Steve H.L. Liang
Xin Wang
Christophe Claramunt (Eds.)

Web and Wireless Geographical Information Systems

LNCS 7820

12th International Symposium, W2GIS 2013
Banff, AB, Canada, April 2013
Proceedings

Springer

Lecture Notes in Computer Science 7820

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Alfred Kobsa
University of California, Irvine, CA, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
TU Dortmund University, Germany

Madhu Sudan
Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

Gerhard Weikum
Max Planck Institute for Informatics, Saarbruecken, Germany

Steve H. L. Liang Xin Wang
Christophe Claramunt (Eds.)

Web and Wireless Geographical Information Systems

12th International Symposium, W2GIS 2013
Banff, Canada, AB, April 4-5, 2013
Proceedings

Springer

Volume Editors

Steve H. L. Liang
University of Calgary, Geomatics Engineering
Calgary, AB T2N 1N4, Canada
E-mail: steve.liang@ucalgary.ca

Xin Wang
University of Calgary, Geomatics Engineering
Calgary, AB T2N 1N4, Canada
E-mail: xcwang@ucalgary.ca

Christophe Claramunt
Naval Academy Research Institute
Lanvéoc-Poulmic BP 600, France
E-mail: christophe.claramunt@ecole-navale.fr

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-37086-1 e-ISBN 978-3-642-37087-8
DOI 10.1007/978-3-642-37087-8
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013932566

CR Subject Classification (1998): H.2.8, H.2-5, H.5, D.2, C.2

LNCS Sublibrary: SL 3 – Information Systems and Application,
incl. Internet/Web and HCI

© Springer-Verlag Berlin Heidelberg 2013

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in ist current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.
The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

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

Preface

This volume contains the papers selected for presentation at the 12th International Symposium on Web and Wireless Geographical Information Systems (W2GIS 2013), hosted by the University of Calgary in Banff, Alberta, Canada, in April 2013.

W2GIS 2013 aimed at providing a forum for discussing advances in theoretical, technical, and practical issues in the field of wireless and Internet technologies suited for the dissemination, usage, and processing of geo-referenced data. W2GIS is a unique forum to address new research challenges in the development of location-based and GIS. It now represents a recognized and regular event in the research community that continues to develop and expand. W2GIS 2013 was the first W2GIS event held in North America, demonstrating its establishment worldwide. Before W2GIS 2013, W2GIS was a series of events alternating between Europe and East Asia.

For the 2013 edition, we received 28 submissions from 16 countries in four continents. Each paper received three reviews and based on these reviews, 11 full papers and five short papers were selected for presentation at the symposium and inclusion in Springer's LNCS collection.

The accepted papers are all of excellent quality and cover topics that range from mobile GIS to sensor Webs. The program covered a wide range of topics including spatial semantics and databases, location-based services and applications, trajectory representation and sensor Web, spatial analysis and systems, and map generation and modeling. We also had the privilege of hosting two invited talks by Carl Reed, CTO of Open Geospatial Consortium, and Sheelagh Carpendale, Professor in Computer Science at the University of Calgary.

We wish to thank all authors that contributed to this symposium for the high quality of their papers and presentations. Our sincere thanks go to Springer's LNCS team. We would also like to acknowledge and thank the Program Committee members for the quality and timeliness of their reviews. Finally, we would like to thank the Steering Committee members for providing continuous support and advice.

January 2013

Steve Liang
Xin Wang
Christophe Claramunt

Organization

General Chairs

Steve Liang	University of Calgary, Canada
Xin Wang	University of Calgary, Canada
Christophe Claramunt	Naval Academy Research Institute, France

Steering Committee

Michela Bertolotto	University College Dublin, Ireland
James Carswell	Dublin Institute of Technology, Ireland
Christophe Claramunt	Naval Academy Research Institute, France
Max Egenhofer	NCGIA, USA
Ki-Joune Li	Pusan National University, Korea
Kazutoshi Sumiya	University of Hyogo, Japan
Taro Tezuka	University of Tsukuba, Japan
ChristelleVangenot	University of Geneva, Switzerland

Program Committee

Masatoshi Arikawa, Japan
Scott Bell, Canada
Alain Bouju, France
Thomas Brinkhoff, Germany
Elena Camossi, Italy
Rolf de By, The Netherlands
Sergio Di Martino, Italy
Matt Duckham, Australia
Peter Fröhlich, Austria
Yoshiharu Ishikawa, Japan
Hassan Karimi, USA
Kyoung-Sook Kim, Japan
Barend Kobben, The Netherlands
Miguel R. Luaces, Spain
Gavin McArdle, Ireland
Eoin Mac Aoidh, Italy
Herve Martin, France
Pedro R. Muro-Medrano, Spain
Kostas Patroumpas, Greece
Mathieu Petit, France
Dieter Pfoser, Greece
Cyril Ray, France
Markus Schneider, USA
Yannis Theodoridis, Greece
Martin Tomko, Australia
Taketoshi Ushiama, Japan
Huayi Wu, China
Stephan Winter, Australia
Phil Yang, USA

Table of Contents

Session 1: Spatial Semantics and Databases

Session 2: Location-Based Services and Applications (Part I)

Session 3: Location-Based Services and Applications (Part II)

Session 4: Trajectory Representation and Sensor Web

Session 5: Spatial Analysis and Systems

Session 6: Map Generation and Modeling

Grounding Linked Open Data in WordNet: The Case of the OSM Semantic Network*

Andrea Ballatore,[1] Michela Bertolotto,[1] and David C. Wilson[2]

[1] School of Computer Science and Informatics
University College Dublin, Ireland
{andrea.ballatore,michela.bertolotto}@ucd.ie
[2] Department of Software and Information Systems
University of North Carolina, Charlotte, NC
davils@uncc.edu

Abstract. In recent years, the linked open data (LOD) paradigm has emerged as a promising approach to structuring, publishing, and sharing data online, using Semantic Web standards. From a geospatial perspective, one of the key challenges consists of bridging the gap between the vast amount of crowdsourced, semi-structured or unstructured geo-information and the Semantic Web. Notably, OpenStreetMap (OSM) has gathered billions of objects from its contributors in a spatial folksonomy. The contribution of this paper is twofold. First, we add a piece to the LOD jigsaw, the OSM Semantic Network, structuring it as a W3C Simple Knowledge Organization System (SKOS) vocabulary, and discussing its role in the constellation of geo-knowledge bases. Second, we devise *Voc2WordNet*, a mapping approach between a given vocabulary and WordNet, a pivotal component in the LOD cloud. Our approach is evaluated on the OSM Semantic Network against a human-generated alignment, obtaining high precision and recall.

Keywords: Geo-semantics, OpenStreetMap, Linked Open Data, OSM Semantic Network, WordNet, Semantic alignment, Semantic mapping, Voc2WordNet.

1 Introduction

Since its invention in the early 1990s, the World Wide Web (WWW) has enabled an unprecedented growth of digital data, offering a platform for publishing, retrieving, and sharing any type of data across the globe. An enormous volume of data has been disseminated online in a variety of formats, resulting in an archipelago of incompatible data spaces. A crucial limitation to the full exploitation of this ocean of heterogenous data is the lack of clear semantics, which hinders the ability to analyse, explore, and discover unexpected connections and relations between entities.

* The research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support.

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 1–15, 2013.
© Springer-Verlag Berlin Heidelberg 2013

A prominent attempt to overcome this structural limitation of the WWW, and provide a unified platform for data semantics, is Berners-Lee's Semantic Web [10]. One of the most successful outcomes of this ambitious initiative is the so-called linked open data (LOD) paradigm, with the purpose of creating a unified data space. To be classified as LOD, data must be released under open licenses; saved in a machine-readable digital format; stored in non-proprietary formats; accessible via URIs; and linked to other LOD [9].

As LOD is generated and published online, a growing web of inter-linked datasets has emerged, resulting in the LOD cloud, also referred to as the Web of Data, defined by Bizer et al. [11] as "a web of things in the world, described by data on the Web" (p. 2). The more linked data is available, the more connections can be discovered between datasets, exploiting network effects to deliver rich and relevant results to users [21]. Large linked data repositories are maintained online.[1] Recently, the commercial potential of the paradigm has been highlighted by Google's Knowledge Graph, a large semantic artifact that utilises Freebase, an LOD resource, to semantically enrich the search engine's results [30].

As a large part of online data involves a spatial component, geographic open data is a first class citizen in the LOD cloud [6]. Semantics is key to enabling the usage, integration, and exploration of geographic data [1, 21]. The advantages of the LOD paradigm applied to geographic information are particularly evident in the context of geographic information retrieval (GIR), where existing techniques have shown limited effectiveness [28]. A linked data search engine such as DBpedia Faceted Search promises – and often returns – highly relevant results to complex geospatial queries, such as 'Rivers that flow into the Rhine and are longer than 50 kilometers.'[2]

The emergence of the LOD infrastructure has a great potential for the dissemination of geographic data. A prominent example is found in the British Ordnance Survey, which has embraced the paradigm and released some of its resources as linked data [19]. In parallel, volunteered geographic information (VGI) is gaining credibility as a source of detailed information generated by non-expert users through crowdsourcing [13]. Challenging traditional top-down cartographic engineering, OpenStreetMap (OSM) provides an open platform to build a world map, tapping its contributors' knowledge of their local geographic context [12]. To date, a gap between VGI datasets and the LOD cloud exists, and constitutes a barrier to the integration and usage of the data.

In this paper, we contribute to bridging the gap between VGI and the LOD cloud in two ways. First, we describe how we have structured the OSM Semantic Network using the W3C Simple Knowledge Organization System (SKOS), and published online as LOD. The OSM Semantic Network offers a machine-readable, structured, open conceptualisation of OSM semantics, and constitutes a semantic support tool to interpret, search, and tap the project's vast vector dataset. We originally extracted the network from the OSM Wiki website and other sources to compute the semantic similarity of geographic terms [5]. Second, we outline and

[1] See for example `http://thedatahub.org` (acc. Oct 30, 2012).
[2] `http://wiki.dbpedia.org/FacetedSearch` (acc. Oct 30, 2012).

evaluate *Voc2WordNet*, a semantic mapping technique to connect OSM terms to WordNet synsets, enabling the discovery of rich semantic relations between terms such as part-whole (e.g. part-of relations) and subsumption (e.g. is-a relations). This semantic mapping is not a goal in itself, but can enable a number of search operations on both OSM and WordNet.

The remainder of this article is organised as follows. Section 2 reviews relevant work in the areas of LOD, open geo-knowledge bases, OSM semantics, semantic mapping, and WordNet. Section 3 presents an LOD resource extracted from OSM semantics, the OSM Semantic Network. Section 4 describes and formalises a generic approach to semantic mapping onto WordNet. Subsequently, we report on the evaluation of the approach, executed on a subset of terms from the OSM Semantic Network (Section 5). This paper concludes with a summary of results and directions for future research in Section 6.

2 Related Work

OSM has received wide attention, generating a large number of academic studies and commercial projects. This section surveys related work relevant to the OSM Semantic Network, VGI, and WordNet, with respect to geo-semantics and the LOD paradigm.

2.1 OpenStreetMap Semantics

From its foundation in 2004, OSM has established itself as the most ambitious VGI project [12]. From a semantic viewpoint, OSM is a semi-structured folksonomy, which allows contributors to create any new term to describe the objects that they find worth mapping [32]. This radically open approach to geo-semantics is supported by the fact that an all-encompassing geographical ontology is an unrealistic endeavour, and that a bottom-up negotiation allows for more experimentation, and attracts non-expert contributors. As project founder Steve Coast [12] succinctly put it, "to dictate [terms] as in a top-down ontology would have been nuts." The downsides of the adoption of a semi-structured folksonomy include wide variability and ambiguity in the interpretation of terms, proliferation of near-synonym terms, and lack of explicit semantic relations, resulting in a 'spatially rich and semantically poor' dataset [4].

In recent years, efforts have been undertaken to strengthen the thin semantic ground on which OSM rests, including LinkedGeoData [2], and OSMOnto.[3] Baglatzi et al. [3] devised an approach to grounding the OSM folksonomy on the DOLCE upper-level ontology [17]. Acknowledging the extreme difficulty in implementing such semantic mapping in an automatic way, they designed a game with a purpose (GWAP) to crowdsource a human-quality mapping. In our previous work, we devised an initial semantic integration between OSM and DBpedia, geared towards exploratory navigation of Web maps [4].

[3] `http://wiki.openstreetmap.org/wiki/OSMonto` (acc. Oct 30, 2012).

To tap the knowledge contained in the OSM Wiki website, we extracted the OSM Semantic Network via a dedicated open source crawler. An early, off-line version of the semantic network was utilised to compute the semantic similarity of OSM terms using link-based measures [5]. In this paper, we extend the OSM Semantic Network by re-structuring it as a SKOS vocabulary, integrating it in the LOD cloud, and devising a mapping technique to WordNet.

2.2 WordNet as Semantic Ground

Since the early 1990s, WordNet has been a precious semantic resource for many applications in natural language processing and artificial intelligence [16]. The core element of WordNet is the 'synset,' a concept that represents set of synonymous words, called 'word senses.' WordNet has found particular success in the areas of word sense disambiguation and semantic similarity [26, 7]. Different components of the network have been exploited to model the semantic similarity of its synsets, tapping its deep taxonomy, and the word definitions, called 'glosses' [e.g. 29]. Although the semantic network was not designed for this purpose, it has been frequently used as an upper level ontology, i.e. a general-purpose semantic ground, for example to discover semantic connections in unstructured data [22].

From a geospatial viewpoint, GeoWordNet aggregates WordNet synsets with the open gazetteer GeoNames [18]. To date, none of the numerous alternative semantic resources has yet managed to dethrone WordNet from its leading position as a general-purpose semantic ground. In the context of the LOD cloud, WordNet is used as a high-quality primary information source in many projects [6]. The lexical database is a well-established linked dataset, wired to a number of open knowledge bases.[4] These resources are inter-linked with DBpedia, a core node of the LOD cloud. In this paper, we devise a general technique to map a vocabulary onto WordNet, using it as a limited, and yet rich semantic ground.

2.3 Open Data Integration

To generate LOD, it is necessary to link the new entities to existing ones in the LOD cloud, a process often called 'bootstrapping' [23]. The identification of the same concepts and entites in heterogenous data spaces is crucial to supporting the Semantic Web. Merging different conceptual schemas is a time-honoured challenge in computer science, started well before the advent of the WWW. Logical reasoning, machine learning, and statistical analysis have been utilised to tackle the problem in the context of database schemas [27].

The Ontology Alignment Evaluation Initiative (OAEI) has proposed benchmarks and performance metrics specifically tailored to the area of ontology alignment and integration [15]. Several approaches to generating a mapping have been devised, both from an intensional and an extensional viewpoint. *Terminological* methods rely on simple string matching between the terms, while *semantic* methods compare the representation of terms in formal semantic models. Furthermore, *internal* methods observe aspects of the terms in isolation, such as

[4] `http://www.w3.org/2006/03/wn/wn20/` (acc. Oct 30, 2012).

Table 1. Namespaces of the OSM Semantic Network and related datasets

Abbr.	Description	URI
osn	OSM Semantic Network	`http://spatial.ucd.ie/lod/osn/`
owl	OWL	`http://www.w3.org/2002/07/owl\#`
rdfs	RDF schema	`http://www.w3.org/2000/01/rdf-schema\#`
dc	Dublin Core	`http://purl.org/dc/elements/1.1/`
skos	SKOS	`http://www.w3.org/2004/02/skos/core\#`
wn	WordNet synset	`http://www.w3.org/2006/03/wn/wn20/instances/synset-`
ws	– word sense	`http://www.w3.org/2006/03/wn/wn20/instances/wordsense-`
wns	– schema	`http://www.w3.org/2006/03/wn/wn20/schema/`
lgdo	LinkedGeoData	`http://linkedgeodata.org/ontology/`

the attribute ranges. By contrast, *external* methods analyse the relational structure of the ontologies, comparing the position of the terms relative to the other terms. Finally, *extensional* methods perform the alignment based on distributional properties of term instances.

Despite the variety of existing mapping techniques, to the best of our knowledge, a semantic mapping technique between a vocabulary and WordNet, geared towards the 'bootstrapping' of the vocabulary in the LOD cloud, has not been devised. *Voc2WordNet* has the purpose of filling this specific gap. As described in Section 4, *Voc2WordNet* performs the semantic mapping between a vocabulary term and a specific WordNet word sense from an intensional (i.e. lexical overlap between the lexical definitions) and an extensional perspective (i.e. the usage frequency). The next section describes our contribution to the area of VGI in the LOD cloud.

3 The OSM Semantic Network as Open Data

The OSM Semantic Network is a semantic artifact containing the conceptualisation of OSM tags, which we developed in our previous work to provide a semantic support tool for OSM.[5] The artifact can be used to compute the semantic similarity of tags [5]. In this section, we report on how the OSM Semantic Network has been structured using W3C Simple Knowledge Organization System (SKOS), and published online in the LOD cloud.

From a semantic viewpoint, OSM is a semi-structured folksonomy. The terms are documented on the OSM Wiki website, in an open process of semantic negotiation and consensus-building. Unsurprisingly, the consistency in the actual usage and intended meaning of these terms is rather low, resulting in semantic ambiguity that hinders the possibility of exploiting the project's rich vector dataset [25]. The OSM Semantic Network provides a machine-readable structure that can support the automatic manipulation of OSM features in data mining, GIR, and information integration.

Initially developed as an offline dataset, the OSM Semantic Network has been integrated in the LOD cloud. In order to facilitate the exploration and usage of

[5] `http://wiki.openstreetmap.org/wiki/OSMSemanticNetwork` (acc. Oct 30, 2012).

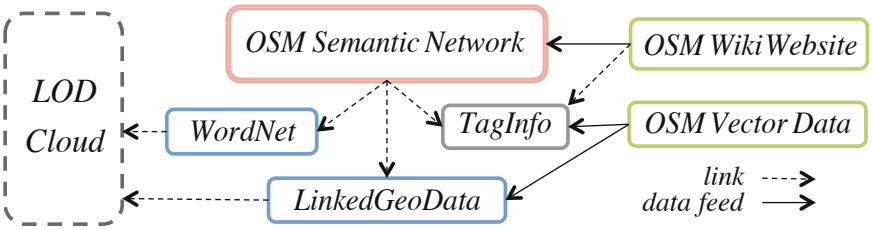


Fig. 1. The OSM Semantic Network in context

the network, we have published it online with a human-readable web interface.[6] Figure 1 shows the location of the OSM Semantic Network in the context of LOD, and the data flow from and towards related projects, including OSM, LinkedGeoData, WordNet, and TagInfo. For the sake of brevity, all the URIs in the remainder of this article are shortened (see Table 1).

