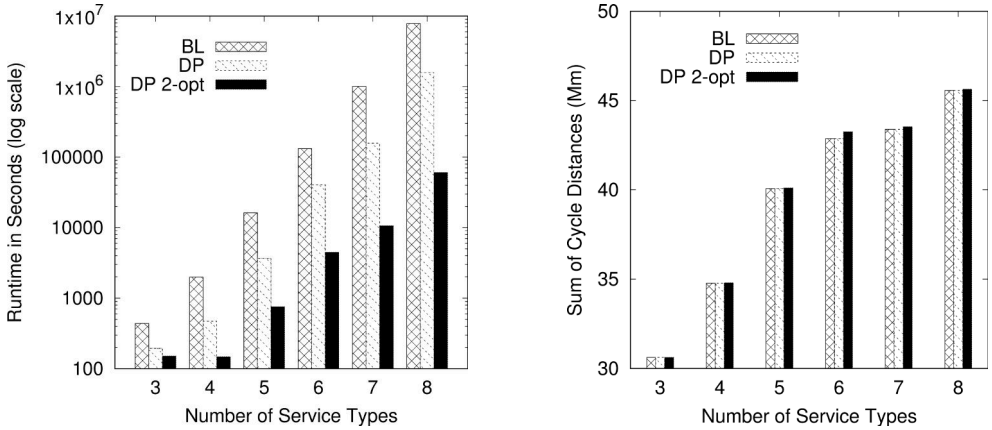
Figure 8 Experiment Layout.

6.2 Experiment Results and Analysis

We experimentally evaluated the proposed algorithms by comparing the impact on performance of (1) number of service types, (2) number of service centers per type and (3) size of the transportation network. We extracted the locations of service centers from OpenStreetMap datasets and then randomly chose a set of service centers from extracted ones to vary the number of service centers. The algorithms were implemented in Java with a 32 GB memory run-time environment. All experiments were performed on an Intel Core i5 machine running Windows 10 with 32 GB of RAM.

6.2.1 Effect of Number of Service Types

The first set of experiments evaluated the effect of the number of service types on the performance of the algorithms. We used a Florida road map consisting of 460, 791 nodes and 653, 392 edges. We fixed the number of nodes to 5, 000 and the number of service centers to 3. We varied the number of service types from 3 to 8. We randomly chose the locations of service centers and constructed 45 test cases. Performance measurements were execution time and the sum of the cycle distances. The performance measurements were averaged over 45 test runs. Figure 9a gives the execution times. As can be seen, the DP approaches outperforms the baseline approach. This is because the number of combinations for the cycle distance computation increases as the number of service types increases. DP with 2-opt heuristic outperforms other approaches because it can reduce the computational cost for the cycle distance. When comparing the sum of the cycle distances, we see that the DP approach produce the optimal solution (see Figure 9b). This means that SDP and TDP have no effect on the solution quality. DP with 2-opt heuristic (DP 2-opt) performs almost identically to the optimal approaches. As the number of service types increase, the sum of the cycle distances increases.



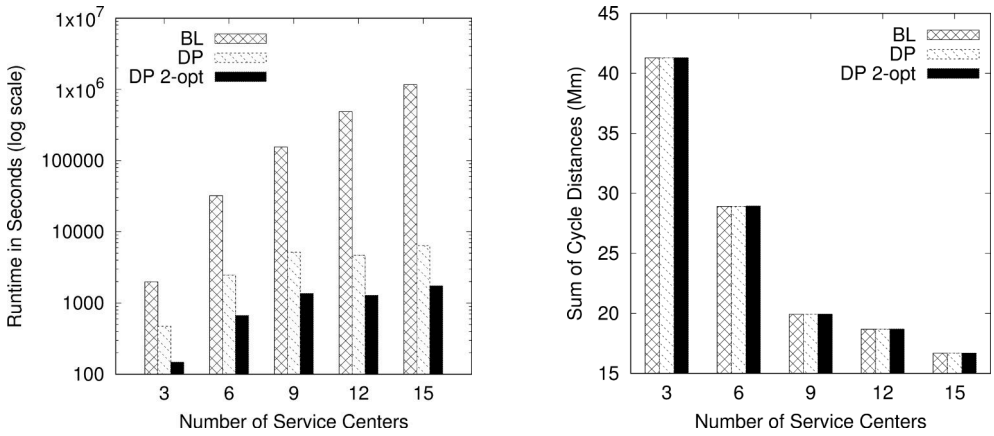
(a) Run Time Comparison.

(b) Comparison of Solution Quality.

Figure 9 Effect of number of service types ($n = 5,000$, $c = 3$).

6.2.2 Effect of Number of Service Centers

The second set of experiments evaluated the effect of the number of service centers on the performance of the algorithms. Performance measurements were execution time and the sum of the cycle distances. We fixed the number of nodes to 5, 000 and the number of service types to 4. The number of service centers was varied from 3 to 15. Locations of service centers were randomly chosen in 54 test cases. Figure 10a shows that the DP approaches significantly outperform the baseline approach. The performance gap increases as the number of service centers increases. This is because the number of combinations for the cycle distance computation increases as the number of service centers increases. DP 2-opt significantly outperforms other approaches due to the reduced computational cost for the cycle distance. Figure 10b shows that the DP approach performs exactly the same as the baseline approach. We can see that DP 2-opt was faster than DP, albeit slightly lower performance in terms of sum of cycle distances. As the number of service centers increases, the sum of cycle distance decreases.

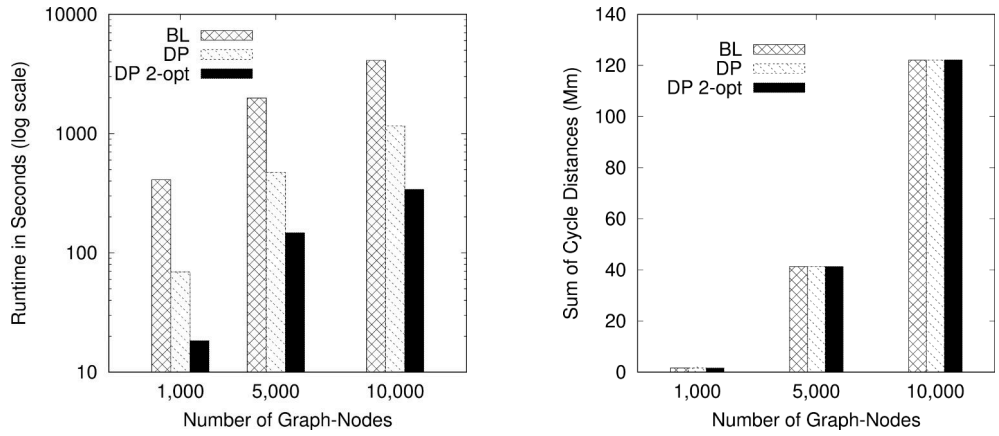


(a) Run Time Comparison. **(b)** Comparison of Solution Quality.

Figure 10 Effect of number of service centers ($n = 5,000$, $k = 4$).

6.2.3 Effect of Network Size

The third set of experiments evaluated the effect of the network size on algorithm performance. We fixed the number of service types to 4 and the number of centers per type to 3. We increased the number of nodes from $1,000$ to $10,000$. Service center locations were chosen randomly and execution times were averaged over 30 test runs for each road network. Figure 11a shows that the DP approaches significantly outperforms the baseline (BL) approach. This is because the size of the Service Areas increases as the number of nodes increases. DP 2-opt outperforms others due to the reduction of computational cost for the cycle distance. Figure 11b shows that BL and DP perform identical. DP 2-opt performs almost identically to the optimal approaches. As the number of nodes increases, the sum of cycle distance increases.



(a) Run Time Comparison **(b)** Comparison of Solution Quality.

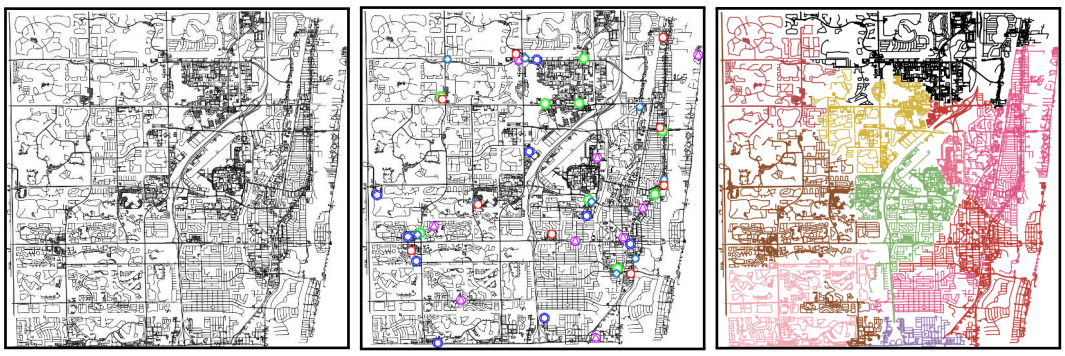
Figure 11 Effect of network size ($k = 4$, $c = 3$).

6.2.4 Discussion

The proposed DP approach achieves a significant computation performance gain over the optimal baseline approach. This improvement was obtained by using three key components: 1) Straight-Distance bounded Pruning (SDP), 2) Triangle-Distance Bounded Pruning (TDP), and 3) 2-opt cycle route computation. The baseline approach computes the optimal cycle distances using dynamic programming; However the computational cost of the baseline approach is prohibitive on the large sized networks [12]. To remedy this issue, SDP creates a Set Window (SW) and reduces the number of computations for the cycle distance by using the bound constraints. TDP reduces the number of computations by identifying the anchor nodes that violate the upper-bound constraint. The 2-opt cycle route computation further reduces the computational cost by utilizing the Tabu-search method. The experimental results shows that the proposed approaches significantly reduce the computational cost to create a MRNVD. The source code is available on our research group website [3].

7 Case Study with Boca Raton road network

In our case study, we created a MRNVD that can identify a set of Service Areas (SAs) to minimize the travel time for citizens to visit all required service centers. For transportation network, we used a Boca Raton, FL road map consisting of $18,679$ nodes and $25,835$ edges (Figure 12a). We chose five different service types (i.e. grocery stores, gas stations, pharmacies, healthcare facilities and law enforcement departments) and nine service centers for each service type. Each circle symbol represents a different type of service centers (Figure 12b). Figure 12c shows the MRNVD constructed thirteen Service Areas that can minimize the total cycle distances. The sum of cycle distances using DP and DP 2-opt are $159,872km$ and $160,021km$ respectively. Our case study showed that the run-time of the baseline approach took 4 hours to produce a MRNVD. DP took 6 minutes whereas DP 2-opt took 1 minute. The solution of the DP approach is exactly the same as that produced by the baseline approach.



(a) Boca City Road Network. **(b)** Locations of Service Centers. **(c)** MRNVD with 13 Service Areas.

Figure 12 Case Study: Boca Raton, FL road map (Best in Colors).

8 Conclusion and Future work

We presented the problem of creating a Multiple Resource Network Voronoi Diagram (MRNVD). An important societal application of MRNVD is promoting transportation resiliency before or after a disaster. The MRNVD problem is challenging due to multiple

different types of resources. General Network Voronoi Diagram uses the distance metric and divides the region based on the closest service center. However, the distance for multiple resources cannot utilize the absolute distance metric because it uses the cycle distance metric for visiting all required service centers. In this paper, we introduced a novel Distance bounded Pruning (DP) approach for creating a MRNVD that can minimize the total cycle distances of graph-nodes to allotted k service center nodes. We presented experiments and case study using a Boca Raton road map.

In future work, we plan to further explore new optimal pruning methods to reduce the computational cost for creating a MRNVD. In addition, we will develop a parallel formulation of the propose approaches to handle continental-sized transportation networks. We will also investigate the effect of applying spatial filters on reducing the size of the network and improving the performance of MRNVD. MRVND with Monte Carlo simulation may solve the facility location problem. We will study new method that determines the near-optimal positions of service facilities. Lastly, we plan to design new MRNVD problem that includes the capacity constraint for each service center and the directional constraint based on directed graphs.

References

1 David L Applegate, Robert E Bixby, Vasek Chvatal, and William J Cook. *The traveling salesman problem: a computational study*. Princeton university press, 2006.

2 Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM (JACM)*, 9(1):61–63, 1962.

3 MRNVD Source Code. `http://faculty.eng.fau.edu/yangk/home/NSF_Career_Projects.html`,Retrieved Jun. 2020.

4 Georges A Croes. A method for solving traveling-salesman problems. *Operations research*, 6(6):791–812, 1958.

5 Matthew T Dickerson and Michael T Goodrich. Two-site voronoi diagrams in geographic networks. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, pages 1–4, 2008.

6 Matthew T Dickerson, Michael T Goodrich, Thomas D Dickerson, and Ying Daisy Zhuo. Round-trip voronoi diagrams and doubling density in geographic networks. In *Transactions on Computational Science XIV*, pages 211–238. Springer, 2011.

7 Martin Erwig. The graph voronoi diagram with applications. *Networks: An International Journal*, 36(3):156–163, 2000.

8 Merrill M Flood. The traveling-salesman problem. *Operations research*, 4(1):61–75, 1956.

9 Fred Glover. Tabu search—part i. *ORSA Journal on computing*, 1(3):190–206, 1989.

10 Anita Graser. Tessellating urban space based on street intersections and barriers to movement. *GI_Forum 2017*, 5(1):114–125, 2017.

11 Said Hanafi. On the convergence of tabu search. *Journal of Heuristics*, 7(1):47–58, 2001.

12 Michael Held and Richard M Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied mathematics*, 10(1):196–210, 1962.

13 Mohammad Kolahdouzan and Cyrus Shahabi. Voronoi-based k nearest neighbor search for spatial network databases. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 840–851, 2004.

14 Ickjai Lee, Kyungmi Lee, and Christopher Torpelund-Bruin. Raster voronoi tessellation and its application to emergency modeling. *Geo-spatial Information Science*, 14(4):235–245, 2011.

15 Atsuyuki Okabe, Barry Boots, and Kokichi Sugihara. Nearest neighbourhood operations with generalized voronoi diagrams: a review. *International Journal of Geographical Information Systems*, 8(1):43–71, 1994.

16 Atsuyuki Okabe, Toshiaki Satoh, Takehiro Furuta, Atsuo Suzuki, and Kyoko Okano. Generalized network voronoi diagrams: Concepts, computational methods, and applications. *International Journal of Geographical Information Science*, 22(9):965–994, 2008.

17 Atsuyuki Okabe and Kokichi Sugihara. *Spatial analysis along networks: statistical and computational methods*. John Wiley & Sons, 2012.

18 OpenStreetMap. `http://goo.gl/Hso0`,Retrieved Feb. 2020.

19 Daniel J Rosenkrantz, Richard E Stearns, and Philip M Lewis, II. An analysis of several heuristics for the traveling salesman problem. *SIAM journal on computing*, 6(3):563–581, 1977.

20 Shigeru Tsubakitani and James R Evans. An empirical study of a new metaheuristic for the traveling salesman problem. *European Journal of Operational Research*, 104(1):113–128, 1998.

21 Gerhard J Woeginger. Exact algorithms for np-hard problems: A survey. In *Combinatorial optimization—eureka, you shrink!*, pages 185–207. Springer, 2003.

22 KwangSoo Yang, Apurv Hirsh Shekhar, Dev Oliver, and Shashi Shekhar. Capacity-constrained network-voronoi diagram. *IEEE Transactions on Knowledge and Data Engineering*, 27(11):2919–2932, 2015.

LSTM-TrajGAN: A Deep Learning Approach to Trajectory Privacy Protection

Jinmeng Rao
GeoDS Lab, Department of Geography, University of Wisconsin-Madison, WI, USA

Song Gao
GeoDS Lab, Department of Geography, University of Wisconsin-Madison, WI, USA

Yuhao Kang
GeoDS Lab, Department of Geography, University of Wisconsin-Madison, WI, USA

Qunying Huang
Department of Geography, University of Wisconsin-Madison, WI, USA

Abstract

The prevalence of location-based services contributes to the explosive growth of individual-level trajectory data and raises public concerns about privacy issues. In this research, we propose a novel LSTM-TrajGAN approach, which is an end-to-end deep learning model to generate privacy-preserving synthetic trajectory data for data sharing and publication. We design a loss metric function TrajLoss to measure the trajectory similarity losses for model training and optimization. The model is evaluated on the trajectory-user-linking task on a real-world semantic trajectory dataset. Compared with other common geomasking methods, our model can better prevent users from being re-identified, and it also preserves essential spatial, temporal, and thematic characteristics of the real trajectory data. The model better balances the effectiveness of trajectory privacy protection and the utility for spatial and temporal analyses, which offers new insights into the GeoAI-powered privacy protection.

2012 ACM Subject Classification Security and privacy → Privacy protections; Computing methodologies → Artificial intelligence

Keywords and phrases GeoAI, Deep Learning, Trajectory Privacy, Generative Adversarial Networks

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.12

Funding Support for this research was provided by the University of Wisconsin - Madison Office of the Vice Chancellor for Research and Graduate Education with funding from the Wisconsin Alumni Research Foundation.

1 Introduction

The increasing location-based services (LBS) have generated large-scale individual-level trajectory data (i.e., a sequence of locations with attributes) through mobile phones, wearable sensors, GPS devices, and geotagged social media [19]. Such trajectory big data provide new opportunities to study human mobility patterns and human-environment interactions [11], disaster responses [12, 27] and public health issues [17, 25]. It also introduces grand challenges regarding the protection of geoprivacy and broader behavioral, social, ethical, legal and policy implications [14]. Generally speaking, trajectory privacy refers to an individual's rights to prevent the disclosure of individual trajectory identity and associated personal sensitive locations [15, 2, 5].

Due to the data breach concerns and increasing public awareness of location privacy protection, many approaches have been proposed to prevent users' trajectories from being identified. A common practice is to remove the identifiers (e.g., user name or ID number) from the trajectory data. However, it turned out that such "de-identified" trajectories may

still cause serious privacy threats since the spatial, temporal and thematic characteristics of trajectories can still be used as strong quasi-identifiers for linking the trajectories to their creators [2]. Another commonly used method is to aggregate trajectory points into geographic or administrative units so that their original locations are not revealed. Nevertheless, recent studies show that aggregation may fail to preserve user privacy and reduce the spatial resolution and effectiveness of spatial analysis [3, 28, 5]. For example, De Montjoye et al. [3] lower the resolution of a human mobility trace dataset through spatial and temporal aggregation to prevent the individuals from being identified, but the coarsened dataset only provides little anonymity. Thus, in order to achieve trajectory privacy protection more efficiently, we need to deal with the spatial and temporal characteristics of trajectory data more comprehensively.

Current trajectory privacy protection studies focus on two research streams. One is the differential privacy approach to grouping and mixing the trajectories from different users so that the identification of individual trajectory data is converted into a k-anonymity problem [23, 31]. For example, the spatial cloaking approach mixes together the trajectory points between k users using k-anonymous cloaked spatial regions, making these trajectories k-anonymized [7]. Also, the mix-zones approach anonymizes the trajectory points in a mix-zone using pseudonyms and breaks the linkage between the former segment and the latter segment of the same trajectory that passes through a mix-zone [24]. Alternatively, the generalization-based approach first divides the points of k trajectories into different k-anonymized regions, and then reconstructs k new trajectories by uniformly selecting points from each k-anonymized regions and linking them together [22].

Another research stream is called geomasking, which blurs the locations of original trajectory data by utilizing perturbation on the spatial dimension so that the original locations can be hidden or modified while spatial patterns may not be significantly affected [9, 5]. For example, Armstrong et al. [1] explored the privacy preservation ability and spatial analysis effectiveness of several types of geomasks. Kwan et al. [15] evaluated the spatial analysis effectiveness of three different random perturbation geomasks on lung-cancer deaths. Seidl et al. [26] applied grid masking and random perturbation on GPS trajectory data and evaluated the privacy protection performance. Gao et al. [5] investigated the effectiveness of random perturbation, gaussian perturbation, and aggregation on Twitter data as well as explored the privacy, analytics, and uncertainty level of each method.

While these approaches all show the capabilities to protect trajectory privacy, they also expose several limitations. First of all, despite the diversity, the goal of these approaches largely is to obfuscate the trajectory locations and add more uncertainty to preserve privacy. However, the trade-off between the effectiveness of trajectory privacy protection and the utility for spatial and temporal analyses is still hard to control [18], and this issue has not been fully discussed or evaluated. Besides, current studies mainly focus on the spatial dimension of trajectory data whereas other semantics (e.g., temporal and thematic attributes) are rarely considered. In fact, these characteristics have been proven to be crucial for trajectory user identification [21]. Moreover, current approaches rely heavily on manually designed procedures. Once the procedure is disclosed, one may have the chance to recover the original trajectory data [28] (e.g., using reverse engineering). The "black-box" machine learning models may help to solve this issue.

To this end, this research aims to explore the effectiveness of state-of-the-art deep learning approaches for trajectory privacy protection. We propose a novel LSTM-TrajGAN model that combines the Long Short-Term Memory (LSTM) recurrent neural network and the Generative Adversarial Network (GAN) structure together to generate privacy-preserving synthetic trajectories as alternatives to real trajectories for trajectory data sharing and publication. Two research questions (RQ) will be investigated in this work.

RQ 1: How effective is the proposed LSTM-TrajGAN model in protecting the trajectory creators from being re-identified? (i.e., privacy protection effectiveness)

RQ 2: Can the synthetic trajectories preserve the semantic features (spatial-temporal-thematic characteristics) compared to real trajectories? (i.e., utility)

The main contributions of our work are fourfold: (1) we propose an end-to-end deep learning approach to generating privacy-preserving trajectory data. The procedure is simple and highly secure (a GeoAI "black-box"); (2) we introduce a trajectory encoding model for semantic trajectory encoding; (3) we design a new TrajLoss metric function to measure the trajectory similarity losses for training deep learning models; and (4) we evaluate the privacy protection effectiveness and the utility of the proposed model using real-world LBS data and explore the trade-off between them.

The remainder of the paper is organized as follows. Section 2 introduces our methodological framework, including a trajectory encoding model, the LSTM-TrajGAN model, and the TrajLoss function design. In section 3, we train and test our model using a city-scale weekly trajectory dataset and compare with other commonly used trajectory privacy protection methods. Both privacy protection effectiveness and utility are evaluated and compared with baseline approaches. In section 4, we discuss the factors affecting privacy protection effectiveness, the trade-off between privacy protection and utility, and the limitations of our model. Section 5 summarizes this research and outlines the future work.

2 Method

Inspired by the vision of the TrajGANs [18], we propose a new approach consisting of three main components: (1) a Trajectory Encoding Model, which encodes GPS location coordinates, temporal attributes, and other attributes such as point of interest (POI) category; (2) a Trajectory Generator, which takes random noise and original trajectories as inputs to generate synthetic trajectories as outputs; and (3) a Trajectory Discriminator, which takes trajectories as inputs and determines them as "real" or "synthetic".

The overall workflow is described in Figure 1. The goal is to train an "intelligent" trajectory generator that generates "realistic" synthetic trajectories to replace the original trajectories, which preserves differential privacy in trajectory analysis tasks such as Trajectory-User Linking (TUL) and trajectory data mining (e.g., work/home location clustering). Meanwhile, it ensures the quality of multiple spatial or temporal summary analysis tasks. Such a framework can serve as a trajectory privacy protection layer in trajectory data acquisition, processing, and publication pipelines, which publish the synthetic alternatives rather than the real trajectory data that may disclose individual privacy.

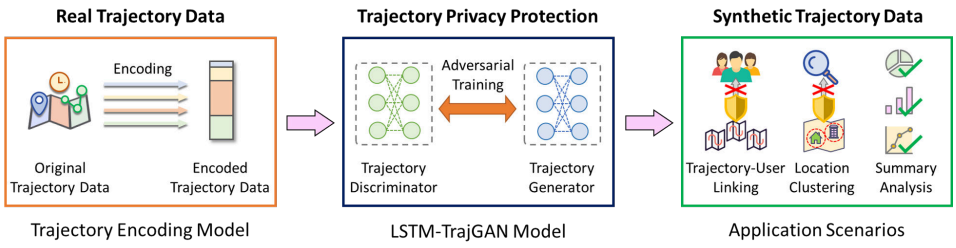


Figure 1 The overall workflow of the proposed LSTM-TrajGAN approach.

2.1 Trajectory Encoding

First, we introduce a trajectory encoding model that converts the original trajectory data to a specific format that serves as the inputs for the LSTM-TrajGAN model. The main reason for the encoding process is that the trajectory data usually contain various types of attributes such as interval data (e.g., GPS coordinates, date and time), nominal data (e.g., POI category), ordinal data (e.g., POI rating), and these data need to be converted into valid numerical representations for training the deep learning model. Our trajectory encoding model includes two parts: trajectory point encoding and trajectory padding.

Trajectory Point Encoding

The trajectory point encoding process is illustrated in Figure 2. A semantic trajectory point contains the following attributes: location, time, user id, trajectory id, and other optional attributes such as POI category. For the location attribute, we standardize all the latitudes and longitudes using the centroid of all the trajectories in the dataset to obtain the deviations of the latitudes and longitudes from the centroid. In this way, the model can better learn the spatial deviation pattern between different trajectory points. These deviation values will be used as the numerical representations of the trajectory points for constructing spatial embeddings [20].

For the temporal attributes and categorical attributes, we use the one-hot encoders (i.e., a representation process using dummy variables in machine learning) to encode the attributes into high-dimensional binary vectors based on their vocabulary sizes. For example, the "Day" attribute is encoded into 7-dimensional binary vectors, and "Monday" is represented as $[1, 0, 0, 0, 0, 0, 0]$. Likewise, the "Hour" attribute is encoded into 24-dimensional binary vectors, and the "Category" attribute is encoded into 10-dimensional binary vectors. Note that we don't encode the User ID and the Trajectory ID since they are only used to indicate the user and the trajectory that the point belongs to.

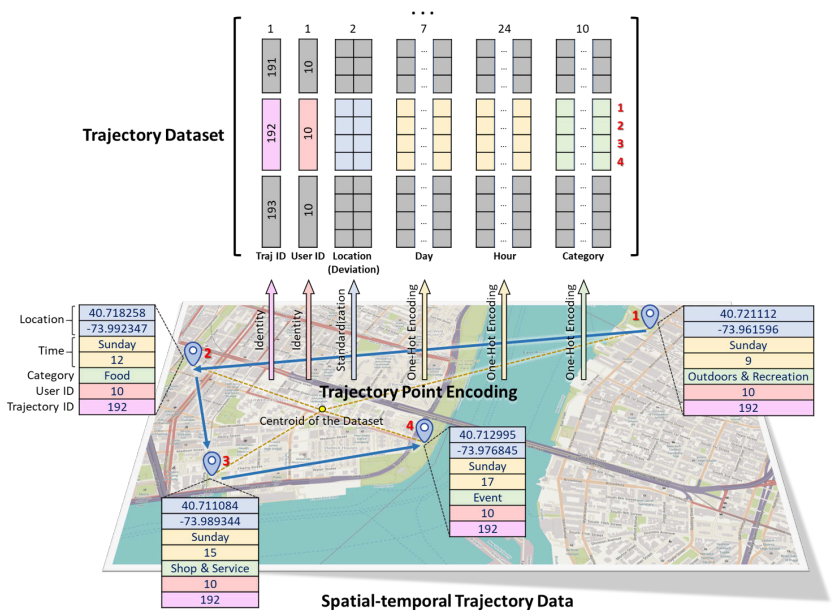


Figure 2 An example for the trajectory point encoding process.

Trajectory Padding

After the trajectory point encoding process, all the spatial, temporal, and thematic attributes of a trajectory are stored in a multidimensional matrix, whose first dimension indicates the index for each trajectory. Since the length of each trajectory data (i.e., the number of the trajectory points) is a variable, we then apply the trajectory padding technique to make sure all the trajectories have the same length as the longest trajectory. Specifically, we use zero pre-padding to pad empty trajectory points (i.e., the points whose attributes are all set to zero) to each trajectory until all the trajectories reached the same length as the longest trajectory in the dataset. The main reason is that the data with the same size can be utilized for batch processing and training the deep learning model, which would speed up the training process. During the model training and inference processes, these padded trajectory points will be masked (i.e., cut) and they won't actually influence the neural network weight updates and the derived results.

2.2 LSTM-TrajGAN Model

Figure 3 describes the neural network structure of the LSTM-TrajGAN Model. The trajectory generator captures the data distribution and pattern of the real trajectory data and generates synthetic trajectory data based on their corresponding original trajectory data and random noise. In addition, the trajectory discriminator distinguishes whether the trajectory samples come from the training set (i.e., real trajectory data) or the trajectory generator (i.e., synthetic trajectory data). The goal of the trajectory generator is to generate "high-quality" synthetic trajectories that can "fool" the trajectory discriminator, which leads to a two-player minimax game between them. The generated synthetic trajectories aim to be competent for spatial and temporal summary analysis, while having some degree of uncertainty and randomness to protect the user privacy in trajectory analysis tasks with privacy issues involved. This idea is reflected in the design and optimization of the LSTM-TrajGAN model.

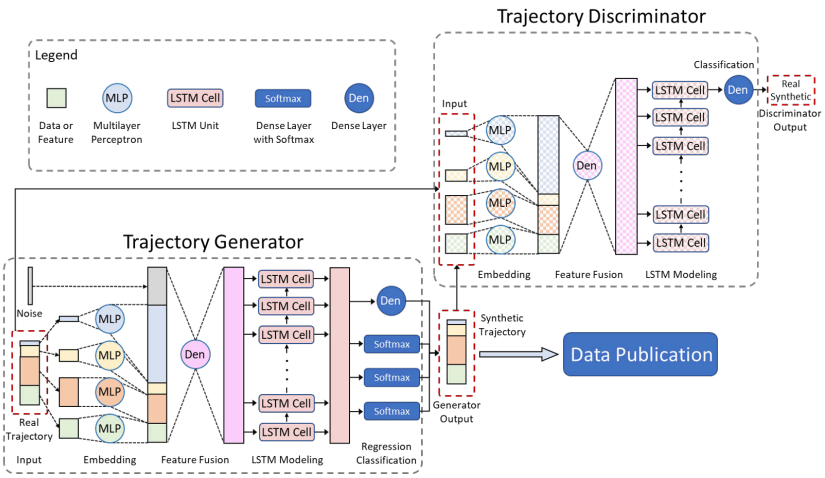


Figure 3 The neural network structure of the LSTM-TrajGAN Model.

Trajectory Generator

As is shown in Figure 3, the trajectory generator consists of five functional layers: the input layer, the embedding layer, the feature fusion layer, the LSTM modeling layer, and the regression/classification layer. The generator first takes the encoded real trajectories and

random noise as inputs, and embeds trajectories using Multilayer Perceptrons (MLPs) [10]. For the spatial dimension of a trajectory (i.e., pairs of latitude and longitude deviations), we embed each pair of them using a MLP to get 64-dimensional vectors. For the temporal dimension (e.g., day and hour) and categorical attributes (e.g., POI category), we use MLPs to embed them respectively and get fixed-length vectors based on their vocabulary sizes:

$$e_i^{spatial} = \phi^s(\Delta lat_i, \Delta lon_i; W_{es}) \tag{1}$$

$$e_i^{day} = \phi^d(v_i^{day}; W_{ed}) \tag{2}$$

$$e_i^{hour} = \phi^h(v_i^{hour}; W_{eh}) \tag{3}$$

$$e_i^{category} = \phi^c(v_i^{category}; W_{ec}) \tag{4}$$

Where Δlat_i and Δlon_i stand for the latitude and longitude deviation of the i-th trajectory point; v_i^{day}, v_i^{hour}, and $v_i^{category}$ stand for the one-hot vectors for the day, hour, and category attributes of the i-th trajectory point; ϕ^s, ϕ^d, ϕ^h, and ϕ^c stand for the MLPs with an activation function – the Rectified Linear Unit (ReLU) for embedding the spatial, daily, hourly, and categorical attributes; W_{es}, W_{ed}, W_{eh}, and W_{ec} are the embedding weight matrices for these MLPs; $e_i^{spatial}$, e_i^{day}, e_i^{hour}, and $e_i^{category}$ are the embedded vectors for each attribute respectively. Note that the embedding weight matrices are shared among all trajectory points.

After the embedding process, we further concatenate all the vectors and the random noise, and then use a dense layer to fuse them into 100-dimensional vectors. By leveraging the feature fusion, we take the advantage of all the spatial, temporal, and categorical characteristics of each trajectory point and fuse them together to support spatiotemporal trajectory modeling and generation. In the LSTM modeling layer, we use a many-to-many LSTM structure that takes a sequence with specific time steps as the input and generates a sequence with the same time steps as the output. Recurrent models such as LSTM are proven to be efficient in spatial-temporal sequence modeling and prediction [8, 21]. Given the dimension of the fused feature, we assign 100 units in the LSTM model and feed the fused features to the model:

$$H = LSTM(F; W_{lstm}) \tag{5}$$

Where F represents for the fused features of all the trajectory points in a trajectory (i.e., $F = [f_0, f_1, ..., f_{maxlength-1}]$), in which f_i is the fused feature vector for the i-th trajectory point; H is the output of the LSTM model, which has the same time step dimensions as the input (i.e., $H = [h_0, h_1, ..., h_{maxlength-1}]$, in which h_i is the modeling output vector for f_i); W_{lstm} is the weight matrix of the LSTM model.

Finally, we decode the synthetic trajectory data from the output H of the LSTM modeling layer. Each feature vector h_i in H is a 100-dimensional vector containing the spatial, temporal, and categorical characteristics of a synthetic trajectory point. To decode the latitude and longitude deviations, we use a dense layer with two units and use the *tanh* hyperbolic tangent function. In addition, we further stretch the output range to make sure its range covers all

the possible deviation values. To decode the day, hour and category attributes, we use dense layers that have as many units as the vocabulary sizes, and use the $softmax$ normalized exponential function to recover the one-hot representation of these attributes:

$$(\Delta lat'_i, \Delta lon'_i) = D^s(h_i; W_{ds}) \tag{6}$$

$$v_i'^{day} = D^d(h_i; W_{dd}) \tag{7}$$

$$v_i'^{hour} = D^h(h_i; W_{dh}) \tag{8}$$

$$v_i'^{category} = D^c(h_i; W_{dc}) \tag{9}$$

Where $\Delta lat'_i$ and $\Delta lon'_i$ are the latitude and longitude deviations of the i-th synthetic trajectory point; $v_i'^{day}$, $v_i'^{hour}$, and $v_i'^{category}$ represent the one-hot vectors for the day, hour, and category attributes of the i-th synthetic trajectory point; D^s, D^d, D^h, and D^c represent the dense layers with a $tanh$ or $softmax$ function for decoding the location, day, hour, and category attributes; W_{ds}, W_{dd}, W_{dh}, and W_{dc} are the decoding weight matrices for these dense layers; Note that the decoding weight matrices are shared among all trajectory points.

Trajectory Discriminator

As is shown in Figure 3, the trajectory discriminator has a very similar structure as the trajectory generator. The major differences between them are:

(1) The discriminator only takes trajectory data as the input (no random noise needed);

(2) We use a many-to-one LSTM model that takes the features with time steps as the input and make one scalar as the output:

$$h = LSTM(F; W_{lstm_d}) \tag{10}$$

Where F represents for the fused features of all the trajectory points in a trajectory (i.e., $F = [f_0, f_1, ..., f_{maxlength-1}]$), in which f_i is the fused feature vector for the i-th trajectory point; W_{lstm_d} is the weight matrix of the LSTM model; and h is the output scalar of the LSTM model.

(3) We use a one-unit dense layer with the $sigmoid$ activation function to make binary classification (real or synthetic) on the scalar output:

$$O_d = D^{bc}(h; W_{bc}) \tag{11}$$

Where D^{bc} is the one-unit dense layer with a $sigmoid$ function used to make binary classification, and W_{bc} is its weight matrix; O_d is the final output of the discriminator.

2.3 TrajLoss for Measuring Trajectory Similarity Losses

The original GAN is designed to optimize the following objective function [6]:

$$O(D, G) = \min_G \max_D (\mathbb{E}_{x \sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))]) \tag{12}$$

Where $p_{data}(x)$ represents the distribution of the real data samples; $p_z(z)$ represents a prior on noise variables; $D(x)$ represents the probability that x came from $p_{data}(x)$; $G(z)$ represents a mapping from $p_z(z)$ to $p_{data}(x)$. The generator aims to minimize $\mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$ while the discriminator aims to maximize $\mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$, leading to a two-player minimax game.

According to the objective function $O(D, G)$, the loss function for the discriminator can be considered as a Binary Cross-Entropy (BCE) loss function (L_{BCE}), which will also be used in training the generator. However, different from the original GAN, we need the real trajectory data as inputs. Thus, we design a new loss metric function named TrajLoss to further measure the similarity losses between the real trajectory data and the synthetic trajectory data in spatial, temporal and categorical dimensions, and use this loss function to train the generator. The TrajLoss is defined as follows:

$$TrajLoss(y^r, y^p, t^r, t^s) = \alpha L_{BCE}(y^r, y^p) + \beta L_s(t^r, t^s) + \gamma L_t(t^r, t^s) + cL_c(t^r, t^s) \quad (13)$$

Where y^r and y^p represent the ground truth label and the prediction result of the trajectory by the discriminator, respectively; t^r and t^s represent the real trajectory and the corresponding synthetic trajectory; L_{BCE} is the original binary cross-entropy loss from the discriminator; L_s, L_t, and L_c are the spatial similarity loss, temporal similarity loss, and the categorical similarity loss between the real and synthetic trajectories, respectively; α, β, γ, and c are the weights for these losses and can be assigned differently for different scenarios.

In this paper, we use the L2 loss (i.e., least square errors) for L_s as a recent study [8] shows that the L2 loss is effective in measuring trajectory spatial similarity. Besides, we choose the Softmax Cross-Entropy (SCE) as the loss function for L_t and L_c since they are all regarded as multi-classification problems in this framework, and thus can be optimized using SCE. During the model training, the weights of the generator will be updated by the TrajLoss to improve the quality of the synthetic trajectory data.