We have structured the OSM Semantic Network as a SKOS vocabulary [24]. SKOS is a semantic formal language designed to allow the publication and sharing of technical vocabularies, taxonomies, and classification systems. In a SKOS scheme, the main semantic unit is the *skos:Concept*. A concept is a term that can be defined using lexical definitions and linked to other concepts through semantic relations. The semantic relations are explicitly left as generic as possible. Concepts can be more general or specific than other concepts (*skos:broader* and *skos:narrower*), and can be semantically related (*skos:related*).

Hence, each term defined in the OSM Wiki website corresponds to a SKOS concept. As the URIs are a key asset in LOD, the mapping between OSM tags and OSM Semantic Network terms is direct and intuitive. For example, the tag *waterway=river* corresponds to the term *osn:term/k:waterway/v:river*. The quality of the SKOS vocabulary was assessed based on the criteria outlined by Suominen and Hyvönen [31]. The OSM Semantic Network is linked to the LinkedGeoData ontology, via about 660 *skos:exactMatch* relations. Our approach to grounding a given vocabulary in WordNet is described in the next section.

4 *Voc2WordNet*, a Semantic Mapping Algorithm

This section presents *Voc2WordNet*, an algorithm devised to generate a semantic mapping between a vocabulary and the lexical database WordNet. The algorithm generates a semantic mapping between a given vocabulary V containing a set of terms (e.g. a SKOS vocabulary), and WordNet synsets that are semantically similar. *Voc2WordNet* can be used to map any vocabulary onto WordNet, enabling some degree of interoperability. More formally, a semantic mapping m between term $t \in V$ and synset $s \in W$ with relation r has the form of a triple $< t, r, s >$. In the OSM Semantic Network, we define a fine-grained semantic

[6] Pubby, available at `http://www4.wiwiss.fu-berlin.de/pubby` (acc. Oct 30, 2012).

mapping, based on the SKOS mapping relations.[7] Hence, *Voc2WordNet* generates three symmetric mapping relations:

Exact (*skos:exactMatch*): Identical terms that can be used interchangeably with high confidence (e.g. 'university' in OSM and LinkedGeoData). This relation is logically equivalent to *owl:sameAs*.

Close (*skos:closeMatch*): Similar terms that might contain some contradiction, and therefore cannot engage in identity (e.g. 'wood' in OSM and 'forest' in WordNet).

Related (*skos:relatedMatch*): Terms that are semantically related by a non-hierarchical relation (e.g. 'power station' in OSM and 'electricity' in WordNet). This relation is non-transitive.

The purpose of *Voc2WordNet* is to obtain correct mappings $m =< t, r, s >$ between the vocabulary V and the WordNet synsets W. For example, the definition of *wn:gallery-noun-3* is "a room or series of rooms where works of art are exhibited." By contrast, *wn:gallery-noun-1* is defined as "spectators at a golf or tennis match," and *wn:art-noun-1* as "the products of human creativity; works of art collectively." Hence, the desired mappings are *<osn:Art_gallery close wn:gallery-noun-3>* and *<osn:Art_gallery related wn:art-noun-1>*.

Voc2WordNet generates a set M of mappings m between a given vocabulary V and the set of WordNet synsets W. Given a term $t \in V$, *Voc2WordNet* utilises a lexical matching function on the words contained in t, taking compound words into account (e.g. 'swimming pool'), and then splitting them if not defined in WordNet (e.g. 'swimming' and 'pool'). If the set of matching wordsenses ws is not empty, the algorithm relies on three indicators of semantic salience:

Word sense frequency f: The usage frequency f of a WordNet word sense is correlated with its semantic salience. In the context of a shared vocabulary, common word senses are more likely to be correct than uncommon word senses. For example, for $t =$'field', *ws:field-noun-1* ("a piece of land cleared of trees and usually enclosed") has a usage frequency $f = 49$, whilst *ws:field-noun-12* ("all of the horses in a particular horse race") has $f = 1$. Indeed, this assumption can be false in the context of open text.

Lexical overlap *ol*: Similar terms tend to be defined using the same words. The lexical overlap *ol* is the number of word shared by two terms. Terms showing high lexical overlap are more likely to be salient than terms that do not show overlap. The overlap is considered after the removal of stopwords, and lemmatisation, excluding the term that is being defined. For example, the overlap between the definitions of term t ("A river is a body of water") and *wn:river-noun-1* ("Rivers are natural streams of water") is equal to 1.

Salient taxonomy Θ: If a vocabulary is domain specific, the mapping can be restricted to a salient taxonomy Θ, i.e. a subset of WordNet. Salient word senses tend to engage in semantic relations with salient synsets. Looking at the noun taxonomy of WordNet, it is possible to select high-level synsets that

[7] `http://www.w3.org/TR/skos-reference/\#mapping` (acc. Oct 30, 2012).

are salient to the vocabulary's domain. If the candidate synsets engage in some relation with such salient taxonomical roots, they are more likely to be valid than synsets that do not. For example, let us choose *wn:artifact-noun-1* as a salient root, and 'shelter' as t. It is possible to infer that *ws:shelter-noun-2* ("protective covering that provides protection from the weather") is related to the salient root through a path of transitive subsumption relations (*wns:hyponymOf*), while *ws:shelter-noun-4* ("a way of organizing business to reduce the taxes it must pay on current earnings") is not.

Formally, we define t as the input term, C_t as the set of candidates for term t, ws as the candidate word sense, s as the corresponding synset, and Θ as a manually selected salient taxonomy. The non-negative θ is set to 1 if $s \in \Theta$, and 0 otherwise. The salience of the three indicators are captured in a normalised score σ as follows:

$$\sigma(t, ws, s) = \frac{2|C_t| - rank(f(ws)) - rank(ol(t, s)) + \theta}{2|C_t| - 1} \tag{1}$$
$$\sigma \in [0, 1],\ rank \in [1, |C_t|]$$
$$\theta = 1\ if (s \in \Theta), \theta = 0\ otherwise$$

The salience score σ captures the semantic similarity between term t and the synset s, through the word sense ws, relative to the set of candidates C_t. The ranking function $rank$ is applied on the set C_t, and returns an integer between 1 and $|C_t|$. The score falls in the interval $[0, 1]$, where 0 indicates no salience, and 1 maximum salience. For example, given a C_t with three candidates, if ws and s have the highest frequency ($rank(f) = 1$), the second highest overlap ($rank(ol) = 2$), and s belongs to the salient taxonomy Θ ($\theta = 1$), then $\sigma = .8$.

In order to provide more flexibility, the algorithm filters out candidates based on a minimum frequency (f_{min}), and a minimum overlap (ol_{min}). Once the candidate having the highest σ has been selected, an appropriate relation r must be chosen from the set { *exact*, *close*, *related* }. As a selection heuristic, we define three boolean conditions, i.e. $rank(f) = 1$, $rank(ol) = 1$, and $s \in \Theta$. If all of the three conditions are true, $r = exact$; if at least two conditions are true, $r = close$; othwerwise $r = related$. The detailed workings of the algorithm are outlined in Algorithm 1. In the next section, *Voc2WordNet* is evaluated on a real-world scenario, i.e. a subset of the OSM Semantic Network.

5 Evaluation

This section describes a preliminary experimental evaluation of *Voc2WordNet*, applying the semantic mapping technique to the OSM Semantic Network. The technique obtains a high-precision mapping between the terms defined by the OSM Semantic Network and WordNet. First, we generate an evaluation dataset M_h (Section 5.1). Second, we define performance measures (precision and recall) that compare the machine-generated mapping M with the human mapping

Algorithm 1. *Voc2WordNet*$(V, W, ol_{min}, f_{min}, \Theta)$

input : vocabulary V, set of synsets W, min overlap ol_{min}, min word sense frequency f_{min}, salient taxonomy Θ
output: Set M of semantic mappings $m < \ldots t, r, s >$

```
M ← ∅
foreach term t ∈ V do
    m ← findSemanticMapping(t, W_t);
    add m to M;
    extract terms from lexical definition of t to set D_t;
    foreach term d ∈ D_t do
        m_d ← findSemanticMapping(d, W_t)
        add m_d to M;
return M.
```

Function findSemanticMapping(t, W_t)

```
C_t ← ∅
foreach ws ∈ W_t do
    find set of matching word senses ws ∈ W_t with lexicalMatch;
    find synset s corresponding to ws in WordNet;
    fetch word sense frequency f(ws) from WordNet;
    compute lexical overlap between definitions ol(s, t);
    apply filters f_min and ol_min;
    compute salience score σ(s, ws, t);
    add pair < s, ws > to candidate set C_t;
select best candidate s_b ∈ C_t having max(σ(s, ws, t));
select relation r ∈ { exact, close, related };
generate mapping m =< t, r, s_b > and return it.
```

M_h (Section 5.2). An experiment on a number of parameter combinations is executed (Section 5.3), and the performance of *Voc2WordNet* is discussed and summarised (Section 5.4).

5.1 Ground Truth

To construct a mapping gold standard, we select a random sample of 30 terms from the OSM Semantic Network, corresponding to the 0.6% of the entire dataset. The sample terms were manually mapped to semantically salient WordNet synsets. By manually selecting correct mappings between the 30 terms from the OSM Semantic Network and WordNet synsets, we obtain a human-generated mapping M_h, which includes 114 correct mappings. This dataset can be utilised as a ground truth to evaluate *Voc2WordNet*, our semantic mapping technique.

5.2 Evaluation Measures

To evaluate the performance of *Voc2WordNet*, we define the following performance measures. Following Euzenat [14], we assume that a correct mapping m belongs both to the machine mapping M and the human mapping M_h ($m \in M \wedge m \in M_h$). By contrast, an incorrect mapping only belongs to the machine mapping ($m \in M \wedge\ m \notin M_h$). Hence, we define precision P and recall R of mapping M as:

$$P_M = \frac{|M \cap M_h|}{|M|} \quad R_M = \frac{|M \cap M_h|}{|M_h|} \quad P_M, R_M \in [0,1] \tag{2}$$

All these measures fall in the interval $[0,1]$, with 1 as the best possible result ($M \equiv M_h$), and 0 as the worst ($M \cap M_h = \emptyset$). These measures will be used as indicators of the quality of the semantic mapping in the next sections.

5.3 Experiment Set-Up

The algorithm *Voc2WordNet* takes five parameters: V, W, ol_{min}, f_{min}, and Θ (see Section 4). Keeping the vocabulary V and WordNet W constant, we want to assess the impact of the other three parameters, ol_{min}, f_{min}, and Θ. Hence, we define the following parameters:

- Salient taxonomy Θ: either $\Theta \equiv W$ (i.e. taxonomy disabled), or a taxonomy of geographic terms (2 options);
- Minimum lexical overlap ol_{min}: $\{0,1,2\}$ (3 options);
- Minimum word sense frequency f_{min}: $\{0,1,2\}$ (3 options).

These parameters result in 18 unique combinations of parameters. A random disambiguation approach is added as a baseline. In order to disambiguate the terms from the OSM Semantic Network to the corresponding word sense in WordNet synsets, we select a subset of the WordNet taxonomy Θ that is relevant to the OSM context, i.e. entities and processes that are employed to describe OSM objects.

By manually observing the upper level of WordNet (i.e. synsets with depth ≤ 3), we selected eight synsets as roots of the salient taxonomy (see Table 2). All children synsets were subsequently recursively extracted, resulting in a salient taxonomy Θ of 6,312 noun synsets, navigating the *wns:hyponymOf* and *wns:partMeronymOf* relations. The salient taxonomy corresponds to about 7% of the entire WordNet noun taxonomy. The algorithm *Voc2WordNet* was executed on the 18 parameter combinations.

Table 2. Salient synsets in the upper part of the WordNet taxonomy

Salient taxonomical roots in WordNet	
wn:location-noun-1	*wn:artifact-noun-1*
wn:land-noun-2	*wn:activity-noun-1*
wn:ecosystem-noun-1	*wn:water_system-noun-1*
wn:natural_object-noun-1	*wn:natural_phenomenon-noun-1*

5.4 Experiment Results

The experiment generated 18 mappings of the OSM Semantic Network on WordNet synsets. Each mapping was compared with the human-generated dataset described in Section 5.1, obtaining precision and recall values. In order to analyse the impact of each parameter on the results, we summarise the performance indicators in Table 3, showing the mean precision $\bar{P}_M$ and recall $\bar{R}_M$. As expected, precision and recall are inversely proportional. All of the three filters (Θ, f_{min}, ol_{min}) have a positive impact on the precision, and a negative impact on the recall. The filter based on the salient taxonomy Θ improves the mean precision $\bar{P}_M$ from .72 to .81, with a minimal loss of recall. Similarly, the filter based on f_{min} and ol_{min} increases the mean precision at the expense of the mean recall. These results support the validity of the key ideas behind *Voc2WordNet*, described in Section 4.

Table 3. Experiment results of *Voc2WordNet* on the OSM Semantic Network. (*) Best precision and recall.

Parameter name	Parameter value	Mean $\bar{P}_M$	Mean $\bar{R}_M$
Random baseline	–	.21	.34
Taxonomy Θ	*off*	.79	.5*
	on	.88*	.49
Min frequency f_{min}	*(off)* 0	.82	.56
	1	.84*	.56
	2	.84*	.54
Min lexical overlap ol_{min}	*(off)* 0	.7	.82*
	1	.75	.81
	2	.87*	.49
Upper bounds	–	.88	.82

Considering the upper bounds obtained in this preliminary experiment ($P = .88$, $R = .82$), we consider *Voc2WordNet* to be a promising approach to grounding a vocabulary such as the OSM Semantic Network in WordNet. The optimal choice of the three parameters largely depends on the specific context in which *Voc2WordNet* is being applied. Based on specific users' needs, precision could be favoured over recall, or vice-versa. In order to extend this initial evaluation further, more terms could be included in the dataset, and the manual mapping could be performed and validated by a group of independent human subjects. In addition, the optimal parameters could be obtained using machine learning techniques on a desired training set of mappings.

6 Conclusions

Linked open data (LOD) constitutes a promising paradigm to create a shared semantic space, in which heterogenous geospatial datasets can inter-operate. In

the LOD cloud, WordNet can be used as a shared semantic ground to enable inter-operability between heterogenous vocabularies. In this paper, we described our two-fold contribution to the LOD cloud. First, we described the structuring of the OSM Semantic Network as LOD, using the W3C Simple Knowledge Organization System (SKOS). Second, we outlined and evaluated a semantic mapping algorithm, *Voc2WordNet*, which aimed at mapping a given vocabulary onto WordNet. The following conclusions can be drawn:

- The OSM Semantic Network bridges the semantics of OSM data and the LOD cloud. The network is extracted from the OSM Wiki website, a repository where contributors define, edit, and document the semi-structured folksonomy of tags. The dataset is structured as a SKOS vocabulary of terms utilised to describe OSM geographic features. We made the OSM Semantic Network freely available online,[8] and we linked it to existing semantic resources, including LinkedGeoData and TagInfo.
- Despite the advances reported in this article, the OSM Semantic Network presents a number of open challenges. As happens with crowdsourced resources, the network inevitably contains some degree of noise, ambiguity, and incorrect semantic mappings. Being a folksonomy, the OSM Semantic Network does not necessarily reflect ontological commitments in the vector data, and should therefore be utilised taking into account the intrinsic uncertainty of VGI.
- Our algorithm *Voc2WordNet* offers a general semantic mapping technique between a specialised vocabulary and the well-known lexical database WordNet. Given an input term from the vocabulary, *Voc2WordNet* identifies salient synsets in WordNet using three salience indicators: (1) the usage frequency of a term; (2) the term overlap between the lexical definition of the given term and the WordNet definition; and (3) a manually selected salient taxonomy. These indicators can be combined to increase precision, with a minor loss in recall. *Voc2WordNet* was tested on the OSM Semantic Network, obtaining high precision (.88) and recall (.82). A more extensive evaluation is necessary to demonstrate the effectiveness of *Voc2WordNet* across different vocabularies.

The OSM Semantic Network provides general-purpose semantic support for exploiting OSM data in geo-applications. Its integration with LinkedGeoData and WordNet enables the discovery of implicit semantic relations between map features, e.g. subsumption or meronomy, as well as the discovery of affordances, a promising approach to modelling the role of places. The network can support a number of semantic tasks, facilitating the computation of semantic similarity of geographic terms, and the matching of the same entities across LinkedGeoData, DBpedia, GeoNames, and other geo-knowledge bases [6].

Similarly, using GeoSPARQL [8] and federated queries over the LOD cloud,[9] it is possible, for example, to retrieve the schools from LinkedGeoData within

[8] `http://wiki.openstreetmap.org/wiki/OSMSemanticNetwork` (acc. Oct 30, 2012).
[9] `http://www.w3.org/TR/sparql11-federated-query` (acc. Oct 30, 2012).

a given geographic location, and to use the OSM Semantic Network to perform a semantic query expansion to features semantically related to school, such as kindergardens, highschools, and colleges.

Structuring VGI according to the LOD paradigm provides a valuable contribution to deliver richer, more structured geospatial information to both humans and machines. However, the LOD cloud presents a number of limitations that need to be addressed, in particular in relation to the management of identity [20], and spatio-temporal reasoning [21]. These issues notwithstanding, the LOD cloud already provides an open laboratory to a growing community of scientists, software developers, and GIS specialists. The OSM Semantic Network and *Voc2WordNet* constitute two further steps towards the inclusion of VGI into this vast semantic space.

References

1. Ashish, N., Sheth, A. (eds.): Geospatial Semantics and the Semantic Web: Foundations, Algorithms, and Applications, vol. 12. Springer, New York (2011)
2. Auer, S., Lehmann, J., Hellmann, S.: LinkedGeoData: Adding a Spatial Dimension to the Web of Data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 731–746. Springer, Heidelberg (2009)
3. Baglatzi, A., Kokla, M., Kavouras, M.: Semantifying OpenStreetMap. In: Proceedings of the 5th International Terra Cognita Workshop 2012 - Foundations, Technologies and Applications of the Geospatial Web. CEUR Workshop Proceedings, vol. 901, pp. 39–50 (2012)
4. Ballatore, A., Bertolotto, M.: Semantically Enriching VGI in Support of Implicit Feedback Analysis. In: Kim, K.-S. (ed.) W2GIS 2011. LNCS, vol. 6574, pp. 78–93. Springer, Heidelberg (2010)
5. Ballatore, A., Bertolotto, M., Wilson, D.: Geographic Knowledge Extraction and Semantic Similarity in OpenStreetMap. In: Knowledge and Information Systems, pp. 1–21 (2012)
6. Ballatore, A., Wilson, D., Bertolotto, M.: A Survey of Volunteered Open Geo-Knowledge Bases in the Semantic Web. In: Advanced Techniques in Web Intelligence - 3: Quality-based Information Retrieval. SCI. Springer (2012) (in Press)
7. Ballatore, A., Wilson, D.C., Bertolotto, M.: The Similarity Jury: Combining Expert Judgements on Geographic Concepts. In: Castano, S., Vassiliadis, P., Lakshmanan, L.V.S., Lee, M.L. (eds.) ER Workshops 2012. LNCS, vol. 7518, pp. 231–240. Springer, Heidelberg (2012)
8. Battle, R., Kolas, D.: Enabling the Geospatial Semantic Web with Parliament and GeoSPARQL. Semantic Web 3(4), 355–370 (2012)
9. Berners-Lee, T.: Linked Data (2006), `http://www.w3.org/DesignIssues/LinkedData.html`
10. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284(5), 28–37 (2001)
11. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)

12. Coast, S.: OpenStreetMap - The Best Map (February 19, 2010), OpenGeoData http://opengeodata.org/openstreetmap-the-best-map
13. Elwood, S., Goodchild, M., Sui, D.: Researching Volunteered Geographic Information: Spatial Data, Geographic Research, and New Social Practice. Annals of the Association of American Geographers 102(3), 571–590 (2012)
14. Euzenat, J.: Semantic precision and recall for ontology alignment evaluation. In: Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI), pp. 348–353 (2007)
15. Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., Trojahn, C.: Ontology Alignment Evaluation Initiative: Six Years of Experience. In: Spaccapietra, S. (ed.) Journal on Data Semantics XV. LNCS, vol. 6720, pp. 158–192. Springer, Heidelberg (2011)
16. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
17. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening Ontologies with DOLCE. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 166–181. Springer, Heidelberg (2002)
18. Giunchiglia, F., Maltese, V., Farazi, F., Dutta, B.: GeoWordNet: A Resource for Geo-spatial Applications. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part I. LNCS, vol. 6088, pp. 121–136. Springer, Heidelberg (2010)
19. Goodwin, J., Dolbear, C., Hart, G.: Geographical Linked Data: The Administrative Geography of Great Britain on the Semantic Web. Transactions in GIS 12, 19–30 (2008)
20. Halpin, H., Hayes, P.J., McCusker, J.P., McGuinness, D.L., Thompson, H.S.: When owl:sameAs Isn't the Same: An Analysis of Identity in Linked Data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 305–320. Springer, Heidelberg (2010)
21. Janowicz, K., Scheider, S., Pehle, T., Hart, G.: Geospatial Semantics and Linked Spatiotemporal Data: Past, Present, and Future. In: Semantic Web – Special Issue on Linked Spatiotemporal Data and Geo-Ontologies, pp. 1–13 (2012)
22. Lin, H., Davis, J., Zhou, Y.: An Integrated Approach to Extracting Ontological Structures from Folksonomies. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 654–668. Springer, Heidelberg (2009)
23. Mendes, P., Jakob, M., García-Silva, A., Bizer, C.: DBpedia Spotlight: Shedding Light on the Web of Documents. In: Proceedings of the 7th International Conference on Semantic Systems, pp. 1–8. ACM (2011)
24. Miles, A., Matthews, B., Wilson, M., Brickley, D.: SKOS Core: Simple Knowledge Organisation for the Web. In: International Conference on Dublin Core and Metadata Applications, DC-2005, pp. 3–10. DCMI Publications (2005)
25. Mooney, P., Corcoran, P.: Characteristics of heavily edited objects in OpenStreetMap. Future Internet 4(1), 285–305 (2012)
26. Navigli, R.: Word sense disambiguation: A survey. ACM Computing Surveys 41(2), 1–10 (2009)
27. Noy, N.: Semantic Integration: A Survey Of Ontology-Based Approaches. SIGMOD Record 33(4), 65–70 (2004)
28. Purves, R., Jones, C.: Geographic Information Retrieval. SIGSPATIAL Special 3(2), 2–4 (2011)

29. Ramage, D., Rafferty, A., Manning, C.: Random walks for text semantic similarity. In: Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing, pp. 23–31. ACL (2009)
30. Singhal, A.: Introducing the Knowledge Graph: things, not strings (May 16, 2012), http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html
31. Suominen, O., Hyvönen, E.: Improving the Quality of SKOS Vocabularies with Skosify. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) EKAW 2012. LNCS, vol. 7603, pp. 383–397. Springer, Heidelberg (2012)
32. Vander Wal, T.: Folksonomy (2007), http://vanderwal.net/folksonomy.html

A Strategy for Optimizing a Multi-site Query in a Distributed Spatial Database

Saad Zaamout and Wendy Osborn

Department of Mathematics and Computer Science, University of Lethbridge, Lethbridge, Alberta, T1K 3M4, Canada

Abstract. In this paper, we present a novel strategy for distributed spatial query optimization that involves multiple sites. Most previous work in the area of distributed spatial query processing and optimization focuses only on strategies for performing spatial joins and spatial semijoins, and distributed spatial queries that only involve two sites. We propose a new strategy, called the Restricted strategy, for optimizing a distributed spatial query. It uses spatial semijoins and can involve any number of sites in a distributed spatial database. The Restricted strategy improves upon an existing strategy by sending group approximations, instead of sending approximations for all objects, in order to reduce the number of comparisons between objects and thereby minimize the CPU and data transmission cost. A performance evaluation of our strategy finds that it significantly minimizes the number of data comparisons and CPU time of distributed spatial queries.

Keywords: distributed query processing, spatial data.

1 Introduction

A distributed spatial database system such as a Geographical Information System [13] contains several spatial database sites that are dispersed geographically. Each site manages its own collection of spatial data, but work collectively for processing inter-site queries and transactions. For example, suppose we have three sites in a province, where one site manages regions of population growth, one manages regions of Lyme disease, and one manages concentrations of wildlife. Each site would be managed separately, but could be use collectively to answer queries such as, "How many areas of population could be affected in the future with Lyme disease given animal migration patterns?".

An important requirement of a distributed spatial database is the ability to efficiently process a query that requires spatial data from multiple sites. Historically, research in distributed relational databases focused on generating query execution plans that minimized the cost of data transmission over the network [3,11]. However, spatial data is more complex than alphanumeric data, which thereby increases the complexity of joining spatial relations. CPU and I/O costs should also be considered when processing a distributed spatial query [13].

Most existing strategies for distributed spatial queries only work for two sites. One exception to this does not handle spatial joins, while the other exception

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 16–24, 2013.
© Springer-Verlag Berlin Heidelberg 2013

has only looked at minimizing data transmission cost. Therefore, we propose a new, improved multi-site distributed spatial query processing strategy. We find through our performance evaluation significant improvements in the number of comparisons that take place for a semijoin, which lowers both the CPU and data transmission costs.

This paper proceeds as follows. Section 2 presents related work in distributed spatial query processing. Section 3 presents our novel strategy for optimizing distributed spatial queries. Section 4 presents a performance evaluation of our strategy versus a recently proposed strategy that works for multiple sites. Finally, Section 5 concludes the paper and presents directions of future work.

2 Related Work

Previously proposed strategies in distributed spatial query processing focus on the use of spatial joins, spatial semijoins, and Bloom filters for processing queries. For all strategies below, with the exception of [6,12], all proposed strategies for query processing work for two sites only.

2.1 Spatial Join

A spatial join [13] takes two spatial relations R and S, and relates pairs of tuples between them, using a spatial predicate that is applied to the spatial attribute values. An example of a spatial join is an overlap spatial join, where each pair of objects is tested for overlap. Most spatial join algorithms are designed for a centralized system [7]. One application of a distributed spatial join comes from Kang et al. [8]. Their parallel strategy has two phases. The first phase is data redistribution, where on each site, the space that contains objects is partitioned into regions. A subset of regions is transmitted between sites so that each site has the same regions for both data sets. This subset is chosen so that the lowest overall response time is achieved. The second phase is filter and refinement, where the spatial join is performed on each site. An experimental evaluation shows that the parallel spatial join technique has a 33% faster response time over a semijoin-based strategy. One limitation of the strategy is no discussion of an extension to more than two sites.

2.2 Spatial Semijoin

A spatial semijoin [2] is performed by projecting the spatial attribute from one relation, transmitting it to the site that contains the other spatial relation, and performing a spatial join between the spatial projection and relation. Then, the qualifying tuples from second site are shipped back to the first site for joining. Existing work in distributed spatial semijoins focuses on modifying the operator [2,14,10] for a two-site query, with one additional work applying the spatial semijoin in a multiple-site query [12].

Abel *et al.* [2,14] combine the spatial semijoin operator that with the filter stage of spatial query processing to reduce the data transmission, I/O and CPU costs. Two adaptations of the spatial semijoin are proposed. The first is a "projection" of a set of MBRs (minimum bounding rectangles) from one spatial relation that is transmitted to the second site and applied to the other relation. The second is a "projection" consisting of a single-dimensional mapping that represents the objects in each relation. A performance evaluation between both approaches shows that: 1) for datasets with very large spatial descriptions, both strategies perform the same, 2) for datasets with smaller spatial descriptions, a semijoin that uses single-dimensional mapping works best, 3) using the R-tree for retrieving MBRs incurs significant CPU costs, and 4) single-dimensional mapping causes more false drops than MBRs.

The version from Karam and Petry [10] takes MBRs from different levels of the R-tree for the spatial semijoin, instead of from the same level. A performance evaluation shows that their spatial semijoin outperforms the Naïve spatial join (i.e. the whole relation is shipped to the other site for joining) when applied to real world data, but not when applied to randomly distributed rectangle sets. Other limitations of their work are no comparison versus other strategies and no consideration of CPU time.

Osborn and Zaamout [12] propose a distributed query processing strategy that works across multiple sites. Their approach is to apply semijoins by shipping smaller spatial projections to the sites of larger relations. A performance evaluation shows significant improvements in data transmission costs over the Naïve approach of shipping all relations directly to the query site. However, a limitation of this work is the lack of CPU and I/O cost evaluation.

2.3 Bloom Filters

A Bloom filter [1] is a hashed bit array that provides a compact but imprecise representation of the values of a joining attribute. A '1' bit represents the possible existence of a joining attribute value, while a '0' bit represents the absence of the value. Two proposed strategies [9,6] propose augmentations to the Bloom filter in a distributed environment.

Karam [9] propose a 2-dimensional bit-matrix approach for performing a semijoin of two relations that focuses on minimizing the data transmission, I/O and CPU costs. A 2-dimensional space is partitioned into equal-sized regions, with each region mapping to a bit in a 2-dimensional array. If a region contains objects, the corresponding bit is set to 1. This bit-matrix is transmitted to the site containing the other relation, and is applied by testing each region containing objects for the existence of a '1' bit in the bit-matrix. Any qualifying objects are sent to the first site. A performance evaluation shows that this approach shows the best improvement when applied to real world data. One Limitation of this work is no evaluation against the original spatial semijoin.

Hua et al. [6] propose the BR-tree, which is an R-tree augmented with Bloom filters to support exact-match queries. Along with an MBR, each node contains a Bloom filter that also represents the same set of objects represented by the MBR.

In a leaf node, a Bloom filter is created by taking each object and producing k filter bits. In a non-leaf node, a Bloom filter is created by intersecting the Bloom filters in its child node. Although the BR-tree supports exact-match queries by using Bloom filters, it still requires the MBRs for region and point queries. To process distributed region, point, and exact-match queries, the root of every BR-tree is duplicated across every site in the distributed database. Any objects that pass the test against a root node is shipped to the site containing the original BR-tree. This strategy works for any number of sites. However, a significant limitation is a lack of support for spatial joins.

3 Restricted Strategy

In this section, we present our strategy for processing and optimizing a distributed query. The Restricted strategy works for any number of sites, with the only requirement being that the number of sites must be a multiple of two. In addition, it improves upon the Optimized strategy [12] by reducing the number of object comparisons required in order to minimize both the data transmission cost and the CPU cost of the query. This is achieved by utilizing an optimization technique from the SpatialJoin2 join algorithm, which is proposed by Brinkhoff *et al.* [4]. We first provide a brief summary of both the Optimized strategy and SpatialJoin2 algorithm before presenting the Restricted strategy.

3.1 Background

The Optimized algorithm [12] uses spatial semijoins to reduce the data transmission cost and I/O cost of a distributed spatial query. This is accomplished by sending smaller projections of spatial attributes to larger relations for semijoin processing. To reduce I/O costs, the spatial projection is obtained by traversing the leaf level of an R-tree [5], which is used to index the spatial attribute of the relation on the site. This is significantly cheaper in terms of I/O than retrieving the entire relation and obtaining the spatial attribute from each tuple.

Given n sites that each contains a spatial relation, the Optimized strategy has four main steps [12]:

1. *Sorting and grouping by increasing spatial attribute cardinality.* After sorting all participating relations by increase spatial attribute cardinality, two groups of relations is formed - one containing the smaller cardinality relations, and the other containing the larger cardinality relations.
2. *Transmission of spatial attributes.* The spatial attribute with the smallest cardinality relation from the first group is sent to the site with the smallest cardinality relation from the second group. This strategy is further applied to the next smallest cardinality relations in both sets, and so on until all relations are processed.
3. *Semijoin execution.* Next, spatial semijoins are performed in parallel, and the identifiers of the spatial projection that qualified in the spatial semijoin are sent back to the originating site.

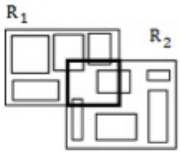


Fig. 1. Intersection Region

4. *Transmission of qualifying tuples to query site for the final join and processing.* From all sites, all tuples that qualified in their respective semijoins are shipped to the query site for final processing.

The algorithm SpatialJoin2 utilizes the concept of *intersection regions*. Given two leaf nodes and the corresponding minimum bounding rectangles (MBRs) that contain the objects in each leaf node, the intersection region is the area that is overlapped by both MBRs. The idea is that any object in either leaf node must overlap the intersection region in order to potentially qualify for a join with any other object. This can significantly reduce the number of comparisons between objects between different sites that must take place. Figure 1 depicts an intersection region with the rectangles that do, and do not, overlap with it.

3.2 Restricted Strategy

Given n participating sites, where each contains one spatial relation, we have the following steps:

1. Sorting and grouping by spatial attribute cardinality,
2. Creation and Transmission of MBRs that bound subsets of objects,
3. Calculation of intersection regions,
4. Identification of objects that intersect the intersection regions,
5. Transmission of qualifying tuples to query site for the final join.

Steps 1 and 5 are the same as those in the Optimized Strategy. We include them in the overall description for completeness.

First, the relations are ordered by increasing cardinality of their spatial attributes, and then are grouped into sets P (lower half of ordering, smaller spatial attribute cardinality) and Q (upper half of ordering, larger spatial attribute cardinality). Second, for each relation in P and Q, a set of MBRs is created for each subset of spatial objects in the relation.

If the spatial attribute is indexed with an R-tree, then the set of MBRs from each leaf node (which can be obtained from the parent node) can be used for the required set of MBRs. Figure 2 depicts an R-tree that is indexing the given set of objects (from [5]). In the Optimized strategy [12], the leaf-level entries (MBRs and identifiers) would be obtained and used as the required spatial projection. In the new Restricted Strategy, the non-leaf-level entries (in the example, the entries in the root node), are the ones obtained and sent as the set of MBRs

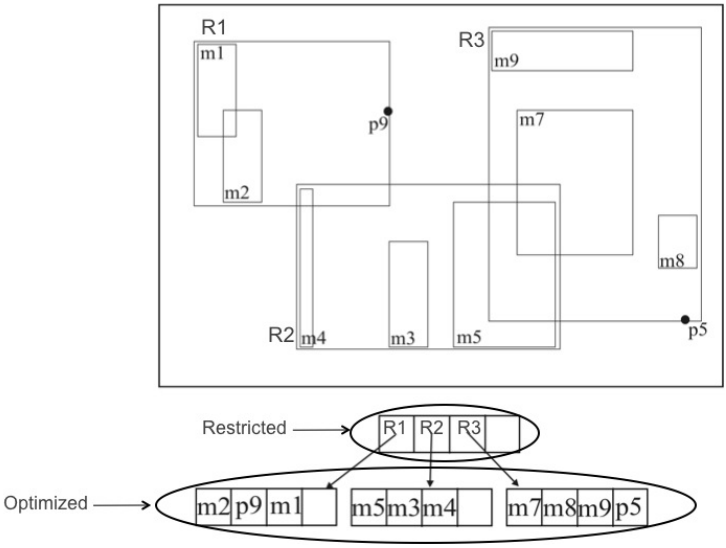


Fig. 2. Obtaining Spatial Projections from R-tree

that bound subsets of objects. In the example, we see that bounding box R1 contains the objects m1, m2 and p9. Similarly, R2 and R3 bound the objects in their respective leaf nodes. Therefore, the set of R1, R2 and R3 are selected as the set of MBRs that represent all objects.

Then, this set of MBRs from each site in P is transmitted to a site in Q in the following manner:

- The MBR set from the relation with the smallest spatial cardinality in P is sent to the relation with the smallest spatial attribute in Q,
- The MBR set from the relation with the next smallest spatial cardinality in P is sent to the relation with the next smallest spatial attribute in Q,
- and so on... until,
- The MBR set from the relation with the largest spatial cardinality in P is sent to the relation with the largest spatial attribute in Q.

For the third step, for each relation in Q, all MBRs from the transmitted set are intersected with the MBRs from the local set to create the set of intersection regions. Readers can refer back to Figure 1 for an example intersection region.

In the fourth step, the spatial objects from the relation in Q are tested for overlap with the appropriate intersection region. At the same time, the intersection regions are sent back to the relation from P that sent the original MBR set. Then, for each relation in P, each object is tested for overlap with the appropriate intersection region.