3 Experiments

To address the abovementioned RQ1, this section first evaluates the effectiveness of trajectory privacy protection using the proposed LSTM-TrajGAN model on a classic LBS task: Trajectory-User Linking (TUL), which identifies users from trajectories and link trajectories to them [4]. TUL is an essential task in geo-tagged social media applications and receives increasing privacy concerns [4, 30, 21]. The evaluation can be regarded as an adversarial experiment: we train the LSTM-TrajGAN model and use the generated synthetic trajectories to suppress the accuracy of a state-of-the-art TUL algorithm. We also compare our approach with the other two commonly used location privacy protection methods: Random Perturbation and Gaussian Geomasking.

Meanwhile, to address the RQ2 for verifying the utility of the proposed model (i.e., the usefulness of the synthetic trajectories in analysis), we also explore the spatial and temporal characteristics of the synthetic trajectories to see if they preserve sufficient information from the original trajectories to further support spatial and temporal analyses.

3.1 Trajectory-User Linking

Dataset

We use the Foursquare weekly trajectory dataset in New York City (NYC) provided by Petry et al. [21], which is extracted from the Foursquare NYC check-ins dataset [29]. We only keep the user ID, trajectory ID, location, hour, day, and category attributes and remove

other attributes (e.g., price tier, rating, weather). The summary of the attributes is shown in Table 1. There are 193 users, 3,079 trajectories and 66,962 trajectory points in total in the dataset. We use 2/3 of the trajectories for training the LSTM-TrajGAN model and 1/3 for testing as suggested in [21].

Training and Evaluation

We train the LSTM-TrajGAN model on the training set for 2,000 epochs with several default training hyperparameters (e.g., we use an adam optimizer with a learning rate of 0.001 and set the batch size to 256). After the training process, the trajectory data from the test set as well as random noise are then used as the input of the generator to get synthetic trajectory data. A visualization example of a real trajectory from the test data and its corresponding synthetic trajectory generated by our model is shown in Figure 4 as a comparison. Next, we use the MARC (Multiple-Aspect tRajectory Classifier [21]), a start-of-the-art TUL algorithm, to perform the TUL task on both the test data and our synthetic data. Same as [21], we evaluate the TUL accuracy with five commonly used metrics: ACC@1 (Top-1 Accuracy, showing the model's ability to have the correct label to be the most probable label candidate), ACC@5 (Top-5 Accuracy, showing the model's ability to have the correct label among the top 5 most probable label candidates), Macro-P (Macro Precision, the mean precision among all classes), Macro-R (Macro Recall, the mean recall among all classes), and Macro-F1 (the harmonic mean of Macro-P and Macro-R). For comparison, we also evaluate the privacy protection effectiveness of Random Perturbation (spatial filter: within 1 km; temporal filter: within 24 hours) and Gaussian Geomasking (spatial filter: standard deviation = 0.001; temporal filter: within 24 hours).

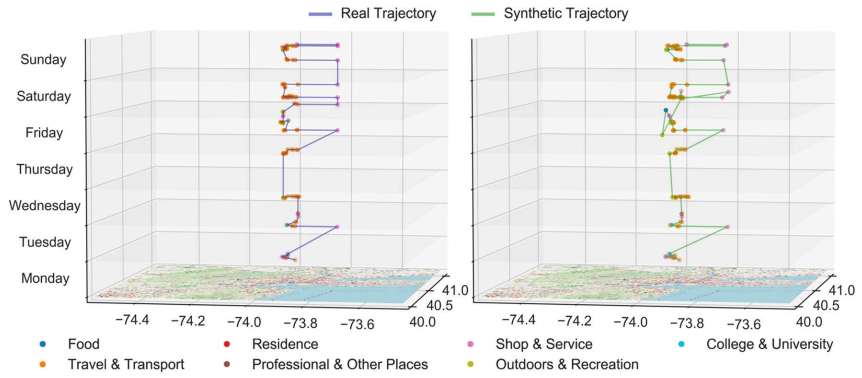


Figure 4 The visualization example of a real trajectory from the test data and its corresponding synthetic trajectory generated by our model.

The results are shown in Table 2. The higher the TUL accuracy, the worse the capability for trajectory privacy protection. One can conclude that the synthetic data generated by the LSTM-TrajGAN successfully suppress the scores in the four metrics (ACC@1, Macro-P, Macro-R, and Macro-F) from over 0.900 to around 0.400. The Top-5 Accuracy is decreased from over 0.976 to 0.722. The results show that our model can effectively prevent users from being identified by analyzing the trajectories. Additionally, Random Perturbation has limited effectiveness in protecting trajectory privacy regarding the TUL task, and Gaussian Geomasking works better while still has higher scores than our model. The results also indicate that leveraging both spatial and temporal dimensions of the trajectories simultaneously leads to better privacy-preserving performance than using only the spatial dimension.

Table 1 The summary of the Foursquare NYC weekly trajectory dataset.

Attribute	Type	Number / Range
Trajectory ID	integer	3,079
User ID	integer	193
Latitude	float	(40.550852, 40.988332)
Longitude	float	(-74.269644, -73.685767)
Hour	integer	24
Day	string	7
Category	string	10

Table 2 The privacy protection effectiveness of different privacy protection methods on the TUL task (RP stands for Random Perturbation; Gaussian stands for Gaussian Geomasking).

Method	ACC@1	ACC@5	Macro-F1	Macro-P	Macro-R
Original	0.938	0.976	0.925	0.937	0.927
RP (Spatial Only)	0.777	0.934	0.758	0.806	0.764
RP (Spatial-Temporal)	0.668	0.888	0.640	0.711	0.654
Gaussian (Spatial Only)	0.561	0.832	0.522	0.573	0.537
Gaussian (Spatial-Temporal)	0.486	0.766	0.431	0.488	0.470
LSTM-TrajGAN	**0.459**	**0.722**	**0.381**	**0.429**	**0.428**

3.2 Synthetic Trajectory Characteristics Analysis

Here, we analyze the spatial and temporal characteristics and other properties of the synthetic trajectories generated by the LSTM-TrajGAN to evaluate its utility (RQ2).

Spatial Characteristics

The spatial characteristics are explored based on two metrics: the Hausdorff Distance and the Jaccard Index. The Hausdorff Distance is a metric for measuring the distance between two point sets in a metric space and has been widely used for measuring the spatial dissimilarity between two trajectories. The Jaccard Index, also known as the Intersection over Union, is an efficient metric for measuring how much the two sample sets or regions overlap, and we use this to indicate the similarity of the activity spaces between two trajectories [18]. We calculate the Hausdorff Distance between each pair of the original and the synthetic trajectories. Likewise, we also calculate the Jaccard Index between the convex hulls of them since the convex hull can generally represent the activity space of LBS users [16]. Table 3 presents the summary of these metrics.

It shows that Random Perturbation has the smallest average Hausdorff Distance (0.004) and the largest average Jaccard Index (0.763), which makes sense since it only makes a limited influence on the spatial dimension of the trajectories. While such a method could preserve spatial similarity well, it sacrifices the location privacy. Our model performs better than Gaussian Geomasking on these two metrics and also better suppresses the abovementioned TUL metrics, which strikes a better balance between spatial similarity and location privacy.

Temporal Characteristics

We also explore the temporal characteristics based on the visualization of two summary indicators: temporal visit probability distribution for each POI category, and overall temporal visit frequency distribution. We count the frequencies of visits to each POI category at each

Table 3 Spatial characteristics evaluation based on Hausdorff Distance and Jaccard Index (RP stands for Random Perturbation; Gaussian stands for Gaussian Geomasking).

Method	Hausdorff Distance				Jaccard Index			
	Min	Max	Std	Mean	Min	Max	Std	Mean
RP	**0.001**	**0.006**	**0.001**	**0.004**	**0.000**	**0.977**	**0.194**	**0.763**
Gaussian	0.001	0.034	0.005	0.014	0.000	0.933	0.231	0.478
LSTM-TrajGAN	0.001	0.046	0.006	0.012	0.000	0.951	0.234	0.582

hour in original trajectories and the synthetic trajectories using three different approaches, and convert them into probability distribution matrices (Figure 5), in which the temporal patterns and the temporal similarity can be analyzed and compared.

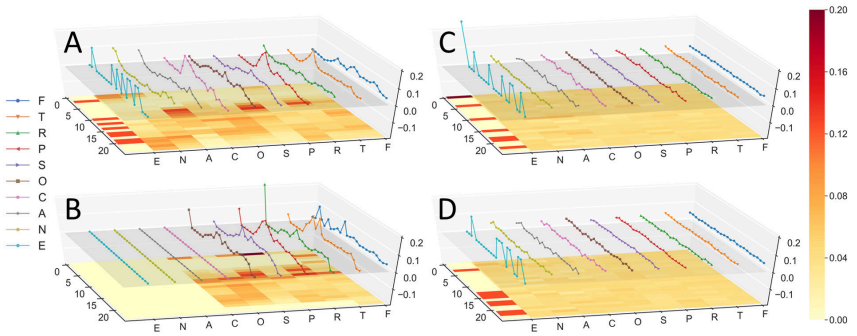
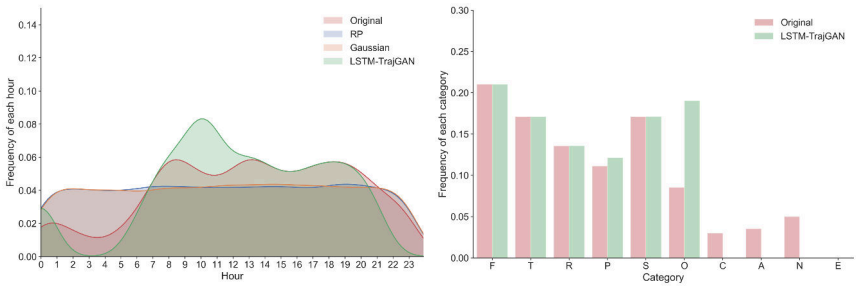


Figure 5 The hourly temporal visit probability distribution for each POI category by (A) Original data, (B) LSTM-TrajGAN, (C) Random Perturbation (within 24 hours), and (D) Gaussian Geomasking (within 24 hours) data (F: Food; T: Travel & Transport; R: Residence; P: Professional & Other Places; S: Shop & Service; O: Outdoors & Recreation; C: College & University; A: Arts & Entertainment; N: Nightlife Spot; E: Event).



(a) Overall temporal visit frequency distribution.

(b) Overall categorical visit frequency distribution.

Figure 6 Overall temporal visit frequency distribution and overall categorical visit frequency distribution (RP stands for Random Perturbation; Gaussian stands for Gaussian Geomasking).

It shows that the temporal visit probability distribution from LSTM-TrajGAN shares a large commonality with that from the original data, which embodies a significant temporal similarity. Some parts of the result by LSTM-TrajGAN (i.e., categories C and E) have near zero visit probability since these categories rarely appear in training data and thus the model doesn't learn sufficient information to make intelligent predictions on them. As

the comparisons, the temporal visit probability from Random Perturbation and Gaussian Geomasking show neither temporal similarity with the original data nor significant temporal patterns over 24 hours (except for the Event category).

Besides, we investigate the overall temporal and categorical visit frequency distribution (Figure 6a and Figure 6b). The overall temporal visit frequency distribution from our model can better fits the original data (Pearson Coefficient: 0.761) than Random Perturbation (0.536) and Gaussian Geomasking (0.535). The overall categorical visit frequency distribution also fits well (0.889). Hence, we conclude that our model generally well preserves both temporal and categorical characteristics.

4 Discussion

This section discusses the factors that may affect the privacy protection effectiveness of the LSTM-TrajGAN model, and the trade-off between the privacy protection effectiveness and the utility. Finally, we discuss the limitations of our approach.

4.1 Factors Affecting Privacy Protection Effectiveness

Training and Optimization Settings

We first explore how the different learning rates, loss metric functions, and random noise data affect the metric scores in the TUL task compared with the baseline setting (i.e., the learning rate = 0.001; the spatial dimension = 64; and the TrajLoss metric function during training). As shown in Table 4, different random noise data have small influences on the metrics, which in fact contributes to the potential generalizability of the proposed approach for generating privacy-preserving trajectory data. We also found that the selection of the learning rate may have a great influence on the metrics. A higher learning rate (0.002) makes the model converge faster, generating the synthetic trajectories that have less uncertainty and share more characteristics with the original trajectories, leading to higher TUL metric scores and vice versa. Although this is not always the case, the learning rate should be carefully set to balance the trajectory utility and privacy protection effectiveness.

Table 4 The metrics in the TUL task based on the synthetic trajectories by LSTM-TrajGAN using different training and optimization settings as well as different spatial embedding dimensions.

LSTM-TrajGAN	ACC@1	ACC@5	Macro-F1	Macro-P	Macro-R
Baseline	0.459	0.722	0.381	0.429	0.428
Different Random Noise	0.466	0.741	0.398	0.451	0.436
Higher Learning Rate (0.002)	0.841	0.959	0.824	0.855	0.828
Lower Learning Rate (0.00002)	0.055	0.157	0.029	0.047	0.054
Higher Spatial Dimensions (128)	0.510	0.811	0.504	0.513	0.513
Lower Spatial Dimensions (32)	0.426	0.703	0.396	0.402	0.392
TrajLoss without Spatial Loss	0.047	0.176	0.030	0.037	0.042
TrajLoss without Temporal Loss	0.093	0.252	0.076	0.119	0.089
TrajLoss without Categorical Loss	0.354	0.623	0.311	0.386	0.346
No TrajLoss	0.010	0.032	0.002	0.001	0.007

In addition, we also investigate how the TrajLoss metric function contributes to the training. When removing the Spatial Loss or the Temporal Loss from the TrajLoss function, the metric scores fall dramatically, implying that the synthetic trajectories fail to preserve

the spatial or temporal characteristics of the original trajectories. By comparison, removing the Categorical Loss only has a limited impact on the metric scores. Not surprisingly, removing the whole TrajLoss function results in losing spatiotemporal characteristics and thus getting the lowest TUL metric scores. We conclude that the spatial and the temporal dimensions represent the essential characteristics of a trajectory and hence need to be taken into consideration explicitly in the privacy protection approaches.

Spatial Embedding

Since the embedding of temporal attributes and categorical attributes is based on their vocabulary sizes, we mainly discuss the spatial embedding. The commonly used methods for spatial embedding are Multilayer Perceptron (MLP) and the Geohash algorithm. For example, Gupta et al. [8] use a MLP to embed the location of each person to obtain a fixed-length vector and use the vector as the input for an LSTM model to generate human trajectory. Petry et al. [21] introduce a binary Geohash algorithm, in which they first use the Geohash algorithm to divide the area into grid cells and then encode the latitude and longitude as a character string, and finally convert the string into a binary fixed-length vector as the representation for the spatial dimension of each trajectory point.

We use MLPs in the generator and the discriminator to embed the spatial dimension, but we implement them in a different way. Instead of directly embedding the coordinates, we first derive the deviations of latitudes and longitudes from the centroid of all trajectory locations, and then we embed these deviations into 64-dimensional vectors using MLP. There are two considerations: (1) On the one hand, unlike the trajectory classification task in [21], our goal is to generate synthetic trajectories, which means we need to decode the coordinates out from the hidden features in the model, and therefore using binary Geohash may lead to difficulties in learning the valid representation of coordinates, in designing the proper spatial loss, and in back-propagating the errors; and (2) On the other hand, unlike the restricted prediction area described by a Cartesian coordinate system in [8], the prediction area in our task is on the city scale, and the difference between two GPS coordinates only appears after the decimal point. It would be a grand challenge for the model to learn and predict the coordinates with only subtle changes. As such, we standardized the coordinates to make the difference between two locations more significant for the model to learn. Recent studies also indicate that scattering the locations based on deviations may help preserve privacy [5].

We also explore how the spatial embedding dimensions affect the metrics in the TUL task. As is presented in Table 4, embedding the location information into a vector with higher dimensions (e.g., 128) improves the TUL metric scores and vice versa. This makes sense since vectors in a higher-dimensional space are usually able to extract and embed more information than that in a lower-dimensional space. However, this also involves a trade-off between location accuracy and computational effort due to the limitation of physical devices.

4.2 The Trade-off between Privacy Protection Effectiveness and Utility

Generally speaking, specific trajectory analysis tasks may rely on different types of trajectory data (e.g., POI-based or road network-based) or different requirements (e.g., road extraction requires the location of each trajectory point to be precise), making it challenging to design a generic privacy protection method. However, we can evaluate a method by some specific criteria to determine its application scenarios, and even design a method based on this consideration to cover as many scenarios as possible. Inspired by the evaluation framework that involves the privacy, analytics, and uncertainty [5], we investigate the relationship

between privacy protection effectiveness and utility. Figure 7a demonstrates the performance of each method. It is worth noting that the placement of each method is estimated from our experiment. We believe that the consideration of this relationship would help choose and design proper trajectory privacy protection methods for specific scenarios.

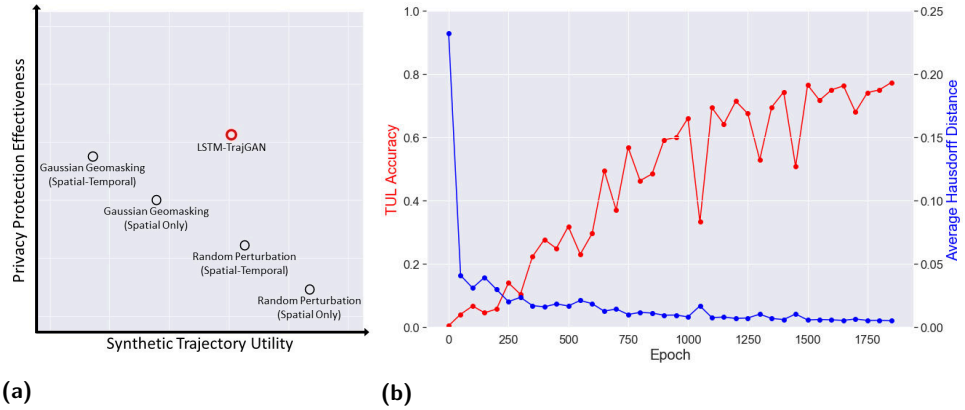


Figure 7 (a) The performance of each method in privacy protection effectiveness and utility; (b) The trade-off between the effectiveness of privacy protection (presented by TUL Top-5 Accuracy) and the preservation of spatial characteristics preservation (presented by Average Hausdorff Distance).

Sometimes the relationship between the privacy protection effectiveness and the utility is somewhat contradictory: we hope that the synthetic data are less similar to the original data to protect privacy while still preserving some similarities as good alternatives for spatiotemporal modeling or analyses. This may result in a "catch-22 situation". As an end-to-end deep learning model, the LSTM-TrajGAN is able to monitor and quantify this relationship during training and help to find the best-balanced parameter settings. For example, as training progresses, the TUL accuracy (Top-5 Accuracy) increases while the Average Hausdorff Distance decreases (Figure 7b). Carefully selecting the model weight from different epochs based on this relationship could ensure that the synthetic trajectories preserve spatiotemporal characteristics to some extent while maintaining a low TUL accuracy as needed, thereby balancing the privacy protection effectiveness and the utility of synthetic trajectories.

4.3 Limitations

Several limitations exist in our current approach. First, compared to traditional geomasking techniques that blur the existing trajectories, our deep learning model that generates new trajectories leads to a much higher computational effort and also needs an additional training process before its deployment in applications. Second, we focus on the TUL task and analyzed spatial and temporal characteristics of the synthetic trajectories, which reflects their potential for privacy-preserving trajectory analysis, but more specific evaluations are not investigated yet. Third, our model generates only the synthetic trajectories that have the same length as the original trajectories. Finally, our model currently focuses on city-scale trajectories, and the deviation-based location representation may not be suitable for global-scale trajectories. These limitations will be further explored in our future work.

5 Conclusion and Future Work

This research proposes a novel LSTM-TrajGAN approach, i.e., a deep learning model that combines the LSTM recurrent neural network and the GAN structure to generate privacy-preserving synthetic trajectories for trajectory data publication. We utilize the idea of adversarial training in the model design, train our model on a Foursquare NYC weekly trajectory dataset, and evaluate its privacy protection effectiveness in the TUL task. Regarding the two research questions we posed at the beginning of this research, the results show that (RQ1) our model can generate the spatial-temporal synthetic trajectories that prevent the trajectory creators (i.e., users) from being re-identified to certain degree and (RQ2) keep some spatial, temporal, and thematic characteristics of the original trajectories. Additionally, the results show that the model has the potentials for supporting further spatial or temporal analyses. Lastly, we explored the factors affecting the privacy protection effectiveness and discussed the trade-off between model effectiveness and utility in general. The design of a new loss function TrajLoss offers new insights into the development of spatially explicit artificial intelligence techniques for advancing GeoAI [13].

Our future work will focus on improving the trajectory similarity loss metric function, extending our framework to global-scale trajectory datasets, generating custom variable-length synthetic trajectory data, exploring potential privacy attack and defense strategies, and evaluating the privacy protection effectiveness and utility of our model in other trajectory data mining and analysis tasks.

References

1 Marc P Armstrong, Gerard Rushton, Dale L Zimmerman, et al. Geographically masking health data to preserve confidentiality. *Statistics in Medicine*, 18(5):497–525, 1999.

2 Chi-Yin Chow and Mohamed F Mokbel. Trajectory privacy in location-based services and data publication. *ACM SIGKDD Explorations Newsletter*, 13(1):19–29, 2011.

3 Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3:1376, 2013.

4 Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang. Identifying human mobility via trajectory embeddings. In *International Joint Conferences on Artificial Intelligence*, pages 1689–1695, 2017.

5 Song Gao, Jinmeng 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.

6 Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

7 Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM, 2003.

8 Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.

9 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.

10 Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

11 Qunying Huang and David WS Wong. Modeling and visualizing regular human mobility patterns with uncertainty: An example using twitter data. *Annals of the Association of American Geographers*, 105(6):1179–1197, 2015. doi:10.1080/00045608.2015.1081120.

12 Qunying Huang and Yu Xiao. Geographic situational awareness: Mining tweets for disaster preparedness, emergency response, impact, and recovery. *ISPRS International Journal of Geo-Information*, 4(3):1549–1568, 2015.

13 Krzysztof Janowicz, Song Gao, Grant McKenzie, Yingjie Hu, and Budhendra Bhaduri. GeoAI: Spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond, 2020.

14 Carsten Keßler and Grant McKenzie. A geoprivacy manifesto. *Transactions in GIS*, 22(1):3–19, 2018.

15 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.

16 Jae Hyun Lee, Adam W Davis, Seo Youn Yoon, and K. G. Goulias. Activity space estimation with longitudinal observations of social media data. *Transportation*, 43(6):955–977, 2016.

17 Mingxiao Li, Song Gao, Feng Lu, Huan Tong, and Hengcai Zhang. Dynamic estimation of individual exposure levels to air pollution using trajectories reconstructed from mobile phone data. *International Journal of Environmental Research and Public Health*, 16(22):4522, 2019.

18 Xi Liu, Hanzhou Chen, and Clio Andris. trajGANs: Using generative adversarial networks for geo-privacy protection of trajectory data (vision paper). In *Location Privacy and Security Workshop 2018 in conjunction with GIScience '18*, pages 1–7, 2018.

19 Yu Liu, Xi Liu, Song Gao, Li Gong, Chaogui Kang, Ye Zhi, Guanghua Chi, and Li Shi. Social sensing: A new approach to understanding our socioeconomic environments. *Annals of the Association of American Geographers*, 105(3):512–530, 2015.

20 Gengchen Mai, Krzysztof Janowicz, Bo Yan, Rui Zhu, Ling Cai, and Ni Lao. Multi-scale representation learning for spatial feature distributions using grid cells. In *The Eighth International Conference on Learning Representations*. openreview, 2020.

21 Lucas May Petry, Camila Silva, Andrea Esuli, Chiara Renso, and Vania Bogorny. Marc: a robust method for multiple-aspect trajectory classification via space, time, and semantic embeddings. *International Journal of Geographical Information Science*, pages 1–23, 2020.

22 Mehmet Ercan Nergiz, Maurizio Atzori, and Yucel Saygin. Towards trajectory anonymization: a generalization-based approach. In *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, pages 52–61, 2008.

23 Ben Niu, Qinghua Li, Xiaoyan Zhu, Guohong Cao, and Hui Li. Achieving k-anonymity in privacy-aware location-based services. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 754–762. IEEE, 2014.

24 Balaji Palanisamy and Ling Liu. Mobimix: Protecting location privacy with mix-zones over road networks. In *2011 IEEE 27th International Conference on Data Engineering*, pages 494–505. IEEE, 2011.

25 Yoo Min Park and Mei-Po Kwan. Individual exposure estimates may be erroneous when spatiotemporal variability of air pollution and human mobility are ignored. *Health & place*, 43:85–94, 2017.

26 Dara E Seidl, Piotr Jankowski, and Ming-Hsiang Tsou. Privacy and spatial pattern preservation in masked GPS trajectory data. *International Journal of Geographical Information Science*, 30(4):785–800, 2016.

27 Zheye Wang, Xinyue Ye, and Ming-Hsiang Tsou. Spatial, temporal, and content analysis of twitter for wildfire hazards. *Natural Hazards*, 83(1):523–540, 2016.

28 Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng Jin. Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1241–1250, 2017.

29 Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):129–142, 2014.

30 Fan Zhou, Qiang Gao, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. Trajectory-user linking via variational autoencoder. In *International Joint Conferences on Artificial Intelligence*, pages 3212–3218, 2018.

31 Huaijie Zhu, Xiaochun Yang, Bin Wang, Leixia Wang, and Wang-Chien Lee. Private trajectory data publication for trajectory classification. In *International Conference on Web Information Systems and Applications*, pages 347–360. Springer, 2019.

Analyzing Trajectory Gaps for Possible Rendezvous: A Summary of Results

Arun Sharma[1]
Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN, USA
sharm485@umn.edu

Xun Tang
Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN, USA
tangx456@umn.edu

Jayant Gupta
Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN, USA
gupta423@umn.edu

Majid Farhadloo
Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN, USA
farha043@umn.edu

Shashi Shekhar
Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN, USA
shekhar@umn.edu

Abstract

Given trajectory data with gaps, we investigate methods to identify possible rendezvous regions. Societal applications include improving maritime safety and regulations. The challenges come from two aspects. If trajectory data are not available around the rendezvous then either linear or shortest-path interpolation may fail to detect the possible rendezvous. Furthermore, the problem is computationally expensive due to the large number of gaps and associated trajectories. In this paper, we first use the plane sweep algorithm as a baseline. Then we propose a new filtering framework using the concept of a space-time grid. Experimental results and case study on real-world maritime trajectory data show that the proposed approach substantially improves the Area Pruning Efficiency over the baseline technique.

2012 ACM Subject Classification Information systems → Data mining; Computing methodologies → Spatial and physical reasoning

Keywords and phrases Spatial data mining, trajectory mining, time geography

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.13

Funding This material is based upon work supported by the US Department Of Defense Grant No. HM04762010009 and National Science Foundation under Grants No. 1737633.

Acknowledgements We would also like to thank Kim Koffolt and spatial computing research group for their helpful comments and refinements.

1 Introduction

Given multiple trajectories which have gaps due to weak signals, instrument malfunction or malicious interference we find possible times and places where moving objects (e.g. ships) rendezvous or meetup. Figure 1 shows an example of the input and output. For simplicity, we are using one-dimensional geographical space along with the dimension of time. Object 1

[1] Corresponding author

is shown in blue and Object 2 is shown in red. The gaps are shown in a dotted form and lie between P3, P4 for blue and P4, P5 for red. Object 1 has a maximum speed of 1 and Object 2 has a maximum speed of 2. The output shows the candidate active volume (CAV) for each object. A CAV is the region in a gap that represents all the possible locations of the object during a missing time interval. The intersection of the two CAVs is the possible rendezvous region termed as a spatio-temporal intersection (STI).

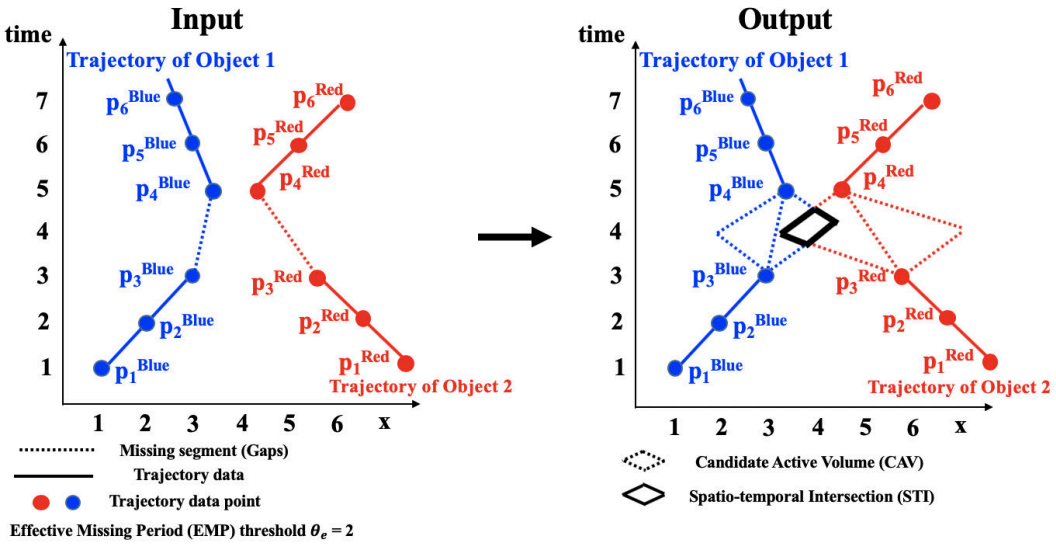


Figure 1 An illustration of rendezvous region detection (Best in color).

Analysis of gaps in trajectories has many societal applications related to maritime safety, homeland security, epidemiology, and public safety. For example, maritime safety and regulation enforcement are important for global security for concerns such as illegal oil transfer and trans-shipments. Such activities can be restricted and managed by identifying frequent missing signals from GPS trajectories of oil vessels with gaps. However, the trajectories can be spread over a large geographical space and manual inspection for gaps to detect rendezvous can be time intensive. Computational methods that detect possible rendezvous regions can substantially reduce the preliminary work for human analysts.

The problem is computationally challenging because the gaps may cause traditional trajectory mining approaches [27] to underperform or fail where they assume the availability and preciseness of trajectories. A second challenge is that the data has a large volume and is spread over considerable geographical space. For example, MarineCadastre [15] is an automatic identification system (AIS) dataset that contains records for more than 30 attributes (e.g., location, draught) for 150,000 ships taken every minute during the years 2009 to 2017. Its total size is about 600 GB and covers all the waters around the US.

The literature on movement pattern analysis [2] and trajectory mining [27, 3] interpolates gaps in trajectories without considering the full range of an object's movement possibilities. The approach provides an approximate solution and may miss possible rendezvous points. Areal interpolation of the gaps has been done through various prism models (e.g., space-time prism [16], kinetic prism [9]). Overlapping space-time prism for two or more objects can be thought of as a potential rendezvous region, or meeting place for moving objects. It may be computationally modeled as a spatio-temporal intersection of trajectories and can alternatively be called a spatio-temporal co-occurence.

In this paper, we study the computational cost of determining potential rendezvous regions. The baseline method uses a plane sweep [18] technique to identify the potential rendezvous regions. To improve the efficiency of plane sweep, we propose partitioning the space into space-time grids. The grids are used to cover all the possible paths the object can take. The grid overlapping two or more objects is a rendezvous region. The proposed framework ensures completeness by finding all the possible rendezvous regions, and ensures correctness because the rendezvous point is bound within the region. We further reduce the geographical search space through time slicing techniques. We use plane-sweep as the baseline to compare our results. Experimental results show that the proposed approach gives tighter bounds. Further, results from time-slicing techniques improve as we increase the time-slicing factor or use a finer time scale.

Contributions.

- We formally define the problem of rendezvous region detection for spatio-temporal trajectories with gaps.
- We propose a space-time partitioning approach to detect rendezvous regions. The approach is further refined to give more accurate approximation using time-slicing techniques.
- We propose and use a new evaluation metric, area pruning effectiveness (APE), to compare the methodologies.
- We compare the proposed approach with the plane sweep based baseline on various relevant evaluation parameters (e.g., study area) and metrics. Results (Section 4.2 show that the proposed approach has better APE values.
- We provide a case-study on ship trajectories from the Bering Sea to show the effectiveness of the proposed approach on a real-world dataset. We find that the proposed approach gives better results on the study area.

Scope. In this work, we do not study kinetic prisms [9]. Further, the proposed framework has multiple phases and we limit this work to the filter phase. The refinement phase requires input from a human analyst and is not addressed in this work. Furthermore, the calibration of cost model parameters is outside the scope of this work. In addition, we do not model rendezvous areas of trajectories without gaps which are involved in intersection. Finally, we do not address the issue of positional accuracy while modeling the trajectories.

Organization. The paper is organized as follows: Section 2 introduces basic concepts and the problem statement. Section 3 describes the proposed framework and approach used in our work. Experiment design, results, and a brief discussion on the computation cost of our approach are reported in Section 4. Section 5 reviews the related work (in more detail). Finally, Section 6 concludes this work and briefly lists future work.

2 Basic Concepts and Problem Statement

2.1 Basic Concepts

This section reviews several key concepts in the rendezvous detection problem and presents a formal problem formulation.

▶ **Definition 1.** *A **study area** is a two-dimensional rectangular area where the input data are located. It usually complies with the (latitude, longitude) coordinate system.*

▶ **Definition 2.** *A **spatial trajectory** is a trace generated by a moving object in a geographic space, that is usually interpreted as a series of chronologically sorted points, for instance, $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$, where each point ($p_i$) is associated with a geospatial coordinate set (x, y) and a time stamp (t).*

▶ **Definition 3.** *An **object maximum speed** ($Smax$) is the maximum speed of an object based on the domain knowledge.*

For maritime data, S_{max} can be identified from publicly available vessel databases [15]. For vehicles, humans, or animals, we can use the maximum physically allowed speed.

▶ **Definition 4.** *An **effective missing period (EMP)** is a time period when the signal is missing for longer than a user-specified EMP threshold (θ_e) .*

As shown in Figure 1 the EMPs for Object 1 and Object 2 is between timestamps 3 and 4, Here, we assume $\theta_e = 2$.

▶ **Definition 5.** *A **candidate active volume (CAV)** is the spatio-temporal volume where an object is possibly located during an EMP [4, 8, 11, 16, 17]. A CAV is based on a space time prism using conical shape and is derived from an EMP.*

2.2 Problem Formulation

The problem to identify optimized rendezvous patterns in a spatio-temporal domain is formulated as follows:

Input:

1. A study area S,
2. A set of $|N|$ trajectories $T = t_1 \ldots t_{|N|}$, each associated with an object,
3. An object maximum speed (S_{max}) for each object,
4. An effective missing period threshold (θ_e).

Output: Approximate geometry intersection of two gaps.
Constraints: Minimal Filter Storage Cost.
Objective: Improve area pruning effectiveness (APE)

For example, Figure 1 illustrates the one-dimensional representation of two gaps involving a spatiotemporal intersection given the study area (one-dimensional), two trajectories, two object maximum speeds, and $\theta_e = 2$. The output is the STI represented by the triangle shown in the figure.

Area pruning effectiveness (APE) is the ratio of the total study area and minimum bounded area inside the filtered region (i.e. approximate CAV).

$$area\ pruning\ effectiveness\ (APE) = \frac{total\ study\ area}{area\ bounded\ inside\ the\ region} \tag{1}$$

If the value of APE is higher, the solution quality is better since the minimum bounded area enclosed within a CAV will be lower.

3 Approach

We begin with an overview of our framework. Then we describe the baseline algorithm (plane sweep), a naive Spatio-temporal Grid Traversal (SGT), and the proposed Spatio-temporal Grid Traversal algorithm with time slicing (SGT-TS) in detail with their corresponding execution trace.

3.1 Framework

Our aim is to identify possible rendezvous regions on a given set of trajectories through a two-phase *Filter* and *Refine* approach. Figure 2 shows a representation of the proposed framework. The framework includes three algorithms: a baseline plane sweep algorithm [18], a naive spatio-temporal grid (SGT) algorithm, and a spatio-temporal grid algorithm with time slicing (SGT-TS). Plane sweep [18] is a basic computational geometry concept for finding intersections (e.g., line segments, polygons). We used the plane sweep algorithm to extract a minimum orthogonal bounding region (MOBR). SGT partition the study area into 3-dimensional (3D) grid cells (x,y, and time), where we approximate two endpoints based on the maximum speed for each trajectory, and in total four endpoints including the starting, and ending points of a gap segment that illustrates a possible rendezvous region. SGT-TS adds a time-slicing technique for selecting a more accurate region that helps reduce data redundancy and storage cost. The output from the filter phase is given to the refinement phase where we can find the exact geometry of the cone intersect using accurate modeling of the space-time prism of each object. The exact geometry can then be used by human analysts for ground truth verification via satellite imagery.

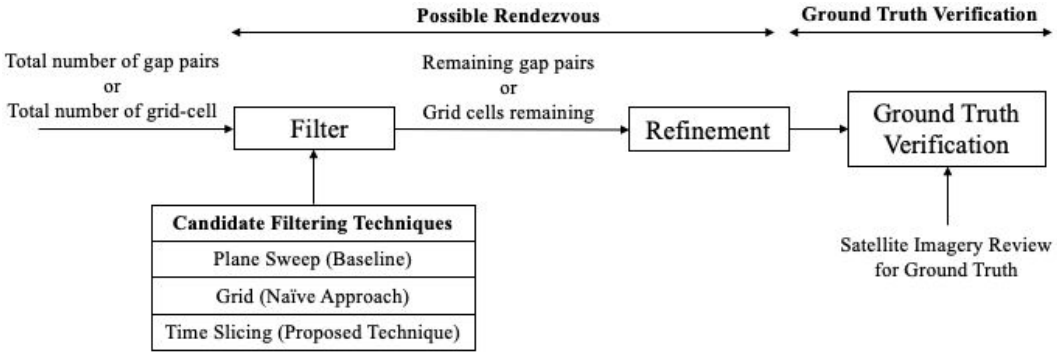


Figure 2 Framework for detecting possible rendezvous regions to reduce manual inspection by analyst.

3.2 Baseline Algorithm

Gaps in a trajectory can be analyzed by computing the minimum orthogonal bounding region (MOBR) over the set of gaps for an individual trajectory. We use the plane sweep algorithm [18] for extracting MOBRs. It is a filter and refine approach [5] where the given study area is projected into a lower-dimensional space. In the filtering phase, all gaps are sorted based on x or y coordinates. Ordering on one dimension reduces the storage and I/O cost, and further allows the computation of intersections in a single pass. In the refinement phase, the gaps are extracted based on the start and end times of their respective effective missing periods (EMPs). The segments are further approximated using MOBRs over each candidate active volume (CAV). The following text describes the algorithm in detail.

Step 1: Sort the endpoints of all the effective missing periods (EMPs). First, we sort the endpoints of all EMPs based on one of the coordinates. An endpoint is represented by three coordinates, namely x, y, and time, and either x or y can be the sorting coordinate. For consistency, we use x throughout this paper.

Step 2: A plane orthogonal to the x-axis sweeps along the sorted EMPs. The second step conducts the sweeping. Imagine there is a plane parallel to the y-t plane and orthogonal to the x-axis sweeping from the low to the high end along x-axis. The sweeping plane stops at both start and end points of each EMP. Note that "start" and "end" refer to the order of sweeping, which is irrelevant to the temporal dimension. An Observed Object List is maintained to store CAVs being currently crossed by the sweeping plane.

When stopping at the start of an EMP, the algorithm first determines if the gap is larger than the given EMP threshold (θ_e). If it is, a new CAV is constructed for that object along with an approximate MOBR around the new CAV as discussed later in Section 3.3. The CAV along with its MOBR is then saved inside the observed list and a check is done to see if any other CAV inside the list is intersecting with the given CAV. If it is, a common MOBR around the pair of intersecting CAVs is added to the observed list as well as the output. The sufficient and necessary condition for the spatiotemporal intersection of two CAVs is explained in 3.3. On the other hand, when stopping at the end of an EMP, the algorithm removes all the STIs that involve the EMP from the list. Note that each of these CAVs has different corresponding time periods indicating when the possible rendezvous may happen. We introduce how to compute the time period in the following Section 3.3.

Line	Observed Object List
a (start of 1)	1
b (start of 2)	1, 2, <1, 2>
c (start of 3)	1, 2, <1, 2>, 3, <1, 3>, <2, 3>
d (start of 4)	1, 2, <1, 2>, 3, <1, 3>, <2, 3>, 4, <3, 4>, <2,4>, <1,4>
e (end of 1)	2, 3, <2, 3>, 4, <3, 4>, <2,4>
f (end of 2)	3, 4, <3, 4>
g (end of 4)	3
h (end of 3)	Empty

Outputs: <1, 2>, <1, 3>, <2, 3>, <2, 4>, <3, 4>, <1, 4>

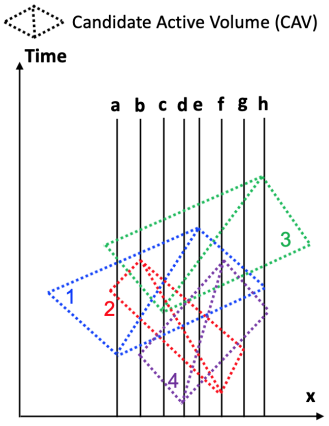
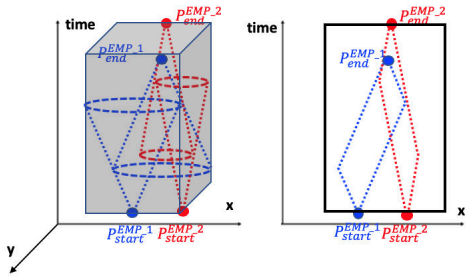


Figure 3 Plane sweep execution trace with a test case.

An execution trace of Plane Sweep. Figure 3 shows a dataset containing EMPs and their corresponding CAVs from four objects. For illustration, we simplify the study area into one-dimensional geographical space. A vertical line sweeps from left to right and stops at the endpoints (from a to h) of each EMP. The table on the left shows the elements in the observed object list after each stop. For example, when stopping at Line d (start of 4), the algorithm determines whether the incoming EMP $< 4 >$ intersects with any element in the observed object list, namely $< 1 >$, $< 2 >$, and $< 3 >$. Hence, $< 1, 4 >$, $< 2, 4 >$ and $< 3, 4 >$ are added to the observed list and the output list. When stopping at Line e (end of 1), the algorithm removes all the elements in the list involving EMP $< 1 >$ which includes $< 1, 2 >$, $< 1, 3 >$, and $< 1, 4 >$. The last stop is at Line h (end of 3). The Observed Object List becomes empty and the final output STIs include: $< 1, 2 >$, $< 1, 3 >$, $< 2, 3 >$, $< 2, 4 >$, $< 3, 4 >$, and $< 1, 4 >$.

3.3 Constructing a Minimum Orthogonal Bounding Region and Spatiotemporal Intersection

In section 3.2, we explained the intuition behind the baseline algorithm along with its corresponding execution trace. Now we explain in more detail the creation of candidate active volumes(CAVs) and new minimum orthogonal bounding regions (MOBRs). A loop goes over all the EMPs with gaps and checks if the given EMP is greater than the missing threshold. If it is, then a new CAV is constructed bounded by the coordinates attained through maximum speed and a check is done to see if any other CAV in the observed list intersects with the new given CAV. If it does, a new common MOBR is constructed around the intersection of the two CAVs using their maximum and minimum coordinates. The sufficient and necessary condition of whether two CAVs intersect is discussed later in this section.



	Bead 1 (blue)	Bead 2 (red)
Start	$P_{start}^{EMP\_1} = (x_{start}^{EMP\_1}, y_{start}^{EMP\_1}, t_{start}^{EMP\_1})$	$P_{start}^{EMP\_2} = (x_{start}^{EMP\_2}, y_{start}^{EMP\_2}, t_{start}^{EMP\_2})$
End	$P_{end}^{EMP\_1} = (x_{end}^{EMP\_1}, y_{end}^{EMP\_1}, t_{end}^{EMP\_1})$	$P_{end}^{EMP\_2} = (x_{end}^{EMP\_2}, y_{end}^{EMP\_2}, t_{end}^{EMP\_2})$
Object max speed	S_{max}^1	S_{max}^2

Figure 4 Cone Intersection of Effective Missing Period (EMPs).

As defined in Section 3.2, an effective missing period (EMP) is the intersection of two cones (i.e., a bead) vertexed at the endpoints of a CAV [26]. Therefore, in order to construct a new CAV during each iteration of this loop, we first need to determine the coordinates of the end points attained via maximum speed along with their respective radius. As shown in Figure 4, we first construct a CAV using maximum speed and radius where the radius of each cone at time t is the product of the object max speed S_{max} and the time difference from the start point of the EMP. For example, the radius of the cone vertexed at $P_{End}^{EMP_1}$ at time t: $r_{end}^{P1} = (t_{end}^{P2} - t) \times S_{max}^1$. Then, we find the end points by calculating the distance covered at maximum speed from each start point to its respective end point. This operation is done for both CAVs and creates a common MOBR using the extreme coordinates bounded by each CAV.

To compute the intersection between two EMPs during each iteration of this loop, we need to determine the intersections between four cones (i.e. two beads). We first check whether two beads have an overlapping time range. If not, then the beads are guaranteed to be not intersecting. Otherwise, the following geometric property is used to determine the intersection. Figure 4 shows two beads generated from two EMPs. EMP_1 starts at $P_{start}^{EMP_1}$ and ends at $P_{end}^{EMP_1}$, while EMP_2 starts at $P_{start}^{EMP_2}$ and ends at $P_{end}^{EMP_2}$. Each point is represented by three coordinates. For example, start point $P_{start}^{EMP_1} = (x_{start}^{EMP_1}, y_{start}^{EMP_1}, t_{start}^{EMP_1})$ is presented by two spatial coordinates $x_{start}^{EMP_1}$ and $y_{start}^{EMP_1}$ as well as the temporal coordinate $t_{start}^{EMP_1}$. The radius of each cone at time t is the product of the object max speed S_{max} and the time difference from the start point of the EMP. For example, the radius of the cone vertexed at $P_{end}^{EMP_1}$ at time t: $r_{end}^{EMP_1} = (t_{end}^{EMP_1} - t) \times S_{max}^1$. Now, we formulate the sufficient and necessary condition for two beads intersect as follows:

$$r_{start} + r_{end} \leq dis(start, end), \tag{2}$$

where index $start = \{start^{EMP_1}, start^{EMP_2}\}$, index $end = \{end^{EMP_1}, end^{EMP_2}\}$, and $dis_{start,end}$ is the Euclidean distance between points P_{start} and P_{end}. Appendex A includes a detailed proof for Equation 2.

3.4 Spatio-temporal Grid Traversal Algorithm

Computing possible rendezvous regions is challenging due to the high computational cost over a large set of trajectories as discussed in Section 1. The plane sweep algorithm provides an axis parallel MOBR which reduces the search space for finding the possible rendezvous regions for the refinement phase. However, this baseline approach proves to be inefficient when pruning a set of gap pairs both having higher positional displacement and EMPs. This may result in the construction of a common MOBR with size similar to the entire study area even if the actual rendezvous region between the gap pairs is relatively small compared to their respective MOBRs. Thus, we propose a spatio-temporal grid traversal algorithm (SGT) that aims to identify possible rendezvous regions using location, time and maximum speed.

Spatio-temporal grid traversal (SGT) is based on the idea of a 3D filtering technique by leveraging spatiotemporal properties and additional attributes of the space-time prism model to get a better geometric approximation of a bounded region. SGT starts by creating a spatiotemporal grid and applying the baseline approach for constructing CAVs of incoming gaps followed by the a common minimum object bounded rectangle (MOBR). However, inside each MOBR, we compute the linear bounds of the cones generated from the start and end point of the individual gaps and determine which grid cells reside inside the linear bounds of each CAV by checking each of the cell's corner points. Then we check the common cells residing in both CAVs. This operation is performed for every slice in the third dimension. Algorithm 1 shows the pseudo-code for SGT.

Step 1: Create common minimum orthogonal bounding rectangles (MOBRs). First, we create a spatio-temporal grid having the size of the given study area and a specified spatial and temporal resolution. Then, we apply the plane sweep algorithm to get common MOBR around gap pairs as described in Section 3.3 and save them in a common MOBR list.

Step 2: Create linear bounds within each common MOBR. For each common MOBR inside the common MOBR list, we index its endpoints inside the spatio-temporal grid along with the start and end points of the individual gaps to their respective nearest cell. Next we derive CAV linear bounds from the start point and end point of each gap based on the object's maximum speed. The linear bounds can be defined as the geometric interpretation of the slant height of the cone derived from the object's maximum speed. The maximum speed provides the slope i.e. the angle between the slant height and the time axis.

Step 3: Filter remaining grid cells qualified within CAVs. The filtering step linearly checks whether each cell inside the MOBR resides in the given CAV linear bounds. The linear bounds can be further divided into lower bounds and the upper bounds which can be derived from the start and end points of the gap respectively using object's maximum speed. In order to satisfy this condition, at least one of the corner points should be positioned higher than the lower bound but lower than the upper bound for each individual CAV. During filtering, we further refine the intersection by concurrently checking if any of those cells reside in both the CAVs. If they do, we filter out the remaining cells into the output list.

Algorithm 1 Spatio-temporal Grid Traversal (SGT).

1: $Spatial\ Resolution \leftarrow M$
2: $Temporal\ Resolution \leftarrow T$
3: Create Spatiotemporal Grid
4: $CMOBR\ list \leftarrow \emptyset$
5: $Output \leftarrow \emptyset$
6: Apply Plane Sweep Algorithm for Commom MOBRs
7: $CMOBR\ list \leftarrow Common\ MOBRs$
8: **for each:** $CMOBR_i$ *in* $CMOBR\ list$ **do**
9: $\quad Index\ CMOBR_i\ over\ Spatial\ Temporal\ Grid$
10: $\quad Calculate\ Linear\ Bounds\ of CAVs$
11: $\quad$ **for each:** $Cell_i$ *inside Common MOBR* **do**
12: $\quad\quad$ **if** $Cell_i$ *resides in both CAVs* **then**
13: $\quad\quad\quad Cell_i \rightarrow Output\{\}$
14: $\quad\quad$ **end if**
15: $\quad$ **end for**
16: **end for**

Time Slicing. Time slicing is an intermediate filtering phase which bounds each cone slice by a rectangle which is tighter than the corresponding slice of the space-time grid, thereby achieving higher efficiency. Hence, increasing the number of slices greater than the spatial resolution extent results in finer pruning that filters out any extra space between the CAV bounds and MOBR. Algorithm 2 provides a modification of step 2 in Algorithm 1 where we increase the temporal resolution greater than its respective spatial resolution.

Algorithm 2 Spatio-temporal Grid Traversal with Time Slicing (SGT-TS).

1: $Spatial\ Resolution \leftarrow M$
2: $TemporalResolution \leftarrow t > T$
3: Run Algorithm 1 (line: 3 to 16)

Execution trace of SGT and SGT-TS. Figure 5 shows the execution trace of Algorithms 1 and 2 over a 2D grid with 64 cells taking the x-axis as longitude (or latitude) and the y axis as time. For a given pair of CAVs, we first create a common MOBR and approximate its end points to their respective nearest cells. Figure 5(a) shows a 16x4 example grid with a CAV surrounded by its MOBR approximated by a grid where each grid cell is marked in yellow. During the filtering phase, SGT checks whether the corners of each cell qualify to be included in the given CAV. Figure 5 (b) shows the resulting grid after filtering where cells marked in red do not qualify and yellow cells represent the cells residing in the CAV. After getting all the cells inside the MOBRs, we check whether each remaining cell (yellow) resides in both the CAV pairs and output them as the final output. Figure 5 (c) shows the blue cells which take part in the intersection of two CAVs. For SGT with time slicing, we increase in temporal resolution providing a better filter as compared to the original SGT. Figure 5 (d) shows the final grid cells remaining in yellow and discarded cells in red. As compared to SGT, SGT-TS gives more refined results since more extra space has been discarded. Figure 5 (e) shows a greater number of intersecting cells, represented in blue, and indicating a better approximation of the intersection area.

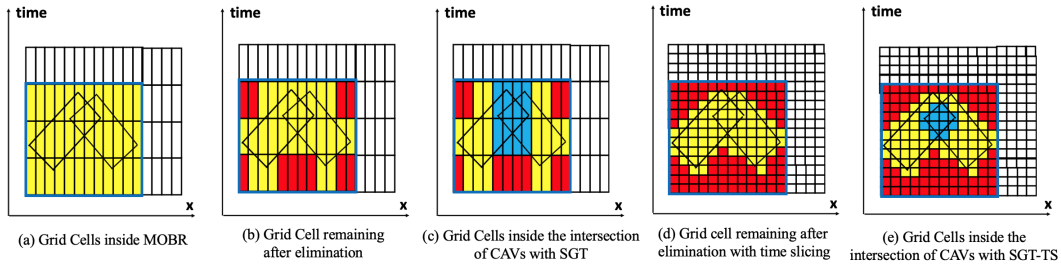


Figure 5 Execution Trace of SGT and SGT-ST.

4 Validation

In this section, we validated our approach using real world data in a case study as well as experimentally by varying different parameters such as study area and number of objects.

4.1 Experiment Design

Dataset. The dataset used in the experiments was MarineCadastre [15] which contains records of more than 30 attributes (e.g., Maritime Mobile Service Identity (MMSI), longitude, latitude, speed over ground (SOG), course over ground (COG) etc.) for 150,000 objects (i.e. ships) taken every minute from 2009 to 2017. The dataset is based on the WGS 1984 coordinates system with a geographical extent of 180W to 66W degrees in longitude and 90S to 90N degrees in latitude covering waters around the US.

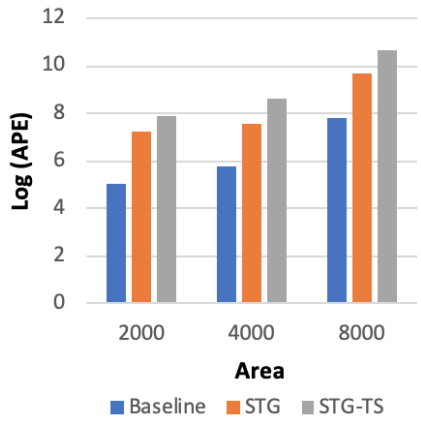
Experimental goal. The goal was to evaluate the performance of the proposed baseline, SGT and SGT-TS under different parameters. Our research questions are as follows:
(1) How does the size of the study area affect the APE ? and (2) How does an increase in the number of objects affect the APE on a given study area ?
Our evaluation metric was area pruning effectiveness (APE) which is the ratio of the total study area and minimum bounded area inside the filtered region (i.e. approximate CAV) as discussed in Section 2.2.

Computing Resources. All the experiments were conducted using Python and performed on an Intel Core i5 2.5GHz CPU and 16GB memory.

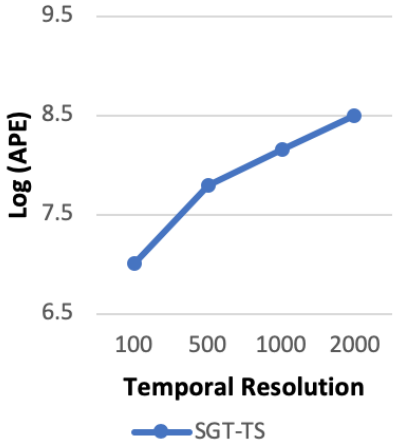
4.2 Experimental Results

Effect of the size of study area. In this experiment, we tested three study area sizes: $2000km^2$, $4000km^2$ and $8000km^2$. The number of GPS points varied for each study area as 1.5×10^5, 3×10^5, and 6×10^5. The density of GPS points remained consistent for the different areas. The results in Figure 6a show that SGT and SGT-TS are always more accurate than the baseline, especially as the study area increases. Figure 6b shows the APE values for SGT-TS with increasing time-slicing factor. Again, the SGT-TS algorithm outperforms the baseline with the APE improving as the time-slicing factor is increased.

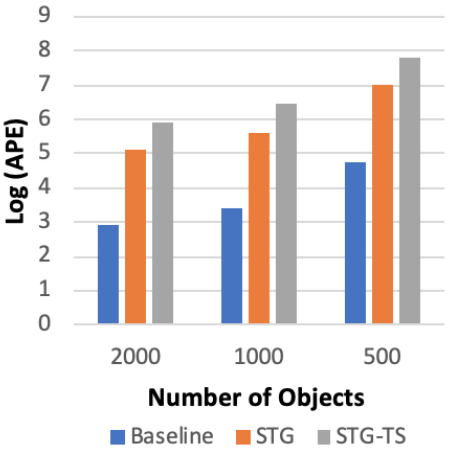
Effect of the number of objects. In this experiment, we set the study area size to $8000km^2$ and varied the number of object pairs (i.e. ships) from 500 to $2,000$. We also varied the number of GPS points respectively from 2.5×10^4 to 4.5×10^4. This was realized by picking



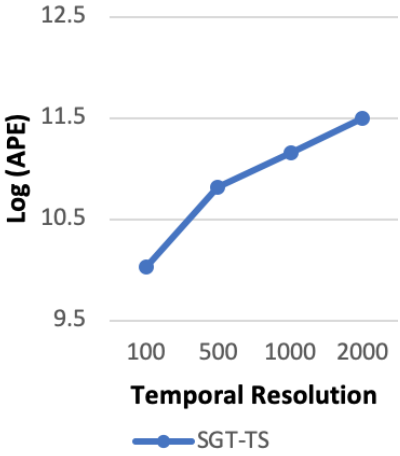
(a) Comparison based on Study Area.



(b) Change in Time Slicing (8000 km^2).



(c) Comparison based on Object number.



(d) Change in Time Slicing (300 obj.).

Figure 6 Effects on area pruning effectiveness (APE).

same study area from the original dataset for each varied number of object pairs resulting in different density of GPS points. The results in Figure 6c show that the increase in average pruning effectiveness (APE) is significantly greater in SGT and SGT-TS as compared to plane sweep for different number of objects. Figure 6d shows the further improvement of the APE as we increase the time-slicing factor.

4.3 Interpretation of Experimental Results

Scanning an entire study area can be exponentially expensive in terms of computational cost and human effort. The filter phase provides approximate regions within a trajectory gap for filtering possible rendezvous regions. However, the refinement phase can be expensive due to uncertainty in modeling the exact geometry of cones. Exact geometry involves inclusion of many real world physics based parameters (e.g. speed, acceleration) [9] which add complexity due to the need to solve quadratic equations. Hence, our main intuition is to reduce the total refinement cost in terms of computation of each cell per unit area by providing a tighter and

more accurate filter in the filtering phase. Equation 3 shows the relationship of filter and refinement in terms of computational cost.

$$C_f + C_r = C_t, \tag{3}$$

where C_f is the cost of filtering, C_r is the cost of refinement and C_t is the total cost to prune the given study area. The cost of refinement C_r decreases as we increase the filtering efficiency which often requires high computational cost in terms of prepossessing, model refinement, etc. However, if we are not considering any filtering then C_r will be equal to C_t.

4.4 Case Study on Real Automatic Identification System (AIS) data

We conducted a case study on data from MarineCadastre, a popular real world AIS dataset [15] to find possible rendezvous regions using the algorithms proposed in the paper. The approaches were applied on a study area ranging from $179.9W$ to $171W$ degrees in longitude and from $50N$ to $58N$ degrees in latitude in the Bering Sea, shown in Figure 7 (a). The dataset contained ~1.4×10^6 GPS readings from 72 ships that traveled during January, 2014. The EMP threshold (θ_e) was set to 30 minutes by which only the top 0.5% longest missing periods were considered as EMPs. We focused on various different trajectories that formed two STI clusters, one near Idak, and Atka Islands. These clusters accord with reports by the Marine Traffic Agency [24] that near an island, AIS systems tend to switch back and forth between terrestrial-AIS and satellite-AIS. Ships moving across the boundary of the effective zone may put out weak and unstable signals. The STI clusters we identified, which are near islands, likely represent such areas of weak signaling. Figure 7 (b) shows the zoomed in region near Atka Island where we selected to study our case. Figure 7 (c) shows the voyage of two vessels in that region categorized by a unique identifier (MMSI) each represented by a unique color indicating the start and end points of the EMPs. Figure 7(d) shows the output of the plane sweep algorithm, which construct a common MOBR around the CAV's (in green). Figure 7 (e) shows the common region (shown in red) after applying SGT and SGT-TS inside the MOBR where both the CAVs intersect, providing a significantly smaller region and better area effective pruning (APE) than the baseline approach.

5 Related Work and Limitations

Due to recent advancements in location-acquisition services, and mobile computing research, an extensive amount of trajectory data is available which serve different research purposes such as pattern recognition, anomaly detection, etc. The work in [27] provides a comprehensive survey of trajectory data mining and also explores the connection between different research topics and existing methodologies. Reconstruction techniques in [27] illustrate a variety of frameworks for modeling uncertainty and noise in trajectory data. However, all the techniques are based on assumptions related to linear interpolation or shortest path discovery. These approaches are not designed to detect patterns when the trajectories of the moving objects are missing (e.g., due to weak signaling) in which case the objects possibly move far from the shortest path. Trajectory data mining has also motivated interdisciplinary research in other fields such as geography and ecology. In [2], the authors provide a unified taxonomy of moving objects concerning their movement patterns by classifying them into generic and behavioral patterns, which encompass patterns such as co-locations, co-occurrence, etc.

Movement behavior patterns such as evasive patterns are used to detect potential anomalies. Analyzing maritime trajectory data with gaps is a particular case of an evasive pattern. A recent survey, [21] provides a panorama of existing techniques to identify anomalous

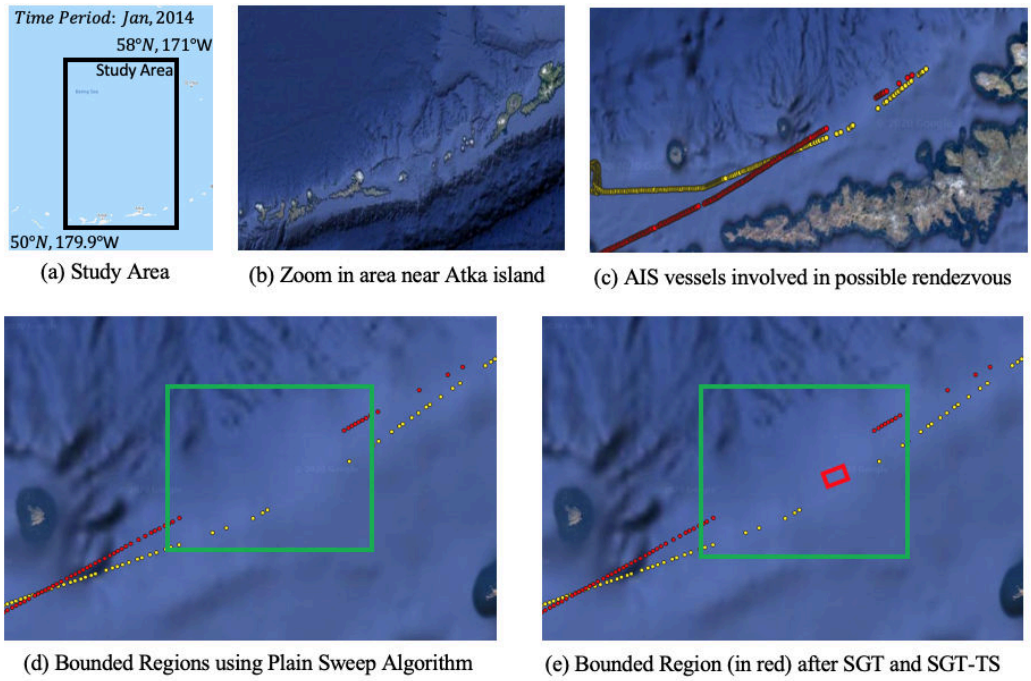


Figure 7 Spatio-temporal intersections detected in Bering Sea (best in color). The background imagery is not taken at the same time when the vessels were traveling in January 2014.

patterns in maritime trajectories by classifying them as data-driven, signature-based, and hybrid methods. Many frameworks [12, 1, 23, 13] have been proposed for analyzing evasive patterns in maritime trajectories. For instance, the authors in [13] proposed a method for determining if the vessel is anomalous by considering longitude, latitude, speed, and direction for each trajectory point and providing a three-division distance that can detect anomalous navigational behaviors. Stop and move [25] is an another conceptual model which analyzes anomalous behavior based on DBSCAN [19], and speed and direction [22], when it comes to ship trajectories data. These techniques do not apply to our work, which is focused on interpreting gaps in trajectory data, rather than detecting anomalous behaviour.

More realistic solutions for modeling gaps in trajectory data are contextual models such as space-time prisms [16, 6] that construct an areal interpolation of the gaps using coordinates and maximum speed of the objects. More recently, the kinetic prism model [9] provides a better estimation by considering other physical parameters such as uncertainty and acceleration. However, applying these models can be computationally expensive. One way to address the cost is through spatial indexing. Many spatial indexing techniques such as 3D R-Trees [28] or many others as described in [28], [20], [14] could be used to index trajectories efficiently. Other spatial indexing techniques, such as Hilbert Curve [20] have also been used to provide a computational speedup. The literature related to space-time prisms addresses computational speedup by using an alibi query for checking whether two space-time prisms intersect theoretically [10] which have also been applied in road networks [7]. In this work, we introduce the use of space-time prisms with grid-based indexing for detecting possible rendezvous patterns over maritime trajectories.

6 Conclusion and Future Work

In this paper, we introduced the problem of rendezvous detection in trajectory data with gaps. We proposed a baseline algorithm based on a plane sweep approach which first sorts and does a linear scan over a set of gaps and then provides a minimum orthogonal bounding region around the gaps. We proposed a spatio-temporal grid traversal (SGT) that provides tighter MOBRs, which in turn provides a more approximate shape of the candidate active volumes (CAVs). We further add efficient pruning based on time-slicing (SGT-TS) by adding a finer temporal resolution that gives a more accurate approximation bounded by the intersection of two cones. The results show relatively better area density ratio in SGT with time slicing (SGT-TS) as compared to SGT with significantly better APE when compared to the baseline.

Future Work. We plan to further implement the refinement phase where we refine the process of finding the exact geometry of spatiotemporal intersection since it is hard to find the exact geometry of the cone intersect due to its complexity. Computing approximate regions is very expensive in terms of time complexity and modeling them in regional space is challenging. Hence, we plan to further address the computational cost for extracting gaps and extend the proposed work in regional space as described in [10] [7]. We will also study if there an empirical threshold or ratio beyond which the gap may be too large to be meaningfully estimated via space-time prism. In addition, we will create synthetic dataset and then remove data points to make the data coarse for evaluating precision and recall with known rendezvous regions. Finally, we will analyze more interesting rendezvous patterns which involve more than two objects where the intersection of multiple space-time prisms takes place.

References

1 Elena Camossi, Paola Villa, and Luca Mazzola. Semantic-based anomalous pattern discovery in moving object trajectories. *arXiv preprint arXiv:1305.1946*, 2013.

2 Somayeh Dodge, Robert Weibel, and Anna-Katharina Lautenschütz. Towards a taxonomy of movement patterns. *Information visualization*, 7(3-4):240–252, 2008.

3 Emre Eftelioglu, Xun Tang, and Shashi Shekhar. Avoidance region discovery: A summary of results. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 585–593. SIAM, 2018.

4 Kathleen Hornsby and Max J Egenhofer. Modeling moving objects over multiple granularities. *Annals of Mathematics and Artificial Intelligence*, 36(1-2):177–194, 2002.

5 Edwin H Jacox and Hanan Samet. Spatial join techniques. *ACM Transactions on Database Systems (TODS)*, 32(1):7–es, 2007.

6 Hyun-Mi Kim and Mei-Po Kwan. Space-time accessibility measures: A geocomputational algorithm with a focus on the feasible opportunity set and possible activity duration. *Journal of geographical Systems*, 5(1):71–91, 2003.

7 Bart Kuijpers, Rafael Grimson, and Walied Othman. An analytic solution to the alibi query in the space–time prisms model for moving object data. *International Journal of Geographical Information Science*, 25(2):293–322, 2011.

8 Bart Kuijpers, Harvey J Miller, Tijs Neutens, and Walied Othman. Anchor uncertainty and space-time prisms on road networks. *International Journal of Geographical Information Science*, 24(8):1223–1248, 2010.

9 Bart Kuijpers, Harvey J Miller, and Walied Othman. Kinetic prisms: incorporating acceleration limits into space–time prisms. *International Journal of Geographical Information Science*, 31(11):2164–2194, 2017.

10 Bart Kuijpers and Walied Othman. Modeling uncertainty of moving objects on road networks via space–time prisms. *International Journal of Geographical Information Science*, 23(9):1095–1117, 2009.

11 Mei-Po Kwan. Gis methods in time-geographic research: Geocomputation and geovisualization of human activity patterns. *Geografiska Annaler: Series B, Human Geography*, 86(4):267–280, 2004.

12 Po-Ruey Lei. A framework for anomaly detection in maritime trajectory behavior. *Knowledge and Information Systems*, 47(1):189–214, 2016.

13 Bo Liu, Erico N de Souza, Cassey Hilliard, and Stan Matwin. Ship movement anomaly detection using specialized distance measures. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1113–1120. IEEE, 2015.

14 Ahmed R Mahmood, Walid G Aref, Ahmed M Aly, and Saleh Basalamah. Indexing recent trajectories of moving objects. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 393–396, 2014.

15 Marinecadastre.gov. URL: `https://marinecadastre.gov/ais/`.

16 Harvey J Miller. Modelling accessibility using space-time prism concepts within geographical information systems. *International Journal of Geographical Information System*, 5(3):287–301, 1991.

17 Tijs Neutens, Tim Schwanen, and Frank Witlox. The prism of everyday life: Towards a new research agenda for time geography. *Transport reviews*, 31(1):25–47, 2011.

18 Jürg Nievergelt and Franco P. Preparata. Plane-sweep algorithms for intersecting geometric figures. *Communications of the ACM*, 25(10):739–747, 1982.

19 Andrey Tietbohl Palma, Vania Bogorny, Bart Kuijpers, and Luis Otavio Alvares. A clustering-based approach for discovering interesting places in trajectories. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 863–868, 2008.

20 Dieter Pfoser, Christian S Jensen, Yannis Theodoridis, et al. Novel approaches to the indexing of moving object trajectories. In *VLDB*, pages 395–406, 2000.

21 Maria Riveiro, Giuliana Pallotta, and Michele Vespe. Maritime anomaly detection: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(5):e1266, 2018.

22 Jose Antonio MR Rocha, Valéria C Times, Gabriel Oliveira, Luis O Alvares, and Vania Bogorny. Db-smot: A direction-based spatio-temporal clustering method. In *2010 5th IEEE international conference intelligent systems*, pages 114–119. IEEE, 2010.

23 Hamed Yaghoubi Shahir, Uwe Glässer, Narek Nalbandyan, and Hans Wehn. Maritime situation analysis: A multi-vessel interaction and anomaly detection framework. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, pages 192–199. IEEE, 2014.

24 Katerina Sofrona. Why cannot i see a vessel on the live map?, October 2017. URL: `https://help.marinetraffic.com/hc/en-us/articles/203990958`.

25 Stefano Spaccapietra, Christine Parent, Maria Luisa Damiani, Jose Antonio de Macedo, Fabio Porto, and Christelle Vangenot. A conceptual view on trajectories. *Data & knowledge engineering*, 65(1):126–146, 2008.

26 Goce Trajcevski, Alok Choudhary, Ouri Wolfson, Li Ye, and Gang Li. Uncertain range queries for necklaces. In *2010 Eleventh International Conference on Mobile Data Management*, pages 199–208. IEEE, 2010.

27 Yu Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):1–41, 2015.

28 Qing Zhu, Jun Gong, and Yeting Zhang. An efficient 3d r-tree spatial index method for virtual geographic environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(3):217–224, 2007.

A Necessary and Sufficient Condition of Spatiotemporal Intersection

Lemma. Two beads must be intersected if and only if

$$r_{start} + r_{end} \leq dis(start, end) \tag{4}$$

where index $start = \{start^{EMP_1}, start^{EMP_2}\}$, index $end = \{end^{EMP_1}, end^{EMP_2}\}$, and $disstart, end$ is the Euclidean distance between points P_{start} and P_{end}.

Proof. For the necessary condition there must be at least one timestamp when the sections of these two beads intersect i.e. if two beads have an overlapping time range, at least one-time stamp (start-point or the end point) of one of the gap segments must be between the time range of the other. To prove this, we take two data gaps, EMP_1 and EMP_2 having time range (t^1_{start}, t^1_{end}) and (t^2_{start}, t^2_{end}) respectively and check if the difference between (t^1_{start}, t^2_{end}) or (t^1_{end}, t^2_{start}) ≥ 0. If true, then the two EMPs satisfy the necessary condition of a two EMP intersect.

For the sufficient condition, if there is one timestamp that is between the overlap of the time gaps, the two beads must intersect. In order to satisfy this condition, we use the radius information of two cones from a different gap and check whether their respective radii will overlap with each other. According to the condition whether two circles overlap, the sum of their radii must be smaller than the distance between their respective centers. As stated in Equation 4, the sum of the radius of the cone from start point r^1_{start} of EMP_1 and end point r^2_{end} of EMP_2 must be less than the distance between their respective radii centers $dist(s, e)$. Using known t and S_{max}, when $S^1_{max} \geq S^2_{max}$, Equation 4 is further derived into:

$$t \geq \frac{d_{s,e} + t_s \times S^1_{max} + t_e \times S^2_{max}}{S^1_{max} + S^2_{max}} \rightarrow \text{where } s = start^{EMP_1},\ e = start^{EMP_2} \tag{5}$$

$$t \geq \frac{d_{s,e} + t_s \times S^1_{max} - t_e \times S^2_{max}}{S^1_{max} - S^2_{max}} \rightarrow \text{where } s = start^{EMP_1},\ e = end^{EMP_2} \tag{6}$$

$$t \leq \frac{d_{s,e} + t_s \times S^2_{max} - t_e \times S^1_{max}}{S^2_{max} - S^1_{max}} \rightarrow \text{where } s = end^{EMP_1},\ e = start^{EMP_2} \tag{7}$$

$$t \leq \frac{d_{s,e} + t_s \times S^1_{max} - t_e \times S^2_{max}}{-S^1_{max} - S^2_{max}} \rightarrow \text{where } s = end^{EMP_1},\ e = end^{EMP_2} \tag{8}$$

In the case that $S^1_{max} < S^2_{max}$, the conditions are derived by swapping the variables.
If the conditions are all satisfied, we know these two EMPs intersect. In contrast, if any of the conditions is not satisfied, the two EMPs do not intersect.