Finally, in the fifth step, the tuples of all objects that overlapped with an intersection region are shipped to the query site for the final join.

Table 1. Two-site Restricted Query vs. Optimized Query

Site 1	Site 2	TT-O	TT-R	%Imp	TC-O	TC-N	%Imp
100	400	1876	183	90.24	40000	9650	75.87
100	600	2847	260	90.86	60000	11642	80.59
100	800	3868	317	91.80	80000	11007	86.24
100	1000	4619	316	93.15	100000	7689	92.31
200	400	3782	1460	61.39	80000	51250	35.93
200	600	5635	1526	72.91	120000	47829	60.14
200	800	7447	1163	84.30	160000	47179	70.51
200	1000	9526	1133	88.10	200000	40407	79.79

4 Evaluation

In this section, we present our experimental methodology and results of our performance evaluation of the Restricted strategy. Our evaluation compares the Restricted strategy against the Optimized strategy [12] to determine which achieves both the lower CPU cost and the lower the number of object comparisons. Although we do not directly compare the data transmission costs between the two strategies, the fact that the Restricted strategy is transmitting fewer representative MBRs instead of entire spatial attributes can lead to data transmission cost savings.

Our experimental framework consists of a simulated distributed spatial database. For the purpose of our experiments, we used six relations. Each relation contains 100, 200, 400, 600, 800, and 1000 tuples respectively. Each tuple consists of a spatial object (in the form of a square), and some non-spatial attributes. Also, each relation is indexed with an R-tree [5], which uses nodes that can hold 50 entries. Each relation will be its own 'site' in our simulated system.

Our evaluation criteria will be evaluated using the following. The CPU time will be tracked using the system clock for all semijoin operations. The number of comparisons made during all semijoin operations will be tracked up to (but not including) the final query site join.

4.1 Two-Site Restricted Query Test

The first set of comparisons of our Restricted strategy is for distributed queries involving two sites. Table 1 shows the pairs of relations that were evaluated, along with the total number of comparisons that were made (column TC-O for Optimized, column TC-R for Restricted), and the total CPU cost (column TT-O for Optimized, column TT-R for Restricted). We also record the percentage improvement in the Restricted strategy over the Optimized strategy for both cost factors.

Results show that a very significant improvement is achieved by the Restricted strategy in both the number of comparisons that are being performed and the CPU time of the distributed query. For both the CPU time and comparison count, the improvement is more significant when there exists a significant difference in the size of the spatial relations between the two sites. In addition, when

Table 2. Four-Site Restricted Query vs. Optimized Query

Site 1	Site 2	Site 3	Site 4	TT-O	TT-N	%Imp	TC-O	TC-N	%Imp
100	200	400	600	7497	1388	81.49	160000	57479	64.08
400	600	800	1000	43361	22942	47.09	920000	501520	45.49
100	200	800	1000	13117	1578	87.97	280000	51414	81.64

the smaller of the two sites contains only 100 spatial objects, the improvement in CPU time is always over 90%. The improvements in the number of comparisons that are being carried out are smaller than the CPU time improvements, but are still very significant nonetheless.

4.2 Four-Site Restricted Query Test

For our second set of tests, we compared the evaluation of the strategies for four-site queries. Table 2 shows the sites involved and the CPU times and total number of comparisons. Again, we find that the Restricted strategy outperforms the Optimized strategy in both CPU time and the number of comparisons. We also find in the situation where a significant size difference exists between the relations - again, 100, 200, 800, and 1000 tuples - the greatest improvement is achieved for both factors.

4.3 Six-Site Restricted Query Test

Finally, we performed one test that compares the Restricted strategy and the Optimized strategy when all six sites are involved. We found that the Restricted strategy performed 204,777 comparisons in 8.7 seconds, while the Optimized strategy required 620,000 comparisons and 24.7 seconds to perform them. Therefore, the restricted strategy required 66% fewer comparisons and 70% less time than the original Optimized strategy.

5 Conclusion

In this paper, we propose a new strategy for performing distributed spatial query processing that involves more than two sites. The Restricted strategy improves upon the existing Optimized strategy [12] by applying a technique for minimizing the number of object comparisons that must be carried out, which in turn reduces both CPU and data transmission costs. A performance evaluation studied the improvement of the Restricted strategy over the Optimized strategy with respect to CPU and comparison costs for two-, four- and six-site queries. Results show that the Restricted strategy outperforms the Optimized strategy with respect to the number of comparisons and CPU cost in all cases. In particular, when there is a significant size difference between the participating relations, the improvements CPU costs and the number of comparisons are the most significant.

Some of our directions of future research include the following. The first is investigating the performance (with respect to the number of comparisons) of the Restricted strategy with different numbers of MBRs in the MBR sets. It is probable that sending fewer representative MBRs may result in more savings in CPU and data transmission costs. The second is deriving more strategies for distributed spatial query processing that work for an arbitrary number of sites. In particular, we want to study the use of Bloom filters for reducing the data transmission, CPU and I/O costs of a distributed spatial query. This work will ultimately lead to the best future strategies for processing and optimizing distributed spatial queries.

References

1. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Communications of the ACM 13, 422–426 (1970)
2. Abel, D.J., Ooi, B.C., Tan, K.-L., Power, R., Yu, J.X.: Spatial Join Strategies in Distributed Spatial DBMS. In: Egenhofer, M., Herring, J.R. (eds.) SSD 1995. LNCS, vol. 951, pp. 348–367. Springer, Heidelberg (1995)
3. Apers, P.M.G., Hevner, A.R., Yao, S.B.: Optimization algorithms for distributed queries. IEEE Transactions on Software Engineering 9, 57–68 (1983)
4. Brinkhoff, T., Kriegel, H.-P., Seeger, B.: Efficient processing of spatial joins using R-trees. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, New York, USA, pp. 237–246 (1993)
5. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data, pp. 47–57 (1984)
6. Hua, Y., Xiao, B., Wang, J.: BR-Tree: a scalable prototype for supporting multiple queries of multidimensional data. IEEE Transactions on Computers 58, 1585–1597 (2009)
7. Jacox, E., Samet, H.: Spatial join techniques. ACM Transactions on Database Systems 34, 1–44 (2007)
8. Kang, M.-S., Ko, S.-K., Koh, K., Choy, Y.-C.: A Parallel Spatial Join Algorithm for Distributed Spatial Databases. In: Andreasen, T., Motro, A., Christiansen, H., Larsen, H.L. (eds.) FQAS 2002. LNCS (LNAI), vol. 2522, pp. 212–225. Springer, Heidelberg (2002)
9. Karam, O.: Optimizing Distributed Spatial Joins using R-trees. Ph.D. Thesis, Tulane University (2001)
10. Karam, O., Petry, F.: Optimizing distributed spatial joins using R-trees. In: Proceedings of the 43rd ACM Southeast Regional Conference, pp. 222–226 (2006)
11. Özsu, M.T., Valduriez, P.: Principles of Distributed Database Systems. Springer, New York (2011)
12. Osborn, W., Zaamout, S.: Multiple-Site Distributed Spatial Query Optimization using Spatial Semijoins. In: Proceedings of the 10th International Baltic Conference on Databases and Information Systems, Vilnius, Lithuania (2012)
13. Shekhar, S., Chawla, S.: Spatial Databases: A Tour. Prentice Hall, New Jersey (2003)
14. Tan, K.-L., Ooi, B.C., Abel, D.J.: Exploiting spatial indexes for semijoin-based join processing in distributed spatial databases. IEEE Transactions on Knowledge and Data Engineering 12, 920–937 (2000)

The Impact of Spatial Resolution and Representation on Human Mobility Predictability

Weicheng Qian[1], Kevin G. Stanley[1], and Nathaniel D. Osgood[1,2]

[1] Department of Computer Science, University of Saskatchewan, Saskatoon, Canada
[2] Department of Community Health and Epidemiology, University of Saskatchewan, Saskatoon, Canada
{weicheng.qian,kevin.stanley,nathaniel.osgood}@usask.ca

Abstract. Western society is distinguished by its mobility. At no time in our history have we enjoyed the capacity to travel as rapidly, conveniently and safely. On the surface, this might suggest that human mobility patterns are highly irregular and impossible to predict. Drawing on a detailed multisensory positioning data set, we replicate earlier cell tower based predictability analyses with granular spatial and temporal multisensory data, and demonstrate a spatial resolution dependence of entropy, while reinforcing the claims of inherent predictability of human mobility advanced in early works. We demonstrate that mobility entropies reported with GPS data remain essentially unchanged with pruning of noisy GPS signals, lending additional credence to our methodology. We further compare cell tower results to those from WiFi-based localization for exactly the same time periods and participants, and demonstrate that the finer spatial resolution of WiFi also results in reported entropy exceeding that from cell tower traces, indicating that the resolution dependence observed for GPS data is not entirely due to GPS noise or discretization representation. This work represents a significant step towards fundamental understanding of human mobility patterns, which serve as key mediators to policy design in fields as diverse as public health, urban planning, and delay tolerant networks.

Keywords: Measurement, Experimentation, Human Factors.

1 Introduction

Understanding the fundamental properties of Western mobility is a core problem in several disciplines. Flight and bus schedules, traffic density, source-destination and census data can each provide large scale approximations of human mobility but miss the finer scale of pedestrian traffic and cannot easily trace individual agents. Patterns of human mobility are echoes of the shapes of individual lives, and while characterized by some regularity imposed by social, economic and physical constraints, also exhibit sufficient unpredictability to make reliable prediction of a person's location or trajectory challenging. Given that these patterns of mobility underlie the use of space and its associated physical, social and information infrastructure, identifying limits on the predictability of human mobility could have far ranging impacts.

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 25–40, 2013.
© Springer-Verlag Berlin Heidelberg 2013

Several authors have attempted to use portable electronic or communications devices to record or derive human mobility patterns. Song et al. [24] used cellphone records to determine the location of heavy mobile phone users and attempted to track users by cell tower, and then used entropy measures to infer the underlying predictability of human mobility patterns [23]. Yuan et al. used similar call logs to attempt to discern the "rhythm of a city" in China [25]. Based on a series of single day GPS traces, the SLAW [16] and SWIM [14] algorithms advanced the study of human mobility by proposing a robust algorithm to generate simulated mobility patterns grounded in empirical data. This work was extended by Hossmann et al. to better replicate human contact patterns in the synthetic agents [8]. Jensen et al. [11] examined the predictability with relative cellular, WiFi and Bluetooth records, but did not possess the spatial ground truth to relate their findings to spatial resolution.

Entropy, or the smallest amount of information required to completely characterize the system, was employed by Song et al. to determine the predictability of motion of individuals around a major urban center. Their analysis demonstrated a modest entropy which was independent of the typical range of a user's trajectory. However, it is unclear whether this entropy was due to their population, the method of position collection, or a more fundamental property of mobility in Western society. In this paper, we analyze a smaller but high-resolution data set [7] to determine the impact of spatial data encoding, resolution, and filtering on the mobility entropy of the study population. The data set includes GPS traces from 38 participants over 4 weeks cross referenced with both WiFi and Bluetooth proximity measures. We demonstrate that the design of the experiment and method of data collection can impact aggregate entropy calculations. We further distinguish between the magnitude and nature of the effects on entropy calculations and discuss the relevance of these findings for the design of future human mobility studies. We report several important findings.

- We confirm with new data existing findings [24] that human predictability has a high upper bound.
- We demonstrate that mobility entropy is sensitive to the scale of the discretization cell, with entropy increasing as the bin size decreases.
- We find that entropy is not sensitive to reasonable discretization cell shape, because Voronoi cells created using a spatial decomposition based on beacon (cell tower or WiFi router) locations produce similar entropy values to GPS grid-based decompositions of similar scale.

2 Related Work

Cell tower call and contact records form an attractive data source for studying human mobility because they are collected automatically, and cover thousands or millions of subscribers over prolonged sampling periods. For this reason, several notable contributions [24,10,6,25] have used traces that reflect connectivity to existing infrastructure (e.g., Access Points (APs) or cell towers) as their primary data source.

However, the tower locations map to a Voronoi diagram with a spatial extent dependent on geography and population density, and position updates for call records are only available when a user initiates or receives a call or a text message [24,25].

The global positioning system (GPS) can provide near-continuous data with an accuracy resolution down to a few meters. However when users move indoors, GPS suffers a significant reduction in accuracy due to fading and multipath effects, up to the point of complete loss of signal within large facilities. Studies have employed GPS data sets with a temporal resolution of hours [24], down to minutes [7,16,1]; study durations from a day [16], to over a month [7,1]; and study population from dozens [7], to hundreds [16,1,24]. To compensate for the poor resolution of cell tower based measures and the reduced coverage of GPS data, researchers have turned to shorter-distance radio-based devices such as WiFi [15], or RFID [19,12,20] systems.

3 Experimental Setup

Our experiment and analysis involves the post-processing of a previously collected multi-sensor dataset of human mobility and contact patterns, primarily collected for public health modeling [7], using techniques drawn from and extending those reported in [24].

3.1 SHED1 Data Collection

Central to our analysis is the Saskatchewan Health Ethology Dataset 1 (SHED1) collected from 39 participants including graduate students and departmental staff over 4 weeks in April and May of 2011 [7]. This study was approved by our institutional ethic review board and data was collected with the knowledge and consent of the participants. This dataset contains traces recorded from HTC Magic Android smartphones and includes direct (through GPS) and indirect (through WiFi and Bluetooth traces) information on participant geographic location. This data was crossed with recorded estimated positions of WiFi routers and cell towers in the city of Saskatoon [17]. One participant's GPS unit experienced a hardware failure, leaving 38 participants with useful data.

The phone was programmed to collect data in bursts with a 5-minute duty cycle to manage data size and battery life. The phone started to log at the beginning of each duty cycle, with GPS records lasting for 2 minutes, accelerometer records for 1 minute, Bluetooth contacts for 1 minute, WiFi contacts for 3 seconds and battery state for 10 records. Additional details on the dataset may be found in [7].

3.2 SHED1 Mobility Analysis

Three different data streams from the SHED1 dataset were employed to analyze human mobility patterns and entropy. GPS was used for the base mobility estimate. Bluetooth signal strength, which can be used to estimate device proximity [11,5,7],

was used to enhance the number of location records. Finally, WiFi signal strength, which can be used to estimate the device proximity to fixed infrastructure [3,7], particularly in indoor spaces, was used as a comparator.

The selected GPS data in the SHED1 dataset was comprised of 1,277,520 records. Our GPS space-time logs are elements of location time series for each participant at each duty cycle over the whole 34-day experiment. For our experiment, a full collection should include 372,134 space-time logs. The 1.2 million records correspond to 126,815 space-time logs, as each duty cycle often contained several GPS records for each participant.

The number of GPS space-time logs can be increased by leveraging Bluetooth proximity data. When a device detects another device's Bluetooth beaconing signal with an RSSI of at least -80 dBm, then these two devices can be inferred to be proximate within approximately 8 meters. It is reasonable to assume that the two devices are in the same cell, given the cell sizes considered here. Applying this logic to pairs of phones in the dataset results in an additional 18,054 (14%) space-time logs.

WiFi signals have been used for both indoor and outdoor localization in both academia [2] and industry [22]. In its simplest form, WiFi localization can be regarded as a variant of cell-tower based analysis, where each WiFi router forms a smaller (< 500 meter radius) cell, generating a Voronoi diagram similar in structure, but with much higher resolution than the one described in [24]. Examining the entropy of the WiFi traces can therefore provide a counterpoint to the GPS grid and cell tower based analysis, disambiguating contributions of cell size and shape to the overall entropy. There were 19,080 unique routers spotted by 38 participants during 192,976 duty cycles for which we received WiFi records in the SHED1 dataset; however, participants were closest (minimum RSSI among multi-scans of routers) to only 4937 routers. We have constrained our analysis to the subset of routers that would appear on the Voronoi described above. By estimating router location as the GPS location of the phone which detected the strongest signal from the router over the course of the experiment, we can roughly localize 95% of the routers in our dataset.

3.3 Spatial Representation

When computing the Lempel-Ziv entropy for mobility data, it is necessary to discretize space to represent location history as a string. The form of this discretization depends on the data source.

Square Grid.
In the absence of any other information, a uniform sampling strategy is the most principled. However – in an effect similar to the binning problem [21] – the grid resolution can have a significant impact on the resulting entropy calculations. We divided the world into square grid cells with 16.625m, 31.25m, 62.5m, 125m, 250m, 500m, 1km, 2km and 4km sides, corresponding to a discretization range from a GPS

accuracy floor to a cell tower coverage area ceiling. To reduce computation, and to maintain consistency with other calculations, only those bins which contained at least one record were included as possible states in the Lempel-Ziv entropy calculation.

Cell Tower Voronoi.
For this study, no cell tower contact or call records were directly available, but we inferred them from the GPS data. Specifically, cell tower location data was extracted from [17], a publicly available website for browsing tower locations and frequencies. We included all cell towers that intersected our study area, without regard to the cellular provider that owned or operated the tower. This methodology provides an overly optimistic estimate of the resolution of cell tower-based localization because it considers a larger set of towers than would be available to any specific call-log dataset. Based on cell tower locations, cellular zones were created by computing the Voronoi diagram of the cell tower locations. GPS locations recorded in the SHED1 dataset were individually assigned to each cellular zone by finding the closest cellular tower based on Euclidean distance. Based on the definition of Lempel-Ziv entropy, only cell towers which were closest to at least one GPS record were included in the Voronoi diagram and therefore entropy calculation.

WiFi Voronoi.
WiFi nodes which publish their Medium Access Control (MAC) address can also be used as beacons. In a particular duty cycle, participants were assigned to the MAC address of the WiFi router to which they were closest, as determined by the minimum value (least negative) of the received signal strength indicator (RSSI). By marking participants' real-time location with routers using RSSI, we were able to determine the Lempel-Ziv entropy from the list of possible routers directly, providing a second, but higher resolution Voronoi representation of space.

3.4 Data Filtering

Exploiting the cross-linked multi-sensor data of SHED1, we used Bluetooth data as ground truth to investigate the fidelity of a GPS data-based inter-contact time computation approach. We found that in 15% of the duty cycles where at least one of a pair of participants could record the Bluetooth beacon of the other, the raw reported GPS displacement between them exceeded 250 meters.

To filter the raw GPS data, we used average speed as a limiting criteria. By setting a threshold that limited the maximum displacement to be 4 kilometers for a participant both within a duty cycle (that is, a 2 minutes interval that GPS data collecting functionality was on) and across a duty cycle (which is 5 minutes in duration), we filtered all GPS records. The filtering discarded 10% of the GPS records, and more than halved the number of those reporting detection of Bluetooth connections spanning more than 250 meters.

3.5 Trip Length

Trip length usually refers to the displacement of a participant's continuous trace with minimal stops. The distribution of trip lengths could be used to describe the characteristic of trajectories [4,13]. According to [4], the distribution of trip length tends to a truncated power-law distribution, reflecting the limited range of human mobility.

To calculate trip length given a representation of space, one must first define what is meant by a trip. Given that all of our representations are inherently rasterized, we chose to represent a trip as any sequence of records for an individual where the location changes between each record. The length of the trip was either summed over cell sizes for the grid representation, or calculated from transmitter to transmitter in the cell tower and WiFi cases. For example, if there were 5 adjacent cells, A, B, C, D, and E, then sequential records across duty cycles containing ABCD or ABD would correspond to a trip of length 4, while ABBC would correspond to two trips of length 2. Making a conservative assumption, missing position data in a duty cycle ended a trip, so the sequence AB?C corresponded to a single trip of length 2.

3.6 Entropy Calculation

To estimate entropy of each person's trajectory, we replicate the real entropy estimation calculation of [24], using Lempel-Ziv approximation, as specified with following equation:

$$S^{est} = \left(\frac{1}{l} \sum_{i}^{l/2} \Lambda_i \right)^{-1} \ln n \tag{1}$$

where Λ_i is the length of the shortest substring starting at position *i* which doesn't previously appear from position 1 to $i-1$, and n is the length of all the duration that we have records. Here, we execute the approximation only to $l = n/2$ primarily to avoid the risk of underestimating Λ_i when $i > n/2$. The rapid convergence of Lempel-Ziv to the actual entropy as n approaches infinity helps to reduce costs caused by noise that is generated by half the time series length.

We also replicated the upper bound of predictability calculation in [24]. $\Pi^{\max}(S, N)$ varied with both real entropy S and the number of distinct places N a person visited during the experiment period. $\Pi^{\max}$ was solved using:

$$\begin{aligned} S &= H\left(\Pi^{\max}\right) + \left(1 - \Pi^{\max}\right) \log_2 (N-1) \\ H\left(\Pi^{\max}\right) &= -\Pi^{\max} \log_2 \left(\Pi^{\max}\right) - \left(1 - \Pi^{\max}\right) \log_2 \left(1 - \Pi^{\max}\right) \end{aligned} \tag{2}$$

To parallelize the computation, we assigned 1 of the 36 nodes of the Socrates cluster (2x Quad core Intel Xeon L5420 at 2.5GHz, and 2GB memory) to calculate the entropy over all participants for a single spatial representation.

4 Results

4.1 SHED1 Data Characteristics

The SHED1 dataset is characterized by high-fidelity multivariate data on human interaction and mobility over a significant duration. It is also characterized by a degree of human error, particularly with respect to compliance. Only half the total possible data was collected, and only one third of the total possible data contained GPS traces, as much of the participants' time was spent in laboratories and offices at the university. Even with the lower than desired compliance, the dataset still represents a significant high-fidelity sampling of the participants' lives, and while the data may not permit minute by minute tracking of each participant over the course of the entire study, we posit that the aggregate distributions derived from this data closely approximate the overall life patterns of the participants, as contended by [24].

4.2 SHED1 Spatial Data Characteristics

Spatial Distribution.
To illustrate the impact of spatial representation on the collected dataset, we plotted heatmaps summed across the entire 34 day period for cell tower Voronoi, WiFi Voronoi, and 1km grid representations of space, as shown in Figure 1. The heatmaps visualize a count summed over participant count by duty cycle: every time any participant was in a location in a given duty cycle, the value of the location's count was incremented. The heatmap is displayed as a logarithmic grey scale of that count, with the extremes of white and black corresponding to zero and 100,000 counts, respectively.

Figure 1.a and b show both large and small-scale Voronoi diagrams representations of the probability of finding a participant at a given location. Within these two Voronoi diagrams, we marked cell towers as red points and Voronoi cell as blue lines. The larger region shown in Figure 1.a represents the extent of ground travel engaged in by the participants, covering an area of approximately 395,000 km^2. While trips further East and West were made, they were made by air, and contain limited path data, as participants had to turn off their smartphones on an aircraft. The smaller area in Figure 1.b represents central Saskatoon, covering an area of approximately 570km^2, where the 95% of records were concentrated. The Voronoi regions corresponding to cell towers are also shown in these figures to provide a sense of scale. Figure 1.c, e and g correspond to heatmaps of the cell tower Voronoi, 1 km grid and WiFi Voronoi for the large scale map, respectively. Figure 1.d, f, and h correspond to the heatmaps of the cell tower Voronoi, 1km grid and WiFi Voronoi over the university area of central Saskatoon shown in Figure 1.b, respectively. The WiFi map has an inset showing the area of the university, due to elevated WiFi node density in the region.

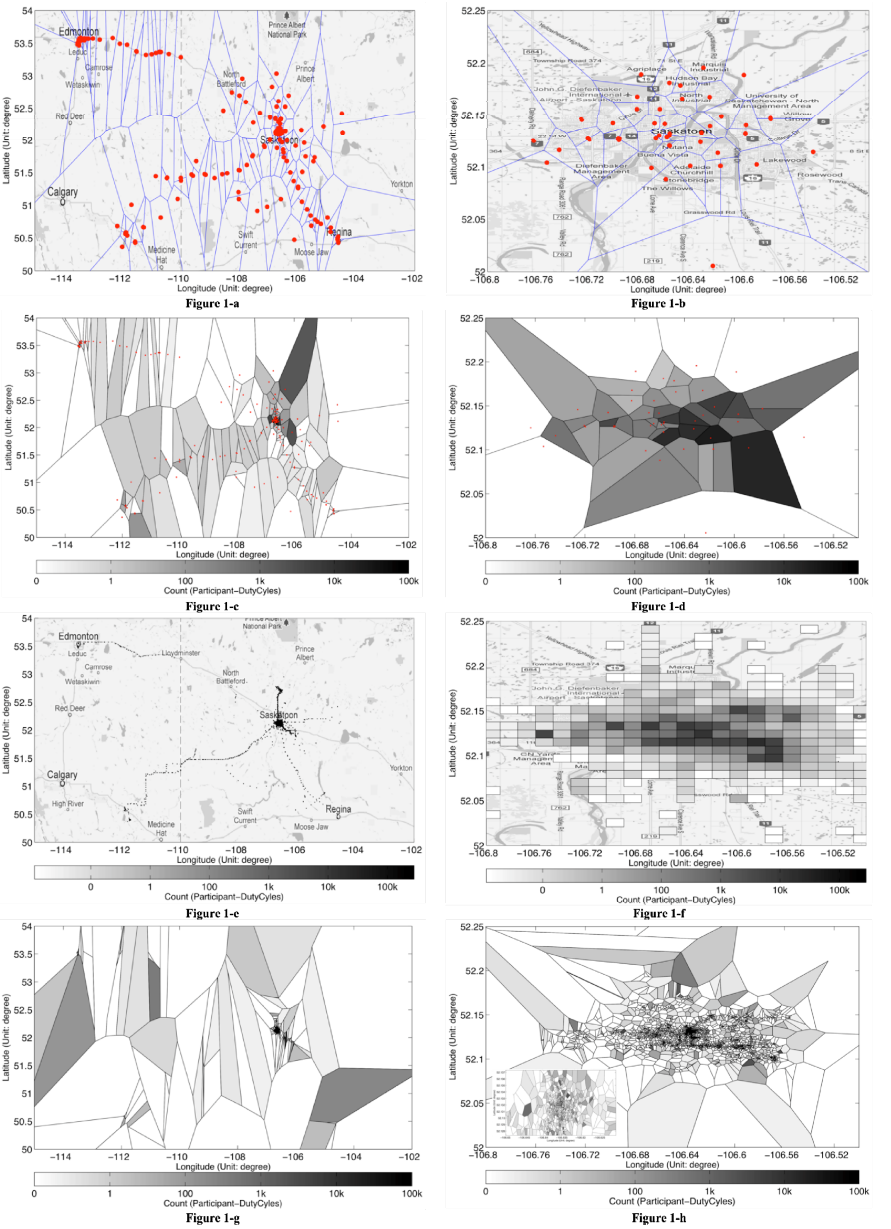


Fig. 1. a) Map showing trips taken by participants and cell tower Voronoi diagram; b) University area of central Saskatoon map showing observed cell tower Voronoi diagram; c) Heatmap of cell tower cell occupancy over area in (a); d) Heatmap of cell tower cell occupancy over area in (b); e) Heatmap of 1 km grid cell occupancy over area in (a); f) Heatmap of 1 km grid cells over area in (b); g) Heatmap of observed WiFi cell occupancy over area in (a); h) Heatmap of observed WiFi cell occupancy over area in (b) with inset of university

Unsurprisingly, the cell towers provide the coarsest representation of space, with the entire university area of central Saskatoon covered by 46 nodes. In all three representations, occupancy is concentrated around the university, with moderately high levels at the university itself and high levels within the surrounding regions where students tend to live.

The large scale map demonstrates some of the shortcomings of cellular and WiFi representations of space. While both provide adequate coverage of the city area, there are an insufficient number of WiFi nodes or towers to capture the long distance travel through rural areas. The cell tower Voronoi generally captures the long distance trips to Edmonton and Regina, but smears the contacts over a large rural area, creating significant uncertainty as to exactly where participants were, while the grid representation shows them following the highway.

To quantify the spatial extent, we calculated the areas of the cell towers and WiFi Voronoi cells for the cell tower and WiFi cases for from the polygon convex hull for the greater Saskatoon area region that contains 97% of our location records. As shown in Figure 1-b, this region extended 1.3 degree (roughly 87km) on longitude and 0.7 degree (roughly 57km) on latitude around the university area of central Saskatoon as shown in Figure 1b, this area contains 97% of our location records.

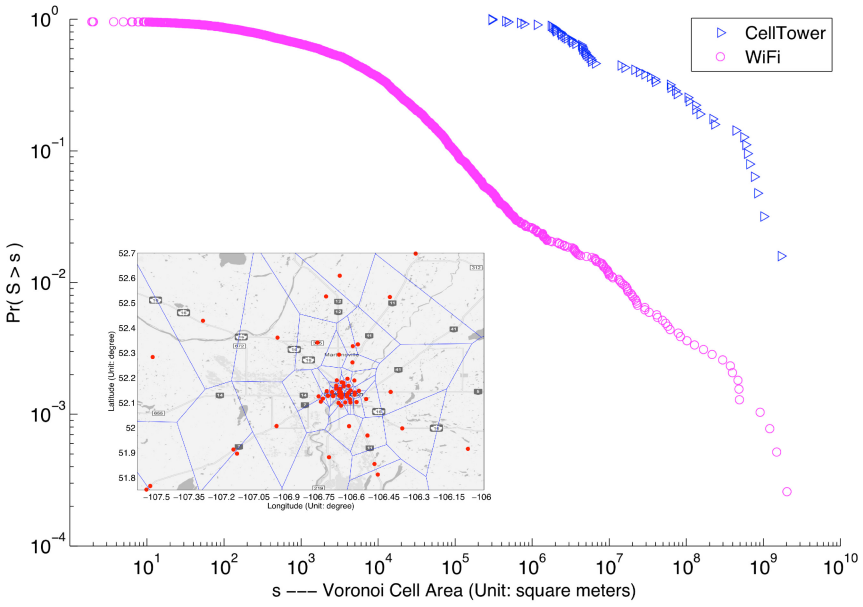


Fig. 2. CCDF of Voronoi Convex Hull Area

To avoid overestimating cell size, the analysis excluded cells that intersected the margin of the area shown in the inset. The distribution of cell sizes resembles an exponential rather than a power-law distribution. As shown in the complementary cumulative distribution functions (CCDFs) in Figure 2, for the cell tower Voronoi

map, over 90% of cells are larger than 1 km^2. The WiFi Voronoi map is more fine-grained; only 2.5% of cells are larger than 1 km^2. Nearly 5% of the cells of the WiFi Voronoi map are smaller than 10 m^2, primarily due to the impact of projecting multiple WiFi nodes from different floors onto a 2D plane, particularly at in the WiFi rich dense university and downtown areas. Figure 2 also reflect the density of landmarks such as cell towers and Wi-Fi becaons, where landmark-dense regions, are characterized by small Voronoi cells.

Trip Length.

The impact of spatial representation on trip length can provide some insight into the potential impacts of resolution on entropy. If trip length distributions tend to zero beyond a minimum, then little change in entropy would be expected for increasing resolution beyond that point. However, if the minimum trip length corresponds to the resolution of the spatial representation, we cannot assume that entropy will saturate with resolution. Trip length complementary cumulative distributions (CCDFs) for 15.625 m, 62.5 m, 250 m and cell tower grids are shown in Figure 3.

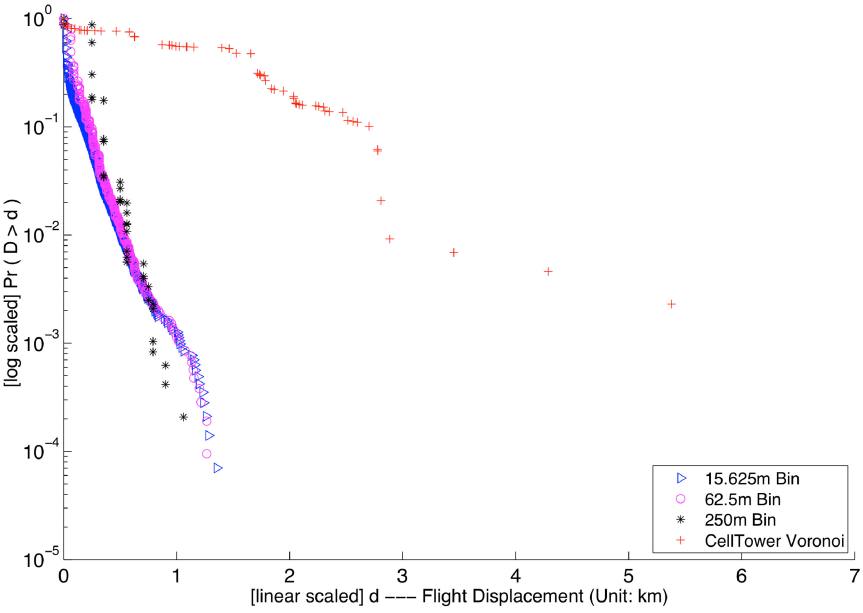


Fig. 3. CCDF of Flight Displacement

Most of the curves are characterized by an exponential relationship over the 0.1 to 1 km interval, with the exception of the cell tower trip length which is characterized by a much steeper decay in the vicinity of 1km. This is clearly a spatial sampling effect, as the underlying data is the same in all cases. Because of the size of the cell towers, fewer trips are recorded, because state changes on the scale of a cell tower Voronoi region in sequential duty cycles are less common.

The saturation effect evident in all cases at around 2 km is likely an artifact of our sample, composed primarily of graduate students with a dependency on public transport, and the modest size of the city, which can easily be traversed by bus in less than an hour. Our strong trip definition – which split trips on either a repeated cell or empty duty cycle also likely split some of the longer trips evidenced in Figure 1.c, e and g into smaller trips.

The inflection in the two-piece curve evident in the 15.625 m grid case inflecting at approximately 75% cumulative probability is likely due to two effects – an increase in the probability of shorter trips as noted in [16], and noise contributed from the accompanying GPS sensor readings, providing the illusion of many short trips. In the 15.625m and 62.5m and 250m grid resolutions, almost 90% of the probability exists between the minimum grid size and 300 m.

4.3 Entropy Analysis

If spatial resolution and representation have little impact on entropy, then we can conclude that Song et al. were correct in characterizing Western mobility as inherently predictable, and we can employ much coarser probes – such as cell tower records – to investigate mobility patterns for planning at all scales. If, however, [16] is correct, then there could be a pronounced resolution effect on entropy. We calculated the entropy for all representations described here. The results are presented in Figure 4 as boxplots, where each distribution is computed across participants.

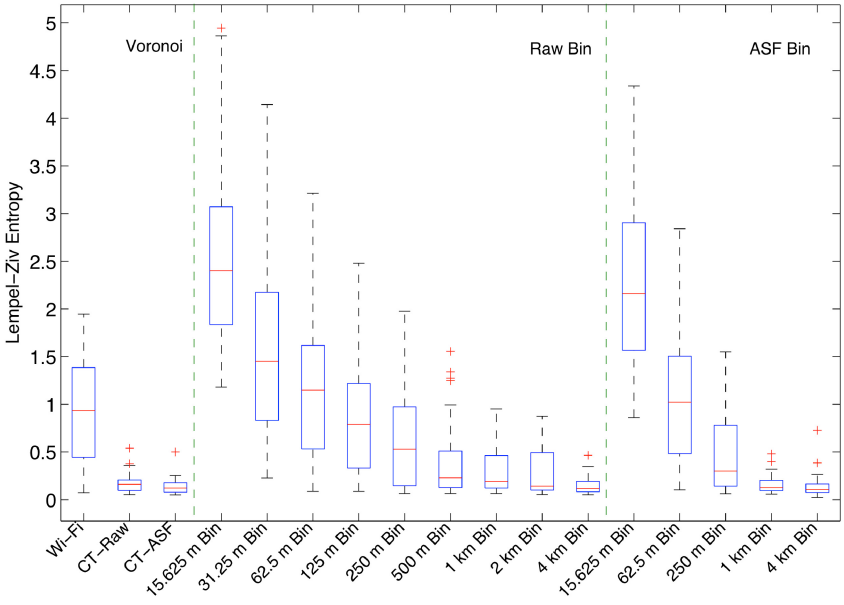


Fig. 4. Boxplot of Lempel-Ziv Entropy, Raw Bin indicating raw location data was used for Square Discretization and similarly ASF Bin indicating Average Speed Filtered data was used.