You Are Not Alone: Path Search Models, Traffic, and Social Costs

Fateme Teimouri
Department of Computing Science, Umeå University, Sweden
https://www.umu.se/en/staff/fateme-teimouri/
fateme.teimouri@umu.se

Kai-Florian Richter
Department of Computing Science, Umeå University, Sweden
https://www.umu.se/en/staff/kai-florian-richter/
kai-florian.richter@umu.se

Abstract

Existing cognitively motivated path search models ignore that we are hardly ever alone when navigating through an environment. They neither account for traffic nor for the social costs that being routed through certain areas may incur. In this paper, we analyse the effects of "not being alone" on different path search models, in particular on fastest paths and least complex paths. We find a significant effect of aiming to avoid traffic on social costs, but interestingly only minor effects on path complexity when minimizing either traffic load or social costs. Further, we find that ignoring traffic in path search leads to significantly increased average traffic load for all tested models. We also present results of a combined model that accounts for complexity, traffic, and social costs at the same time. Overall, this research provides important insights into the behavior of path search models when optimizing for different aspects, and explores some ways of mitigating unwanted effects.

2012 ACM Subject Classification Human-centered computing; Information systems → Geographic information systems; Information systems → Location based services

Keywords and phrases wayfinding, navigation complexity, spatial cognition, social costs

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.14

Funding This research has been funded by the Swedish Research Council (Vetenskapsrådet) under grant 2018-05318.

1 Introduction

We are not alone in this world. This is not a new or surprising insight. However, if we look at cognitively motivated path-search models published in the literature (e.g., [4, 6, 7, 16]), there seems to be an underlying assumption that we are. They seem to take roads to be empty. These models account for all kinds of aspects that may cause navigation to become difficult, aiming for the least complex paths. But they ignore fellow travelers on the road. They do not account for varying degrees of traffic and the complexity such traffic may add to the navigation process – let alone the potentially significant increase in travel time.

Commercial navigation systems, which usually calculate the shortest or fastest route, do account for traffic and the delays it may cause. They adapt suggested routes according to changing situations on the road. Thus, they do not make the same "empty roads" assumption. However, they might make other assumptions of "being alone." Namely, they might ignore that for some parts of the road network social conventions tell that they are not meant for the general public to drive (or even walk) through even if there are no legal restrictions preventing this [9]. Ignoring these social conventions is a deficit commercial systems share with the research models.

In this paper, we will explore the effects of these assumptions of "being alone." That is, we will analyze what effects taking into account traffic and social costs have on the computed routes. In particular, we will test how the fastest and the least complex routes may change under avoiding traffic in terms of their complexity and violation of said social costs. We will also present results of a combined model, i.e., a model that accounts for complexity, traffic, and social costs at the same time. Accordingly, this paper offers important insights into the behavior of path search algorithms when optimizing for different aspects, and explores some ways of mitigating unwanted effects.

In the next section, we discuss relevant related work. We will then introduce the different path search models used in our analysis in Section 3. Methods and results of the analysis are presented in Section 4 and discussed in Section 5. Section 6 concludes the paper with suggesting some future work.

2 Related work: Cognitively motivated path search algorithms

Several different models have been proposed in the literature that adapt path search to factors of human cognition, preferences, and environmental layout in order to reduce navigation complexity. Roughly, these models can be divided into three categories: 1) choosing routes that are easiest to describe; 2) integration of and routing along landmarks; 3) adaptation to environmental structure.

The main focus of the first category is on simplifying the instructions needed to guide a wayfinder from origin to destination. These models are inspired by the fact that people often prefer to direct wayfinders along routes that are easy to describe instead of the shortest ones [13, 20], aiming to simplify (or minimize) the amount of information these wayfinders need to remember. For example, Duckham and Kulik [4] proposed the simplest paths algorithm, which essentially implements Mark's complexity model [13]. Mark's model assigns to each wayfinding action a number of required so-called slots to represent said action. Richter and Duckham [16] then took this approach further by employing more realistic instruction generation mechanisms – including references to landmarks and spatial chunking [10].

Models in the second category specifically focus on the integration of landmarks in calculated paths. They aim to exploit the importance of landmarks for human navigation in reducing wayfinding complexity. Richter and Klippel [17] proposed a methodology for generating easier-to-remember wayfinding instructions. For a given route through an environment, their method finds the minimal number of chunks, i.e., the minimal number of instructions, required to fully describe the route. The chunking mechanisms heavily rely on landmarks to anchor actions in space. Caduff and Timpf [1] introduced the landmark spider approach, which calculates paths through a network using edge weights that account for the presence of landmarks. Weights are computed based on the distance between landmark and wayfinder, the direction between landmark and wayfinder, and the salience of the landmark itself. As a consequence, the "shortest landmark spider" path, i.e., the one with the lowest costs, is a path that passes many relevant landmarks.

The third category of models focuses on the complexity of the environment, in particular the structure of decision points and the path network. Such models aim to avoid complex parts of an environment, and also to avoid ambiguity resulting from its structure. For example, Haque, Kulik and Klippel's model [7] computes instruction equivalence for the different turns at an intersection (e.g., two different turns may be both seen as "left"). In path search, the model minimizes ambiguity (or unreliability). Richter [15] proposed a regionalized path planning algorithm based on environmental structure and decision point complexity.

This model computes a complexity measure for each node of the path network. Then nodes are clustered into different regions based on complexity threshold values (a complex region and an easy region in the simplest case). The model allows different cost functions for each region, for example, shortest path for the easy region and simplest paths [4] for the complex one. Manley, Orr and Cheng [12] proposed a hierarchical route choice model using heuristic selection processes. Based on the idea of regionalized path planning [19], among others, the model first determines which regions to travel through. In a second step, this gets refined to major nodes to path through, and then which actual roads to take in the third step. The different selection processes make use of "human" heuristics, such as minimization of angular deviation.

As stated in the introduction, none of these approaches accounts for other people on the road, i.e., traffic, nor for social costs, i.e., avoiding to route through areas that are residential and not meant for higher traffic volumes. In an earlier study, Johnson et al. [9] found that scenic routing and – to a lesser degree – safe routing, i.e., optimizing paths for scenic routes or to avoid "unsafe" regions, leads to these routes becoming more complex, but also to redirecting traffic into areas that are not supposed to take high traffic volume, for example, parks or slower neighborhood roads. In a similar vein, in this paper we explore what it means that we are not alone in the world for different path search optimization criteria, particularly for fastest and least complex paths.

3 Path search models

In this section, we will present the different models that are used to calculate paths of varying kind. Specifically, we present a model that accounts for different aspects of wayfinding complexity (inspired by [6]). In addition, we present a model simulating different traffic loads in a road network, thus, allowing for calculating the least-traffic path, and a model that accounts for the previously discussed social costs of navigating urban environments. These different models can then be integrated into a combined model that allows for flexibly using just one, some, or all of these aspects (to varying degree) in computing the costs of traversing a road network. They are all based on Dijkstra's shortest path algorithm [3].

We are aware that some of these models use relatively simple heuristics. This is done because we are interested in showing the principal effects of the various parameters, rather than providing the most realistic modeling possible. Since all of the models, as well as the combined model, are modular it would be straightforward to replace some of the aspects with more complex models in the future.

3.1 Complexity model

In order to make paths as easy to follow or remember as possible, we need to know about the complexity of a path's components, particularly its decision points as here decisions about how to continue need to be made. Following [6], there are three categories of complexity factors (see Figure 1): 1) environmental complexity; 2) those related to how instructions are provided; 3) factors inherent to the wayfinders. Some of these factors are assigned to the decision points (nodes), some to the road segments (edges).

3.1.1 Environmental complexity

Already Lynch [11] considered environmental complexity, or legibility, an important factor for the ease of navigating an environment. In his empirical studies, Weisman [18] found a direct relationship between environmental legibility and wayfinding behavior. However, while

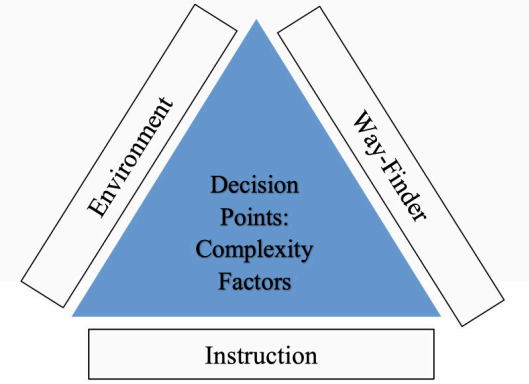


Figure 1 Three different categories of wayfinding complexity factors: environmental complexity; instruction complexity; factors inherent in the wayfinder.

environmental legibility is arguably the richest factor for environmental complexity, it is also the most poorly understood [14]. In our model, we approximate this factor by using three parameters based on [15] to capture environmental complexity:

- *Number of branches* is the number of road segments that meet at a decision point, i.e., the node degree in network terms. With an increasing number of branches it becomes more likely that a wayfinder takes a wrong turn at a decision point. Node degree is inherent to the nodes and, thus, a parameter assigned to the decision points.
- *Deviation from prototypical angles* is the deviation of a turn from 45 and 90 degree turns. Humans conceptualize turns usually as these prototypical angles. The larger the deviation from these prototypes, the more difficult it may become to identify the correct turn. We assign the mean deviation, i.e., the average over all turn angles between a decision point's branches, as a decision point parameter.
- *Road length* is the length of a road segment originating at the decision point at hand. With longer road segments, wayfinders travel further till the destination without the need to make decisions (i.e., until the next decision point), compared to many short segments, and consequently, the fewer chances they have to take a wrong turn. Length is inherent to a road segment and, thus, stored with an edge.

These factors are calculated as in the following. The number of branches at a decision point is simply the node degree of the corresponding node in the road network. The deviation from prototypical angles is computed as the difference of the *bearing* between the decision point at hand and the decision points at the other end of each branch:

$$\begin{aligned} bearing &= atan2(x, y) \\ x &= \cos(lat_1) * \sin(lat_2) - \sin(lat_1) * \cos(lat_2) * \cos(lon_2 - lon_1) \\ y &= \sin(lon_2 - lon_1) * \cos(lat_2) \end{aligned}$$

with lat_1, $long_1$ the latitude and longitude of the start node and lat_2, $long_2$ the latitude and longitude of the end node. Road length is the distance between the two nodes forming an edge (branch). It is computed as the following:

$$\begin{aligned} x &= \sin^2((lat_2 - lat_1)/2) + \cos(lat_1) * \cos(lat_2) * \sin^2((long_2 - long_1)/2) \\ y &= 2 * atan2(\sqrt{x}, \sqrt{1-x}) \\ distance &= radius * y \end{aligned}$$

where *radius* is earth's radius (mean radius = 6,371km).

3.1.2 Complexity related to instructions

Wayfinding instructions can be more or less helpful in finding the way, i.e., they may differ in their understandability and interpretability. Our model utilizes three factors to compute this complexity related to instructions: 1) instruction equivalence according to [7]; 2) the number of items to remember according to [4, 13]; 3) the presence of relevant landmarks according to [1].

- *Instruction equivalence* means how many turns at a decision point can be described with the same linguistic label. For example, in historic city centers with 6-way intersections, the instruction "turn right" may apply to several turns; there may be several roads that lead to the "right." Instruction equivalence is calculated by checking the bearing of all branches at a decision point for whether they are in the same quadrant of the bearing coordinate system, which we take as being instruction equivalent. We use quadrants instead of half-planes to allow for distinguishing different linguistic turn direction concepts (e.g., "veer left" vs. "sharp left"). This parameter is assigned to the decision points.
- *Instruction complexity* corresponds to the slot values as in [4]. These slot values reflect the complexity of performing (correctly) different navigation maneuvers, such as going straight (1 slot) vs. turning at a t-intersection (6 slots) vs. turning at a four-way (or more) intersection (5 + node degree slots). We compute the average instruction complexity for all possible turns at a decision point and assign this value to the corresponding node.
- *Landmark complexity* is computed as a combination of the distance between landmark and wayfinder, and the salience of the landmark itself (cf. [1]). All landmark objects in a radius of 50 meters or half of the length of the longest road segment – whichever is smaller – around a decision point are extracted. The distance to the decision point is multiplied by the landmark's salience value. The mean value for all landmarks is stored as the landmark complexity with the decision point at hand. The smaller this value, the more wayfinding is supported by landmarks at this decision point.

3.1.3 Wayfinder-related factors

Individual characteristics and differences among wayfinders is another important factor in wayfinding [2]. This factor relates to an individual's ability to, for example, stay oriented, build up a mental representation of an environment, or to understand instructions. In our model, we represent these individual differences with the Santa Barbara Sense of Direction (SBSOD) scale [8]. This self-report measure reliably captures people's spatial abilities. For people with a lower score, wayfinding is more difficult and, thus, their ability to correctly navigate complex decision points reduces. The SBSOD score is a number between 1 and 7, where 7 indicates high spatial abilities. We normalize the score to lie between 0 and 1 and take $1 - SBSOD$ as the complexity value, i.e., the higher somebody's spatial abilities, the less complex navigation is for them.

3.1.4 The final model

Table 1 provides a summary of the different parameters used in calculating the complexity of decision points.

All of these parameters get normalized to values between 0 and 1 in their computation. They are then combined in a weighted sum model as follows:

$$C_c = \frac{w_e * Complexity_e + w_w * Complexity_w + w_i * Complexity_i}{w_e + w_w + w_i}$$

Table 1 Parameters for computing decision point complexity.

Complexity factor	Parameters
Environment	Number of branches (node degree), Deviation from prototypical angles, Road length
Instruction	Instruction equivalence, Instruction complexity, Landmark complexity
Wayfinder	Santa Barbara Sense Of Direction

where w_e, w_w, w_i are the weights of the environmental complexity factor, wayfinder-related factor, and factor related to instructions, respectively. The environmental complexity $Complexity_e$ is computed as another weighted sum of its individual parameters:

$$Complexity_e = \frac{w_{nd} * nd + w_{dv} * dv + w_l * (1 - length)}{w_{nd} + w_{dv} + w_l}$$

with nd being the node degree of the decision point at hand, dv the deviation from prototypical angles, and $length$ the normalized length of a branch; w_{nd}, w_{dv}, w_l are the according weights. The instruction complexity is computed accordingly:

$$Complexity_i = \frac{w_{ie} * ie + w_{ic} * ic + w_{lm} * lm}{w_{ie} + w_{ic} + w_{lm}}$$

, with ie being the number of instruction equivalent turns, ic the complexity of describing the turn to take, and lm the complexity of landmarks. Finally, in the current model wayfinder-related factors are only the (normalized) SBSOD score, thus:

$$Complexity_w = 1 - SBSOD$$

3.2 Social model

This model accounts for the social costs of traveling certain roads. Since it is difficult (if not impossible) to know these costs for each and every place just based on road network data, we use a simple heuristics. We employ road category as a stand-in for social costs, with the higher the category the lower the social costs (see Table 2). For example, motorways as the highest category would have a cost of 1, while residential roads would have significantly higher costs – for example, when using parts of the OpenStreetMap (OSM) road hierarchy as we do in our evaluation (see Section 4) these costs may be 6. The higher the costs of a road, the less socially appropriate it is to use it. Thus, the social model aims at using higher category roads since these roads have less social costs associated with them.

3.3 Traffic model

We use a simple heuristic algorithm to assign traffic load to the different roads in a road network. We create a breadth-first search tree to traverse all the roads based on how they connect. The root of the tree is selected randomly from the list of all decision points. At depth zero (the root level) we randomly assign a number between 0 and 1, with 0 corresponding to “no traffic” and 1 to “heavy traffic”. Values in the range of [0,0.3) correspond to slight traffic, the range [0.3,0.7) to moderate traffic. Heavy traffic is defined by the range [0.7,1].

We traverse the tree level by level, assigning to each edge the average traffic of the preceding connected roads, plus a variation factor in the range [-0.4,0.4], which is randomly chosen. The variation factor includes negative numbers because otherwise traffic load would only ever increase for roads further down in the tree. Accordingly, we take the maximum of 0 and the calculated traffic load as the actual value, to avoid negative numbers as traffic load. This traffic load is then used as the costs for traversing an edge in "least traffic" path search.

3.4 Combined model

The models presented so far all account for different single factors important in navigation. It seems reasonable to assume that they are independent from each other, thus, they can all be combined using a weighted sum. This way, different factors can be assigned higher weight, i.e., taken to be more important, but the default assumes equal weight, and hence equal importance, of all factors. This results in the following *costs* for traversing an edge in the combined model:

$$costs = \frac{w_c * C_c + w_s * C_s + w_t * C_t}{w_c + w_s + w_t} \tag{1}$$

with $w_c, w_s, w_t = 1$ as a default. Here, w_c, w_s, w_t are the different weights for complexity, social costs, and traffic, respectively. Accordingly, the different C are the respective costs for traversing an edge in the different models.

3.5 Fastest path model

The fastest path model computes the fastest path between some origin and destination, as the name implies. To that end, it calculates the time it takes to traverse a road segment based on its length and an assumed average speed, which depends on the road type. This time is then the costs for an edge used in the "fastest path" search.

3.6 Shortest path model

In some of our experiments, we also use the shortest path (shortest distance) in the comparisons. This is simply computed using the standard Dijkstra algorithm [3].

4 Analysis

In this section, we detail the analysis of the effects of not "being alone" on the various path search models. We first explain the methods employed in the analysis, and then present the results.

4.1 Methods

4.1.1 Data

We extract road network data from OpenStreetMap[1] (OSM). OSM data has three main elements, namely *nodes*, *ways* and *relations*. To form a road network, we identify decision points (intersections), which are those *nodes* shared by two or more *ways* elements [5]. Figure 2 shows decision points for a part of New York. The decision points correspond to the nodes and the ways to the edges in the road network, resulting in a directed graph.

[1] `https://www.openstreetmap.org/`

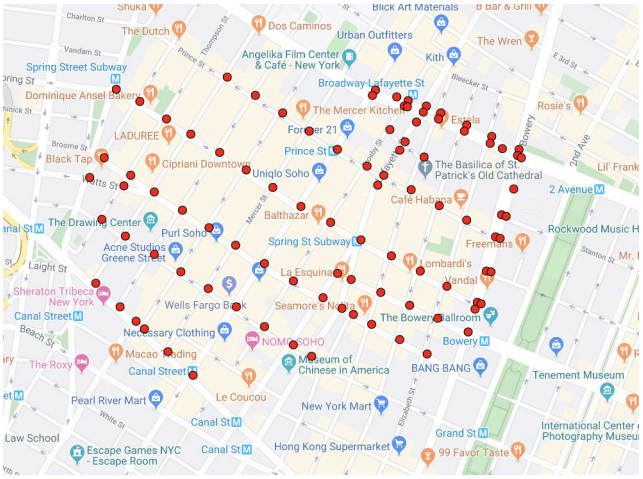


Figure 2 Identified decision points as red circles for a part of New York, defined by the bounding box $(-74.00433, 40.72038, -73.99263, 40.72545)$.

For our analysis, we selected four different city environments: (parts of) New York, Stockholm, London, and Paris. These cities have been chosen because they differ in their structure, but also because they provide good OSM data quality. Whereas New York exhibits a (well-known) grid structure, Stockholm is similar in the eastern part, but less structured towards the west. The road network of Paris is almost radial with connecting roads forming a "spider web". Finally, London has several areas of local roads loosely connected via some major roads. Figure 3 shows the road networks for the four cities. Here, width and color of the edges represent the road type and traffic load, respectively. The wider an edge, the higher the road type (i.e., residential roads are the thinnest). Slight, moderate, and heavy traffic are shown as green, yellow, and red edges, respectively.

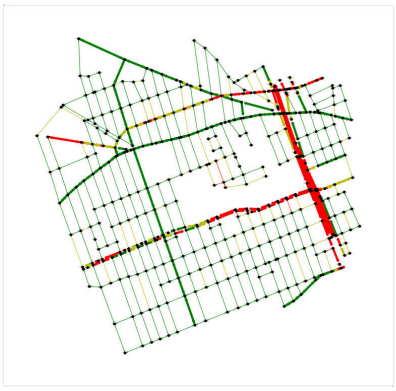
4.1.2 The implemented models

Section 3 presented the principal, generic models. For the analysis, they need to be implemented based on the available data. Thus, certain restrictions and simplifications may be made, as well as specific parametrizations for some of the factors. For all models, all parameter weights are set to 1, which makes them all equally important. At this point, we do not have any indications otherwise, and we are interested in the models' general behavior.

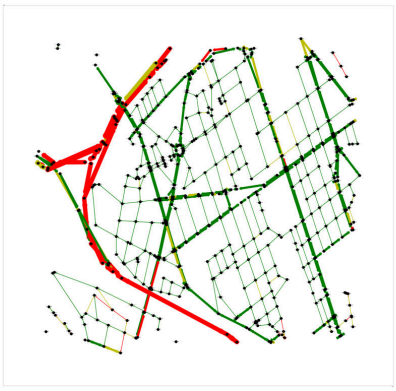
The implemented model for least complex paths does not account for wayfinder-related factors, i.e., SBSOD scores. There are no actual wayfinders involved in the analysis, and as said we are interested in general behavior. To account for landmarks in the model, we use all OSM objects tagged as *amenity* as a stand-in[2]. For reasons of simplicity, each landmark has a salience of 1. We use OSM's API to find all landmarks around a decision point using the procedure explained in Section 3.

The social, traffic, and fastest path models refer to the OSM road type hierarchy, identified via the *highway* tag of the different *ways* objects[3] to distinguish road types. We use six types of *ways* (see Table 2): motorway, trunk, primary, secondary, tertiary, and residential. The higher the category (motorway being the highest), the more social it is to use this road (i.e., the lower the social costs), and the higher the average speed to traverse it.

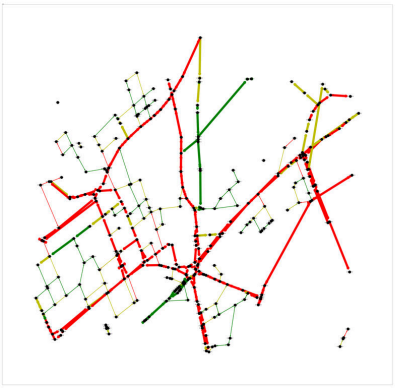
[2] `https://wiki.openstreetmap.org/wiki/Category:Amenities`
[3] `https://wiki.openstreetmap.org/wiki/Key:highway`



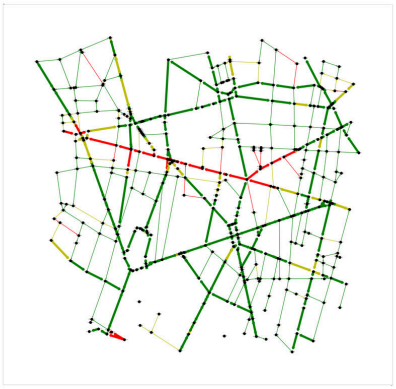
(a) New York.



(b) Stockholm.



(c) London.



(d) Paris.

Figure 3 Road networks for the four different city areas used in the analysis. The width of an edge represents the road type; the color the traffic load.

Table 2 The different road types used in the analysis, the average speed assigned to them, and their social costs. "Road type" corresponds to OSM *highway* tag value.

Road Type	Speed (km/h)	Social costs
Motorway	100	1
Trunk	80	2
Primary, **Secondary**	60	3, 4
Tertiary	50	5
Residential	30	6

4.1.3 Procedure

For each of the four different road networks (city environments), we randomly chose 100 origin / destination pairs, between which we compute the various different path types under investigation. In other words, for each of the different path search models, we compute 100 paths in each of the four different environments. We perform three analysis steps:

1. We compare paths between five different models, namely the complexity model, the traffic model, the social model, the combined model, and – in addition – the shortest path model. Over all 100 different paths for each model, we average path length, paths' complexity value, paths' traffic load and paths' social costs.
2. We investigate the effects of accounting for traffic on the complexity and social costs of the fastest paths. In order to add the effects of traffic to this model, we modify the (assumed) average speed (see Table 2) along a road segment by a "traffic" factor. We multiply this speed by 1, 0.75, and 0.4 for slight, moderate, and heavy traffic, respectively. We average the paths' complexity value and social costs over all 100 different paths.
3. We investigate the effects of accounting for traffic on the complexity and social costs of the least complex paths. To add traffic as a factor to the least complex paths, we use the combined model with according weight settings: $w_c = 1, w_t = 0, w_s = 0$ for paths without traffic; $w_c = 1, w_t = 1, w_s = 0$ for those with traffic. Again, we average the paths' complexity value and social costs over all 100 different paths.

4.2 Results

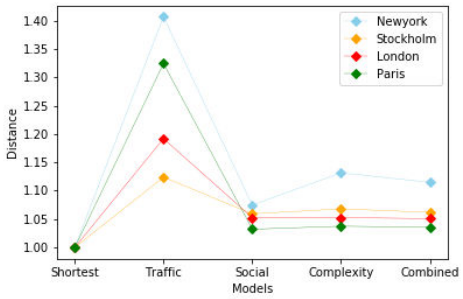
In the following, we present the results of the three analysis steps. These are then further discussed in Section 5.

4.2.1 Effects of the different path search models

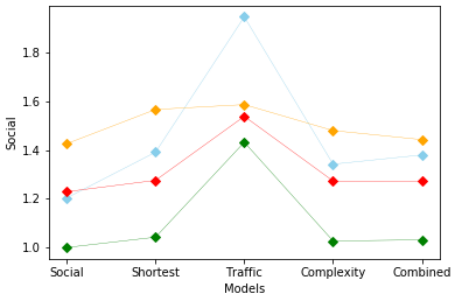
Table 3 and Figure 4 illustrate the effects the different path search models have on distance (a), social costs (b), traffic (c), and complexity (d), for each of the four environments. In each figure, the absolute lowest value is used as a reference value, set to 1, and all others are scaled relative to this reference value. That is, a value of 1.5 would mean that the respective value is 50% higher than the reference value. This is done globally, i.e., across the four environments, except for distance, where for each environment the respective average shortest path is used as a reference value. This is done to more clearly show relative increase of path length when accounting for other factors than distance.

We can see that the traffic model, which finds paths with the least traffic load, results in potentially large detours compared to the shortest path, with the increase depending on the environment (for London on average about 20% longer paths, for Paris more than 30%, for New York more than 40%). The other three models (social, complexity, combined) only lead to minor increases of path length (15% or less on average).

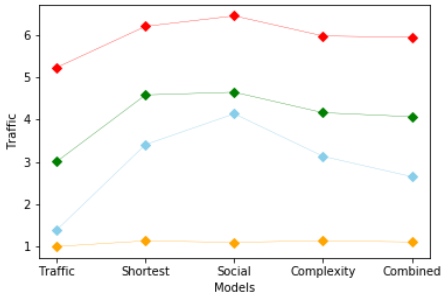
Figures 4b and 4c show the relation between social costs and traffic load for the paths calculated by the different models. First, we can observe that there are differences in the "baseline" between the different environments. For example, traffic load is approximately five times higher in London compared to New York or Stockholm even in the optimal cases, and the paths calculated with the social model have significantly lower social costs in Paris compared to the most social paths in Stockholm. That is, again we see an impact of the environment, but also of the distribution of traffic, on path search behavior. Further, the traffic model results in the paths with the highest social costs (Figure 4b), whereas the social



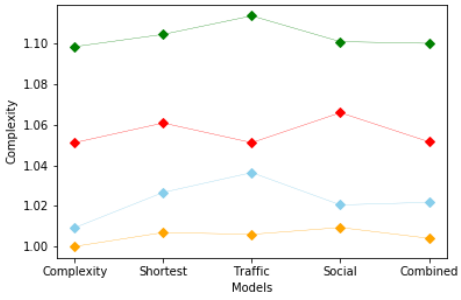
(a) Average path length of the different models.



(b) Average social costs of the paths computed by the different models.



(c) Average traffic load of the paths computed by the different models.



(d) Average path complexity of the paths computed by the different models.

Figure 4 Relative increase of the different factors under investigation depending on the applied path search model, for the four different environments: (a) average distance, (b) average social costs, (c) average traffic load, (d) average complexity. The different colors always represent the same environment across the diagrams.

model results in paths with the most traffic load (Figure 4c). For all other models, there seems to be a rather low increase in the social costs, however, (except for Stockholm) they all suffer a rather drastic increase in traffic load, even if it is lower than for the social model.

In Figure 4d we can see that complexity also depends on the environment, i.e., the least complex path in Stockholm is less complex on average than that in Paris, for example. These differences are less pronounced than for the environments' impact on the social and traffic model, though. Generally, the differences in complexity across the different models are small.

4.2.2 Effects on fastest paths

Figure 5a shows the effects accounting for traffic has on the (average) social costs of the fastest paths, whereas Figure 5b shows the same for the average complexity. Table 4 presents the average and standard deviation values. Similar to the previous analysis step, we can see that the social costs increase when accounting for traffic. In terms of path complexity, there is hardly an observable difference of paths with or without traffic for Stockholm and Paris. However, for New York and London complexity decreases slightly when avoiding traffic. Thus, again, the structure of the environment has an impact on the results.

Table 3 Average (Avg, and standard deviation; Std) path length (a), social costs (b), traffic load (c), and complexity (d) of the different path search models for the four different environments, as relative values (i.e., scaled to the lowest one).

(a) Path length.

	New York		Stockholm		London		Paris	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Shortest	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
Traffic	1.406	0.398	1.123	0.151	1.191	0.257	1.324	0.416
Social	1.074	0.113	1.059	0.098	1.052	0.101	1.032	0.079
Complexity	1.131	0.216	1.068	0.106	1.052	0.127	1.03	0.092
Combined	1.114	0.182	1.061	0.096	1.050	0.129	1.035	0.090

(b) Social costs.

	New York		Stockholm		London		Paris	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Shortest	1.392	1.602	1.567	1.687	1.276	1.214	1.042	1.04
Traffic	1.948	2.253	1.587	1.712	1.539	1.686	1.432	1.866
Social	1.203	1.192	1.426	1.380	1.229	1.169	1.0	1.0
Complexity	1.342	1.391	1.481	1.499	1.273	1.224	1.026	1.035
Combined	1.380	1.519	1.443	1.412	1.272	1.23	1.031	1.037

(c) Traffic.

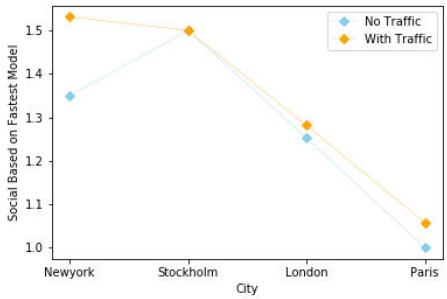
	New York		Stockholm		London		Paris	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Shortest	3.409	13.524	1.133	1.134	6.205	10.376	4.583	12.047
Traffic	1.391	4.713	1.0	1.178	5.232	9.763	3.008	8.839
Social	4.137	12.9	1.097	1.065	6.453	9.104	4.648	11.325
Complexity	3.131	11.744	1.137	1.058	5.978	10.333	4.163	11.259
Combined	2.65	10.928	1.106	1.0	5.945	10.374	4.068	11.394

(d) Complexity.

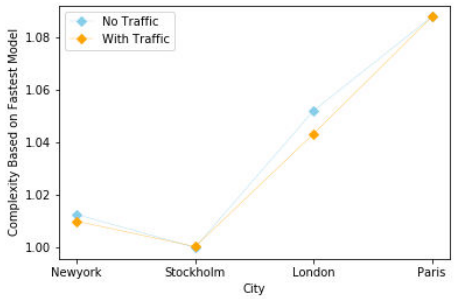
	New York		Stockholm		London		Paris	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Shortest	1.026	1.014	1.006	1.295	1.06	1.122	1.104	1.281
Traffic	1.036	1.133	1.005	1.297	1.051	1.129	1.113	1.331
Social	1.02	1.0	1.009	1.309	1.066	1.165	1.101	1.288
Complexity	1.009	1.001	1.0	1.292	1.051	1.167	1.098	1.282
Combined	1.021	1.083	1.004	1.31	1.051	1.179	1.1	1.28

4.2.3 Effects on least complex paths

Figures 6a and 6b show the effects of accounting for traffic on average social costs and path complexity, respectively, when computing paths using the least complex paths model (see Table 5 for average and standard deviation values). Again, aiming to reduce traffic load increases social costs. However, average path complexity essentially remains the same.



(a) The effects of accounting for traffic on social costs, for the four environments.



(b) The effects of accounting for traffic on path complexity, for the four environments.

Figure 5 Effects of accounting for traffic on (a) social costs and (b) path complexity for the fastest paths.

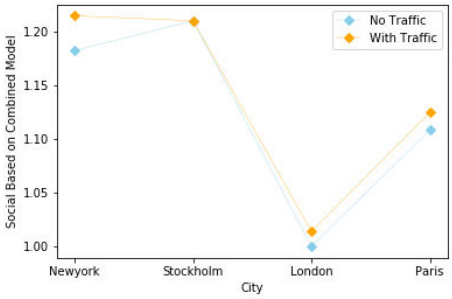
Table 4 Average and standard deviation of accounting for traffic on (a) social costs and (b) path complexity for the fastest paths.

(a) Social costs.

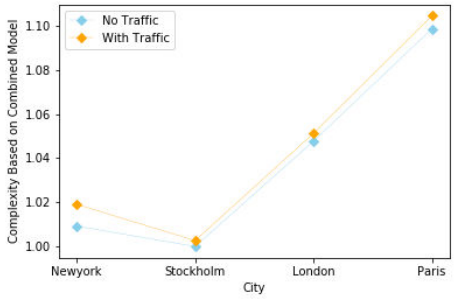
	New York		Stockholm		London		Paris	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Fastest Model (Without Traffic)	1.349	1.521	1.499	1.353	1.252	1.137	1.0	1.0
Fastest Model (With Traffic)	1.531	1.796	1.5	1.347	1.281	1.201	1.056	1.009

(b) Complexity.

	New York		Stockholm		London		Paris	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Fastest Model (Without Traffic)	1.012	1.131	1.0	1.02	1.052	1.101	1.087	1.186
Fastest Model (With Traffic)	1.009	1.145	1.0	1.0	1.04	1.081	1.087	1.186



(a) Effects of accounting for traffic on social costs for the least complex paths, for the four environments.



(b) Effects of accounting for traffic on path complexity for the least complex paths, for the four environments.

Figure 6 Effects of accounting for traffic on (a) average social costs and (b) average path complexity for the least complex paths, for the four environments.

Table 5 Average and standard deviation of accounting for traffic on (a) social costs and (b) path complexity for the least complex paths.

(a) Social costs.

	New York		Stockholm		London		Paris	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Combined Model (Without Traffic)	1.182	1.387	1.21	1.0	1.0	1.489	1.108	1.506
Combined Model (With Traffic)	1.214	1.326	1.21	1.0	1.013	1.496	1.124	1.503

(b) Complexity.

	New York		Stockholm		London		Paris	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Combined Model (Without Traffic)	1.009	1.159	1.0	1.0	1.047	1.104	1.098	1.243
Combined Model (With Traffic)	1.019	1.105	1.002	1.003	1.051	1.138	1.104	1.27

5 Discussion

The results of our analysis show that accounting for traffic in path search has a clear negative effect on social costs. In avoiding traffic, wayfinders may easily end up in small, residential roads, thus, being redirected into areas that are not built for taking larger amounts of traffic. In some ways, such a result was to be expected, and it confirms similar unwanted consequences as discovered in [9]. This effect goes both ways, i.e., accounting for social costs drastically increases average traffic load. On the other hand, all other tested models mostly have only minor impact on social costs, i.e., seem to implicitly avoid these small, residential roads. Such roads are often rather short and residential areas may involve many turns, thus, navigating through them increases complexity (according to our model) and would often not provide the most direct connection, i.e., increase distance traveled. But ignoring traffic in computing paths may well mean that you end up being stuck in it. In other words, the average traffic load for all other models is also significantly higher than that for the least traffic model, even if this impact is smaller than for the social model. Thus, these results clearly show that assuming to "being alone" is problematic for path search models.

Interestingly, accounting for traffic hardly seems to influence path complexity. In fact, for the fastest path it decreases for some environments, which might be explained by some "simpler", but generally slower, roads becoming now faster to traverse due to lower traffic load. Still, this result seems to contradict the findings in [9]. But they used other path search criteria (scenic and safety) and also their path complexity measure is simpler than ours, only using node degree and turn/no turn, which might explain some of these differences.

The combined model does not lead to much of an increase of either complexity or social costs – at least for most environments. Only for New York the social costs increase from about 1.2 to nearly 1.4. Traffic load does increase significantly, though (except for Stockholm). But this increase is smaller than for any of the other models, so adding traffic as a parameter into the combined model does have the wanted effect. Increasing the weight and, thus, importance of this parameter would help to reduce the increase in traffic load further, though likely with increased social costs as a consequence. It would take a careful calibration of weights to find the ideal balance here, which would also depend on the context.

We observe a major impact of environmental structure on our results. And this impact is amplified by the distribution of traffic. For example, in Stockholm heavy traffic is restricted to the major roads to the west, thus, does not impact paths through the regular, grid-like area to the east. On the other hand, in London all major roads that connect the somewhat

dispersed local areas have heavy traffic load, which results in significantly higher traffic load for all paths compared to, for example, Stockholm. And in Paris there is a relatively dense, "spider-web"-like network of major roads interwoven with residential roads, which allows for largely avoiding these small roads in optimizing for social costs, while there are significantly fewer of these major roads in Stockholm, for example.

While our analysis provides some important insights into the effects of "not being alone" it also has some limitations. Notably, the environments we used are fairly small. Larger environments may amplify the differences between the different path search models, but also the effects of traffic on the resulting paths. Further, the model for least complex paths combines several parameters in an overall complexity measure. Arguably, all of them are relevant for wayfinding complexity, but in their combination they may also mask each other to some degree. Thus, a more systematic analysis of their individual impact may be interesting.

6 Conclusions and future work

In this paper, we analyse different path search models with respect to the fact that we are not alone while navigating through road networks. "Not being alone" is an aspect that has been neglected so far in most existing models. Overall, we find a significant effect of accounting for traffic, in particular on social costs (and vice versa), however, interestingly, hardly any changes of wayfinding complexity when accounting for either traffic or social costs.

Future work includes analysing further environments to gain more insights into the effects of environmental structure, and in particular, using larger areas in this analysis. We also plan to incorporate individual differences in the analysis, i.e., SBSOD scores but also preferences for certain road types, to evaluate their effects on path search results, in particular complexity and social costs. Finally, looking at real traffic patterns would be interesting to gain better insights into the actual, real-world effects of this parameter.