In the boxplot in Figure 4, boxes represent the 25th and 75th percentiles of participant mobility entropy and the maximum whisker corresponds to approximately 99.3% coverage. The line in the center of the box represents the mean. It is readily apparent from Figure 4 that while the pruning for the average speed filter removed a significant amount of data, it did little to change the entropy distribution at any resolution. This effect gives us confidence that the measurements represent a viable sample of participant mobility entropies, as selectively removing those points least likely to be correct had a limited impact.

As we expected, there were significant resolution impacts, with larger bins – either Voronoi or grid – producing smaller entropies. There appeared to be little sensitivity to the geometry of the representation, as cell tower entropy closely resembles the 4 km bin size – approximately the average scale of the Voronoi cell size considered in the Greater Saskatoon area in which most of the records fell, and the WiFi Voronoi approximates the 250 m grid, which is a typical nominal range for WiFi routers.

The large variance in the entropies for the smallest grids (15.625 m and 62.5 m, respectively) are likely indicative of both greater mobility at small scales and the increasing impact of sensor noise on the distribution. At any given duty cycle, the signal could then wander randomly about its neighbors, giving rise to an artificially higher entropy.

There is a pattern apparent in the relationship between the entropy and bin size in Figure 4. In fact, the relationship between bin area and entropy seems to follow an inverse relationship, in that increasing the bin size decreases the entropy, but by a smaller factor.

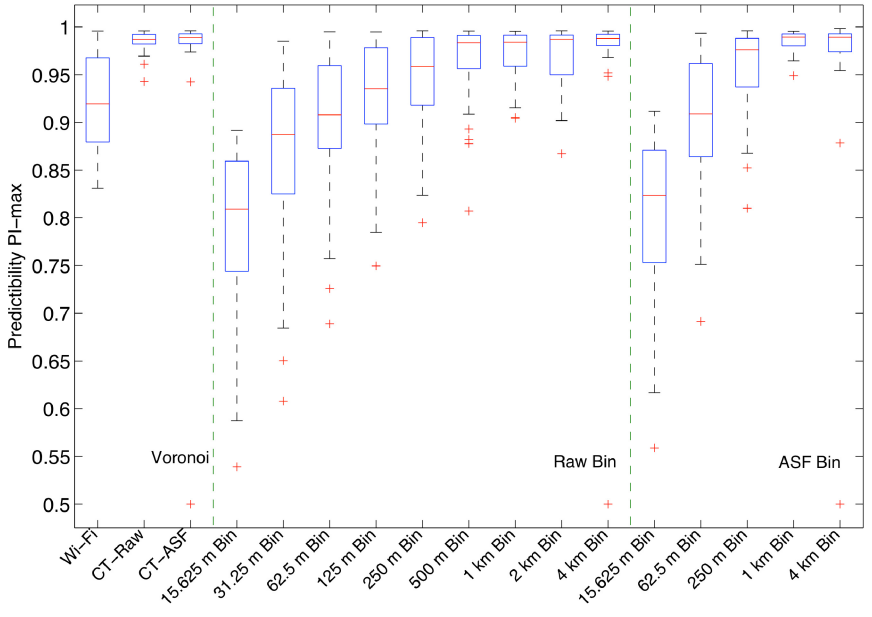


Fig. 5. Boxplot of Predictability

To facilitate comparison with other work [24], we also plotted the upper bound of predictability (Π^{max}), which denotes the upper bound for the predictability of an event based on the entropy. A box plot of predictability is shown in Figure 5.

The plot shows the same trend as the entropy, and is largely consistent with the values reported in [24]. Human mobility in the limit is typically very predictable. While individual trajectories may be difficult to interpret, in the limit, Western mobility patterns seem regular.

5 Discussion

5.1 Contributions

We can conclude that as we look at human motion in greater detail, we find that more information is required to capture that mobility. There is no entropy saturation point observable in our data – suggesting, from a purely analytic perspective, there is no scale examined here at which no additional data is required to represent motion as scale decreases. However, we have noted that the entropy grows much slower than resolution, indicating that, for our data at least, there is a declining return for incremental investment in improving tracking resolution, as the additional variability captured will grow more slowly than the sensing effort employed.

We noted no interaction between scale size and spatial representation. Voronoi regions generated similar entropies as rectangular bins of similar scale. This finding is important, as it allows experimental designers to choose the most appropriate spatial partition opportunistically based on available infrastructure.

We have also exploited data fusion between sensors during our experiment, to estimate the reliability of GPS coordinates using Bluetooth records; to determine the location of WiFi router by find joint GPS location records with Wi-Fi connection records; and to complete missing data by copying GPS location data between Bluetooth sensor spotted pairs.

5.2 Recommendations

Our most fundamental finding – that entropy scales in a regular fashion with resolution (but at slower rate) – provides guidance to intelligent system design based on the detection of human patterns, such as those described in [9,18]. Our analysis suggests that the complexity of representing mobility should scale slower than the resolution, implying that the memory and computational requirements will also scale in a relatively favorable manner, and designers should not be inhibited from employing higher-fidelity data to meet their system's needs.

We were able to conclude that there was little difference in predictability due to spatial representation, and that the dominant parameter was the size of a cell, not the shape of a cell, at least between the two most common representations: grid and Voronoi. Given the relative costs in terms of hardware and battery life, and the extremely dense WiFi networks common in even moderately sized cities, we believe

future researchers should preferentially leverage WiFi Voronoi representations for urban mobility data, supplementing with GPS only when in rural areas lacking adequate WiFi coverage.

5.3 Limitations

While our research has provided significant insight into the role of spatial resolution and representation on the apparent entropy of human mobility, it suffers from distinct shortcomings related to the structure of the underlying SHED1 dataset. First, the dataset is predominantly comprised of graduate students, who have unstructured lives and may exhibit a bias towards a higher mobility entropy than average citizens. However, even with this highly flexible population, we found a lower entropy than Song et al. [24], whose reliance on highly-mobile participants was built into their sampling protocol. Secondly, our sample size – while comprising a significant proportion of graduate students in the Department of Computer Science – is insufficient to provide broadly generalizable empirical conclusions. However, we can conclude that for the population under study, these findings are reasonable, providing strong motivation for studies of larger and broader populations using similar techniques. The third shortcoming of the dataset relates to the compliance of the participants. We captured slightly more than half the available data that would have been obtained with perfect compliance, and only one third of the total possible data had viable GPS measurements. However, we still obtained more than one million GPS records at multi-minute resolution, corresponding to a significant sample of the participants' lives.

Our use of GPS proximity to cell towers as a measure of cell tower based localization – rather than call records from cell towers themselves, as reported in [24] – is also a limitation. However, Song et al. were forced to use call records as their baseline because they did not have access to actual participant positions. In a sense, our representations is an idealized case of what Song et al. were hoping to achieve, and represents an upper bound on the fidelity of data that could be obtained from outdoor call records. However, calls placed from inside a building would register with the cell tower, whereas GPS might not, meaning that our records would be biased towards outdoor motion. However, this is also true of Song's data as few buildings span multiple cell tower Voronoi regoins. The resolution and scope of the WiFi records provides us with additional confidence that our methods were sufficient to capture the overall trends in mobility patterns.

6 Summary and Future Work

In this paper, we used a high-fidelity data set to examine the limits of human mobility predictability and have demonstrated how the entropy of human mobility scales with measurement resolution. While higher resolution mobility patterns are less predictable than coarser ones, the increase in entropy scales more slowly than the number of possible states. The predictability also appears to depend on the size – rather than the

geometry – of the cell. We suspect that this resolution dependence is based on the preponderance of short trips evidenced in our path-length analysis. These results are important because they provide a sound theoretical and empirical basis for future work in the measurement, leveraging or analysis of human mobility patterns. In the future, we will extend this analysis to larger and longer datasets, and employ additional orthogonal sensors to better condition the position data.

References

1. Aharony, N., Pan, W., Ip, C., Khayal, I., Pentland, A.: Social fMRI: Investigating and shaping social mechanisms in the real world. Pervasive and Mobile Computing 7(6), 643–659 (2011)
2. Amundson, I., Koutsoukos, X.D.: A Survey on Localization for Mobile Wireless Sensor Networks. In: Fuller, R., Koutsoukos, X.D. (eds.) MELT 2009. LNCS, vol. 5801, pp. 235–254. Springer, Heidelberg (2009)
3. Bell, S., Jung, W.: Wifi-based enhanced positioning systems: Accuracy through Mapping, Calibration, and Classification. In: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness, pp. 3–9 (November 2010)
4. Brockmann, D., Hufnagel, L., Geisel, T.: The scaling laws of human travel. Nature 439(7075), 462–465 (2006)
5. Eagle, N., Pentland, A.S., Lazer, D.: Inferring friendship network structure by using mobile phone data. Proceedings of the National Academy of Sciences 106(36), 15274–15278 (2009)
6. González, M.C., Hidalgo, C.A., Barabási, A.-L.: Understanding individual human mobility patterns. Nature 453(7196), 779–782 (2008)
7. Hashemian, M., Knowles, D., Calver, J., Qian, W., Bullock, M.C., Bell, S., Mandryk, R.L., Osgood, N.D., Stanley, K.G.: iEpi: an end to end solution for collecting, conditioning and utilizing epidemiologically relevant data. In: Proceedings of the 2nd ACM International Workshop on Pervasive Wireless Healthcare (MobileHealth 2012), pp. 3–8. ACM (2012)
8. Hashemian, M.S., Stanley, K.G., Knowles, D.L., Calver, J., Osgood, N.D.: Human network data collection in the wild: the epidemiological utility of micro-contact and location data. In: Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium (IHI 2012), pp. 255–264. ACM (2012)
9. Hui, P., Crowcroft, J., Yoneki, E.: Bubble rap: social-based forwarding in delay tolerant networks. In: Proceedings of the 9th ACM International Symposium on Mobile ad Hoc Networking and Computing (MobiHoc 2008), pp. 241–250. ACM (2008)
10. Isaacman, S., Becker, R.: Human mobility modeling at metropolitan scales. In: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys 2012), pp. 239–252. ACM (2012)
11. Jensen, B., Larsen, J., Hansen, L.: Predictability of Mobile Phone Associations. In: Inter. Workshop on Mining Ubiquitous and Social Environments (2010)
12. Jiang, X., Liu, Y., Wang, X.: Proceedings of the 2009 WASE International Conference on Information Engineering (ICIE 2009), vol. 1, pp. 169–172. IEEE Computer Society (2009)
13. Karagiannis, T., Le Boudec, J.Y., Vojnovic, M.: Power law and exponential decay of intercontact times between mobile devices. IEEE Transactions on Mobile Computing 9(10), 1377–1390 (2010)

14. Kosta, S., Mei, A.: Small world in motion (SWIM): Modeling communities in ad-hoc mobile networking. In: Proceedings of The 7th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2010), Boston, MA, U.S.A., pp. 2106–2113 (June 2010)
15. Kotz, D., Essien, K.: Analysis of a campus-wide wireless network. Wireless Networks 11(1-2), 115–133 (2005)
16. Lee, K., Hong, S., Kim, S., Rhee, I.: SLAW: A new mobility model for human walks. In: IEEE International Conference on Computer Communications (INFOCOM 2009), pp. 855–863. IEEE (2009)
17. Loxcel. Canadian Cell Tower Map, http://www.loxcel.com/celltower
18. May, R.M.: Network structure and the biology of populations. Trends in Ecology & Evolution 21(7), 394–399 (2006)
19. Ni, L.M., Liu, Y., Lau, Y.C., Patil, A.P.: LANDMARC: indoor location sensing using active RFID. Wireless Networks 10(6), 701–710 (2004)
20. Saab, S., Nakad, S.: A standalone RFID indoor positioning system using passive tags. IEEE Transactions on Industrial Electronics 58, 1961–1970 (2010)
21. Scott, D.W.: On optimal and data-based histograms. Biometrika 66(3), 605–610 (1979)
22. Skyhook. Skyhook, http://www.skyhookwireless.com/
23. Song, C., Koren, T., Wang, P., Barabási, A.-L.: Modelling the scaling properties of human mobility. Nature Physics 6(10), 818–823 (2010)
24. Song, C., Qu, Z., Blumm, N., Barabási, A.-L.: Limits of predictability in human mobility. Science 327(5968), 1018–1021 (2010)
25. Yuan, Y., Raubal, M.: Extracting Dynamic Urban Mobility Patterns from Mobile Phone Data. In: Xiao, N., Kwan, M.-P., Goodchild, M.F., Shekhar, S. (eds.) GIScience 2012. LNCS, vol. 7478, pp. 354–367. Springer, Heidelberg (2012)

A Data-Driven Approach for Convergence Prediction on Road Network*

Qiulei Guo[1,3], Jun Luo[1,2], Guiqing Li[3], Xin Wang[4], and Nikolas Geroliminis[5]

[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
{ql.guo,jun.luo}@siat.ac.cn
[2] Shenzhen Key Laboratory of High Performance Data Mining, Shenzhen 518055, China
[3] South China University of Technology, Guangzhou, China
ligq@scut.edu.cn
[4] Department of Geomatics Engineering, University of Calgary, Canada
xcwang@ucalgary.ca
[5] Urban Transport Systems Laboratory (LUTS), Ecole Polytechnique Fédérale de Lausanne (EPFL)
nikolas.geroliminis@epfl.ch

Abstract. With the rapid development of location sensing technology such as GPS, huge amount of location data through GPS are produced every day. The flood of taxi GPS data make it possible to predict the plentitude of traffic events on road network. In this paper, we propose a data-driven approach for traffic state convergence prediction on road network. We introduce a new method predicting the future location of taxis on road network. Furthermore we propose a statistical model to predict real time convergence on road network. We experimentally demonstrated that our approach achieves high prediction precision on the real world massive taxi GPS data.

Keywords: Convergence Prediction, Taxi Trajectory, Data Mining.

1 Introduction

Road network traffic convergence usually represents some interesting things happening in the city. For example, the traffic jams caused by too many cars' arriving at the same road from different living districts to some working places in the rush hour, and also some interesting major events such as people's gathering for football games, or movie debuts. Convergence prediction on road network is very useful not only for drivers but also for traffic management operation centers since they can make plans before unexpected events happen. Here we define cars

* This research has been partially funded by the International Science & Technology Cooperation Program of China (2010DFA92720) and NSF of China under poject 11271351.

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 41–53, 2013.
© Springer-Verlag Berlin Heidelberg 2013

convergence as for each road segment, there will be certain amount of cars meeting on that road segment at the same time interval. Unlike traditional traffic flow prediction, which needs to consider cars that currently are not moving in the road network but will appear in the future, cars convergence mainly answers the question of how many cars that are currently moving in the road network will meet at road X in the future? Traffic flow prediction is an challenging problem which has been studied for quite a long time. Most of the approaches try to predict the traffic flow of one street based on historic data [11]. But they can not accurately predict those unexpected (random) events or convergences, which have been seldom studied. One big challenge is that sometimes it is very difficult to know when and where those events happen since they happen randomly and we can not tell how many people are on their way to it. Take the sale promotional activities in some big shopping malls as an example. Maybe there are many shopping malls in a city and sale promotional activities happen from time to time. It is difficult or even impossible to collect the news about these activities, let alone to know how many people want to go and when will they go, which would result in accidental convergences (possible traffic jams). If we can predict that too many cars on the move would all arrive at some streets (near some shopping mall) at the same time, which would end up as a convergence and even traffic jam, we can suggest some drivers to avoid those streets.

Convergence as an interesting pattern has been studied in [12]. However authors assume the objects move in free space and convergence area is defined as the disk which intersects with certain number of motion azimuth vectors. To predict the convergence time, they assume each object moves in constant speed. With the constraint of road network, we can not use the motion azimuth vectors to predict the convergence area since a car's motion azimuth is fixed by the road and it could change abruptly when a car travels from one road to the other. Furthermore, a car does not move in constant speed because of the traffic light, congestion, blocking from other cars *etc.*

To predict the convergence on road network, we need to solve the following two problems:

1. Given the trajectory of a car which is at road segment r at time t, where will it be at time $t + \delta t$?
2. Given n cars' trajectories at time t, how many of them will be at road segment r at time $t + \delta t$?

The first problem is a challenging problem since the traffic network is a complicated system. Where the car would arrive at time $t + \delta t$ is not only decided by the car's current location at time t but is also affected by other cars (for example, traffic jams) and traffic lights. However there are still some patterns we can mine out and use to predict the future car's location. Buchin *et al.* develop an efficient algorithm to find the commuting pattern of a person based on his daily driving car's trajectories [5]. In [9], Froehlich and Krumm also show that a large portion of a typical driver's trips are repeated and they use those repeat patterns to predict driver's future route. In [13] Oswald *et al.* also show that traffic is a cyclical system. Therefore, if we can find a similar traffic situation as our current

situation, we can predict the future situation of current traffic based on similar historic traffic. Unfortunately, there are no commuting (or repeat) patterns for individual taxi. But if the origin and destination are fixed, taxi drivers usually take similar paths (shortest, fastest or most economic path). In [16], Ge *et al.* show that the normal taxi drivers always take similar routes (maybe more than one but not many) for fixed origin and destination. Based on this observation, we can use historic trajectories with similar origin and similar route as current trajectory to predict the future trace of current trajectory.

It seems that if we solve the first problem, the second problem can be solved easily. However, since our method is based on huge amount of historic trajectories, for a given trajectory, it could match with many subtrajectories of hot routes. That means the answer to the first problem is the probability of a car on each road segment at time $t + \delta t$ instead of the deterministic road segment. Therefore we need to build a statistical model to predict how many cars will be on road segment r at time $t + \delta t$.

In this paper we come up with a data-driven approach to predict the real time traffic convergence of a road network. We use massive taxi trajectories as our experimental data to predict the convergence. One reason why we use taxi data is that it is difficult to get private cars trajectories in the city. The other reason is that to some extent the trajectory data of large amount of taxis can represent the traffic pattern of the whole city and also reflect some events [10]. Additionally, taxis are moving in the city quite more often than cars and by monitoring a taxi for a long period, it might be equivalent with a couple of dozen cars, in terms of vehicle-kilometers travelled. The contributions of this paper are three-fold:

1. We propose a new method predicting the future location of taxis on the road network without having any prior knowledge about personal driving habits.
2. We propose a statistical model to predict real time convergence on the road network.
3. We develop an efficient algorithm based on the suffix tree to find frequent trajectories and test it using real world massive taxi data.