References

1 David Caduff and Sabine Timpf. The landmark spider: Representing landmark knowledge for wayfinding tasks. In Thomas Barkowsky, Christian Freksa, Mary Hegarty, and Ric Lowe, editors, *Reasoning with Mental and External Diagrams: Computational Modeling and Spatial Assistance - Papers from the 2005 AAAI Spring Symposium*, pages 30–35, Menlo Park, CA, 2005.

2 Laura A. Carlson, Christoph Hölscher, Thomas F. Shipley, and Ruth Conroy Dalton. Getting lost in buildings. *Current Directions in Psychological Science*, 19(5):284–289, 2010. doi:10.1177/0963721410383243.

3 E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

4 Matt Duckham and Lars Kulik. "Simplest" paths: Automated route selection for navigation. In Werner Kuhn, Mike Worboys, and Sabine Timpf, editors, *Spatial Information Theory*, volume 2825 of *Lecture Notes in Computer Science*, pages 169–185, Berlin, 2003. Springer.

5 Paolo Fogliaroni, Dominik Bucher, Nikola Jankovic, and Ioannis Giannopoulos. Intersections of our world. In Stephan Winter, Amy Griffin, and Monika Sester, editors, *10th International Conference on Geographic Information Science*, volume 114 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.GISCIENCE.2018.3.

6 Ioannis Giannopoulos, Peter Kiefer, Martin Raubal, Kai-Florian Richter, and Tyler Thrash. Wayfinding decision situations: A conceptual model and evaluation. In Matt Duckham, Edzer Pebesma, Kathleen Stewart, and Andrew U. Frank, editors, *Geographic In-*

formation Science, pages 221–234, Cham, 2014. Springer International Publishing. doi:10.1007/978-3-319-11593-1_15.

7 Shazia Haque, Lars Kulik, and Alexander Klippel. Algorithms for reliable navigation and wayfinding. In Thomas Barkowsky, Markus Knauff, G'erad Ligozat, and Daniel R. Montello, editors, *Spatial Cognition V*, LNCS4387, pages 308–326, Berlin, 2007. Springer.

8 Mary Hegarty, Anthony E Richardson, Daniel R Montello, Kristin Lovelace, and Ilavanil Subbiah. Development of a self-report measure of environmental spatial ability. *Intelligence*, 30(5):425–447, 2002.

9 I. Johnson, J. Henderson, C. Perry, J. Schöning, and B. Hecht. Beautiful...but at what cost? An examination of externalities in geographic vehicle routing. *Proceedings of the ACM on Interactive, Mobile, Wearable, Ubiquitous Technologies*, 1(2):15:1–15:21, 2017. doi:10.1145/3090080.

10 Alexander Klippel, Heike Tappe, and Christopher Habel. Pictorial representations of routes: Chunking route segments during comprehension. In Christian Freksa, Wilfried Brauer, Christopher Habel, and Karl F. Wender, editors, *Spatial Cognition III*, volume 2685 of *Lecture Notes in Artificial Intelligence*, pages 11–33, Berlin, 2003. Springer.

11 Kevin Lynch. *The Image of the City*. The MIT Press, Cambridge, 1960.

12 E.J. Manley, S.W. Orr, and T. Cheng. A heuristic model of bounded route choice in urban areas. *Transportation Research Part C: Emerging Technologies*, 56:195–209, 2015. doi:10.1016/j.trc.2015.03.020.

13 David M Mark. Automated route selection for navigation. *IEEE Aerospace and Electronic Systems Magazine*, 1(9):2–5, 1986.

14 Daniel R Montello. Spatial cognition and architectural space: Research perspectives. *Architectural Design*, 84(5):74–79, 2014. doi:10.1002/ad.1811.

15 Kai-Florian Richter. Adaptable path planning in regionalized environments. In Kathleen Stewart Hornsby, Christophe Claramunt, Michel Denis, and Gérard Ligozat, editors, *Spatial Information Theory*, volume 5756 of *Lecture Notes in Computer Science*, pages 453–470, Berlin, 2009. Springer.

16 Kai-Florian Richter and Matt Duckham. Simplest instructions: Finding easy-to-describe routes for navigation. In Thomas J. Cova, Harvey J. Miller, Kate Beard, Andrew U. Frank, and Michael F. Goodchild, editors, *Geographic Information Science*, volume 5266 of *Lecture Notes in Computer Science*, pages 274–289, Berlin, 2008. Springer.

17 Kai-Florian Richter and Alexander Klippel. A model for context-specific route directions. In Christian Freksa, Markus Knauff, Bernd Krieg-Brückner, Bernhard Nebel, and Thomas Barkowsky, editors, *Spatial Cognition IV*, volume 3343 of *Lecture Notes in Artificial Intelligence*, pages 58–78, Berlin, 2005. Springer.

18 Jerry Weisman. Evaluating architectural legibility: Way-finding in the built environment. *Environment and Behaviour*, 13(2):189–204, 1981.

19 Jan Malte Wiener and Hanspeter A. Mallot. 'Fine to coarse' route planning and navigation in regionalized environments. *Spatial Cognition and Computation*, 3(4):331–358, 2003.

20 Jan Malte Wiener, Thora Tenbrink, Jakob Henschel, and Christoph Hölscher. Situated and prospective path planning: Route choice in an urban environment. In *CogSci 2008: 30th Annual Conference of the Cognitive Science Society*, 2008.

Enhancing Usability Evaluation of Web-Based Geographic Information Systems (WebGIS) with Visual Analytics

René Unrau
Institute for Geoinformatics, University of Münster, Germany

Christian Kray
Institute for Geoinformatics, University of Münster, Germany

Abstract

Many websites nowadays incorporate geospatial data that users interact with, for example, to filter search results or compare alternatives. These web-based geographic information systems (WebGIS) pose new challenges for usability evaluations as both the interaction with classic interface elements and with map-based visualizations have to be analyzed to understand user behavior. This paper proposes a new scalable approach that applies visual analytics to logged interaction data with WebGIS, which facilitates the interactive exploration and analysis of user behavior. In order to evaluate our approach, we implemented it as a toolkit that can be easily integrated into existing WebGIS. We then deployed the toolkit in a user study (N=60) with a realistic WebGIS and analyzed users' interaction in a second study with usability experts (N=7). Our results indicate that the proposed approach is practically feasible, easy to integrate into existing systems, and facilitates insights into the usability of WebGIS.

2012 ACM Subject Classification Human-centered computing → User studies; Human-centered computing → Usability testing

Keywords and phrases map interaction, usability evaluation, visual analytics

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.15

Supplementary Material The source code is publicly available at `https://github.com/ReneU/session-viewer` and includes configuration instructions for use in other scenarios.

1 Introduction

Geospatial data has become a critical backbone of many web services available today, such as search engines, online booking sites, or open data portals [29]. Frequently, these sites need to visualize geospatial data [11] and enable interaction with the visualizations. The resulting web-based geographic information systems (WebGIS) have proliferated over the last decade though they vary regarding their complexity and purpose – from simple map-visualizations of search results to geographic information systems with extensive functionality (e.g., [4, 14, 25]). The range of users is equally broad, from novices with little to no knowledge about geo-data and geo-visualization to experts, who all expect good usability. This variety of applications and target users for WebGIS gives rise to diverse and potentially conflicting requirements for the UI [7, 26, 28]. Consequently, designing the user interface (UI) of a WebGIS can be quite challenging and affect the overall usability of the website considerably [10, 21, 22].

One of the challenges in this context is the combination of map-based visualizations with more traditional UI elements (such as menus, buttons, or sliders). Besides some basic cartographic understanding, the former requires specific map actions (such as zooming, panning or layer selection) while the latter provides access to map-related and other functionality. Treating map actions like any other functionality can potentially cause problems and misunderstandings. For example, the actual scale and the visible layers of a map may have a

significant impact on how successful users are when accessing a WebGIS and considerably affect which UI elements users use in which order and how often. To better understand these issues and to assess the usability of WebGIS, it is thus essential to evaluate them thoroughly. However, many existing evaluation approaches do not explicitly consider interaction with the geographic content [16], are lacking a visual representation for exploratory usage [4] or do not handle large amounts of data from multiple user sessions [6, 20].

In this work, we hypothesize that usability evaluations of WebGIS could greatly benefit from a holistic and scalable approach that is based on logging map interactions for visual analysis by experts. To evaluate the approach, we implemented it as a prototypical toolkit and integrated it into a realistic A/B testing scenario. We collected interaction data from 60 WebGIS users and conducted an expert study focused on analyzing and comparing usage patterns in both scenarios, thus evaluating the usability of the WebGIS with our toolkit. Unlike alternative methods, such as eye-tracking or screen recordings, our approach explicitly considers map interactions, does not require additional hardware and can be deployed at a large scale. It can thus complement traditional methods (such as questionnaires) through an interactive and profound exploration of usability aspects.

We make two main contributions: (1) we propose a new approach for usability evaluation of WebGIS by applying visual analytics for map interaction data from multiple user sessions through a holistic toolkit with integration capabilities for existing applications; and (2) we evaluate the proposed approach by integrating our toolkit into a realistic WebGIS to collect the required data (N=60) and to then analyze it in an expert study (N=7). In addition, we also present insights into the usability aspects of a geovisualization that we used in the evaluation.

The remainder of this paper is structured as follows: First, we provide an overview of work related to usability evaluation approaches for WebGIS, software instrumentation, and visual analytics. Next, we introduce our approach and briefly discuss its prototypical implementation. Section 4 lists our hypotheses and describes the two studies that we conducted with regular users and experts in order to evaluate our approach. The penultimate sections discuss the implications and limitations of the obtained results and our approach. The paper concludes by summarizing our key findings and contributions.

2 Related Work

2.1 Usability Issues of WebGIS

Various use cases have demonstrated the needs for assessing the usability of WebGIS and their tools. For instance, Lobo et al. [14] investigated different techniques for comparing map layers. Their results showed that specific tools are inferior to others if users have to identify missing or modified features. In a different user study for Ethermap [4], participants were asked to map flooded areas collaboratively. Although only three out of 36 participants did not map actively, the authors could not identify the underlying reasons for this behavior. An analysis of these users' interactions could yield exciting insights into the usability of the WebGIS. May and Gamble [17] conducted three experiments for investigating the impact of automatic map movements after users clicked a point on the map. Their analysis revealed that the evaluation of map movement techniques also depends on the geospatial data that might be outside the visible extent after panning or zooming the map. Frequently, evaluators use a combination of automatic data collection approaches and traditional methods. Manson et al. [16] asked two groups of participants to perform the same tasks in a WebGIS for navigation. They logged mouse actions such as mouse-up time as well as the total number of

mouse interactions and applied eye tracking to test the usability of map navigation schemes. Although they collected information about map interactions, the state of the map (i.e., scale and extent) was not captured. As a result, the data could only be used to reconstruct the users' behavior with traditional UI elements. Ingesand and Golay [8] applied a method that is similar to the one proposed in this paper. In a remote evaluation, they collected detailed interaction logs for measuring the performance of predefined tasks and involved user satisfaction ratings. However, they focused on traditional usability metrics such as error rate and task completion time but did not consider user strategies for map interactions.

A review of GIS usability evaluations that are available in the literature revealed that most findings are related to issues with user guidance and tool usage (53.8% and 51.3% of all reviewed studies) [30]. In contrast, identified issues that are related to the users' strategies were reported only in 15.4% of the reviewed studies. These differences could be related to the choice of evaluation methods or data collection approaches. The combination of qualitative knowledge from usability experts with quantitative data processing might facilitate a better understanding of the underlying user strategies.

2.2 Instrumenting Software for User Testing

Instrumenting software for data collection facilitates the conduction of remote and asynchronous user studies. As a result, the conduction of usability evaluations requires less effort for experimenters: Once developed, instrumented software can be mass deployed to collect the required amount of data with little effort. Target users, as well as first-time users of the software, are tested in their actual real-world environment. Usage data can be collected and analyzed continuously even for longitudinal studies [12]. Subsequently, the datasets may be used to compare changes in the UI or to evaluate the learnability and memorability of users. Finally, instrumented software minimizes experimenter bias and novelty effects. For example, Atterer and Schmidt [2] implemented a proxy for recording detailed usage information. By intercepting requests and responses, they were able to perform usability evaluations. However, graphic-intensive applications, such as web mapping services, pose new problems: "[...] the central part of the user interface does not consist of GUI elements which are given ID values by the application programmer, but of a number of anonymous tiles which contain graphics".

Our method extends a recent tool for visualizing WebGIS sessions to identify usability issues via heatmaps and Sankey diagrams [31]. However, the approach presented in this paper goes beyond standalone visualizations of user interactions. Instead, it explicitly addresses the spatial aspects of map interactions by providing a GIS-like concept to explore and analyze map interactions, thus, applying the concept of visual analytics.

2.3 Visual Analytics

Visual analytics aims to combine data processing and human domain knowledge in interactive visualizations to generate new insights. Keim et al. [9] define visual analytics as an "automated analysis technique with interactive visualizations for an effective understanding, reasoning, and decision making on the basis of extensive and complex data sets." A common application of visual analytics is the analysis of movement data. For example, Rinzivillo et al [23] developed a set of algorithms to cluster large number of trajectories and thus facilitate visual exploration of movement patterns. However, applications of visual analytics for evaluating UI interaction data are rare in the literature, especially for graphic-intensive UIs such as WebGIS. For example, Mac Aoidh et al. [15] made use of visual analytics to

analyze implicit interest indicators for spatial data by visualizing map interactions on top of the actual UI to present their results. The scalability of their approach is limited as the visualization of mouse movements is restricted to display and compare only three user sessions at the same time. Coltekin et al. investigated the use of space-time cubes for exploring eye-tracking recordings which allows users to discover movement patterns in a combined view [13]. While their results show interesting opportunities for usability evaluations, the authors state that many users stills struggle to understand and interact with complex 3D views.

3 Approach

The overall goal of our approach is to explicitly consider the state of the map and the interactions of the user with it while assessing the usability of a WebGIS. The state of the map can strongly affect the users' interaction with a WebGIS. For example, the zoom level of the map may require users to perform many zooming and panning interactions before they can actually complete a task. Depending on their skill level, a disadvantageous zoom level might even lead to errors or delays. Map designers can also realize the map content itself via different geo-visualizations, which in turn may affect user interaction with a WebGIS and thereby the overall usability. Even if the UI is the same, usability and user performance can vary substantially depending on the map scale, region, or chosen geovisualization.

An approach that explicitly captures map-related aspects and interactions has the potential to identify the issues mentioned above, and it thus could help to improve the UI of future WebGIS. To achieve this goal, we combine tailored data collection and visualization techniques in a prototypical toolkit for integration into existing WebGIS. In the following, we provide an overview of our approach and its implementation as a toolkit.

3.1 Data collection

We instrument the code of the investigated WebGIS to log changes of the map's state, such as the current center. This procedure requires access to the source code of the application. The integration of our data-collection component into the existing source code is simple because most web mapping frameworks already provide access to the required events [24] and thus result in minimal augmentation of the existing code. For our initial implementation we logged zoom-in, zoom-out, pan, and select events from the augmented WebGIS. While zooming and panning events represent traditional map interactions selection, in this case, means marking a table entry, that corresponds to a features on the map, via a checkbox. This shows that our tool is capable of logging map interactions as well as interactions with traditional UI elements. Before the data-collection component sends the user interaction data to a central database, it adds a timestamp and an anonymous session ID (randomly generated) to the event.

3.2 Visual Analytics

The session-viewer component of our toolkit provides capabilities similar to a WebGIS as recorded map interactions represent geospatial information themselves and can thus be viewed and analyzed likewise. The collected data is visualized via three analysis layers on top of a basemap and can be toggled on or off via a layer list control as well as spatially explored by zooming or panning the map (Figure 1). As A/B testing is a standard method to compare different scenarios with subtle differences in the UI, our toolkit contains separate map views for each of the two scenarios and synchronizes their state. The interactions of an analyst

with either of the two map views will be synchronized to the other view. Synchronization includes panning and zooming interactions as well as the selection of visible layers and the state of additional controls.

The analysis layers that are provided by our toolkit focus on different aspects of the available map interaction data and allow individual interaction possibilities to filter or highlight subsets of the data (Figure 1). For this purpose, we make use of traditional task metrics (task time and interaction count) as well as additional data from questionnaires (user experience ratings) and combine these with the spatial aspects of map interaction data (center of current map extent). Further data sources can also be included, such as map entries that were part of the original user task and might be used as a reference when analyzing the data. Although other approaches, such as space-time cubes, are possible and should also be considered, we believe that 2D analysis layers minimize visual cluttering, provide well-known forms of interaction, and are thus more intuitive. Our pre-tests with a space-time cube prototype did found that users struggled to compare the position of 3D tracks and point clouds.

Single Metrics Layer. The locations of all map interactions are displayed as points on the map and colored based on one of the metrics that can be selected from a menu. For the evaluation of our toolkit we provided the zoom level, the user interaction count and seconds since session start as well as the pragmatic, hedonic, and overall quality. In contrast to previous visualizations of map interactions in the literature [15], this layer helps overcoming visual cluttering by providing mechanisms to filter and manipulate the representation of the data. First, the analyst can choose between two color themes ("High to Low" and "Above and Below") to highlight outliers or remove noise. Second, a color ramp also acts as a slider for changing the color-stops of the visualization and allows the analyst to determine thresholds to emphasize specific points.

The single metrics layer provides an overview of the spatial distribution of the users' map interactions. This aspect should reveal new insights compared to traditional metrics as the density, the accumulation of clusters, and the detection of regions of interest may be used to detect usability flaws. For example, analysts may identify spatial areas that are important for the task at hand but are not visited by the users' of the investigated WebGIS or only with low zoom levels that cannot reveal much detail.

Relationship Layer. This analysis layer extends the single metrics layer by enabling the combination of two metrics, and thus the investigation of correlations between them. The rationale for this layer is based on the limitations of previous studies that struggled to identify the relationship between usability metrics and user interactions. Using this layer, analysts can combine one of the three traditional metrics (zoom level, interaction count, seconds since session start) with one of the UX ratings (pragmatic, hedonic, overall) in our tool. The session-viewer component of our toolkit automatically creates four categories based on the analyst's selection and applies them as a visual variable to the data points on the map. After selecting a relationship, the widget in the lower-left corner changes to a legend that explains the visual variable.

This relationship layer supports analysts in evaluating the impact of users' map interactions on their experience. As a result, this layer might help answer questions such as: "Where do users who rated the WebGIS as not pragmatic interact with the map initially compared to users who rated the WebGIS as highly pragmatic?"

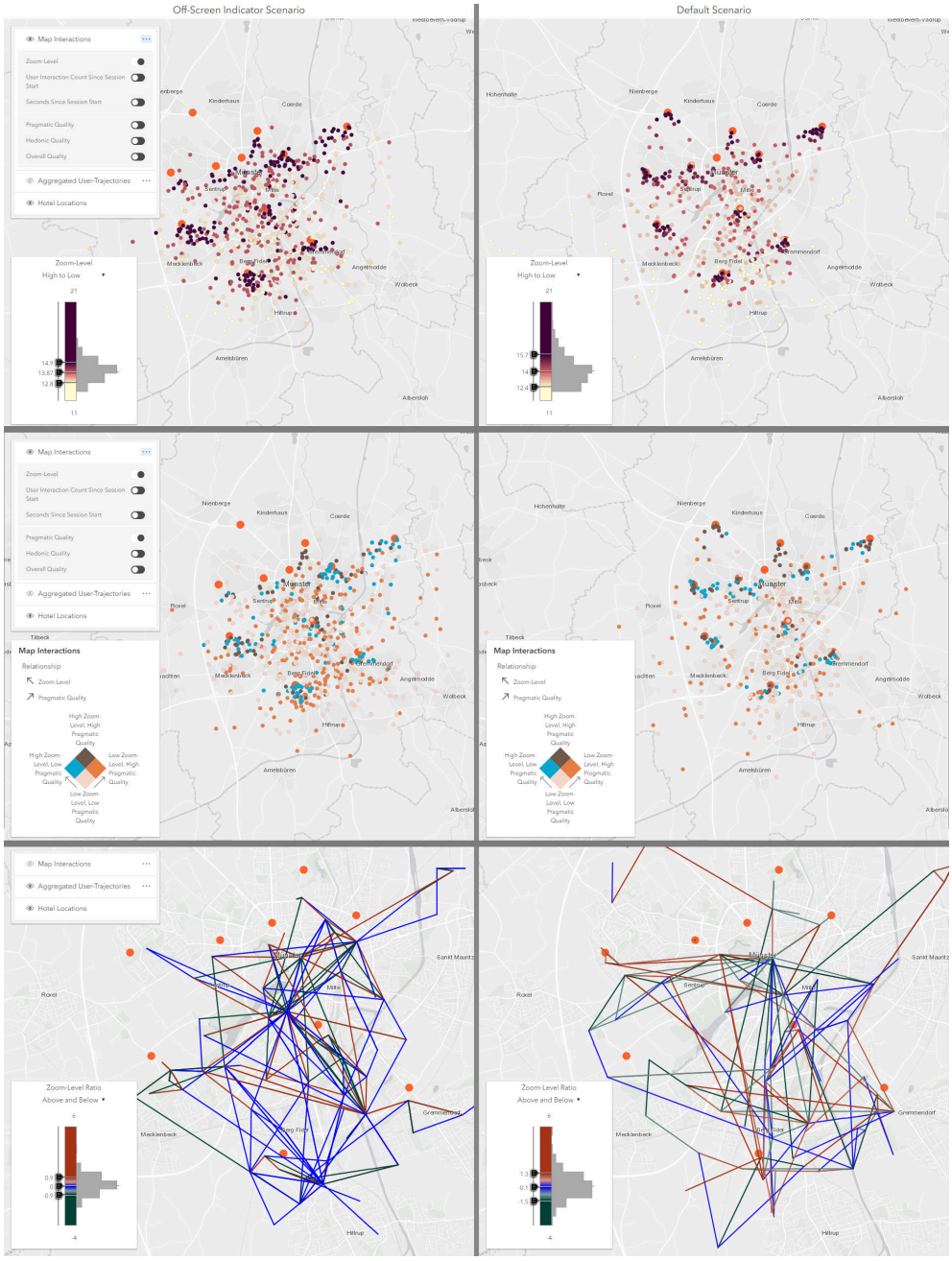


Figure 1 Theses three screenshots of all three analysis layers (top/center/bottom) show the data that was collected in our user study (N = 60). The Session-Viewer component of the toolkit provides two synchronized map views for comparing datasets from A/B testing (left/right). Analysts can choose between three map layers in the list that may expand and provide further controls. Furthermore an additional widget for each map view is used to display a legend or manipulate the parameters. The single metric analysis layer shows the location of map interactions and the corresponding zoom level (top). Analysts can interact with the data by using the color slider and histogram in the additional widget. The relationship layer visualizes the combination of two metrics which can be selected by the analyst (center). The "Aggregated User-Trajectories" layer indicates the users' key movements and characteristic stops while navigating the original WebGIS (bottom). The color variable is used to show whether the movement was a pan or zoom (in/out) action.

Aggregated User Trajectories. This layer combines the users' map interactions to actual trajectories and aggregates these to avoid visual cluttering and allow the detection of key movements. We adapted and extended the summarization algorithm by Andrienko et al. [1] that has been initially developed for movement tracks, as the characteristics of map interactions are similar to such real-world movements. First, the algorithm filters the dataset based on characteristic points that fulfill specific criteria. These criteria are a minimal stop duration and a distance tolerance. If the elapsed time between two subsequent points of a trajectory exceeds the minimal stop duration and their distance is within the tolerance, the first point represents a characteristic point in the dataset. The same procedure can be applied to map interactions by considering the time between map interactions (minimal stop duration) and the distance on the map between two subsequent map extents (distance tolerance). Second, characteristic points are clustered to generalize single points to areas of interest and aggregate the data. Again, this step is reasonable for the evaluated data type as relevant entries on the map result in map interactions that set the current map extent to locations that are close to those entries and thus represent areas of interest. Third and last, the initial trajectories are filtered based on the generated clusters. The algorithm removes every stop that is not inside an area of interest and, thereby, hides short and intermediate stops while still considering the overall movement.

Our toolkit includes the adapted algorithm and extends the resulting key movements with a color variable. To distinguish between zooming and panning interactions, we used the zoom level of both points to calculate the zoom level ratio for the movement. Similar to the single metrics analysis layer, the analyst can change the color of this variable by using a slider with an adjacent histogram.

This visualization may help evaluators to understand the users' approaches for the task at hand, such as the general movement pattern in a user session. For example, if the task requires users to visit multiple locations, the analysts can identify if there is potential to improve the efficiency by optimizing movement patterns between these locations.

3.3 Implementation

The implemented toolkit consists of three components. The data-collection component is implemented in JavaScript and must be imported and used in the targeted WebGIS. The instrumentation requires access to and limited knowledge about the source code of the WebGIS. However, the application programming interface of the data-collection component provides only one method and is thus easy to use and understand. Next, the data-collection component sends the captured interactions to a central database (second component). We used the open-source database engine *Elasticsearch*[1] as it provides a schema-less index with endpoints for posting and retrieving data. Consumers of the data-collection component can, therefore, post custom data fields without adjusting the data model. These capabilities ensure that our approach is customizable and facilitates the realization of further logging scenarios in the future with little effort. Last, we used the *ArcGIS API for JavaScript*[2] to build a WebGIS-like application (session-viewer component) for consuming, processing, and visualizing the data from the database as it provides many built-in capabilities for interactive data visualizations. The source code is publicly available and includes configuration instructions for use in other scenarios[3]. Comparable existing commercial solutions, like Maptiks[4], usually collect

[1] `https://www.elastic.co/elasticsearch/` accessed June 6th, 2020
[2] `https://developers.arcgis.com/javascript/` accessed June 6th, 2020
[3] `https://github.com/ReneU/session-viewer`
[4] `https://maptiks.com/` accessed June 6th, 2020

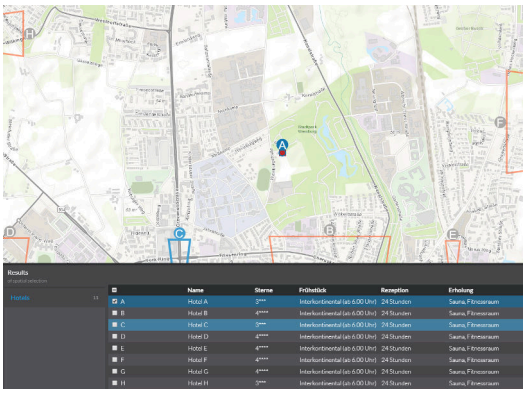


Figure 2 The first group was presented with off-screen indicators that reveal map entries which are not visible in the current viewport of the screen. Our optimized visualization extends the indicators with an alphabetical coding to simplify the assignment of individual values from a table.

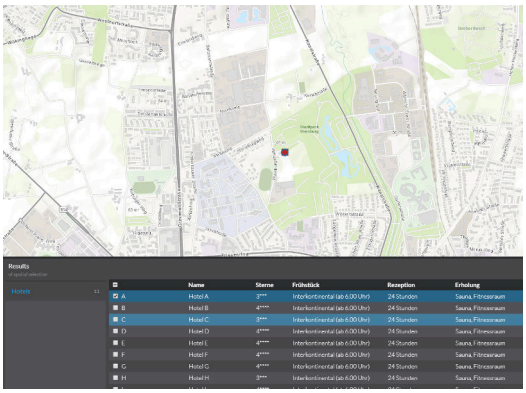


Figure 3 The second group was equipped with a common geovisualization that display map entries as simple dots on the map. The connection between map and table entries is not initially available, and users must hover or select table entries to mentally establish the connection.

aggregated statistics like the average task performance, number of activities, or conversion rates. In contrast, our approach allows analysts to identify the underlying reasons for these metrics by comparing users' individual interactions on an interactive map.

4 Evaluation

In order to evaluate our approach and the prototypical toolkit, we conducted two user studies. First, we instrumented a commercial WebGIS framework and captured map interactions from two different geo-visualization approaches (A/B testing) for a localization task with the data-collection component of our toolkit (user study). The UI for this task was minimal, and the designed task primarily required users to perform map interactions. By choosing a real-world WebGIS framework, we determined the feasibility of integrating our tool into an existing and large code base. The tested geovisualization is an approach for supporting map users in localizing map entries which are outside the currently visible extent. This scenario has been chosen to evaluate changes in the UI that could impact the users' map interactions. In our second study (expert study), usability experts evaluated the resulting datasets of the first study by using the session-viewer component of our toolkit. These experts were given the task to compare both datasets, identify meaningful patterns, and evaluate the usability after being introduced to the previous user study and our toolkit. We selected experts with experience in the field of usability and visual analytics to validate the outcome of our approach. For our studies, we formulated the following three hypotheses:

H0 Off-Screen Indicators increase the efficiency, effectiveness, and satisfaction of users.

H1 The identification of meaningful map interaction patterns via interactive visualizations is effective and comfortable.

H2 The interpretation of map browsing observations can generate useful and deep usability insights.

4.1 User Study: Off-Screen Indicators

For our first study, two groups of participants worked on the same localization task. They were asked to select hotels that are in a quiet location, close to a park and have a star rating of at least four. The WebGIS displayed hotels on the map and attributes, such as the hotel star rating, on a table below it. Participants had to use checkboxes in the table rows to add or remove hotels from the set of selected results. We created this task layout to force participants to make combined use of the map and the table. In total, 11 hotels were available on the map and in the table (A-K), four of them fulfilled the criteria (B, F, G, J). We chose the initial extent of the map view to show only a subset of all entries and, thus, made participants interact with the map via pan or zoom actions. The first group was presented with *Off-Screen Indicators* (OSI) [3]. OSI are a geovisualization type for map entries and consist of triangles whose sides can be traced and extrapolated to locate the off-screen object [5]. Due to the ability of the human brain to recognize shapes, users can estimate where the legs intersect and thus track the relative position of all map entries continuously as well as navigate precisely to the desired entry (Figure 2). Besides, the first group used an alphabetical coding to support the assignment of map entries and table rows. This coding was displayed as an attribute in the table and also next to the OSI on the map. The second group used a geovisualization that is common for dynamic maps and highlights map entries on the map with a symbol. These participants had to mentally match the tabular representation of entries to the ones on the map (Figure 3). In a between-group user study, we randomly assigned participants to one of the two groups.

We completed the implementation and testing of the instrumentation for the WebGIS framework with our data-collection component within less than one day. In total, we added less than 100 lines of code to the source files of the framework.

Participants. We opportunistically recruited 60 participants during a user conference of the tested WebGIS framework to participate in our user study. The primary criteria for participation in our study included basic knowledge and experience with WebGIS (i.e., participants were actual end-users). The resulting sample set of participants consisted of regular users with high levels of motivation and user expertise. We considered the total sample size of $n = 60$ sufficient for two different scenarios, based on recommendations for usability testing [19] and regarding collecting enough data to warrant its non-trivial inspection by usability experts.

Materials and Procedure. We conducted the study during the mentioned conference in a quiet area at the conference venue. Our simple usability setup consisted of a laptop computer, an external monitor, and a mouse that were used by the participants during testing. We completed all sessions within two days, with our setup remaining in the same configuration throughout this period.

Similar to the idea of a usability kiosk [18], we invited passers-by to participate in a 10-minute user study that investigated an experimental design for visualizing the relationship between data in a table and on the map. Before starting with the study, participants were also asked to read and sign an informed consent form about the anonymously collected data. Participants sat down in front of our setup that guided them through the required steps and automatically assigned them to one of our two groups. Next, participants were asked to rate their experience with GIS on a Likert scale based on the following statement: *I have experience in working with GIS* (1: strongly disagree, 5: strongly agree). After the actual task the short version of the User-Experience-Questionnaire (UEQ-S) [27] was filled out by

the participants. This questionnaire asks users to rate their experience based on eight pairs of terms that can later be used to calculate the hedonic, pragmatic and overall quality of the tested system (Likert Scale from 1 to 8). Participants could ask for clarification before pressing a button to start. All participants were able to finish their session with a set of selected hotels.

4.2 Expert Study: Session Viewer

In our second study, we validated the results of our approach by having actual usability experts work with the session-viewer component. During this study, the synchronized map views displayed the collected map interaction data of both groups from the previous user study next to each other. We chose a minimal stop duration of 3 seconds and a distance tolerance of 3 kilometers for the aggregation of user trajectories as these values represent the average values for all map interactions. Experts described, compared, and interpreted the data by using the provided analysis layers and reported their insights as well as their evaluation of the toolkit. We also asked experts to rate the precision, efficiency, comfort, and confidence of their results and the extraction process for each analysis layer. Last, we asked them to choose their preferred visualization for evaluating the usability of the WebGIS.

Participants. We recruited seven usability experts via a regional user experience meetup that aims to connect designers, developers, and researchers. Our criteria required participants to be familiar with GIS software and have experience with usability evaluations of UIs. This narrow definition of experts resulted in a small set of seven participants, though the size is still sufficient based on recommendations for expert reviews [19]. All participants had experience with conducting user studies, 63% had analyzed study results before, and 50% were familiar with creating concepts for usability evaluations. The average age of our participants was 37 ($\sigma = 5.83$), the average experience with GIS 10.6 years ($\sigma = 6.41$), and the average experience with usability evaluations 5.7 years ($\sigma = 6.74$). Participants reported using visual analytics tools in their job, i.e. Google Analytics, The R project, SPSS, and the Microsoft Suite. Independent of any specific tool, the selected experts reported an average of 5.5 years ($\sigma = 5.47$) of experience with visual analytics tools.

Materials and Procedure. The setup for our expert study consisted of two monitors that participants used to work with our session-viewer component and to write down comments. After an informed consent form was signed, we gave participants a questionnaire to enter demographic data as well as their experience with usability evaluations, GIS applications, and visual analysis tools. Next, we introduced them to the previously conducted user study, the concept of OSI, and provided an overview of the traditional task metrics of the user study (see *Result* section). Finally, we introduced the experts to the session-viewer, the overall concept of the synchronized map views and the individual analysis layers in detail by using the same explanation for every participant. After this introduction, we asked participants to describe the differences between both interaction datasets and possible reasons for the underlying user behavior by using our tool and focusing on these three aspects:

1. Spatial distribution of map interactions (extent, density, clusters).
2. "Zoom behavior" of users (order, frequency, zoom level).
3. Relation between map interactions and user experience (spatial correlations).

These aspects were chosen to address the intended purposes of each analysis layer. We asked participants to always prioritize correctness over speed in their answers and allowed them to state additional observations and underlying reasons. Finally, they rated each analysis layer

Table 1 Number of participants (n=60) that selected a hotel as fulfilling the defined criteria. Bold columns represent hotels that actually fulfilled the criteria.

	A	**B**	C	D	E	**F**	**G**	H	I	**J**	K
w/	3	**20**	2	1	4	**24**	**26**	2	11	**26**	12
w/o	0	**14**	0	4	3	**25**	**21**	0	19	**25**	10

on a Likert scale based on the following statement: *Spatial visualizations of [analysis layer] in the used tool allow me to make precise/efficient/comfortable/confident statements about the usability* (1: strongly disagree, 5: strongly agree). The average length of the expert study sessions was 72 minutes ($\sigma = 15.17$).

4.3 Results

Task Metrics and User Experience. For the preceding data collection phase, 60 conference attendees (44 male, 16 female) participated in our study. The mean age of these users was 38.3 ($\sigma = 10.13$), and their self-rated GIS experience on the Likert scale was 4.6 ($\sigma = 0.95$, scale: 1 to 5). The 30 participants who were working on the tasks with OSI (M = 123.5 seconds, SD = 48.6) compared to the 30 participants without OSI (M = 125.9, SD = 71.8) did not demonstrate significantly better task completion times (t(55) = 0.147, p = .8836). About 67% of users in the scenario with OSI selected all hotels that fulfilled the required criteria (Table 1). In the scenario without OSI, the success rate was substantially lower at 47%.

Based on the user experience ratings from our questionnaire, we calculated the pragmatic, hedonic, and overall quality for both scenarios. The results are mapped to ranges with a scale between -3 (horribly bad) and +3 (extremely good). There was no significant difference in the overall scores between with OSI ($\mu = 0.738; \sigma = 1.032$) and without OSI ($\mu = 0.383; \sigma = 0.879$) conditions ($t(58) = 1.432, p = 0.157$). Although the pragmatic value in the condition with OSI represented a positive evaluation ($\mu = 0.925; \sigma = 1.251$), according to Schrepp et al. [27] the difference to the group without OSI ($\mu = 0.558; \sigma = 1.15$) was not significant ($t(58) = 1.182, p = 0.242$). Finally, there was also no significant difference in the hedonic scores between with OSI ($\mu = 0.550; \sigma = 1.21$) and without OSI ($\mu = 0.208; \sigma = 0.933$) conditions ($t(58) = 1.225, p = 0.226$).

In summary, the task was finished with comparable mean completion times (efficiency) by all groups (contrary to our initial hypothesis H0). The results from the UEQ-S did not lead to a significant higher user experience (satisfaction). However, the success rate (effectiveness) for the scenario with OSI was 20% higher than in the scenario without OSI.

Expert Observations and Evaluations. In our subsequent expert study, experts stated that the toolkit helped them to understand that many of the users' map interactions formed clusters around the hotel locations in both scenarios (e.g., Expert 1 and 4). It was clear to the experts that the underlying reason was the users' aim to check the surroundings for the required criteria (E4). However, four out of our seven experts (E2, E3, E6, E7) reported more dense clusters of map interactions in the scenario without OSI (Figure 1). Experts also made use of the capabilities of the single metric layer to gain deeper insights into the users' intentions. For example, the experts reported that the clustered map interactions of the group without OSI occurred on a higher zoom level (i.e., revealed more map details) compared to the remaining map interactions of the same group (E2, E3, E6, E7). In contrast, they said that map interactions of users in the group with OSI generally preferred panning over

zooming ("constant zoom levels") on high zoom levels ("worked more focused"), independent of the proximity to hotels (E2, E4, E5, E6, E7). Thus many experts concluded that their map interactions were less clustered and more evenly distributed in the study area (E2, E3, E4, E5, E6, E7). The majority of experts (four out of seven) stated that this difference was caused by the OSI, as users in the group without OSI have to zoom out once in a while to get an overview of remaining hotels, whereas users in the group with OSI already know the location of the next hotel whose surroundings they want to check (E2, E4, E6, E7). As a result, E2 reported that working with higher zoom levels requires more panning interactions as the distance traveled on the map is smaller for each interaction. The same expert stated that these differences in the users' map interactions have proven that they adapt their approach based on the given geovisualization although they might require some time to learn the new concept. By investigating the user interaction count and the time spent since session start, E1 and E5 detected that users in the scenario with OSI spent more time and performed more interactions close to the initial extent of the session. E2 elaborated on this insight and concluded that while most users with OSI were able to complete the task faster, some users of this group required notable more time because of the learning phase that the expert identified with our visual analytics component.

Usability experts found several correlations between traditional metrics and the users' experience ratings by using the relationship layer. For both scenarios, high overall quality ratings correlated with low zoom levels (E1, E7). The same experts detected that users with OSI often gave low UX ratings if they had to zoom at the hotels while the UX ratings of users without OSI were generally higher around hotels even if they had to zoom in. These experts thus concluded that the OSI generally allowed users to work more efficiently by performing fewer zoom interactions although hotels which required them to zoom in possibly reduced their performance and thus harmed their experience. However, besides this hypothesis, the insights from the relationship layer were few and three of our seven usability experts stated that they could not detect any meaningful patterns.

The aggregated user trajectories layer allowed experts to detect additional differences in the users' behavior and suggest usability improvements. Looking at the network of key movements of the group that made use of OSI, our experts identified areas that contain potential hotels but no interactions of these users. Some experts thus concluded that these users were able to exclude hotels from their search without having to navigate to their location in the first place, by using the alphabetical coding (E1, E2, E4, E5, E6). Besides, the alphabetical coding also helped users to advance more efficiently from the initial extent to the surrounding hotels (E1).

All experts were able to retrieve insights from our session-viewer component and the included analysis layers. E1 stated that "the display and synchronization of the two map views is the ideal solution for comparing A/B testing results for WebGIS". Most experts preferred to work with the single metrics analysis layer as it "was easy to use and returned the most insights" (E1). The visualization of relationships between traditional task metrics and UX ratings achieved the lowest scores (Table 2) as it was "unfamiliar, complex and overwhelming" (E3) while the expert scores of the aggregated user trajectories were similar to those of the single task metrics layer. Three experts understood the trajectories as complementing these metrics by providing a summarized view with fewer details (clustered map interactions) but more context information (movements and change of zoom level). The recruited experts also mentioned that they would like to spend more time with the tool and could think of using it for other scenarios (E2, E4).

Table 2 Average expert ratings for analysis layers (scale: 1 to 5).

	Single Metric	Relationship	Trajectory
Precision	3.58	2.72	3.5
Efficiency	2.58	2.58	3.5
Comfort	3.86	2.72	4
Confidence	3.58	2.58	3.67

5 Discussion

Prior studies on mobile devices revealed that OSI result in faster task completion times and more accurate scores for localization tasks compared to arrow-based or "Halo" interfaces [3, 5]. In contrast, we could not identify significant improvements of the efficiency or satisfaction for users of OSI, which are extended with an alphabetical coding in our desktop scenario (H0).

The visual analysis of map interactions showed that WebGIS users navigate more efficiently as they do not have to zoom out to locate map entries and can omit uninteresting entries earlier. Therefore, we expect improved efficiency over more extended periods if OSI are applied. Besides, the gained insights could result in concrete steps for improving the usability of the evaluated WebGIS. For example, the analysis of the spatial distribution showed that many map interactions are clustered around map entries that were relevant for the task. Further iterations of the adapted geovisualization could include "shortcuts" that allow users to click on indicators to zoom in to the corresponding map entries and thus improve the efficiency. The analysis of the spatial aspects of map interaction patterns shows that our results go beyond traditional metrics and yield deeper insights.

In terms of visualization design, expert ratings (Table 2) show that our choices were successful in providing representations and interaction modes that allow experts to effectively and comfortably identify meaningful map interaction patterns from the given data (H1). However, the insights from the relationship layer were few, the expert ratings for this layer low and three of our seven usability experts stated that they could not detect any meaningful patterns. These ratings could be due to the population that we assessed, with no previous familiarity and only small training with our toolkit, which may have constrained the understanding of this particular visualization. Nevertheless, we are confident that this aspect does not affect the validity of the evaluation. One evidence for this is that experts generated confident and specific usability insights (Table 2, H3).

The expert ratings as well as the provided feedback also showed that the discovery of usability insights benefits from the application of visual analytics. In open comments, several experts reported that they would like to apply the tool in their role for analyzing results from usability studies because of, for example, the immersive experience, easy and interactive manipulation of analysis parameter as well as the immediate and visual feedback. We thus conclude that our approach has the potential to help gaining deeper insights into the usability of WebGIS in real-world scenarios. However, additional data is required to confirm hypotheses that are made via our tool. We thus consider our approach as complementary to existing usability methods, such as think-aloud protocols, experience sampling, or videotaping. By addressing open questions or validating observations from these methods our approach allows decision-makers to conduct more focused studies.

The presented approach is highly scalable to scenarios with many more users as the data is collected automatically and the aggregated user trajectories prevent visual cluttering. We are confident that our toolkit is also capable of providing usability insights for other types of map applications. Although our evaluation only covered four basic interactions, the interaction logging and visualization components can be easily extended to handle other

events as well. In particular, any interactive map application with a two-dimensional view and a given task that focuses on the map element could be instrumented for evaluation via our tool.

Limitations. In terms of instrumentation, one limitation of our evaluation is the fact that one of the authors of this paper augmented the WebGIS source code rather than someone outside the project team. The identification of the relevant code sections and the integration of our data-collection component requires programming skills and thus might represent an obstacle for usability experiments. However, the simple interface of our toolkit should make the instrumentation process straightforward as it does not require extensive knowledge about the system's architecture. Hence, we expect the required effort for instrumenting other WebGIS applications to be similar to the workload reported here.

The selected sample for our study presents an additional limitation of our results. Our participants were experienced GIS users, the UI was reduced to a minimum and the given localization task was short. We were thus not able to test how more data may impact the performance and insights of analysts when working with our toolkit. However, the interactions for longer tasks could be broken down into smaller semantic chunks for analysis via our toolkit. As the experience level did not vary much the interaction pattern were fairly consistent. We plan to address this shortcoming in future work by testing less experienced users with more complex systems and tasks.

Opportunities. Finally, we think that recent developments in Artificial Intelligence research could complement our approach very well but not replace the identification of usability issues by humans entirely. The adaption of the trajectory summarization algorithm for our toolkit leads to promising results and represents the first step in this direction. However, embedding the discovered patterns and anomalies in the task's context requires deep knowledge about the scenario and the users, which is not captured by approaches such as machine learning. In contrast, humans can combine the aggregated data with their knowledge about the participants, tasks, and the visualized information.

6 Conclusion

In this work, we proposed and evaluated an approach that applies visual analytics to map interaction data, aiming to generate deep insights into the usability of WebGIS user interfaces. We reviewed previous literature in usability evaluations for WebGIS and visual analytics and implemented a holistic toolkit that facilitates the conduction and evaluation of usability studies for WebGIS via interactive visualizations.

Even though our work can only be considered a first step into the investigation of the benefits of visual analytics for this domain, analysts were able to generate plausible explanations for differences in descriptive statistics via our tool. In a realistic WebGIS, they compared users' map interactions and user experience ratings resulting from two geovisualizations. The visual analysis of this data facilitated the understanding of the analysts by showing them that one group of WebGIS users navigated more efficiently as they did not have to zoom out to locate map entries and could omit uninteresting entries earlier, which eventually led to higher satisfaction (though not shorter completion times). In addition, they were able to identify learning phases in users sessions that explain initially longer task completion times but may vanish after users get more experienced.

The next steps planned for our research include further detailed evaluations with more complex WebGIS UIs and the application of our approach in a real-world scenario.

References

1 Gennady Andrienko, Natalia Andrienko, and Stefan Wrobel. Visual analytics tools for analysis of movement data. *SIGKDD Explor. Newsl.*, 9(2):38–46, December 2007. doi:10.1145/1345448.1345455.

2 Richard Atterer and Albrecht Schmidt. Tracking the interaction of users with ajax applications for usability testing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 1347–1350, New York, NY, USA, 2007. ACM. doi:10.1145/1240624.1240828.

3 Patrick Baudisch and Ruth Rosenholtz. Halo: A technique for visualizing off-screen objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 481–488, New York, NY, USA, 2003. ACM. doi:10.1145/642611.642695.

4 Thore Fechner, Dennis Wilhelm, and Christian Kray. Ethermap: Real-time collaborative map editing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3583–3592, New York, NY, USA, 2015. ACM. doi:10.1145/2702123.2702536.

5 Sean Gustafson, Patrick Baudisch, Carl Gutwin, and Pourang Irani. Wedge: Clutter-free visualization of off-screen locations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 787–796, New York, NY, USA, 2008. ACM. doi:10.1145/1357054.1357179.

6 Jason I Hong, Jeffrey Heer, Sarah Waterson, and James A Landay. Webquilt: A proxy-based approach to remote web usability testing. *ACM Trans. Inf. Syst.*, 19(3):263–285, July 2001. doi:10.1145/502115.502118.

7 Joseph H. Hoover, Paul C. Sutton, Sharolyn J. Anderson, and Arturo C. Keller. Designing and evaluating a groundwater quality internet gis. *Applied Geography*, 53(Complete):55–65, 2014. doi:10.1016/j.apgeog.2014.06.005.

8 Jens Ingensand and François Golay. Remote-evaluation of user interaction with webgis. In Katsumi Tanaka, Peter Fröhlich, and Kyoung-Sook Kim, editors, *Web and Wireless Geographical Information Systems*, pages 188–202, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. URL: http://dl.acm.org/citation.cfm?id=1966271.1966292.

9 Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. *Visual Analytics: Definition, Process, and Challenges*, pages 154–175. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi:10.1007/978-3-540-70956-5_7.

10 Peter Kiefer, Ioannis Giannopoulos, Vasileios Athanasios Anagnostopoulos, Johannes Schöning, and Martin Raubal. Controllability matters: The user experience of adaptive maps. *GeoInformatica*, 21(3):619–641, July 2017. doi:10.1007/s10707-016-0282-x.

11 Menno-Jan Kraak. The role of the map in a web-gis environment. *Journal of Geographical Systems*, 6(2):83–93, June 2004. doi:10.1007/s10109-004-0127-2.

12 Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. *Research Methods in Human-Computer Interaction*. Wiley Publishing, 2010.

13 Xia Li, Arzu Çöltekin, and Menno-Jan Kraak. Visual exploration of eye movement data using the space-time-cube. In Sara Irina Fabrikant, Tumasch Reichenbacher, Marc van Kreveld, and Christoph Schlieder, editors, *Geographic Information Science*, pages 295–309, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. doi:10.1007/978-3-642-15300-6_21.

14 María-Jesús Lobo, Emmanuel Pietriga, and Caroline Appert. An evaluation of interactive map comparison techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3573–3582, New York, NY, USA, 2015. ACM. doi:10.1145/2702123.2702130.

15 Eoin Mac Aoidh, Michela Bertolotto, and David C. Wilson. Analysis of implicit interest indicators for spatial data. In *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, GIS '07, pages 47:1–47:4, New York, NY, USA, 2007. ACM. doi:10.1145/1341012.1341071.

16 Steven M. Manson, Len Kne, Kevin R. Dyke, Jerry Shannon, and Sami Eria. Using eye-tracking and mouse metrics to test usability of web mapping navigation. *Cartography and Geographic Information Science*, 39(1):48–60, 2012. doi:10.1559/1523040639148.

17 Jon May and Tim Gamble. Collocating interface objects: Zooming into maps. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 2085–2094, New York, NY, USA, 2014. ACM. doi:10.1145/2556288.2557279.

18 Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

19 Jakob Nielsen and Thomas K. Landauer. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 206–213, New York, NY, USA, 1993. ACM. doi:10.1145/169059.169166.

20 Hartmut Obendorf, Harald Weinreich, and Torsten Hass. Automatic support for web user studies with scone and tea. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, pages 1135–1138, New York, NY, USA, 2004. ACM. doi:10.1145/985921.986007.

21 Alenka Poplin. How user-friendly are online interactive maps? survey based on experiments with heterogeneous users. *Cartography and Geographic Information Science*, 42(4):358–376, 2015. doi:10.1080/15230406.2014.991427.

22 Alenka Poplin, Wendy Guan, and Ben Lewis. Online survey of heterogeneous users and their usage of the interactive mapping platform worldmap. *The Cartographic Journal*, 54(3):214–232, 2017. doi:10.1080/00087041.2016.1229248.

23 Salvatore Rinzivillo, Dino Pedreschi, Mirco Nanni, Fosca Giannotti, Natalia Andrienko, and Gennady Andrienko. Visually driven analysis of movement data by progressive clustering. *Information Visualization*, 7(3-4):225–239, 2008.

24 Robert E. Roth, Richard G. Donohue, Carl M. Sack, Timothy R. Wallace, and Tanya M. A. Buckingham. A process for keeping pace with evolving web mapping technologies. *Cartographic Perspectives*, 0(78):25–52, January 2015. doi:10.14714/CP78.1273.

25 Robert E. Roth and Alan M. MacEachren. Geovisual analytics and the science of interaction: an empirical interaction study. *Cartography and Geographic Information Science*, 43(1):30–54, 2016. doi:10.1080/15230406.2015.1021714.

26 Robert E. Roth, Kevin S. Ross, and Alan M. MacEachren. User-centered design for interactive maps: A case study in crime analysis. *ISPRS International Journal of Geo-Information*, 4(1):262–301, 2015. URL: http://www.mdpi.com/2220-9964/4/1/262.

27 Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. Design and evaluation of a short version of the user experience questionnaire (ueq-s). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(6):103–108, 2017. doi:10.9781/ijimai.2017.09.001.

28 Carol Traynor and Marian G. Williams. Why are geographic information systems hard to use? In *Conference Companion on Human Factors in Computing Systems*, CHI '95, pages 288–289, New York, NY, USA, 1995. ACM. doi:10.1145/223355.223678.

29 Nancy Tsai, Beomjin Choi, and Mark Perry. Improving the process of e-government initiative: An in-depth case study of web-based gis implementation. *Government Information Quarterly*, 26(2):368–376, 2009. doi:10.1016/j.giq.2008.11.007.

30 René Unrau and Christian Kray. Usability evaluation for geographic information systems: a systematic literature review. *International Journal of Geographical Information Science*, 33(4):645–665, 2019. doi:10.1080/13658816.2018.1554813.

31 René Unrau, Morin Ostkamp, and Christian Kray. An approach for harvesting, visualizing, and analyzing webgis sessions to identify usability issues. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '17, pages 33–38, New York, NY, USA, 2017. ACM. doi:10.1145/3102113.3102122.

Volume from Outlines on Terrains

Marc van Kreveld
Department of Information and Computing Sciences, Utrecht University, The Netherlands
m.j.vankreveld@uu.nl

Tim Ophelders
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
t.a.e.ophelders@tue.nl

Willem Sonke
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
w.m.sonke@tue.nl

Bettina Speckmann
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
b.speckmann@tue.nl

Kevin Verbeek
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
k.a.b.verbeek@tue.nl

Abstract

Outlines (closed loops) delineate areas of interest on terrains, such as regions with a heightened risk of landslides. For various analysis tasks it is necessary to define and compute a volume of earth (soil) based on such an outline, capturing, for example, the possible volume of a landslide in a high-risk region. In this paper we discuss several options to define meaningful 2D surfaces induced by a 1D outline, which allow us to compute such volumes. We experimentally compare the proposed surface options for two applications: similarity of paths on terrains and landslide susceptibility analysis.

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

Keywords and phrases Terrain model, similarity, volume, computation

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.16

Funding Marc van Kreveld, Tim Ophelders, Willem Sonke, Bettina Speckmann, and Kevin Verbeek are supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 612.001.651 (M.v.K.), no. 639.023.208 (T.O., W.S., and B.S.), and no. 639.021.541 (K.V.).

1 Introduction

Digital elevation data, representing a terrain by DEMs, triangulations, or contour maps, are one of the main types of spatial data. Mathematically speaking, a terrain is a function that maps a 2D point to a value that represents elevation, and a terrain model is a finite representation of such a function. Terrains are analyzed in many different ways, including slope and aspect analysis, viewshed analysis, and natural disaster assessment. Terrain analysis is of prime importance in physical geography, urban planning, and disaster management.

Volume is an important measure used in terrain analysis, for example, to describe the amount of water in a lake, of ice in a glacier, or of contaminated soil in the ground [4, 6, 10, 16]. Geometrically, such a volume lies between two 2D surfaces. In many scenarios, one of the surfaces (frequently the upper one) is clearly determined, but the other surface needs to be (re-)constructed in a meaningful way. Such a reconstruction can, for example, be based on depth measurements (echo sounding). In other situations, however, a suitable surface needs

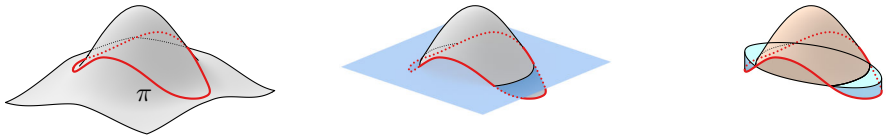


Figure 1 Terrain with path π (left); a base surface (center); earth ( ) and and air ( ) (right).

to be constructed from nothing more than an outline: a closed path (loop) on the surface of the terrain. This outline need not be a contour: its elevation may vary along the outline.

For example, suppose that linear features (ridges, rivulets) or paths (hikers' tracks) on a terrain are to be clustered for further analysis. One could project the paths onto the xy-plane and use an existing polyline similarity measure, like the Hausdorff distance, Fréchet distance, or area in between. However, such measures do not take into account that there is relief between the paths: two straight, parallel paths on a flat terrain may be considered more similar than two paths with the same distance in the projection, but with a ridge in between. Hence the volume of the terrain between the paths might be a better indicator of similarity.

Consider a second example from landslide susceptibility analysis. Here several factors play a role, which can be described by geological, hydrological, land cover, and morphological variables [17]. Thanks to LiDAR and SRTM, many morphological features can be assessed automatically, without a human visiting the area in question. For example, the volume of earth on a slope, rising above a certain region (mapping unit [3]), can be estimated via methods as discussed in this paper and subsequently analyzed for risk of detachment.

Our input is a terrain and an outline, such as the concatenation of two paths or the outline of a region. We wish to compute a meaningful 2D *base surface* induced by the outline so that we can determine a volume of earth above it. In other words, we start with a (1D) outline, and define a (2D) base surface to compute a (3D) volume (see Fig. 1). A given outline may also enclose a significant dent, so there could be air below the base surface and above the terrain surface. In some applications the volume of this air may be relevant. Hence, we will consider not only the earth above a base surface but also the air below it.

In prior work, we proposed several options to define meaningful base surfaces from an outline [20]. In this paper we extend our earlier work in two ways. First, we show how to actually compute the base surfaces and the corresponding volumes from an outline, that is, we describe algorithms that realize our earlier definitions. Second, we experimentally compare the proposed surface options for the two aforementioned application examples: similarity of paths on terrains and landslide susceptibility analysis.

Results and organization. Our input consists of a two-dimensional surface T embedded in three-dimensional space, representing a terrain, and an outline π, or two paths π_1 and π_2 that share their endpoints but are otherwise disjoint. In Section 2 we review the options for base surfaces which we proposed in prior work [20]. We first describe three possibilities for the most basic of surfaces, namely planes, and argue how to place them optimally. Second, we consider three options for more general base surfaces which do contain π, or π_1 and π_2 (which usually cannot be the case with planes). In Section 3 we describe how to compute the base surfaces as well as the volumes of earth and air between the input surface T and the base surfaces. Finally, in Section 4 we investigate the different plane and surface options experimentally, using several real-world datasets as well as informative synthetic data. We observe the results for the different choices and discuss their characteristics, with respect to both similarity measures (paths) and terrain morphology (landslides).

Related work. Similarity measures for linear features (shapes) have been considered in a variety of contexts. Popular geometric measures include the Hausdorff distance, the Fréchet distance, the area-of-symmetric difference, the Wasserstein distance (Earth Mover's Distance), and the turn function distance. Also in GIS, shape similarity measures have been used, for example, in cartographic generalization. Here a city outline or river shape will be displayed with less detail on a smaller-scale map, while still capturing the overall shape well. One needs to measure the similarity between the original shape and the generalized shape to determine how well the generalization still resembles the original [13, 19]. Furthermore, similarity measures are used for trajectory similarity [21], landscape ecology [2], (urban) property analysis [7], spatio-temporal processes [12], and retrieval in spatial databases [15]. There has also been some recent interest in semantic similarity [8, 18], focusing more on cognitive than on geometric aspects of similarity.

There is a huge body of research treating landslide susceptibility analysis, surveyed in [17] (and much earlier in [22]). Morphological factors are an important indicator, but generally, only simple morphological variables are considered, due to their availability in GIS [17]. One notable exception is the work by Völker [23] who uses a 6-step approach utilizing tension surfaces fitted over nearby profiles to reconstruct the ocean floor prior to a submarine landslide. His method strongly relies on the assumption that the terrain before the landslide at the position of the landslide is similar to the surrounding area. As such Völker's methodology does not apply to the predictive analysis of morphological features for landslide risk assessment. To analyze unstable landforms like escarpments or features in poised position [14], we need more advanced terrain shape analysis tools, and we hope that volumes from outlines can contribute to these developments.

2 Preliminaries

We review the options for base surfaces induced by an outline which we proposed in earlier work [20]. Specifically, we describe three linear surfaces (planes) and three general surfaces.

The horizontal averaging plane (**HAP**) is the horizontal plane whose elevation is the average of the elevations of the outline. It minimizes the sum-of-squared vertical distances from the outline to the plane, over all horizontal planes. The regression plane (**RP**) is the (non-horizontal) plane that minimizes the sum-of-squared vertical distances from the outline to the plane. As a third plane choice, we consider the horizontal plane $z = c$ that minimizes the sum of absolute differences: $\int_{p\in\pi} |\, z_p - c \,|\, \mathrm{d}p$. We refer to this plane as the minimizing horizontal plane (**MHP**); it is located at the median height of π. These planes are – within the outline – typically partly above and partly below the surface. So we can measure the volume of earth above and of air below the plane.

General base surfaces can be of various types. For example, we could use a constrained Delaunay triangulation on the vertically projected outline and lift it to 3D, similar to the construction of a triangulated irregular terrain from contour lines. Alternatively, we could use a minimum tension surface or a minimum curvature surface. However, neither of these choices is particularly well motivated by our intended applications; we feel that a minimum area surface (**MAS**) is a more natural choice. For example, in the context of landslide risk assessment, a minimum area surface as a base surface represents the case where the area, and thus the friction between the moving and not moving earth, is minimized.

The four base surfaces described so far can be used both for a single outline and for two paths. We next describe a choice of base surface that applies only to two paths. This surface readily lends itself to measure the similarity between the two paths in a manner which takes the volume of the relief between them into account. Specifically, the so-called

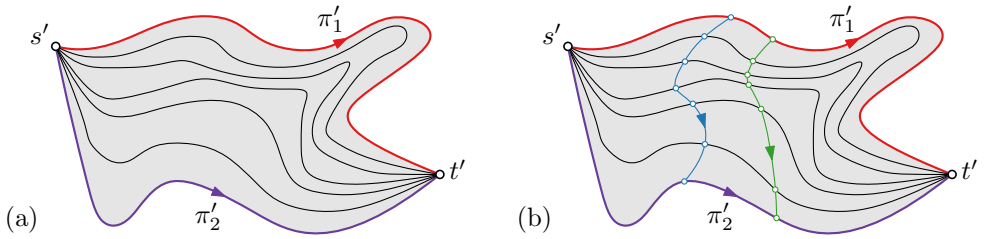


Figure 2 (a) A monotone isotopy illustrated by intermediate paths of the morph between π_1' and π_2'. **(b)** Two transverse curves of this isotopy. The transverse curve at parameter 0 is simply s' and the transverse curve at parameter 1 is t', the projections of s and t (taken from [20]).

water flow surface (**WFS**) models the ease with which one path can morph into the other. It is motivated by morphological processes shaping channels in braided rivers [5, 11]. The WFS models the minimum amount of earth that must be removed for a path to change its course from π_1 to π_2. The WFS cannot be symmetric, for example, if π_1 lies further uphill than π_2, and is hence defined by an ordered pair (π_1, π_2). Note that the volume-based distance function implied by the WFS is not a metric.

The transformation from π_1 into π_2 is essentially a morph. Such a morph should be smooth and should not "double-back" on itself, that is, it should be a *monotone isotopy* (see Fig. 2(a)) in the 2D plane, morphing between the 2D projections π_1' and π_2' of π_1 and π_2. To construct the surface we need to assign suitable elevations to the points on the paths in the morph between π_1' and π_2'. We can view the curves π_1', π_2', and the ones in between in the morph, as parametrized curves where at parameter 0 we are at the common start s', and at parameter 1 we are at the common end t' (see Fig. 2(b)). Points that occur at the same parameter form transverse curves. We now choose the WFS to be the surface on or below the terrain T such that a monotone isotopy exists for which all transverse curves are monotonically decreasing, and among these, the one that has the smallest volume between T and the WFS. A more extensive description and motivation can be found in [20].

We illustrate the difference between the WFS and the RP by a simple example (see Fig. 3). Assume that paths π_1 and π_2 lie in a plane, which is the RP for these paths. The RP gives a volume above it and below T that is the whole bump above the plane (see Fig. 3(b)). The WFS lies higher and defines a smaller volume, namely the volume of the bump above the saddle point (see Fig. 3(c)). Hence, a sloped terrain with some roughness between π_1 and π_2, but no local maxima, gives a volume of 0 between T and the WFS. The RP would give some volume based on the summed volumes of the roughness spots above the regression plane. The WFS distance for the example in Fig. 3(d) is the volume between T and a horizontal plane through s and t on one side of the valley.

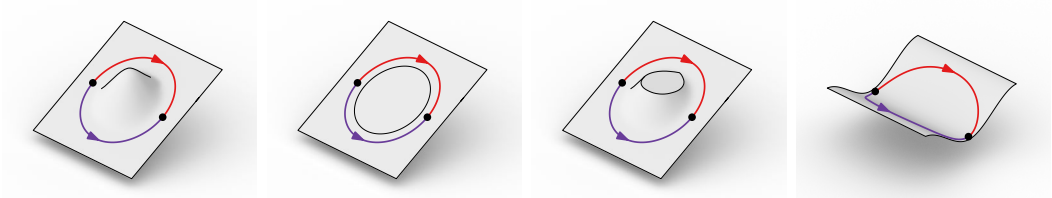


Figure 3 (a) A sloped hill with transverse paths. **(b)** Removing parts by the RP. **(c)** Removing parts by the WFS (above the saddle point) **(d)** A valley with longitudinal paths (from [20]).

As noted, the WFS is asymmetric: exchanging the roles of π_1 and π_2 leads to a different base surface and a different measured volume. Our last base surface is a symmetric version of the WFS which is again defined for an outline. To define the *symmetric flow surface* (**SFS**) we consider paths on or below T from any point p on or below T inside the outline π to the outline π. The SFS is the 2D boundary of the union of all points p for which there is a path on or below T which monotonically increases towards the 1D boundary π. Since these paths can choose any point on π as their destinations, instead of only a specific part (π_1 or π_2), the SFS always bounds less volume than the WFS. The SFS is the same as the WFS in Fig. 3(c), but bounds a volume of 0 in Fig. 3(d), unlike any of the other base surfaces we described.

3 Computing base surfaces and volumes

In this section we describe how to compute the base surfaces as well as the volumes of earth and air between the input surface T and the base surfaces. Recall that our input consists of a two-dimensional surface T, representing a terrain, and an outline π, or two paths π_1 and π_2 that share their endpoints but are otherwise disjoint. For all base surfaces but WFS, we first concatenate π_1 and π_2 into an outline π. We assume that we can model T as a height function $h\colon \mathbb{R}^2 \to \mathbb{R}$ that maps a geographic position (x, y) to its corresponding height value. We further assume that T is represented by a TIN, that is, the terrain model consists of vertices and edges forming triangles, where every vertex v has a position and a height value associated with it, and heights are interpolated linearly over edges and triangles. The outline π by extension is a simple polygonal boundary that lies on the terrain surface. Our methods apply to other elevation models than TINs, such as DEMs and spline surfaces, but some adaptations are needed.

Computing base surfaces. Computing the planes HAP, RP, and MHP from the input is straightforward. We interpret the vertices of the outline π as 3D points, and compute these planes with standard methods.

The base surfaces MAS, WFS, and SFS are functions $\mathcal{B}\colon \mathbb{R}^2 \to \mathbb{R}$. For ease of computation, we assume that each base surface $\mathcal{B}$ is also represented by a TIN, and the projection onto the (x, y)-plane coincides with that of T. In other words, $\mathcal{B}$ has the same vertices as T, but the heights of these vertices may differ from those in T. The minimal area surface (MAS) cannot be represented exactly by a TIN because the surface is curved, but we believe that using the same TIN as T provides an approximation that is comparable to how well T approximates the real-world terrain. Computing an approximation to the MAS is computationally intensive. For a given outline π, we start with a surface based on a weighted average of the vertex elevations defining the outline π (or π_1 and π_2); the weight for a vertex is proportional to $1/d^2$ where d is the distance to that vertex. We further optimize using gradient descent, until a (local) minimum is reached. This is our approximation of the MAS.

For WFS or SFS, the base surface often overlaps with the terrain T, except where locally maximal parts are cut off by a horizontal plane at the elevation of a saddle. The boundary of the base surface uses vertices on the contour line of the saddle. We can therefore represent these base surfaces implicitly (but still exactly), using annotations in the original terrain T. To know at which saddles inside π (or $\pi_1 \cup \pi_2$) we need to trace the contour lines, we use a *highest path tree* which represents all highest paths towards π or π_1. A highest path tree can be computed efficiently [1, 11].

Computing volumes. The volume V of earth above the base surface $\mathcal{B}$ for an outline π, for all six base surfaces, is defined as follows (see Fig. 1):

$$V(\pi) = \int_D \max\big(0, h(x,y) - \mathcal{B}(x,y)\big)\,\mathrm{d}x\,\mathrm{d}y. \tag{1}$$

Here D is the domain of the function that corresponds to the part of the terrain inside π. Alternatively, we can measure the volume V' of air below the base surface:

$$V'(\pi) = \int_D \max\big(0, \mathcal{B}(x,y) - h(x,y)\big)\,\mathrm{d}x\,\mathrm{d}y. \tag{2}$$

We now describe how we can compute these integrals in practice. For that we use the fact that the terrain T is represented as a TIN.

Assuming that T and $\mathcal{B}$ share the same TIN (with different heights), we can compute the integral in Equation 1 (or similarly, in Equation 2) as follows. We split up the domain D into triangles corresponding to the projections of the triangles in T onto the (x,y)-plane. For each such triangular domain, the function $h(x,y) - \mathcal{B}(x,y)$ is linear by the definition of a TIN, and this function is completely determined by the values of $h(x,y) - \mathcal{B}(x,y)$ at the vertices of the triangle. Let $h_1 \le h_2 \le h_3$ be the corresponding values at the vertices of the triangle. We will simply refer to these values as height.

To compute the integral for a single triangular domain, we need two basic building blocks, namely the volume of a prism and the volume of a pyramid:

$$V_{\text{prism}} = Ah; \qquad V_{\text{pyramid}} = Ah/3,$$

where A is the area of the base of the prism/pyramid and h is the height of the prism/pyramid. We now consider several cases. First of all, we assume that $h_3 > 0$ (otherwise the integral is 0) and that $h_1 < 0$ (otherwise we can simply add a prism with height h_1 to the volume). The following cases remain:

Case 1 ($h_1 < h_2 < h_3$): We cut up the triangle into two triangles by cutting it at height h_2. Note that for the two resulting triangles, two vertices share the same height; we can use Case 2, 3 or 4 to compute their corresponding volumes separately.

Case 2 ($0 = h_1 = h_2 < h_3$): The remaining volume is a pyramid.

Case 3 ($h_1 = h_2 < 0 < h_3$): We cut the triangle at height 0 and end up in Case 2.

Case 4 ($h_1 < 0 < h_2 = h_3$): We add a pyramid on top of the triangle to form a prism. We then compute the volume of the prism with height h_3 and subtract the volume of the pyramid, except for the part of the pyramid that is below height 0 (see Fig. 4). The volume of the tip of this pyramid can be computed as in Case 3.

To compute the integral in Equation 1 for the WFS or the SFS, we take a similar approach, using the implicit representation of $\mathcal{B}$ in T. The computations can be simplified because WFS and SFS are never above T.

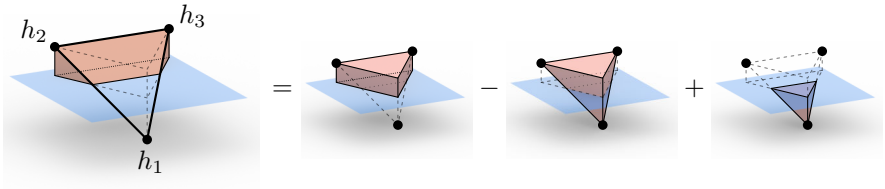


Figure 4 Computing the volume below the triangle and above the zero plane (blue) in Case 4.

4 Experiments

We implemented our methods and investigated the different plane and surface options experimentally, using several real-world datasets as well as informative synthetic data. In this section we observe the results for the different choices of base surfaces and discuss their characteristics, with respect to both similarity measures (Section 4.1) and terrain morphology (Section 4.2).

4.1 Volume-based path similarity on terrains

We used two extracts from the world-wide SRTM elevation dataset: one of the area around Mont Blanc on the border of France and Italy, and one of Grampians National Park, Australia. Both extracts were taken from the void-filled SRTM data sets produced by CIAT [9]. In both datasets we manually drew three input paths π_1, π_2, and π_3 that are of interest. In the Mont Blanc dataset we drew two paths π_1 and π_3 through valleys, and one path π_2 that goes higher along the mountain (but not over the peak, see Fig. 5(a)). In the Grampians dataset we drew two paths π_1 and π_3 along mountain ridges, and one path π_2 in the valley between (see Fig. 5(b)). We also constructed a small set of synthetic datasets that clearly illustrate the features of our base surfaces and the associated volumes.

To measure the volume-based similarity between two paths, we define the distance between them with respect to a particular base surface as the volume of earth and possibly air between this base surface and the terrain. The distances (the computed volumes) are shown in Tables 1, 2, and 3; we express all values in $10^9\,\mathrm{m}^3$.

Mont Blanc dataset. In Table 1 we show the results for the Mont Blanc dataset. The first column shows the distance between π_1 and π_3, the second column the distance between π_1 and π_2, and the third column the distance between π_2 and π_3. Every row in the table shows the result for a different base surface. For each combination of base surface and pair of paths we show: (1) a grayscale figure showing the height of the base surface, (2) a color figure showing the amount of earth (brown) or air (blue) measured with respect to the base surface, and (3) the corresponding volumes measured for earth (⛰) and air (☁), if applicable.