The rest of the paper is organized as follows. The next section briefs related work about hot route detection and trajectory prediction. Section 3 formally defines the problem addressed. Section 4 presents the algorithms for predicting the future location of a taxi and the statistical model for predicting the convergence on road network. In Section 5, we discuss how to improve the efficiency of our algorithms based on the suffix tree. Section 6 covers comprehensive experiment evaluations over real world massive taxi data. Finally, Section 7 concludes and points out future directions for research.

2 Related Work

We first review works about hot routes or hot spots detection and then cover the works about trajectory prediction.

2.1 Hot Routes Detection

Finding hot routes or popular districts from trajectories of cars is a very interesting problem which has attracted a lot of attention. It could reveal many interesting aspects of a city, such as the flow pattern of people from home to work place, and popular places where people go to shop, eat, *etc.*

Liu et al. propose a mobility-based clustering method to find the hot spots in the city through taxis' GPS data[10]. [8] also proposes a method to reveal the pulse of the city and the character of each district, such as residential area, working places, shopping mall and so on. In [18,19], authors try to find the most popular hot routes from trajectories that help reveal the majority of travel patterns of the city, such as which routes most people travel. However, all these works only detect current or historic traffic patterns. They can not predict the future traffic situation and reflect the real time traffic changes.

2.2 Trajectory Prediction

In [1], Monreale *et al.* provide an algorithm to predict the next markable places people would go, such as home, convenience stores, working places, through mining those most similar location patterns. When it comes to road network, Liu *et al.* [17] present two models to predict short term route choice of a car: turning left, right or straight with a multi-way classification. [6,9] try to predict users' mid-term or long-term trajectory assuming that people have their usual habits and regular travel pattern which is not true for taxi drivers. [7] tries to predict the future traffic density with a statistical model. They assume their simulated cars always travel on the shortest paths which is not true in the real world. Combining the trajectory data with the real-world traffic situation, Ide *et al.* [15] predict travel-time for a given origin-destination (OD) pair on a road map based on the Gaussian regression model.

All those previous works do not take the real world situation such as traffic light variability or congestion into consideration. They assume that cars drive at constant speed or the speed proportional to the road speed limit. Furthermore, most of those works only test their methods with synthetic data or just hundreds of personal trajectories which is really difficult to tell whether or not their methods work. To the best of our knowledge, we are the first to propose the method to predict the future trajectory without any prior knowledge of personal travel habits and take the real world situation such as traffic lights and congestion into account, through mining massive trajectory data (more than 18000 taxi trajectories and 300000 trips per day).

3 Notations and Problem Definitions

In this section, we will introduce the notations used in this paper and then give the formal definitions of two problems we plan to solve.

The trajectory data are represented as $tr(< p_1, t_1 >, ..., < p_i, t_i >, ..., < p_n, t_n >)$, where p_i is the location reported by GPS device at time t_i. Then

we match those points to the edges of road network (due to space limitation, we omit the details of our map matching techniques). As a result, the trajectory $tr(< p_1, t_1 >, ..., < p_i, t_i >, ..., < p_n, t_n >)$ turns into $tr(< p'_1, e_1, t_1 >, ..., < p'_i, e_i, t_i >, ..., < p'_n, e_n, t_n >)$, where e_i is the edge of road network and p'_i is the mapped point of p_i on e_i. If p'_i and p'_{i+1} are not on the same edge or not on the neighbor edges, we can use shortest path algorithm to insert the edges between them to ensure that edges are continuous. By assuming linear interpolation, we can also get the exact location of each trajectory at any time given that the speed is constant between p'_i and p'_{i+1}.

The following is the list of notations used in this paper:

Table 1. Notations

tr_i	The trajectory i after map matching.
$tr_i.e_j$	The jth edge of trajectory i.
$tr_i(x, y)$	The subtrajectory of trajectory i that is between (including) $tr_i.e_x$ and $tr_i.e_y$.
$tr_i.len$	The number of edges of trajectory i.
$tr_i.ttype$	The time type for trajectory i, *i.e.* workday, weekend, rush hour or night time (for detailed classification , see section 6).
ctr	All trajectories of loaded taxis currently and we want to predict their future locations.
ctr_i	The ith trajectory of ctr.
$ctr_i.e_{ne}$	The last edge of ctr_i currently.
htr	All historic trajectories.
htr_j	The jth edge of htr.

There are two problems we need to solve in this paper:

Problem 1. **Location Prediction** Given historic trajectories htr and current trajectory ctr_i at time t, what is the probability $P_i^{e_k, t+\delta t}$ of ctr_i traveling on edge e_k at time $t + \delta t$?

Problem 2. **Convergence Prediction** Given historic trajectories htr and all the current trajectories ctr at time t, report e_k such that $\sum_{i=1}^{n} P_i^{e_k, t+\delta t} \geq T$, where n is the number of trajectories of ctr and T is the threshold defined by users.

4 Algorithms

In this section we will describe the algorithm of predicting a taxi's future location. The key idea of our algorithm is to find the similar trajectories as current trajectory and use them to predict the future location of current trajectory. Therefore, we need to define a similarity measurement for a historic trajectory htr_j and a current trajectory ctr_i. Since each trajectory has both spatial and temporal factors, we will take them into consideration correspondingly.

We focus on the spatial factor firstly. Since each trajectory is composed of a sequence of edges and each edge could be represented by a letter, each trajectory is actually a string. One popular similarity measurement for two strings is the *edit distance* [2]. With the idea of edit distance, we define the spatial-distance of trajectories:

Definition 1 (Spatial-Distance of Trajectories). *Given two trajectories tr_1, tr_2, the spatial-distance $gd_{tr_1,tr_2}(i,j)$ between two subtrajectories $tr_1(1,i)$ and $tr_2(1,j)$ is defined as follows:*

$$gd_{tr_1,tr_2}(i,j) = \begin{cases} 0 & ,i=j=0 \\ i & ,j=0 \text{ and } i>0 \\ j & ,i=0 \text{ and } j>0 \\ min \begin{cases} gd_{tr_1,tr_2}(i-1,j)+1 & ,i>0 \text{ and } j>0 \\ gd_{tr_1,tr_2}(i,j-1)+1 & ,i>0 \text{ and } j>0 \\ gd_{tr_1,tr_2}(i-1,j-1)+1 & ,i>0 \text{ and } j>0 \text{ and } tr_1.e_i \neq tr_2.e_j \\ gd_{tr_1,tr_2}(i-1,j-1) & ,i>0 \text{ and } j>0 \text{ and } tr_1.e_i = tr_2.e_j \end{cases} \end{cases}$$

It seems that we can decide whether two trajectories are similar or not based on spatial-distance. However, in the definition above, we do not consider the the length of trajectory. Intuitively, if the lengths of two trajectories are both longer, the two trajectories are more similar for the same spatial-distance than the two shorter trajectories. Otherwise if the difference between two trajectories' lengths is larger, they are less similar. To address the factors of trajectory length, we define spatial-similar trajectories as follows:

Definition 2 (Spatial-Similar Trajectories).
Two trajectories tr_1 and tr_2 are said to be spatial-similar if:

$$\frac{st_{tr_1,tr_2}}{tr_1.len} \geq \phi_1 \ \& \ \frac{st_{tr_1,tr_2}}{tr_2.len} \geq \phi_1 \ \& \ sRatio_{tr_1,tr_2} \leq \phi_2$$

where $sRatio_{tr_1,tr_2} = |\frac{st_{tr_1,tr_2} * (tr_1.len - tr_2.len)}{tr_1.len * tr_2.len}|$, $st_{tr_1,tr_2} = max\{tr_1.len, tr_2.len\} - gd_{tr_1,tr_2}$ and ϕ_1, ϕ_2 are the thresholds given by users, here we set them as 0.6 and 0.2.

Furthermore we need to consider the temporal factor. For example in Figure 1, *ctr* is going either to park or a working place. If the time is the morning of a working day, then there is higher probability that *ctr* is going to turn right to the work place. Otherwise if it is weekend, then the probability that *ctr* would turn left to the park is much higher. That suggests we need to take the day type (working day or weekend) and the time of the day (time stamp) into consideration. What's more, from [11][13] we can see the traffic system is a recurrent system hence the two traffic conditions in a street are more likely the same if they were in the same time interval.

Therefore we give the definition of trajectory similarity between a current trajectory and a history trajectory considering both spatial and temporal factors.

Definition 3 (Similarity With Historic Trajectory). *The similarity S_{ij} between a current trajectory ctr_i and a history trajectory htr_j is defined as follows:*

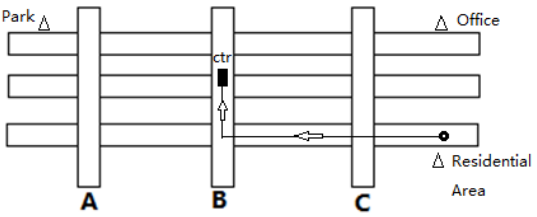


Fig. 1. Route choice with different time and day type

When $ctr_i.e_{ne} = htr_j.e_k$, $ctr_i.dtype = htr_j.dtype$, *and* ctr_i *are spatial-similar with* $htr_j(1,k)$ *:*

$$S_{ij} = w_1 * st_{ctr_i,htr_j(1,k)} + w_2 * exp(-|ctr_i.t - htr_j.t|)$$

Otherwise :

$$S_{ij} = 0$$

Algorithm 1 is to find and return all similar historic trajectories and their similarity for a current trajectory ctr_i.

Algorithm 1. Find Similar Historic Trajectories

Require:
The current trajectory ctr_i that we want to find its similar historic trajectories;
All the history trajectories htr;
Ensure:
All similar trajectories with ctr_i and their similarity value S_{ij};
1: Foreach trajectory htr_j in htr;
2: S_{ij}=0;
3: If($htr_j.ttype != ctr_i.ttype$)
4: continue;
5: Else
6: Foreach edge $htr_j.e_k$ in htr_j
7: If $htr_j.e_k == ctr_i.e_{ne}$
8: If htr_j and ctr_i are similar based on $Definition$ 2
9: Calculate S_{ij} with $Definition$ 3 ;
10: Break;
11: Return htr_j and S_{ij};

After finding all similar historic trajectories for ctr_i, we can compute $P_i^{k,t+\delta t}$ as follows:

$$P_i^{k,t+\delta t} = \frac{\sum_m S_{im}}{\sum_j S_{ij}}$$

where htr_m is the similar historic trajectory which is on edge ctr_i^{ne} at time t and on edge e_k at time $t + \delta t$.

After computing $P_i^{k,t+\delta t}$, for each edge e_k, we can just sum up all $P_i^{k,t+\delta t}$ to get the expected number of cars on e_k at time $t + \delta t$. If this value is larger than predefined threshold T, we report it. The algorithm is described in algorithm 2.

Algorithm 2. Find Road Map Convergence

Require:

All the trajectories ctr that are currently moving;
All the historic trajectories htr;
The current time t that we want to start to predict;
The time period δt that after how long we want to predict;
The road network G;
Threshold T;

Ensure:

All possible convergence roads and corresponding expected expected number of cars;

1: Foreach e_k in G
2: Foreach ctr_i in ctr
3: compute $P_i^{k,t+\delta t}$;
4: $nc_{e_k} = \sum_i p_i^{k,t}$;
5: If($nc_{e_k} > T$) ;
6: report e_k and nc_{e_k};

5 Accelerating Algorithms with Suffix Tree

For the algorithms in Section 4, the most time consuming step is to find historic trajectories for a given ctr_i, which repeat the latest edges of ctr_i because the algorithms need to scan all historic trajectories for each query and the amount of historic trajectories is huge. In order to improve the efficiency of the algorithm, we use the suffix tree to store the historic trajectories and perform the query over the suffix tree.

Suffix tree is a well known technique to solve a large number of string problems that occur in text-editing, free-text search, computational biology and other application areas for its space-efficient storage of massive amounts of string data.[20]. Formally, a suffix tree ST for string $S = S[0..n-1]$ of length n over the alphabet A is a tree with the following properties:

- ST has exactly $n + 1$ leaf nodes, numbered consecutively from 0 to n.
- all internal nodes (except the root) have at least two children.
- edges spell non-empty strings.
- all edges from the same node start with a different element of A.
- for each leaf node i, the concatenation of all edges from the root node to i matches $S[i..n-1]$.

The time for building suffix tree is $O(n)$. For a query string Q with length m, we can find all substrings of S matching Q in $O(m+z)$ time where z is the number of substrings matching Q.

In order to use the suffix tree in our problem, we need to encode each edge as a unique ID. Then the sequence of IDs of a trajectory forms a string. Furthermore, there are multiple trajectories instead of one. We need to concatenate them by putting a different termination symbol between consecutive trajectories. This is called generalized suffix tree [3]. We implement our suffix tree with an online algorithm [4] since in reality, the trajectories appear in online fashion.

6 Experiments

The data set we use is the GPS data of 18000 taxis from a southern city of China. Each GPS device records one data point about every 30-150 seconds with time stamp and the status of taxi (loaded or unloaded). All GPS devices work nonstop 24/7 and produce around 3Gbyte data every day. We extract 15 days data (Sep. 12-26, 2011) for our experiments to build the trajectory database. For the road network, there are 14029 edges and 11124 nodes. Note that we only extract the trajectories with loaded passengers since unloaded taxis do not have specific destinations and they might travel randomly. After preprocessing, we get 4035712 loaded trajectories. As we claimed in section 4, there are different traffic patterns for different day type (working day and weekend) and different time intervals in a day (rush hour, working hour, *etc.*), in order to accelerate our program we first separate those trajectories by the day type then by different time intervals as follows: {workday, 7am-10am}(morning rush hour), {workday, 10am-5pm}(working hour), {workday, 5pm-7pm}(afternoon rush hour), {workday,7pm-10pm}(night time), {workday,10pm-7am}(bed time), {weekend, 8am-10pm}(weekend day time), {weekend, 10pm-8am}(weekend night time). We only compare the similarity of trajectories in the same time group.

Figure 2 shows the number of same sequence trajectories varying with the length of trajectories. Here we define the same sequence trajectories as they travel exactly the same edge sequence on road network. We can see the number of same sequence trajectories decreases for longer trajectories. This is reasonable since the longer distance between origin-destination, the more edge combinations drivers could choose. When the length is larger than 15, the number of the same sequence trajectories is less than 10. As we know, if we do not have enough historic trajectories, the prediction will not be precise. Furthermore, for a current trajectory, we think the subtrajectories close to the current time is more important than those subtrajectories earlier. Therefore, we show the number of spatial-similar trajectories based on the definition2 in figure 3, from which we can see the relationship between the number of spatial-similar trajectories and the trajectories' lengths. Similar to Figure 2, when the length becomes larger, the number of spatial-similar trajectories decreases. But the gradient is decreasing more smoothly, which reflects the interesting fact that the longer similar edge

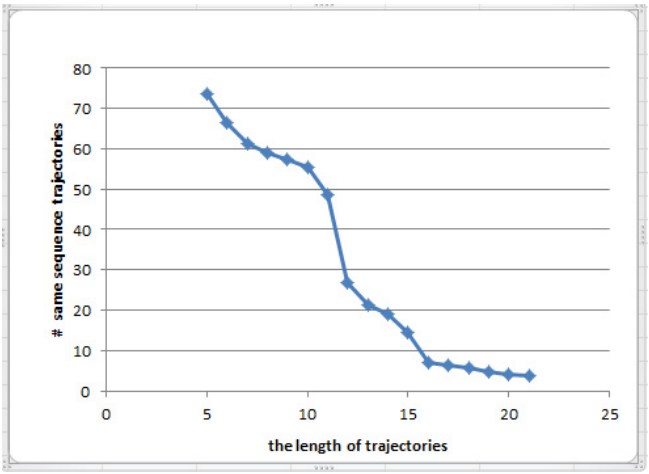


Fig. 2. The number of same sequence trajectories versus the length of trajectories

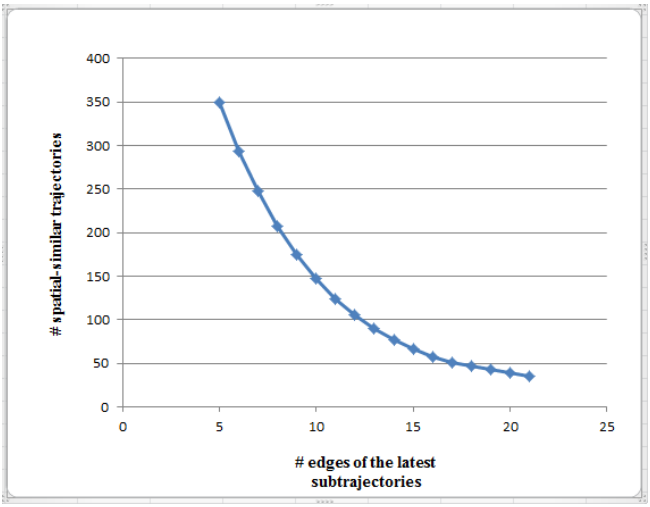


Fig. 3. The number of spatial-similar trajectories versus the number of edges of the latest subtrajectories

sequences some drivers had travelled, the higher probability they would choose the same edge sequence in the future.

We run the Algorithm 2 to predict the future convergence on the road network. We define the precision as follows:

$$Precision = \frac{\sum_k |nc_{e_k} - nc'_{e_k}|}{\sum_k nc'_{e_k}}$$

Where nc_{e_k} is the predicted number (by our algorithm) of taxis on edge e_k at time $t+\delta t$, nc'_{e_k} is the real number of taxis on edge e_k at time $t+\delta t$. We extract "current" trajectories (time t,for instance) on the move from the database as the test data while the rest as training data. nc_{e_k} is the result of our prediction and nc'_{e_k} is the real number of those "current" taxis on the edge e_k in our database after time δt.

We set the threshold λ of convergence as 2. We do not consider those edges that both $nc_{e_k} < \lambda$ and $nc'_{e_k} < \lambda$ since we are only interested in those edges having possible convergence. Figure 4 shows the precision of convergence prediction. The x axis is δt. As we can see, when δt increases, the precision decreases. The precision stays around 60% to 70% when $\delta t < 500$ seconds. When δt reaches 1000 seconds, the precision is around 40%.