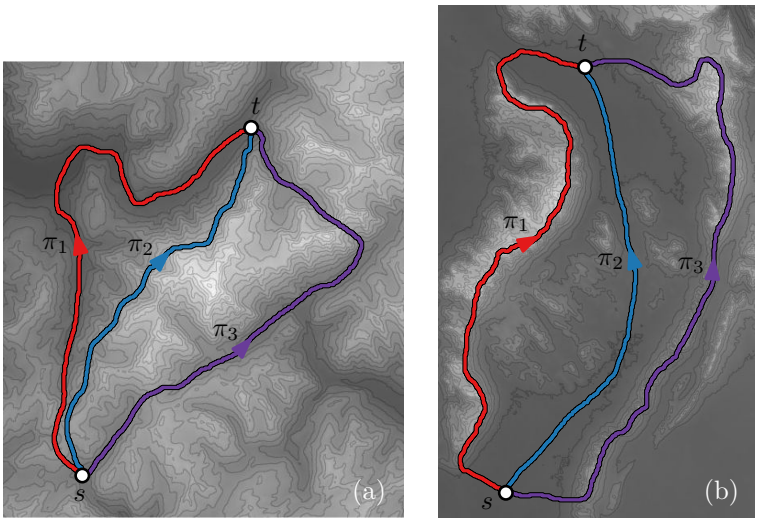
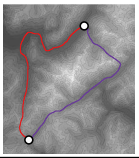
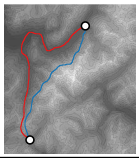
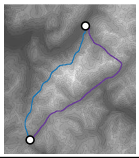
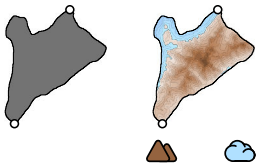
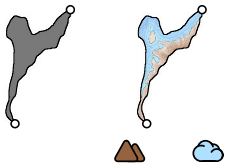
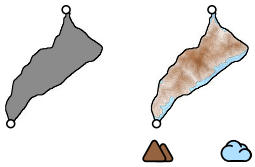
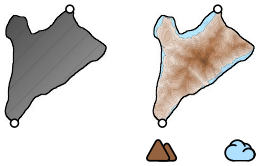
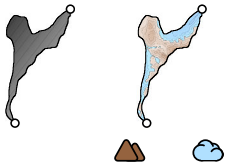
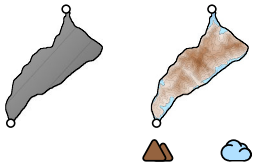
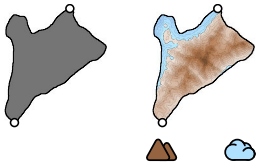
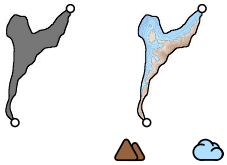
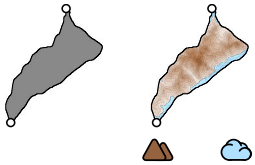
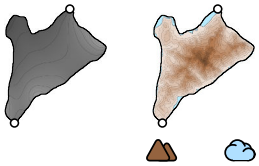
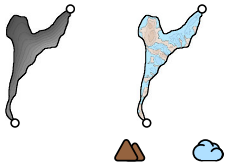
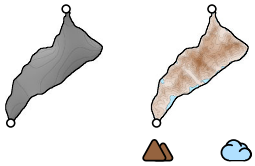
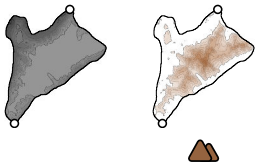
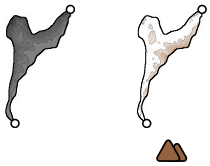
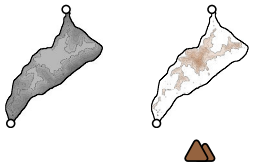
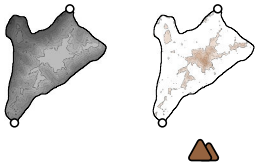
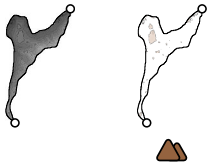
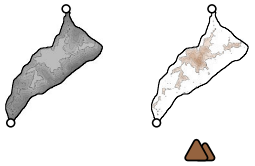
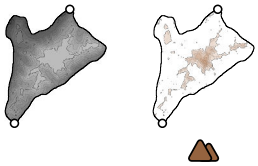
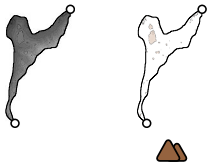
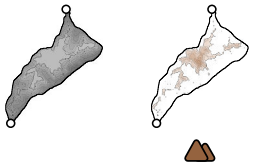


Figure 5 Datasets and paths: **(a)** Mont Blanc. **(b)** Grampians.

Table 1 Mt. Blanc, distances between (π_1, π_3), (π_1, π_2), and (π_2, π_3), earth (▲) and air (☁).

Area (10^8 m^2)							
Projected	6.627		3.508		3.119		
Actual	6.742		3.568		3.174		
HAP			314.10 24.77		25.77 33.72		160.60 9.07
RP			297.87 8.32		29.60 16.39		158.58 6.23
MHP			305.13 26.85		27.73 31.21		169.97 7.47
MAS			310.59 0.90		9.07 10.94		143.35 0.64
WFS →			117.92		15.34		22.36
WFS ←			17.32		0.64		16.69
SFS			17.29		0.60		16.69

Since both paths π_1 and π_3 are in valleys, we expect the base surfaces to separate a lot of earth and not much air. All of the HAP, RP, MHP, and MAS distances appear to have this property, although the horizontal planes separate more air than intuitively desirable. The WFS separates much less of the earth (especially in one direction), and the same holds naturally for the SFS. However, these distances still appear to capture the most important parts of the mountain (at least the WFS ($\rightarrow$) does). These distances are expected to measure less, as they are more conservative and more closely follow the input terrain.

If we consider the distances between π_1 and π_2, then we expect to measure less earth, and relatively more air. This indeed appears to be the case for all distances that also measure air. Also for WFS and SFS the distances clearly appear to measure less earth. There is an interesting difference to notice here between the RP and MAS distances: because the RP is restricted to be a plane, it is well below the higher parts of π_2, thus the RP distance measures much more earth than the MAS distance.

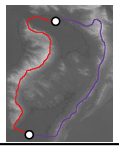
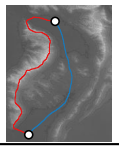
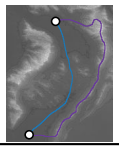
Finally, since the peak is contained between π_2 and π_3, we expect our distances to measure more earth (relative to projected area) than between π_1 and π_2. Again, the HAP, RP, MHP, and MAS distances clearly capture this. The same holds for the WFS ($\leftarrow$) and SFS distances. As WFS ($\rightarrow$) essentially just separates earth that needs to be removed to travel monotonically from π_1 to π_2 (capturing the part of the mountain left from π_2) and from π_2 to π_3 (capturing only the peak to the right of π_2), it is not surprising that the corresponding distances are similar.

It is further interesting that the WFS ($\rightarrow$) distance between π_1 and π_2 (capturing the part of the mountain left from π_2) and the WFS ($\leftarrow$) distance between π_2 and π_3 (capturing the part of the mountain right from π_2) are roughly the same. We would expect the latter to be higher, since that part contains the peak of the mountain. This discrepancy can be explained by the fact that the path π_3 is higher than the path π_1. Finally, note that for most distances, the amount of earth measured between π_1 and π_3 is higher than the the sum of the amounts of earth measured between π_1 and π_2 and between π_2 and π_3. This is desired behavior, as the paths π_1 and π_3 are both in different valleys and thus very different. It also directly implies that these distances do not satisfy the triangle inequality.

Grampians dataset. In Table 2 we show the results for the Grampians dataset, in the same structure as for the Mont Blanc dataset. Since the paths π_1 and π_3 enclose a large valley, we expect to measure a lot of air in this dataset. This indeed appears to be the case for the distances based on HAP, RP, MHP, MAS. Especially the MAS is good at not separating any unnecessary earth near the paths π_1 and π_3. However, none of these distances capture the small hills inside the valley. These hills are measured by the WFS distances, but they also measure some extra volume near the paths π_1 and π_3. The SFS distance picks up the volume of the hills inside the valley only.

Since the path π_2 goes through the valley, we expect the distances between π_1 and π_2 to capture much less air than between π_1 and π_3. This is indeed the case for the HAP, RP, MHP, and MAS distances. However, the MHP distance appears to capture very little air and more earth than expected. This is due to the fact that the MHP uses the median height of the paths. Since more than half of $\pi_1 \cup \pi_2$ lies in the valley, the base surface MHP also lies in the valley, thus separating a large amount of earth. Further, as before, the HAP, RP, MHP, and MAS distances do not measure the hills inside the valley. This volume is measured by the WFS ($\rightarrow$) distance. The WFS ($\leftarrow$) distance also measures it, but it measures a significant amount of extra volume near π_1. Finally, the SFS distance again nicely captures the volume of the small hills in the valley.

Table 2 Grampians, distances between (π_1, π_3), (π_1, π_2), and (π_2, π_3), earth (⛰) and air (☁).

Area (10^8 m^2)							
Projected	6.627			3.508		3.119	
Actual	6.742			3.568		3.174	
HAP							
		6.78	167.63		7.15	47.24		9.75	40.16
RP							
		6.05	168.24		6.65	38.56		4.56	45.96
MHP							
		4.97	191.45		26.33	4.43		21.57	7.61
MAS							
		0.66	173.19		0.65	48.35		1.10	33.49
WFS →							
		23.55			1.89			21.40	
WFS ←							
		23.84			21.59			1.39	
SFS							
		3.26			1.87			1.37	

For the distances between π_2 and π_3 we expect similar results as between π_1 and π_2, and indeed the HAP, RP, MHP, and MAS distances appear to give similar values. We also clearly see the asymmetry of the WFS distance, where the WFS ($\rightarrow$) and WFS ($\leftarrow$) distances have switched roles compared to the values between π_1 and π_2. The SFS distance again nicely captures the volume of the small hills in the valley.

Like in the Mont Blanc dataset, we see that the amount of air measured between π_1 and π_3 is more than the sum of the amounts of air measured between π_1 and π_2 and between π_2 and π_3. Again, this is desired behavior. This dataset also illustrates the usefulness of the WFS and SFS distances. In particular, from the SFS distance we can see that the volume of the hills in the valley to the left of π_2 is more than to the right of π_2, which is almost impossible to see from the HAP, RP, MHP, and MAS distances.

Synthetic datasets. In Table 3 we show the results for the synthetic datasets. The synthetic datasets are (1) a hill, (2) a slope, (3) a set of small hills on a slope, and (4) a valley where the two paths start in the valley, go up different sides of the valley, and then end up down in the valley again.

For the first dataset with the hill, all distances clearly measure the volume of the hill. The distances measure small amounts of air, but these amounts are clearly insignificant.

For the slope dataset there are clear differences. Depending on what behavior is desired, different distances should be chosen. If the paths should be considered to be different, then one should simply measure the area/projected area between the paths, or use the HAP or MHP distances. Because the HAP and MHP distances use horizontal planes as base surface, they cannot capture the slope and must measure significant volumes of both earth and air. The WFS distance is useful for a situation where going up implies more distance than going down. However, given our motivation of contextual volume-based distances, these paths should have distance zero, and this is indeed measured by the RP, MAS,[1] and SFS distances.

The third synthetic dataset also contains a slope, so the HAP and MHP distances give similar results as for the second synthetic dataset. Also, the WFS distance again demonstrates its inherent asymmetry. The remaining distances (based on the RP, MAS, and SFS) all capture the small hills on the slope. SFS is generally more conservative in measuring the earth volume.

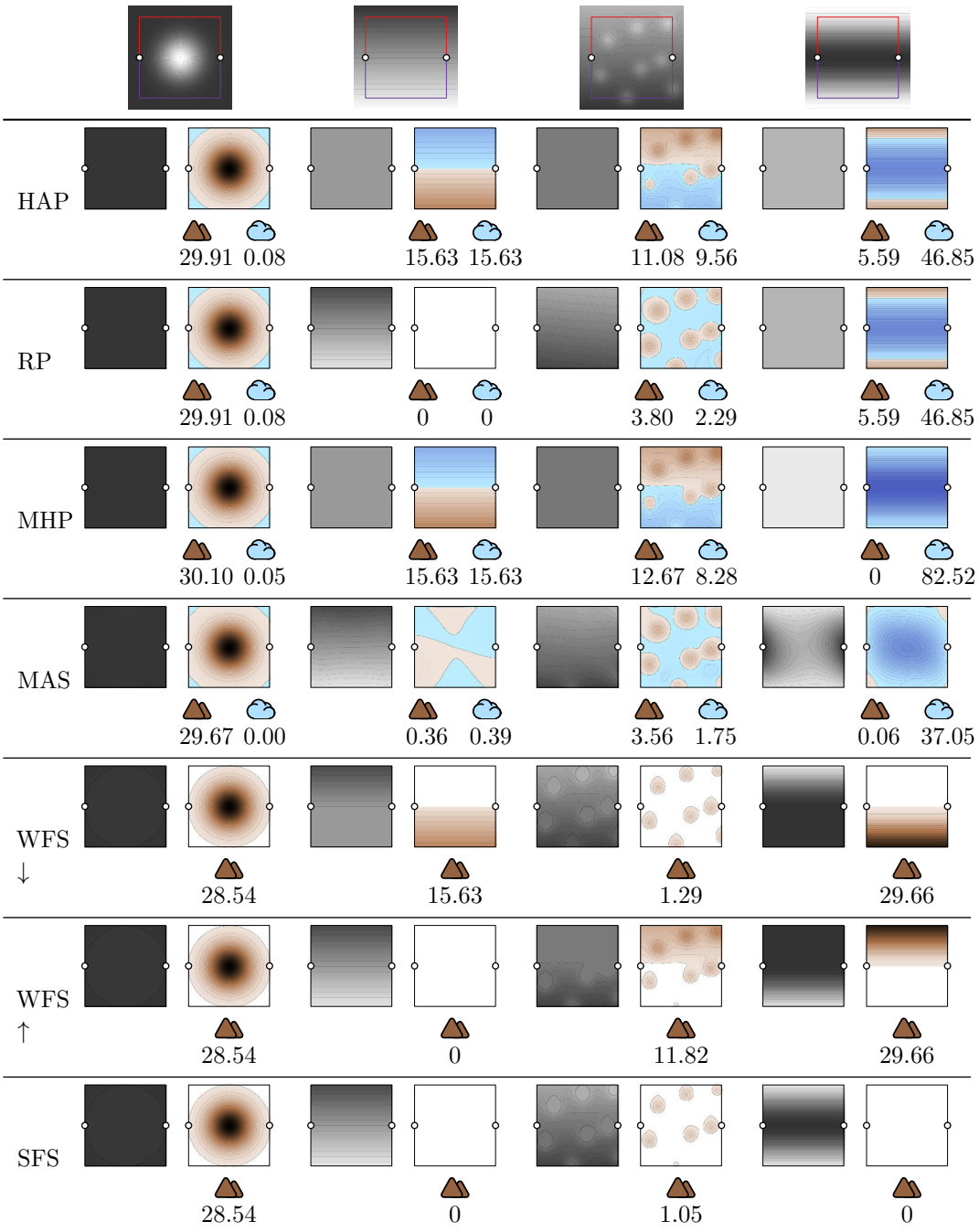
The fourth synthetic dataset shows some interesting differences between the distances. The HAP, RP, MHP, and MAS distances all capture the air in the valley between the high parts of the paths. The HAP and RP distances also capture some earth volume near the high parts of the paths. The WFS distances capture how much "effort" it costs to go from one path to the other, where only moving up requires effort. The effort here is measured as the amount of earth that needs to be removed to eliminate this effort. Finally, the SFS distance measures the amount of effort needed for the two paths to come together.

4.2 Volumes from outlines for landslide risk analysis

We next analyze how our base surfaces and the corresponding volumes may contribute to landslide risk analysis. To that end, we consider a part of real-world terrain (see Fig. 6) that has been identified as high risk for landslides according to the deep-seated landslide susceptibility map for California, USA, created by the California Department of Conservation [24].

[1] The MAS distances are not exactly zero. This is a consequence of approximating a minimum area surface; the real minimum area surface should follow the slope and result in a distance of zero.

Table 3 Synthetic data, distances between π_1 and π_2, earth (⛰) and air (☁).



Specifically, we consider a part of terrain along the California coastline close to Highway 1 (near the mouth of Russian river), which has been assigned the highest risk class X among eight different classes. The classification is based on the slope of the terrain and the soil type, but does not take volume or other aspects of the shape of the terrain into account.

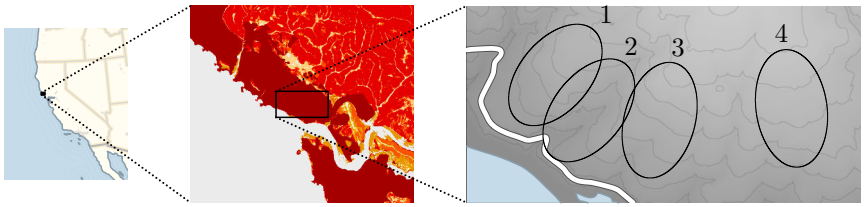


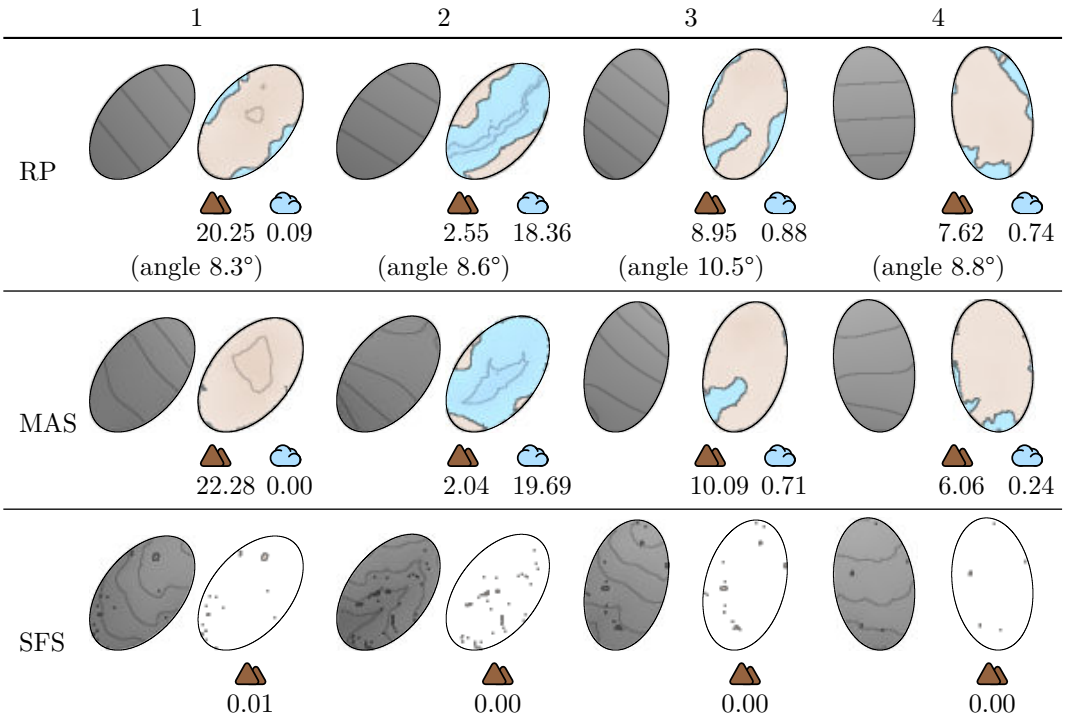
Figure 6 Regions used in our landslide analysis: **(a)** overview map; **(b)** part of the deep-seated landslide susceptibility map for California (image from CGS Map Sheet 58 [24]); **(c)** the four regions.

For a more fine-grained analysis of landslide risk, it is relevant to measure the amount of soil (that is, volume) that may slide and where this soil may be deposited. We believe that (some of) our volumes from base surfaces can contribute to such analysis. To show that, we consider four equal-size ellipse-shaped regions of the terrain that all have the same risk class X in the deep-seated landslide susceptibility map, but contain parts of the terrain with different morphology (see Fig. 6). Specifically, Region 1 contains a convex ridge with relatively high plan curvature, Region 2 contains a valley with negative plan curvature, and Regions 3 and 4 are flatter, but Region 3 borders the valley of Region 2, whereas Region 4 does not. We computed three volumes on the region outlines, using the RP, MAS, and SFS. Note that the volumes from horizontal planes are not meaningful (the terrain is sloped for all regions), and that the WFS requires two paths instead of a single outline, so we did not include them in this analysis. The results are shown in Table 4.

We can immediately see that the SFS volume is small. This is to be expected: the SFS is designed for similarity in the context of water flow, and therefore only separates volume around local maxima (as they impede water flow). Since all regions contain sloped terrain without significant local maxima, the SFS always separates near-zero volumes. The volumes based on the RP and MAS are similar. If we interpret the outline of a region as the boundary of a potential landslide, then indeed we would expect to measure a high volume of earth for the ridge (Region 1). On the other hand, we measure very little earth for the valley (Region 2), as expected. However, the volume of air is very high. This indicates that a large volume of soil can be deposited in this region, but the region does not contribute much soil to any landslide. For Regions 3 and 4, which are flatter than Region 1, the RP and MAS indeed separate less earth volume than for Region 1. The earth volume for Region 3 is somewhat higher than for Region 4, likely because Region 3 borders the valley of Region 2, and hence one part of the outline of Region 3 drops down steeper there. Also note that, purely based on morphology, the outline of Region 4 is not likely the border of a potential landslide, as the profile of the terrain is not changing much across the outline. However, geological, hydrological, and land use factors may indicate otherwise.

Considering the (minor) difference between the RP and MAS volumes, we see that the volumes for the MAS are somewhat "cleaner". For example, the MAS separates no air volume for the ridge (Region 1), as one would expect, whereas the RP separates a small volume of air. The reason for this is that the MAS can completely follow the terrain at the outline, and the RP, which is a flat plane, cannot. This results in small amounts of air and earth volume close to the outline unnecessarily being added to the total earth and air volumes separated by the RP. These amounts become larger as the outline becomes more irregular (less flat). However, we can see that these added amounts are relatively small and do not change the volumes significantly. Given the fact that the RP can be computed much faster than the MAS, the RP volume might be preferable if there is no need for high precision.

Table 4 Landslides, volume measurements in 10^5 m^3 for the four areas, earth (▲) and air (☁).



5 Conclusion

We studied the problem of computing a volume of terrain based on a given outline. After reviewing six possible base surfaces, we showed how to compute them and how to determine the volume between a base surface and the terrain. We highlighted two use cases, namely volume-based similarity and landslide risk assessment, and performed experiments related to both of them. The experiments revealed properties of the different volume computation models so that practitioners can easily select the most suitable one for their application. Summarizing, we have demonstrated that computing terrain volume based on an outline is a useful type of terrain analysis. We presented various options to do so, including the required algorithms, and reported on extensive comparative experiments.

References

1 Mark de Berg and Marc van Kreveld. Trekking in the Alps without freezing or getting tired. *Algorithmica*, 18(3):306–323, 1997.

2 Enrico Feoli and Vincenzo Zuccarello. Spatial pattern of ecological processes: the role of similarity in GIS applications for landscape analysis. In Manfred Fischer, Henk J. Scholten, and David Unwin, editors, *Spatial Analytical Perspectives on GIS*, pages 175–185. Taylor & Francis, 1996.

3 Fausto Guzzetti, Alberto Carrara, Mauro Cardinali, and Paola Reichenbach. Landslide hazard evaluation: a review of current techniques and their application in a multi-scale study, Central Italy. *Geomorphology*, 31(1-4):181–216, 1999.

4 L. A. M. Hendriks, H. Leummens, A. Stein, and P. de Bruijn. Use of soft data in a GIS to improve estimation of the volume of contaminated soil. *Water, Air, and Soil Pollution*, 101(1-4):217–234, 1998.

5 Matthew Hiatt, Willem Sonke, Elisabeth A. Addink, Wout M. van Dijk, Marc van Kreveld, Tim Ophelders, Kevin Verbeek, Joyce Vlaming, Bettina Speckmann, and Maarten G. Kleinhans. Geometry and topology of estuary and braided river channel networks automatically extracted from topographic data. *Journal of Geophysical Research: Earth Surface*, 125(1), 2020.

6 Jeffrey Hollister and W. Bryan Milstead. Using GIS to estimate lake volume from limited data. *Lake and Reservoir Management*, 26(3):194–199, 2010.

7 Alec Holt. Spatial similarity and GIS: the grouping of spatial kinds. In *Proceedings of the 11th Annual Colloquium of the Spatial Information Research Center (SIRC05)*, pages 241–250, 1999.

8 Krzysztof Janowicz, Martin Raubal, Angela Schwering, and Werner Kuhn. Semantic similarity measurement and geospatial applications. *Transactions in GIS*, 12(6):651–659, 2008.

9 A. Jarvis, H.I. Reuter, A. Nelson, and E. Guevara. Hole-filled seamless SRTM data V4, 2008. URL: `http://srtm.csi.cgiar.org`.

10 A. Keutterling and A. Thomas. Monitoring glacier elevation and volume changes with digital photogrammetry and GIS at Gepatschferner glacier, Austria. *International Journal of Remote Sensing*, 27(19):4371–4380, 2006.

11 Maarten Kleinhans, Marc van Kreveld, Tim Ophelders, Willem Sonke, Bettina Speckmann, and Kevin Verbeek. Computing Representative Networks for Braided Rivers. In *Proceedings of the 33rd International Symposium on Computational Geometry (SoCG 2017)*, volume 77 of *LIPIcs*, pages 48:1–48:16, 2017.

12 John McIntosh and May Yuan. Assessing similarity of geographic processes and events. *Transactions in GIS*, 9(2):223–245, 2005.

13 Robert B. McMaster and K. Stuart Shea. *Generalization in Digital Cartography*. Association of American Geographers, 1992.

14 Colin W. Mitchell. *Terrain Evaluation*. Routledge, 2014.

15 Giorgos Mountrakis, Peggy Agouris, and Anthony Stefanidis. Similarity learning in GIS: an overview of definitions, prerequisites and challenges. In *Spatial Databases: Technologies, Techniques and Trends*, pages 294–321. IGI Global, 2005.

16 Pascal Peduzzi, Christian Herold, and Walter Claudio Silverio Torres. Assessing high altitude glacier thickness, volume and area changes using field, GIS and remote sensing techniques: the case of Nevado Coropuna (Peru). *Cryosphere*, 4(3):313–323, 2010.

17 Paola Reichenbach, Mauro Rossi, Bruce D Malamud, Monika Mihir, and Fausto Guzzetti. A review of statistically-based landslide susceptibility models. *Earth-Science Reviews*, 180:60–91, 2018.

18 Angela Schwering. Approaches to semantic similarity measurement for geo-spatial data: A survey. *Transactions in GIS*, 12(1):5–29, 2008.

19 K. Stuart Shea and Robert B. McMaster. Cartographic generalization in a digital environment: When and how to generalize. In *Proceedings of Auto-Carto 9*, pages 56–67, 1989.

20 Willem Sonke, Marc van Kreveld, Tim Ophelders, Bettina Speckmann, and Kevin Verbeek. Volume-based similarity of linear features on terrains. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 444–447, 2018.

21 Kevin Toohey and Matt Duckham. Trajectory similarity measures. *Sigspatial Special*, 7(1):43–50, 2015.

22 David J. Varnes. *Landslide Hazard Zonation: a review of principles and practice*. Number 3 in Natural Hazards. United Nations, 1984.

23 David Julius Völker. A simple and efficient GIS tool for volume calculations of submarine landslides. *Geo-Marine Letters*, 30(5):541–547, 2010.

24 C. J. Wills, F. G. Perez, and C. I. Gutierrez. Susceptibility to deep-seated landslides in California, 2011. California Geological Survey, Map Sheet 58.

Traffic Prediction Framework for OpenStreetMap Using Deep Learning Based Complex Event Processing and Open Traffic Cameras

Piyush Yadav[1]
Lero-SFI Irish Software Research Centre, Data Science Institute, National University of Ireland Galway, Ireland
piyush.yadav@lero.ie

Dipto Sarkar
Department of Geography, University College Cork, Ireland
dipto.sarkar@ucc.ie

Dhaval Salwala
Insight Centre for Data Analytics, National University of Ireland Galway, Ireland

Edward Curry
Insight Centre for Data Analytics, National University of Ireland Galway, Ireland
edward.curry@insight-centre.org

Abstract

Displaying near-real-time traffic information is a useful feature of digital navigation maps. However, most commercial providers rely on privacy-compromising measures such as deriving location information from cellphones to estimate traffic. The lack of an open-source traffic estimation method using open data platforms is a bottleneck for building sophisticated navigation services on top of OpenStreetMap (OSM). We propose a deep learning-based Complex Event Processing (CEP) method that relies on publicly available video camera streams for traffic estimation. The proposed framework performs near-real-time object detection and objects property extraction across camera clusters in parallel to derive multiple measures related to traffic with the results visualized on OpenStreetMap. The estimation of object properties (e.g. vehicle speed, count, direction) provides multidimensional data that can be leveraged to create metrics and visualization for congestion beyond commonly used density-based measures. Our approach couples both flow and count measures during interpolation by considering each vehicle as a sample point and their speed as weight. We demonstrate multidimensional traffic metrics (e.g. flow rate, congestion estimation) over OSM by processing 22 traffic cameras from London streets. The system achieves a near-real-time performance of 1.42 seconds median latency and an average F-score of 0.80.

2012 ACM Subject Classification Information systems → Data streaming; Information systems → Geographic information systems

Keywords and phrases Traffic Estimation, OpenStreetMap, Complex Event Processing, Traffic Cameras, Video Processing, Deep Learning

Digital Object Identifier 10.4230/LIPIcs.GIScience.2021.I.17

Supplementary Material The system was implemented in Python 3 and is available at `https://github.com/piyushy1/OSMTrafficEstimation`.

Funding This work was supported by the Science Foundation Ireland grants SFI/13/RC/2094 and SFI/12/RC/2289_P2.

[1] corresponding author

1 Introduction

OpenStreetMap (OSM) is arguably the largest crowdsourced geographic database. Currently, there are more than 5 million registered users, over 1 million of whom have contributed data by editing the map. Different aspects of OSM data quality have been scrutinized, and OSM data performed well on tests of volume, completeness, and accuracy across several classes of spatial data, such as roads and buildings [18, 11]. Despite standing up to data quality tests, enforcing data integrity rules required for a navigable road map is challenging. Lack of topological integrity and semantic rules such as turn restrictions which enable navigational capabilities is a stumbling block for OSM to be a viable open-source alternative to commercial products like Google Maps and Apple Maps.

Over the last five years, several large corporations have realized the value of OSM and have assembled teams to contribute and improve data on the OSM platform [3]. Backed by large companies, the editing teams are capable of editing millions of kilometers (km) of road data each year. As a result of these efforts, the road network data on OSM is improving rapidly, and the gap in the quality of the data in the developed and developing countries is narrowing. In the foreseeable future, it is expected that widely available open-source navigation services may also be built on top of OSM data. The next frontier to making OSM more usable for navigational purposes is to have real-time traffic information to estimate trip times. This feature is already available in commercial digital navigation maps. However, the estimation and availability of traffic state data relies on platforms (i.e. iPhone and Android) built by the respective companies to feed data to the service.

In recent years, internet-connected devices collectively referred to as the Internet of Things (IoT) have become ubiquitous. With the proliferation of visual sensors, there is now a significant shift in the data landscape. We are currently transitioning to an era of the Internet of Multimedia Things (IoMT) [2, 5] where media capturing sensors produce streaming data from different sources like CCTV cameras, smartphones and social media platforms. For example, cities like London, Beijing and New York have deployed thousands of CCTV cameras streaming hours of videos daily [1]. Complex Event Processing (CEP) is an event-driven paradigm which utilizes low-level data from sensor streams to gain high-level insights and event patterns. CEP applications can be found in areas as varied as environmental monitoring to stock market analysis [10].

In this research, we propose a Complex Event Processing (CEP) framework to estimate traffic using deep learning techniques on publicly available street camera video streams. We process the data on GPU computing infrastructure and expose the processed output to OSM. The proposed method utilizes publicly available data and does not piggy-back on using cell phones. The solution is better for privacy and is also immune to subversion techniques such as the recent case where an artist used 99 Android phones to simulate traffic on Google Maps [12]. Further, the rich data stream from the video can provide multidimensional measurements of traffic state going beyond simple vehicle count-based metrics.

2 Background and Related Work

This section provides the initial background and techniques required for the development of a traffic estimation service for OSM.

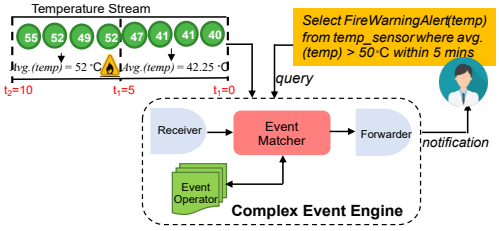


Figure 1 Complex Event Processing Paradigm. Temperature is being monitored from a sensor to issue fire warnings.

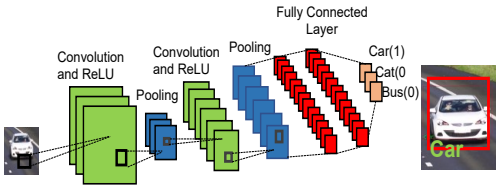


Figure 2 CNN Architecture. A CNN consisting of several connected layers of neuron for object identification from an image.

2.1 Complex Event Processing

Complex Event Processing (CEP) is a specialized information flow processing (IFP) technique which detects patterns over incoming data streams [9]. CEP systems receive data from different sensor streams and then mine high-level patterns in real-time to notify users. CEP systems perform online and offline computations and can handle high volume and variety of data. In CEP, event patterns of interest are expressed using SQL like query language, and once a query is registered, it continuously monitors the data stream. Pattern matching for queried patterns occurs in (near) real-time as the data flows from the Data Producers (sensors) to the Consumers (applications).

Fig. 1 shows a simple CEP system, where a user queries from a temperature sensor to send notification of a *fire warning* alert if the average temperature is greater than 50°C in the last five minutes [28]. The fire warning alert query is registered in the CEP engine which continuously monitors the data from the temperature sensor. The CEP engine will raise a fire warning alert at time $t_1 - t_2$ as the average temperature of incoming streams is higher than 50°C in the last five minutes. Thus, a complex fire warning alert is generated by averaging simple 'temperature event' from the sensor.

2.2 Deep Learning-based Image Understanding

The computer vision domain focuses on reasoning and identifying image content in terms of high-level semantic concepts. These high-level concepts are termed as *objects* (e.g. car, person) which act as building blocks in understanding and querying an image. There are different object detection algorithms like SIFT [16], which classify and localize the objects in the video frames. Convolutional Neural Networks (CNN) based deep learning methods [15] are proficient at identifying objects with high accuracy. It is a supervised learning technique where layers are trained using labelled datasets to classify images. Fig. 2 shows the underlying architecture of a CNN model having different layers to classify and detect an object from the image. CNN based object detection methods like YOLO [20] detect and classify objects by drawing bounding boxes around them.

2.3 Related Work

Traffic Estimation Services. Most of the traffic monitoring related data is collected from sensor devices like GPS embedded in mobile phones and speed detection cameras (loop detectors, camera, infrared detectors, ultrasonic, radar detectors) [4]. The traffic data is displayed on proprietary maps such as Google, Apple and Here Maps. These companies also expose the data through APIs. However, the methodology for traffic estimation is opaque,

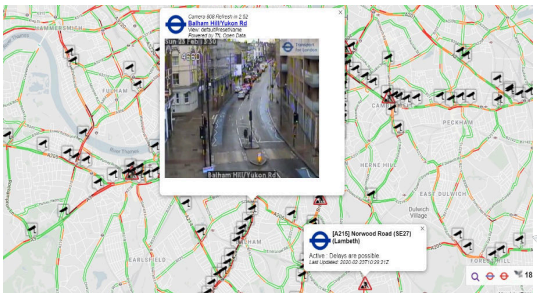


Figure 3 TfL Traffic Camera Dashboard.

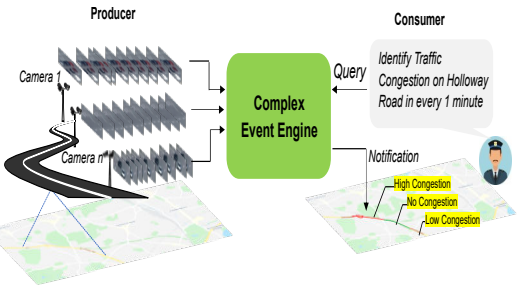


Figure 4 Traffic CEP Overview.

and depending on the service provider, there are costs and restrictions on how a third-party developer can utilize the data. Initiatives such as OpenTraffic[2] are building an open-source data platform and services which collect anonymized telemetry data from vehicles and smartphones and exposes them through OSM API. OpenTraffic is a relatively new platform and is still building up a list of partners (currently three) to gather traffic information. Other attempts to create open traffic data framework for consumption as a map service include the OpenTransportMap[3] project. Their framework is based on pre-calculated estimates of traffic volumes based on demographic data. There are some traffic state prediction works which use interpolation techniques using sensor-based reading like GPS and LIDAR to identify traffic at unknown locations [17, 31, 23]. In this work, instead of telemetry data like GPS, we focused on clusters of openly available video feeds which provide live-streaming updates from multiple locations and update traffic in real-time on OSM.

Video-Based Traffic Estimation. Video streams are an ideal example of BigData as it represents a high volume, high velocity, unstructured source of data. Fatih et al. [19] proposed a Gaussian based Hidden Markov Model (GM-HMM) to estimate traffic over MPEG videos. But their work was limited only to the camera Field of View (FoV) to determine the traffic over a highway segment. In this work, we focused on estimating traffic beyond camera FoV across the whole queried street network even where camera feeds are not available. Kopsiaftis et al. [14] used background estimation over high-resolution satellite video images and performed traffic density estimation by counting vehicles in the given region. Again, their work is limited to only pre-recorded historical video data. On the other hand, our proposed framework estimates traffic over streaming video in near-real-time. Connected vehicles are another data source for traffic estimation. Kar et al. [13] proposed real-time traffic estimation considering vehicles as edge nodes. They performed object and lane detection using dash cameras installed in the vehicles to estimate their speed. The author's future work was focused on sharing such data across vehicles and on a cloud-based map service. Our work builds upon this line of argument by considering traffic camera network as a cluster of edge nodes and then applying data-driven techniques to identify traffic across the road network which is later updated to OSM.

[2] `opentraffic.io`
[3] `opentransportmap.info`

3 Motivation and Problem Scope

3.1 Case Study: London- A City of Open Traffic Cameras

London possesses an extensive camera network. The city is dotted with nearly 500K cameras [26]. With a density of 68.4 cameras per 1000 people, it is estimated that an average Londoner is caught in a camera approx 300 times a day [26]. Live camera feeds stream real-time information of things happening around London's streets, enabling applications like license plate reading. The camera streams API is provided from Transport for London (TfL) and can be accessed by registering with their system. Each video camera comes with metadata including date, timestamp, street name and its geolocation. Availability of open cameras, accessibility of real time[4] and archived data[5], and friendly streaming API makes London ideal for our case study. Fig. 3 shows the screenshot of camera networks across a part of the city with a video clip instance from a particular camera.

3.2 User Scenario

Traditional traffic monitoring using CCTV is mostly a manual effort where traffic personnel monitor the situation by looking at the feeds from cameras installed across the city. This manual approach is time-consuming, tedious, and error-prone as it is difficult for humans to synthesize a large number of events occurring across space and time. Thus, the development of automated techniques is crucial to supplement manual qualitative efforts. Suppose the traffic authority wants to map the busiest routes of the city over the day in real-time and wants to provide the traffic state as a service to the citizens. Fig. 4 shows the authority using a CEP engine to obtain the traffic congestion status over a road segment from a cluster of CCTV cameras installed along the road. Processing this query over multiple video streams requires identifying objects (e.g. cars), determining their properties (e.g. speed), calculating traffic states, interpolation across the road network for continuous estimation, and visualizing the results in real-time with summarized metrics and visualizations overlaid on OSM.

3.3 Multidimensional Traffic Analysis

There are multiple factors related to traffic and untangling the impact of individual factors responsible for congestion is challenging. Congestion is a function of both: the physical way vehicles (and other road users) interact with each other, and the people's perception of congestion (e.g. 'the traffic is terrible today') [24]. We focus on the primary factors related to the physical dimensions of vehicular movement through the road network, namely vehicle count and average vehicle speed. Thus, at each time point, we estimate the number of vehicles at a location and the speed of each vehicle.

While these represent rudimentary aspects related to quantifying traffic state, they can be used as building blocks to more complex metrics such as traffic density, expected travel time, free flow ratio, and estimates of delay [24, 14]. No single indicator can be a 'catch-all' metric to represent the problem. Reliance on a single parameter in describing network performance paints an incomplete, or in some cases an incorrect picture of travel conditions. In this paper, we focus on building the stream data processing architecture and show its efficacy on a real-world application by focusing on calculating the basic metrics of vehicle count and vehicle speed. Table 1 and Table 2 lists the macroscopic traffic metrics and Level of Service(LOS) parameters [8] which are consider for evaluation in this work.

[4] https://www.tfljamcams.net/
[5] http://archive.tfljamcams.net/

Table 1 Traffic Metrics.

Traffic Metrics	Definition
Traffic Flow (TF)	TF is the total count of vehicles that have passed a certain point in both directions for a given time. TF is calculated over short duration's (e.g. two days) and then approximated over days adjusting for seasonal and weekly variations. The TF can be calculated at different timescales ranging from minutes, hours, daily, weekly and yearly level (Annual Average Daily Traffic).
Average Traffic Speed (ATS)	It is the average speed of vehicles on both directions at a point for a given time scale. ATS is measured in m/s, km/hr or mph.
Traffic Density (TD) and Capacity(c)	TD or *volume (V)* is the number of vehicles per unit distance over a given road. It is measure in term of the number of vehicles per unit road (Km or mile). The maximum number of vehicles in a mile per lane which a road can accommodate is termed as *capacity*.

Table 2 Level of Service.

Level of Service	Description	Volume-Capacity Ratio (V/C)
A	Free flowing and highest driving comfort.	<0.60
B	Little delay and high driving comfort.	0.60-0.70
C	Some delay and acceptable level of comfort.	0.70-0.80
D	Moderate delay and some driving frustration.	0.80-0.90
E	High degree of delay and driving frustration.	0.90-1.0
F	Excessive delay and highest level of frustration.	>1.0

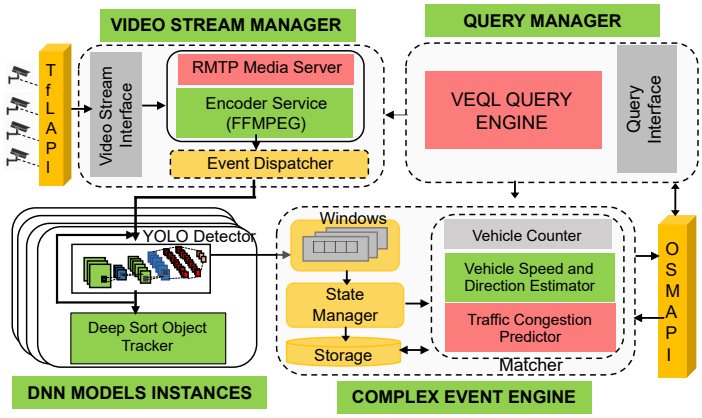


Figure 5 System Architecture Overview.

4 System Architecture

To identify different traffic metrics using multiple cameras, a distributed microservice-based complex event processing system is implemented. Fig. 5 shows the high-level architecture of the traffic estimation system which is divided into four major components. These components are independent microservices wrapped in a container and their instances can be deployed over the cloud or local computing nodes. These components are:

- **Query Manager**: The user can subscribe to different queries using the query manager. The Query manager consists of a Query Interface where users can write queries in Video Event Query Language (VEQL) to detect traffic patterns [29]. VEQL is a SQL-like declarative language where rules and operators can be written to identify pattern over video streams. The VEQL Query Engine creates a query graph to represent patterns. Further details of VEQL can be found in [29]. A sample VEQL query for traffic congestion is as follows:

```
Select Traffic_Congestion(Object) from Brixton Road
WHERE Object = 'Car' OR Object = 'Bus'
WITHIN Time_Window = 5 sec WITH CONFIDENCE >40%
```

  In the above query, the user subscribed for Traffic Congestion for 'Car and Bus' over Brixton Road camera network with an update of every five-seconds. The traffic congestion operator will be discussed in detail in Section 6.
- **Video Stream Manager**: The video stream manager connects to multiple video feeds using a video stream interface which is a network adapter to provide the connection to different streaming API's. The TfL (Transport for London) unified API is used to bring together video data from street cameras to our system. The Query Manager forwards the road information(e.g. Brixton Road) to the stream manager which later uses metadata information from the TfL API to fetch video streams from the corresponding road camera. For example, we pass different lat-long coordinates of Brixton Road (e.g. 51.4812,-0.11065) using OSM which is then passed to the TfL API (`https://api.tfl.gov.uk/Place?lat=51.4812&lon=-0.11065&radius=100&type=JamCam`) to search cameras. The search process continues and iterates until the end of the queried road segment. Similarly, each stream is sent in parallel to the media server using the *GStreamer* library. Finally, the Event Dispatcher sends the received video streams from different cameras to the DNN Model pipeline.
- **Deep Neural Network (DNN) Models Pipelines**: This is a computer vision pipeline which consists of deep learning-based object detector and object tracker. The object detector model receives the video frames from the *event dispatcher* as a feature map and extracts the vehicles in the form of bounding boxes. The object detector is integrated with a DNN based object tracker to track the identified vehicle for a given length of time. The specifics of the object detector and tracker are explained in detail in Section 5.1. Based on the number of cameras on the queried road, separate DNN model instances are created dynamically to process each camera feed in parallel. This is necessary to separate the tracking instances of each identified vehicles in each camera.
- **Complex Event Engine**: The Complex Event Engine is the core component which predicts traffic-related activities. Its sub-components are:
  - **Window and State Manager**: Data streams (i.e. video feeds) are considered as an unbounded timestamped sequence of data [35]. CEP systems work on the concept of

state. Windows capture the stream state by taking the input stream and producing a sub-stream of finite length (equation (eq.) 1 and 2).

$$S_{video} = ((f_1, t_1), (f_2, t_2), ..., (f_n, t_n)) \quad (1)$$

$$TIME\_WINDOW(S_{video}, t_{5sec}) : \rightarrow S' \quad (2)$$

where f_i are video frames and $S^{'} = ((f_1, t_1), (f_2, t_2), ..., (f_n, t_{5sec}))$. In eq. 2, a $TIME\_WINDOW$ of five seconds is applied over an incoming video stream S_{video} (eq. 1) and gives a fixed sub-sequence $S^{'}$ of five seconds of video data. The *State Manager* handles the created window state and sends the state information to the persistent storage and matcher services. The current work focuses on *online* processing of the data in near-real-time. The *Storage* component stores the event state of video feeds for historical batch-based analysis.

- **Matcher**: The Matcher performs traffic operations over the received state from the state manager. The matcher consists of three traffic-related operators 1) Vehicle Counter, 2) Vehicle Direction and Speed Estimator, and 3) Traffic Congestion Estimation. The functionality of these operators is explained in details in Section 5 and 6. The matcher finally sends the results of the CEP engine to the OSM layer through the OSM API.

5 Computer Vision Pipelines for Traffic Classification

Two computer vision pipelines have been developed for the CEP Engine to estimate the traffic service. The first pipeline performs object detection (e.g. cars, buses) and tracking over incoming video streams. The second pipeline involves calculation of traffic-related properties from objects (speed, direction and count) and interpolating traffic information from point sources to create a continuous surface over OSM.

5.1 Pipeline 1: Vehicle Detection and Tracking

Vehicle detection is a common problem in computer vision. Different object detection techniques ranging from feature-based matching like SIFT [16] and complex deep learning models have been used to identify vehicles in the videos. In this work, the YOLO v3 [20], a state-of-the-art object detection model is used for vehicle detection. The YOLO model considers object detection as a single regression problem and divides the image into a SxS grid to predict the objects bounding boxes and class probabilities simultaneously. The model gives real-time performance and processes 45 frames per second(fps) at the rate of 22 milliseconds per frame on modern GPUs and is suitable for processing streaming data like videos. Five classes of vehicles- {*bus, car, truck, bicycle and motorcycle*} are selected as they represent the significant vehicular traffic on the road. We have used the YOLO model pre-trained on the COCO dataset which already consists of all the above five vehicular classes. The model outputs the bounding box coordinates of each detected vehicle with a probability score. The probability score is the model confidence to predict the class of an object (like vehicle), and experimentally we found that score greater than 0.4 detect most of the vehicles accurately.

Videos are timestamped continuous sequence of image frames. The vehicle object detector process frames one-by-one to detect and classify vehicles per frame. Videos are temporally correlated where the same object (vehicles) remain in multiple frames. To avoid repeatedly counting the same vehicle, the vehicle needs to be tracked across the video feed. Deep-

Figure 6 (L-R) a)TfL Camera Image at Brixton Road/Island Place, b)Edge Detection to Identify Lanes, c) Distance Identification Using Google Earth Referencing , d) Object Identification, Tracking, Direction and Speed Estimation.

SORT [27] is a multiobject tracking algorithm which uses Kalman filters and deep neural network to track the detected objects across the image frames. We integrated the DeepSORT tracking model with the YOLO object detection to uniquely identify each vehicle across frames.

5.2 Pipeline 2: Vehicle Direction, Count and Speed Estimation

It is essential to know the movement direction of the vehicle to segregate traffic estimation for different lanes. Direction estimation is challenging in TfL installed cameras as there is no metadata information about the direction of placement of the cameras on the road. After close inspection, we concluded that the cameras are placed over roads in such a way that their FoV covers the length of the road (top-front view). The cameras are placed in South-North or West-East direction such that outgoing traffic is in the left lane while incoming traffic is in the right lane with respect to cameras FoV. The direction of each vehicle can be calculated by measuring the displacement of the centre pixel location of its bounding box across frames for a given time window (eq. 2). Considering the bottom left of the image frame as reference origin, if the displacement of the y-axis value of centre point of bounding box decreases for a given time, then it is considered an incoming traffic and vice-versa. DeepSORT provides each vehicle with a unique id so that the counting can be done for vehicles in both directions.

The speed of the vehicles can be identified using the standard equation of $Distance = Speed * Time$. Thus, the displacement of the centre pixel point of vehicle across frames where it is present for a given time (e.g. 5 sec) can be used to estimate the vehicle speed. But the speed of the vehicle will be calculated in pixels/sec which is not helpful in traffic estimation. As discussed earlier, there are no camera calibration points or pixel geotagging is available in the metadata. In computer vision, object distance from the camera is measured by taking pictures of the object from different angles and then fit into camera calibration equations. It is infeasible to go to every camera points to get object pictures from different angles. We performed a different approach to identify the relative pixels value in terms of metres. As shown in Fig. 6(b), a canny edge detection [7] algorithm is used to identify the lanes of the road using a background frame where no vehicles are present. The number of pixels between two lanes is then calculated by dividing the image into two parts and taking average pixels distance value between lanes to accommodate camera FoV at a different scale. Using the ruler tool available in Google Earth, the distance between lanes is calculated for the same location (figure 6(a,b,c)). For example, the number of pixels between lane is x and google earth distance is y metres, then the size of each pixel is x/y metre. We used the Design Manual for Roads and Bridges (DMRB) CD127 [25] document which provides highway cross-sections and traffic lane width for trunk roads to validate that identified distance is within standard permissible

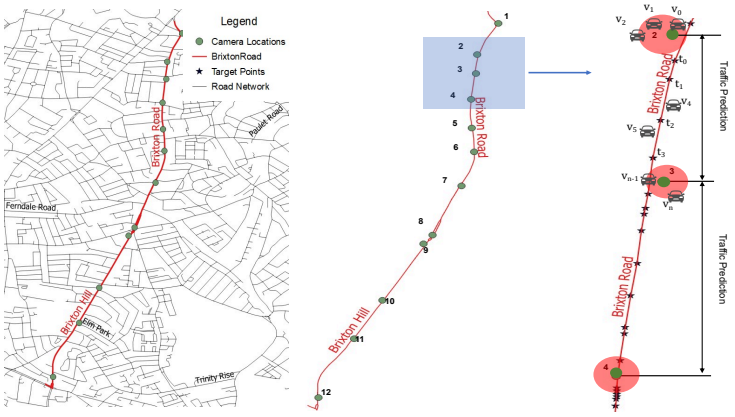


Figure 7 (L-R) a)OpenStreetMap portion of London, b)Brixton Road with Camera Points, c) NB-IDW Technique Shown for Camera 2 and 3.

limits. Finally, for a camera cluster $\{c_0, c_1, c_2, c_3, \cdots\}$ over a road, for each c_i we process the video streams to get information as: $c_i = \{vehicleID : [class, direction, speed], \cdots\}$ (figure 6(d)). This processed data is then used to identify traffic estimates across the street network.

6 Traffic Prediction Over Street Network

Traffic cameras are installed over the street at specified distances on important locations like junctions and roundabouts. For example, in Brixton Road London the cameras are placed at an average of 0.4 -0.7 km apart from each other. The traffic cameras have a limited FoV extending over a few metres of the road. So, the proposed system will perform vehicle detection and tracking within this camera FoV. There is no traffic-related information available for a road segment located between the installed cameras. Thus, traffic interpolation is required to identify the unknown traffic values along with street networks.

Algorithm 1 Traffic Congestion Estimation Operator Algorithm.

Data: $Camera_{Route} \leftarrow \{C_0, C_1, C_2, ..., C_n\}$
Result: Traffic Congestion Overlay on OSM
foreach $(C_i, C_{i+1}) \in Camera_{Route}$ **do**
  $R_{ShortestPath} \leftarrow getShortestPath(C_i, C_{i+1})$;
  $TargetPoints \leftarrow identifyVertices(R_{ShortestPath})$;
  $\{O_0, O_1, O_2, ..., O_n\} \leftarrow getObjectAndDirection(C_i, C_{i+1})$;
  $\{S_0, S_1, S_2, ..., S_n\} \leftarrow estimateObjectSpeed(O_i, O_1, ..., O_n)$;
  $SamplePoints \leftarrow \{O_0, O_1, O_2, ..., O_n\}$;
  $dis_{network} \leftarrow getNetworkDistance(TargetPoints)$;
  $weights_{speed} \leftarrow \{S_0, S_1, S_2, ..., S_n\}$;
  $traffic_{congestion} \leftarrow doIDW(dis_{network}, weights_{speed})$;
  $OSM \leftarrow displayCongestion(R_{ShortestPath}, traffic_{congestion})$;
end

6.1 Network-based Traffic Interpolation

In GIS, interpolation is widely used in mapping terrain, temperature, and pollution levels. Spatial interpolation is a prediction technique to estimate unknown values at different points using values from sample locations. Common strategies for spatial interpolation assumes that points are distributed over a 2D- Euclidean space and that the output is to be a surface spanning the entire space. Measurement of distance between sample and target locations are also done 'as the crow flies' along straight lines assuming that the space is isotropic and can be traversed easily in every direction. These assumptions do not hold in the case of road networks which are essentially 1D and can only be traversed along its length. Traditional spatial interpolation approaches, when applied to network-based structures results in significant errors. We performed a network-based Inverse Distance Weighted (NB-IDW) interpolation method for traffic prediction over the road segment between installed cameras [21]. As per Tobler's Law which underpins spatial interpolation techniques, we model congestion by assuming that nearby cameras that record high vehicle counts and low speed represent areas of high traffic and assume an exponential distance decay function for interpolation.

Fig. 7(a) shows the Brixton Road instance in London over the OSM layer. The road stretch is approximately 2.5 miles (Camberwell New Rd/Brixton Rd to Brixton Hill/Morrish Rd) with 12 CCTV cameras (Fig. 7(b)) with an average density of 4.8 cameras per mile. Fig. 7(c) shows the road segment between camera 2 and 4 and how the traffic values were interpolated across this road section. Let $\{v_0, v_1, v_2, \ldots, v_{n-1}, v_n\}$ be the identified vehicles with speed $\{S_0, S_1, S_2, \ldots, S_{n-1}, S_n\}$ in a specific direction for camera 2 and 3. Let $\{t_0, t_1, t_2, \ldots, t_{l-1}, t_l\}$ the target locations between camera 2 and 3 with unknown traffic congestion values $\{\hat{c_0}, \hat{c_1}, \hat{c_2}, \ldots, \hat{c_n - 1}, \hat{c_n}\}$. Therefore, the observed traffic congestion value $(\hat{c_o})$ is the weighted mean of the speed of nearby identified vehicles as:

$$\hat{c_o} = \sum_{i=1}^{n} d_i S_i \tag{3}$$

In eq. 3 d_i is the network distance and S_i is the speed of the vehicles from the target location. The equation can be expanded as:

$$\frac{\sum_{i=1}^{n} f(dis_{network}(v_i, t_0))S_i}{\sum_{i=1}^{n} f(dis_{network}(v_i, t_0))} \tag{4}$$

In eq. 4, $f(dis_{network}(v_i, t_0))$ is the network distance between the vehicles sample points and a given target location (t_0). The road network is treated as a graph with nodes V and edges E. The $dis_{network}$ is the shortest path distance between the vehicles and target point and is calculated using Dijkstra algorithm while Haversine distance is used to calculate the distance. In IDW, the weights are inversely proportional to the distance and are raised to power value p (-ve for inverse relationship). With the increase in p, the weights of farther points are decreased rapidly (eq. 5). Algorithm 1 details the traffic congestion operator and the steps required to estimate the traffic.

$$\frac{\sum_{i=1}^{n} (f(dis_{network}(v_i, t_0)))^{-p} S_i}{\sum_{i=1}^{n} (f(dis_{network}(v_i, t_0)))^{-p}} \tag{5}$$

The traffic congestion value will be higher if vehicles are nearer to the target points. For example, in Fig. 7(c) traffic congestion value at the target point t_0 will be more as compared to t_1 and t_2 as number of vehicles (v_0, v_1, v_2) near to t_0 is greater. Now the question arises: how many vehicles (such as v_3, v_4 in Fig. 7(c)) are already present in the road segment?

The OSM layer for London provides the maximum speed (S_{max}) of the road (48 mph), so the maximum travel time between camera points can be derived. The speed of the vehicles will lie in the range of $0 \leq S_i \leq S_{max}$. Suppose the camera feeds refresh every $t_{cam-refresh}$ seconds, then the distance covered by vehicle(v_i) will be:

$$D_i = S_i \times t_{cam-refresh} \tag{6}$$

Processing the camera video feeds are computationally intensive and leads to latency. The given time needs to be added to the current time to know the location of the vehicle. So, the total distance covered by the vehicle will be:

$$D_i = S_i \times (t_{Cam-refresh} + processing - latency) \tag{7}$$

If the vehicle(v_i) covers the distance D_i such that $0 \leq D_i \leq max_{road-length}$ where $max_{road-length}$ is the distance between two cameras, then this vehicle will be considered as sample points for traffic calculation. For example, let the distance between camera 2 and 3 is 1 mile (for easier calculation) and vehicle v_0 at camera 2 is moving with speed(S_i) of 40mph which is less than S_{max} (48mph). Suppose the camera feed refreshes ($t_{cam-refresh}$) every 10 seconds and requires 2 seconds to process ($processing - latency$) the new video feed. In such time the v_0 will cover distance(D_i) = (40/3600)*(10+2) i.e. 0.13 mile and will be near point t_0, t_1. Thus, we consider the vehicles which are identified from both the cameras and the vehicles which are present in the road segments identified from previous feeds to perform the an appropriate traffic calculation.

7 Experimental Results

7.1 Dataset and Implementation Details

Traffic Camera Data. For the experiments, two trunk roads (Brixton and Kennington) were selected from the Lambeth borough of London. As mentioned in Section 4, the traffic camera data feed was obtained from the TfL API by passing the camera location and search radius. Table 3 shows the list of cameras (total 22) installed on selected roads. A total 3080 video clips with 140 video clips (9 seconds) per camera are processed to identify the traffic status.

Hardware and Software. The system[6] was implemented in Python 3 over the VidCEP engine [29] running on a 16 core Linux Machine with 3.1 GHz processor, 32 GB RAM and Nvidia Titan Xp GPU. The microservices were wrapped in Docker containers with Redis Stream acting as a messaging service among the containers. The *OpenCV* library was used for image processing and *Darknet* and *PyTorch*[7] deep learning framework were used for object detection and tracking. *GStreamer* was used to stream camera video feeds from the TfL API. *OSMNX* library fetched road network from OSM and calculated network distance [6].

Creation and Updating of Traffic Overlay over OSM. We used the Leaflet[8] JavaScript library to overlay traffic information over OSM. The Node.js server supporting the Leaflet application was connected to the back-end system via Redis Streams. The Leaflet ColorLine class was used to highlight the roads with the given colour intensity.

[6] https://github.com/piyushy1/OSMTrafficEstimation
[7] https://pytorch.org/
[8] https://leafletjs.com/index.html

Table 3 List of Traffic Cameras for Study.

No.	Brixton Road Cameras	Kennington Road Cameras
C1	Camberwell New Rd/Brixton Rd	Kennington Lane/Newington Butts
C2	Brixton Rd/Island Place	Kennington Pk Rd/Penton Pl
C3	A23 Brixton Rd/Vassell Rd	Kennington Pk Rd/Braganza St
C4	A23 Brixton Rd/Hillyard St	Kennington Pk Rd/Kennington Rd
C5	A23 Brixton Rd/Ingleton St	Kenington Pk Rd/Kennington Oval
C6	A23 Brixton Rd/Wynne Rd	A3 Clapham Rd/Elias Place
C7	Brixton Rd/Stockwell Pk	A3 Clapham Rd/Handforth Street
C8	Acre Lane/Coldharbour Lane	A3 Clapham Rd/Crewdson Rd
C9	A23 Brixton Hill/Effra Rd	A3 Clapham Rd/Caldwell St
C10	Brixton Hill /Lambert Rd	A3 Clapham Rd/Landsdowne Way
C11	Brixton Hill / Elm Park	A3 Clapham Rd/
C12	Brixton Hill/Morrish Rd	NA

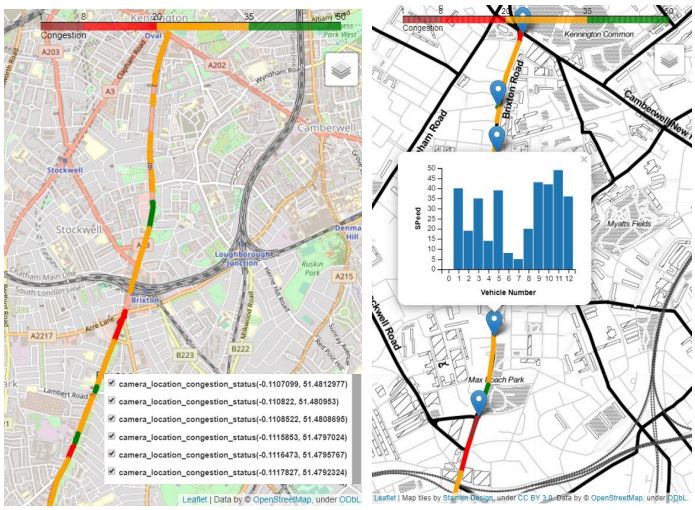
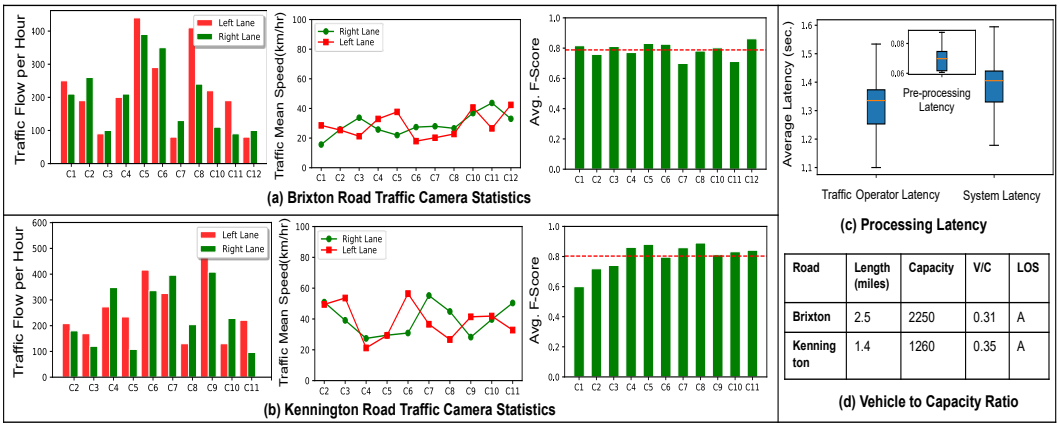


Figure 8 Traffic Congestion and Vehicle Speed Information over OSM.

7.2 Empirical Evaluation

Traffic Congestion Visualization over OSM. The traffic congestion is the interpolated value superimposed over the road segment between the cameras. As discussed in Section 6.1, the number of vehicles, their direction (left or right lane) and speed is calculated. The congestion is estimated for the road segment between the two cameras using e.q 5. The value of p =2 is used as the distance decay factor for interpolation. The TfL API upload the video feeds of 9 seconds (approx. 280 frames), and thus we process them in two TIME WINDOW of 4.5 seconds (140 frames) each. Using eq. 7, for the second time window, we also estimate if any vehicle from the previous window is present on the road segment and take this into consideration for congestion.

Different map services calculate traffic congestion values, but they remain opaque about their method. For example, *Here Maps API* provides *Jam factor* values and divides them into four jam categories without explaining how the values are calculated. In this work, count and speed are considered as parameters for congestion estimation; so if the speed of vehicles is high then the traffic is considered smooth. As per OSM data, the maximum speed



Road	Length (miles)	Capacity	V/C	LOS
Brixton	2.5	2250	0.31	A
Kenning ton	1.4	1260	0.35	A

Figure 9 Different Traffic Statistics for Roads Calculated Using Camera Video Feeds.

limit for the selected roads is 48mph (approx. 77km/hr). In Fig. 8, a four-step colour-based traffic congestion over Brixton Road is shown where- 1)Green - No traffic (45-70 km/hr) 2) Orange - light traffic (30-45 km/hr), 3)Red - moderate traffic(20-30 km/hr) and 4) Brown-high traffic(10-20 km/hr). The above-defined congestion parameters range is not static and can be reconfigured depending on requirements. Fig. 8(right) also shows specific traffic data available by clicking anywhere along the road. Since we could not perform a direct comparison with other map services, we can only visually compare that the traffic dynamics they provide are similar to our proposed technique.

Traffic Flow Rate and Speed. As per Table 1, the Traffic Flow rate (TF) is the number of vehicles in each lane per unit of time. TF is calculated for small periods and then extrapolated to larger time scales. For example, if 20 vehicles observed in 10 minutes, then TF will be 120 vehicles per hour [22]. Fig. 9(a) and (b) shows the hourly TF for Brixton and Kennington roads for both lanes. The camera C5 (Brixton-left lane) and C9 (Kennington-left lane) have max. flow rate of 410 and 572 vehicles per hour. The reason behind such a low flow rate is the lockdown measures currently in place in London because of the COVID-19 pandemic situation. Fig. 9(a) and (b) shows the Traffic Mean Speed (TMS) which is the average speed of vehicles recorded at a given point for a selected period. The Brixton road TMS ranges between 18-46 km/hr while that of Kennington is from 20-58 km/hr. Cameras where speed is below 20 km/hr is due to the video feeds having more red light signals. Fig. 8 shows the TMS example on the OSM map where markers can be clicked to get the current traffic status (e.g vehicle speed and count).

Traffic Density and Volume to Capacity Ratio(V/C). Traffic density is the number of vehicles occupying a unit length of road. The Road Task Force from TfL specifies the capacity of the road as 900 vehicles per mile. Fig. 9(d) shows the V/C ratio for both roads to identify the Level of Service (LOS). During the experimentation time, the V/C ratio for Brixton and Kennington was 0.31 (LOS-A) and 0.35 (LOS-A) respectively. As discussed earlier the lower V/C ratio is due to lockdown measures as a result of COVID-19 restrictions.

Traffic Prediction Accuracy and Latency. The above-discussed traffic estimates are linked to the performance of the object detector (YOLO) and tracker (DeepSORT) models. The error in these models will directly propagate to the traffic estimates and skew the overall

results. F-score is the standard metric to identify the performance of the classifier. It is the harmonic mean of precision and recall and is calculated as:

$$F = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{8}$$

In eq. 8, the precision is the ratio of *relevant events matched* and *matched events* while recall is the ratio of *relevant events matched* and *relevant events*. The mean F-score of each road camera (c_i) is calculated as $F_{mean} = \frac{\sum_{i=1}^{n} F_{ci}}{n}$. A sample of 110 video clips (each of 9 seconds) from 22 camera clusters were taken to identify the F-score. Fig. 9(a,b) shows that the mean F-score of the Brixton and Kennington camera is 0.78 and 0.80 respectively. The error estimation rate which propagates can be calculated as ER = (actual-approx)/actual*100 and is 22% and 20% respectively for both roads camera clusters. The low F-scores for some cameras were due to blurred FoV (like Kennington C1), tree shadows (like Kennington C1) and faulty cameras (e.g. Brixton C9 which was not included).

Latency measures the time required by the framework to process the traffic information and update it on the OSM. The latency can be divided into two stages: 1) Pre-processing Latency- the time required by DNN models to process the video stream to track and extract objects and 2) Traffic Operator Latency- the time required to compute traffic statistics from pre-processed data and update it on the OSM. Fig. 9(c) shows the box plot of pre-processing and traffic and system median latency of 0.071 seconds, 1.36 and 1.42 seconds, respectively which implies a near-real-time performance.

8 Discussion of Limitations

While using camera feeds for traffic estimation poses privacy risks, the cameras are state infrastructure with publicly available video streams. Open-source software built on top of the open data infrastructure is more transparent compared to the opaque black-boxes and data sources which certain corporations have exclusive access. As the quality of OSM data improves, providing value-added services such as traffic estimation is essential for the adoption of OSM as a mainstream routing service.

The uniqueness of this work lies in the query-based approach (VEQL) where users can query multiple video streams by deploying operators (e.g. traffic congestion, flow rate, speed, etc.) to the system. The framework can be deployed over the cloud and can be exposed as an API to provide services over OSM. More complex and potential traffic services like Vehicle Overtake and Lane Change [30] can be queried by creating operators and deploying them to the system. Our method relies on existing cameras installed along road networks and most of the limitations arise from camera deployment. For example, some cameras have sub-optimal FoVs because of obstacles. Cameras in London have a relatively low refresh rate, in the order of several minutes and provides only 9-second clips in each refresh cycle. Most of the CCTV's are installed in well-lit areas so the system can work in the night. Further, incremental weather may result in object mismatch due to lack of training datasets in such conditions. The inability to detect vehicles result in errors that propagate to final traffic computation. We have tested our model on cameras placed over straight roads and the strategy can be generalized for branched road networks.

In terms of generating visualizations for display over maps, it is possible to use more complex spatial interpolation (e.g. kriging) and classification techniques to get better congestion estimates. Using all cameras in a city will significantly improve the sample size and provide better performance in the interpolation step leading to more accurate estimates of travel time. The various measures of traffic we offer can also be used to create comprehensive

dashboards for communicating multi-dimensional aspects of traffic state. Finally, other open data streams (e.g. General Transit Feed Specification) can also be integrated to supplement traffic state calculations and estimation of travel times using different transport modes.

9 Conclusion

We have presented a traffic estimation framework built on open video streams based on open-source deep learning technology. We have exposed the results and data through visualizations on OSM. Exploiting the data from the video streams enabled us to extract traffic-related parameters beyond simple vehicle counts. Combining these multiple parameters provides opportunities to present a multidimensional analysis of the traffic state. For example, during interpolation, we treated each vehicle as a sample point and its speed as weight, thus factoring in both vehicle density and flow in the estimation of traffic. Users can utilize these parameters to calculate their own metrics for congestion. The VEQL query empower users to create their rules and deploy them as services for traffic-related events. We aim to deploy this service across multiple cities which make their traffic camera feeds available to be able to provide a comprehensive traffic state estimation service over OSM. We hope that places that do not make their feeds available publicly will adopt our open-source framework to provide their traffic estimation API to be consumed over OSM. Widespread adoption will provide the currently lacking feature of traffic state information on OSM and will be a step towards making OSM a viable alternative to commercial digital map service providers.

References

1 David Alm. Somebody's Watching You: Ai Weiwei's New York Installation Explores Surveillance in 2017, 2017. URL: `https://bit.ly/2Um1hT5`.

2 Sheeraz A Alvi, Bilal Afzal, Ghalib A Shah, Luigi Atzori, and Waqar Mahmood. Internet of multimedia things: Vision and challenges. *Ad Hoc Networks*, 33:87–111, 2015.

3 Jennings Anderson, Dipto Sarkar, and Leysia Palen. Corporate editors in the evolving landscape of openstreetmap. *ISPRS International Journal of Geo-Information*, 8(5):232, 2019.

4 Waadt Andreas, Wang Shangbo, Jung Peter, et al. Traffic congestion estimation service exploiting mobile assisted positioning schemes in gsm networks. *Procedia Earth and Planetary Science*, 1(1):1385–1392, 2009.

5 Asra Aslam and Edward Curry. Towards a generalized approach for deep neural network based event processing for the internet of multimedia things. *IEEE Access*, 6:25573–25587, 2018.

6 Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 2017.

7 John Canny. An algorithm to estimate mean traffic speed using uncalibrated cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

8 Thurston Regional Planning Council. Appendix o level of service standard and measurements. URL: `https://bit.ly/2z6yvzY`.

9 Gianpaolo Cugola and Alessandro Margara. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)*, 44(3):15, 2012.

10 Alan Demers, Johannes Gehrke, Mingsheng Hong, Mirek Riedewald, and Walker White. Towards expressive publish/subscribe systems. In *International Conference on Extending Database Technology*, pages 627–644. Springer, 2006.

11 Mordechai Haklay. How good is volunteered geographical information? a comparative study of openstreetmap and ordnance survey datasets. *Environment and planning B: Planning and design*, 37(4):682–703, 2010.

12 Alex Hern. Berlin artist uses 99 phones to trick google into traffic jam alert, 2020. URL: `https://bit.ly/34ISrF4`.

13 Gorkem Kar, Shubham Jain, Marco Gruteser, Fan Bai, and Ramesh Govindan. Real-time traffic estimation at vehicular edge nodes. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, pages 1–13, 2017.

14 George Kopsiaftis and Konstantinos Karantzalos. Vehicle detection and traffic density monitoring from very high resolution satellite video data. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1881–1884. IEEE, 2015.

15 Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

16 David G Lowe et al. Object recognition from local scale-invariant features. In *iccv*, volume 99, pages 1150–1157, 1999.

17 Michael Lowry. Spatial interpolation of traffic counts based on origin–destination centrality. *Journal of Transport Geography*, 36:98–105, 2014.

18 Pascal Neis, Dennis Zielstra, and Alexander Zipf. The street network evolution of crowdsourced maps: Openstreetmap in germany 2007–2011. *Future Internet*, 4(1):1–21, 2012.

19 Fatih Porikli and Xiaokun Li. Traffic congestion estimation using hmm models without vehicle tracking. In *IEEE Intelligent Vehicles Symposium, 2004*, pages 188–193. IEEE, 2004.

20 Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

21 Narushige Shiode and Shino Shiode. Street-level spatial interpolation using network-based idw and ordinary kriging. *Transactions in GIS*, 15(4):457–477, 2011.

22 NCSU Transport Tutorial Site. Traffic flow fundamentals: Flow, speed and density. URL: `https://lost-contact.mit.edu/afs/eos.ncsu.edu/info/ce400_info/www2/flow1.html`.

23 Yongze Song, Xiangyu Wang, Graeme Wright, Dominique Thatcher, Peng Wu, and Pascal Felix. Traffic volume prediction with segment-based regression kriging and its implementation in assessing the impact of heavy vehicles. *Ieee transactions on intelligent transportation systems*, 20(1):232–243, 2018.

24 Burkhard Stiller, Thomas Bocek, Fabio Hecht, Guilherme Machado, Peter Racz, and Martin Waldburger. Understanding and Managing Congestion For Transport for London. Technical report, Integrated Transport Planning Ltd., 2017. URL: `http://content.tfl.gov.uk/understanding-and-managing-congestion-in-london.pdf`.

25 Unkown. Design manual for roads and bridges- cd 127 cross-sections and headrooms. Technical report, Department for Transport UK, 2020. URL: `https://www.standardsforhighways.co.uk/ha/standards/`.

26 Website. How many cctv cameras in london? URL: `caughtoncamera.net/news/how-many-cctv-cameras-in-london/`.

27 Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

28 Piyush Yadav and Edward Curry. Vekg: Video event knowledge graph to represent video streams for complex event pattern matching. In *2019 First International Conference on Graph Computing (GC)*, pages 13–20. IEEE, 2019.

29 Piyush Yadav and Edward Curry. Vidcep: Complex event processing framework to detect spatiotemporal patterns in video streams. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2513–2522. IEEE, 2019.

30 Piyush Yadav, Dibya Prakash Das, and Edward Curry. State summarization of video streams for spatiotemporal query matching in complex event processing. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 81–88. IEEE, 2019.

31 Haixiang Zou, Yang Yue, Qingquan Li, and Anthony GO Yeh. An improved distance metric for the interpolation of link-based traffic data using kriging: a case study of a large-scale urban road network. *International Journal of Geographical Information Science*, 26(4):667–689, 2012.