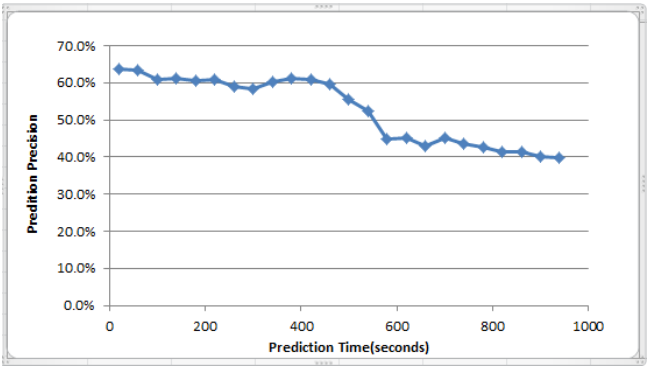


Fig. 4. The precision of convergence prediction

Lastly, we show the efficiency of our accelerating algorithms based on the suffix tree by the simulation in which we ran our algorithm online. We test our algorithms online as follows: once the system receive 1500 trajectories, we update the suffix tree simply by adding those new trajectories [3]. Then we perform the query of finding matched subtrajectories for those unfinished trajectories (the taxis loaded with passengers at that time).

Figure 5 shows the accumulating running time of finding matched subtrajectories versus the increase number of historic trajectories in database. It shows when the number of historic trajectories is small, the algorithm with suffix tree has no advantage over the algorithm without suffix tree. But with the number of historic trajectories growing, the algorithm with suffix tree is much faster than the algorithm without suffix tree. This is because building suffix tree costs some extra time but it is only built once and can speed up the query time dramatically. Therefore the algorithm with suffix tree is suitable for large amount of historic trajectories and queries.

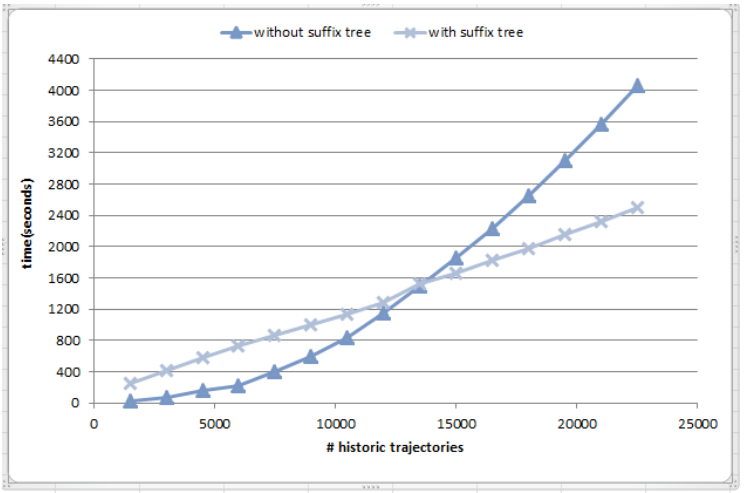


Fig. 5. The time cost of finding similar trajectories

7 Conclusions and Future Work

In this paper, we proposed a data-driven approach for convergence prediction on the road network. We introduced a new method predicting the future location of taxis on the road network without knowing any prior knowledge about personal driving habits. Furthermore we propose a statistical model to predict realtime convergence on road network. We experimentally demonstrated that our approach achieves high prediction precision.

In the future, to achieve better prediction precision, we will use more GPS data to construct the historic trajectory database in the future. Actually, we have already collected about one year GPS data of 18000 taxis and the data are still accumulating every day. For processing so large amount of data, we will also process our data on the cloud computing platform and develop more efficient algorithms to find repetitive subtrajectories.

References

1. Monreale, A., Giannottielli, F.P., Trasarti, R.: WhereNext: a Location Predictor on Trajectory Pattern Mining. In: The ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 637–646 (2009)
2. Atallah, M.J.: Algorithms and Theory of Computation Handbook. CRC Press LLC (1999)
3. Gusfield, D.: Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge Univ. Press (January 15, 1997)
4. Ukkonen, E.: On-line construction of suffix trees. Algorithmica 14(3), 249–260, doi:10.1007/BF01206331.

5. Buchin, K., Buchin, M., Gudmundsson, J., Löffler, M., Luo, J.: Detecting commuting patterns by clustering subtrajectories. Int. J. Comput. Geometry Appl. 21(3), 253–282 (2011)
6. Jeung, H., Liu, Q., Shen, H.T., Zhou, X.: A Hybrid Prediction Model for Moving Objects. In: The International Conference on Data Engineering, pp. 70–79 (2008)
7. Kriegel, H.-P., Renz, M., Schubert, M., Zuefle, A.: Statistical Density Prediction in Traffic Networks. In: The SIAM Conference on Data Mining, pp. 692–703 (2008)
8. Froehlich, J., Neumann, J., Oliver, N.: Sensing and Predicting the Pulse of the City through Shared Bicycling. In: The International Joint Conferences on Artificial Intelligence, pp. 1420–1426 (2009)
9. Froehlich, J., Krumm, J.: Route Prediction from Trip Observations. In: The Society of Automotive Engineers (SAE) World Congress (2008)
10. Liu, S., Liu, Y., Ni, L.M., Fan, J., Li, M.: Towards mobility-based clustering. In: The ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 919–928 (2010)
11. Tan, M., Wong, S.C., Xu, J., Guan, Z., Zhang, P.: An Aggregation Approach to Short-Term Traffic Flow Prediction. The IEEE Transactions on Intelligent Transportation System 10(1) (March 2009)
12. Laube, P., van Kreveld, M., Imfeld, S.: Finding REMO - Detecting Relative Motion Patterns in Geospatial Lifelines. In: The International Symposium on Spatial Data Handling, pp. 201–215 (2005)
13. Oswald, K., Scherer, W.T., Smith, B.L.: Traffic Flow Forecasting Using Approximate Nearest Neighbor Nonparametric Regression. Research Report No. UVACTS-15-13-7 (December 2001)
14. Scellato, S., Musolesi, M., Mascolo, C., Latora, V., Campbell, A.T.: NextPlace: A Spatio-temporal Prediction Framework for Pervasive Systems. In: Lyons, K., Hightower, J., Huang, E.M. (eds.) Pervasive 2011. LNCS, vol. 6696, pp. 152–169. Springer, Heidelberg (2011)
15. Ide, T., Kato, S.: Travel-Time Prediction using Gaussian Process Regression: A Trajectory-Based Approach. In: The SIAM Conference on Data Mining, pp. 1183–1194 (2009)
16. Ge, Y., Xiong, H., Liu, C., Zhou, Z.-H.: A Taxi Driving Fraud Detection System. In: The International Conference on Data Mining, pp. 181–190 (2011)
17. Liu, X., Karimi, H.A.: Location awareness through trajectory prediction. Computers. Environment and Urban Systems 30(6), 741–756 (2006)
18. Li, X., Han, J., Lee, J.-G., Gonzalez, H.: Traffic Density-Based Discovery of Hot Routes in Road Networks. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 441–459. Springer, Heidelberg (2007)
19. Chen, Z., Shen, H.T., Zhou, X.: Discovering Popular Routes from Trajectories. In: The International Conference on Data Engineering, pp. 900–911 (2010)
20. `http://www.allisons.org/ll/AlgDS/Tree/Suffix/`

Tour Suggestion for Outdoor Activities

Joris Maervoet[1,2,⋆], Pascal Brackman[3], Katja Verbeeck[1], Patrick De Causmaecker[2,4], and Greet Vanden Berghe[1]

[1] KAHO Sint-Lieven, Computer Science, CODeS, Gebr. De Smetstraat 1, 9000 Ghent, Belgium
[2] KU Leuven-Kulak, Department of Computer Science, CODeS, Etienne Sabbelaan 53, 8500 Kortrijk, Belgium
[3] RouteYou.com, Kerkstraat 108, 9050 Ghent, Belgium
[4] iMinds - ITEC - KU Leuven

Abstract. The present article introduces the outdoor activity tour suggestion problem (OATSP). This problem involves finding a closed path of maximal attractiveness in a transportation network graph, given a target path length and tolerance. Total path attractiveness is evaluated as the sum of the average arc attractiveness and the sum of the vertex prizes in the path. This problem definition takes its rise in the design of an interactive web application, which suggests closed paths for several outdoor activity routing modi, such as mountain biking. Both path length and starting point are specified by the user. The inclusion of POIs of some given types enrich the suggested outdoor activity experience.

A fast method for the generation of heuristic solutions to the OATSP is presented. It is based on spatial filtering, the evaluation of triangles in a simplified search space and shortest path calculation. It generates valuable suggestions in the context of a web application. It is a promising method to generate candidate paths used by any local search algorithm, which further optimizes the solution.

Keywords: tour suggestion, transportation network graphs.

1 Introduction

The company RouteYou offers recreational navigation for several outdoor activity modes such as hiking and mountain biking. This involves maintenance of a set of transportation network graphs, in which each arc r has a length l_r and attractiveness $0 < a_r \leq 1$. The latter parameter models suitability to the applicable outdoor activity mode, in terms of the arc's scenic context, physical condition and relation to traffic. The motive behind the present paper is the design of a tour suggestion module, which plans attractive round trips for any of the company's outdoor activity modes. This module requires that the user chooses an outdoor activity mode and specifies a target path length and a starting point. Within some seconds, it returns a closed path, consisting of arcs of optimal attractiveness and satisfies the length and starting point constraints. The desirable

⋆ Corresponding author.

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 54–63, 2013.
© Springer-Verlag Berlin Heidelberg 2013

path has some constraints because users tend not to accept paths with a considerable number of self-intersections or with recurring subpaths. Paths in clockwise direction are usually prefered in countries with right-hand traffic because they ease turn traffic manoeuvres. Moreover, the user is able to select one or several POI types of preference, such that the output path results from a trade-off of arc attractiveness and the number of contained POIs of the preferred types. Examples of interesting mode - POI type combinations are hiking with mountains and motorcycling with scenic viewpoints. Another scenario is that external organizations set the mode and POI types of preference in a customized planner. This sort of planner aims at promoting a peculiar type of tourism in a specific region e.g. cycling along the Châteaux of the Loire Valley in France. An extension to this module is the generation of multiple suggestions. This means that the user can browse through a set of m path suggestions, for the same set of preferences and constraints. It involves finding the set of m most attractive tours that are spatially different.

The next section gives a literature overview of models applied in the domain of leisure and tourism. Section 3 introduces the OATSP, which formalizes the tour suggestion problem described above. The approach presented in Section 4 enables generating a set of heuristic solutions to the OATSP in a low computational time. The following section discusses two sets of individual tours obtained by this approach. Section 6 is the conclusion of this work.

2 Tour Suggestion Models for Leisure and Tourism

The tour suggestion problem for leisure and tourism (TSPLT) involves generating a path through a transportation network visiting some arcs and/or points of interest (POIs). The path should optimally match the end user's preferences or some general recreational preference, given a set of constraints (adapted from the itinerary planning problem formulated by Shcherbina and Shembeleva [15]). It has applications in several subdomains of leisure and tourism. *Recreational point-to-point navigation* aims at generating a path from A to B, which is tailored to a specific recreational navigation mode, such as nordic walking. *Individual city trip planning* involves providing a tour schedule along a selection of POIs, satisfying the personal preferences of an individual intending to visit a city for a certain amount of time. This kind of services are often realized as a web-based application or a mobile client-server application (e.g. [1,16]), generating on-the-fly suggestions for the end user. This is usually not required for applications in collective tourism planning (bus tour planning, cruise itinerary planning). The path generated in the TSPLT may be open or closed. In individual city tour planning for instance, the path is - in most cases - closed. The trip typically both starts at and ends in the tourist's hotel, a parking lot or a train station.

A first type of systems model the problem as a shortest path (SP) problem. This model generates exclusively open paths and focuses on the suitability of the arcs for a certain purpose. In the most common recreational SP approach, the inverse suitability is encoded in single arc weights of a directed weighted graph, corresponding to the transportation network. Path generation involves

that common SP algorithms are used to find the path with the lowest cost between two nodes of this graph. Traditional SP algorithms are Dijkstra [3] and its variants [24] and A* [6]. The use of single scenic/attractive weights has often been suggested in this context (e.g. [13,9]). The company presented in the introduction adapted this concept by introducing weight attractiveness for a series of recreational routing modi. Attractiveness also deals with physical conditions and traffic aspects of the roads. Rogers and Langley [12] model the attractiveness of weights by a linear combination of criteria that reflect the end user's preferences. Niaraki and Kim [11] developed an ontology-based technique that generates network weights for personalized routing planning. Tarapata [20] states that single-objective functions are not sufficiently adequate to model real SP problems. He presents a classification of multi-objective shortest path (MOSP) problems, which are used in other real application domains, such as routing with quality-of-service in computer networks. The author identifies six general solution methods to MOSP problems, including mathematical optimization and objective function hierarchization. Hochmair and Navrath [7] argue that the computation of attractive routes is generally beyond the ability of SP algorithms, since an SP algorithm is not able to find a route that maximizes a benefit criterion. However, they demonstrate the practical value of single criterion SP computation for finding this type of routes.

A second type of systems use a problem model that trades off POI selection with time or distance. It originates from the field of Operations Research (OR). It involves selecting a sequence of POIs from an eventually larger input set of POIs. This sequence should meet certain preferences and/or must satisfy a set of constraints. The POI sequence selection is often preceded by a POI filtering mechanism, improving the sequence selection performance. The model takes into account the travel time or distance between candidate POIs, aided by a precalculated travel time/distance matrix or a heuristic estimation function. The resulting path is obtained by concatenating precalculated (shortest) paths between the selected POIs, or by recalculating the complete path using via-points. The latter approach is useful in order to avoid undesirable U-turns and forbidden traffic manoeuvres passing through a selected POI. Godart [5] presented a version of the traveling salesperson problem (TSP) that integrates activity selection and lodging availability for trip planning problems. Deitch and Ladany [2] introduced the bus touring problem (BTP). It requires an undirected graph in which the vertices represent visiting sites and the edges represent connecting scenic routes. Both edges and vertices have associated attractivity values and require traveling/visiting times. The goal is to find a (closed path) bus tour of maximal total attractivity, below a given maximal tour time. The attractivity of recurrently visited vertices and edges is only counted once. The authors show that the BTP can be transformed to the Orienteering Problem (OP). Both Suna and Lee [19] and Maruyama et al. [10] have built tourist trip recommender systems based on variants of the prize collecting TSP (PCTSP). The original model minimizes the total travel cost minus the sum of the benefit criterion values of the POIs along the selected path. Suna and Lee integrate a personal interest factor in the travel cost weights. The two following query models in the field of spatial databases focus on trip planning with *typed* locations. Li et al. [8] introduce the

trip planning query, which involves a request for the shortest route from and to a given point that passes through at least one point of any of the specified set of location types. The optimal sequenced route query [14] looks for the shortest path that visits locations according to a specified sequence of POI types. An example of a POI type sequence for leisure is: (1) hair dresser (2) restaurant (3) cinema. Vansteenwegen and Van Oudheusden [23] introduced the tourist trip design problem (TTDP), which is modelled as an OP with time windows. This model starts from a fully interconnected distance/time-weighted graph in which the vertices represent POIs with a personalized [18] score. It involves finding the sequence of POIs that maximizes the total score of the selected POIs, while the total path weight must not exceed a given value. Each POI can only be visited once. Moreover, certain POIs are only available within certain time windows (cf. opening hours). Very good approximate solutions are found with iterated local search by Vansteenwegen et al. [21,22]. They have shown the practicability of the TTDP in city trip planning. All problems in this second class are $\mathcal{NP}$-hard. Both Shcherbina and Shembeleva [15] and Souffriau and Vansteenwegen [17] provide a more detailed overview of the models and functionalities of this type of systems.

3 The Outdoor Activity Tour Suggestion Problem

The BTP model suits the requirements of company's tour suggestion module best. The objective function takes into account both edge and vertex attractiveness, whereas tour time resides under the constraints. However, the tour suggestion module requires a target path length instead of a maximal travel time. This gives rise to a distance window constraint. In order to model the particular tour suggestion problem, we introduce the OATSP.

This model requires a directed graph $G = (V, A)$, for which

- any arc $r \in A$ is an ordered pair $(v_1, v_2) \in V^2$,
- any arc $r \in A$ has an associated attractiveness $0 < a_r \leq 1$ and length l_r,
- reverse arcs have equal attractiveness and length,
- each vertex v has an associated prize $0 \leq p_v \leq 1$, and,
- a closed path C is a series of arcs, which are circularly subsequent in G. Each element of A appears at most once in C. C_i denotes the arc at the i-th position in C. The sets $A_C \subseteq A$ and $V_C \subseteq V$ consist of the arcs and vertices visited by the closed path C. The set A'_C contains any arc $C_i = (v_1, v_2)$ of C that is not preceded by its reverse arc $C_j = (v_2, v_1)$ with $j < i$.

Given a prefered path length l_p, length tolerance t and starting vertex v_s, a solution to the OATSP is a closed path C in G satisfying

1. C_1 starts in v_s
2. $(1-t) \cdot l_p \leq \sum_{r \in A_C} l_r \leq (1+t) \cdot l_p$
3. $\varphi \cdot \dfrac{\sum_{r \in A'_C} (a_r \cdot l_r)}{\sum_{r \in A_C} l_r} + \sum_{v \in V_C} p_v$ is maximal

The parameter φ determines the relative importance of arc to node attractiveness. This problem is a variant of the PCTSP [4]. This model suits the requirements of

the tour suggestion module. It is able to suggest closed paths of a target length and optimal arc and POI attractiveness, by setting the vertex prizes to the degree of membership to the specified categories. Since recurrent visits to reverse arcs and nodes are penalized by the objective function, solutions tend to avoid U-turns and recurring reverse subpaths. The model does not allow recurring arcs. It does allow multiple vertex and reverse arc visits since the starting vertex or any of the POIs may be located on a subgraph with low connectivity. Planar self-intersections can be avoided by assigning a low threshold value to the prizes of vertices that do not belong to any of the specified categories.

4 Approach

A multiple tour suggestion module has been designed for the web application of the company. It enables generating a set of m heuristic solutions to the OATSP within a low computational time, in an environment without precalculated paths nor distances available. The main algorithm starts by determining a *feasibility window (FW)*. This square area discerns the arcs and nodes reachable in a round trip through v_s given the path length window constraint. The algorithm assumes that, within the FW, (a) only a few (ranging from none to the order of tens) vertices v have $p_v > 0$, (b) there is a diversity of attractiveness a_r amongst the arcs r, and, (c) vertices with $p_v > 0$ have a high probability to be located along trajectories of higher attractiveness than vertices in their local neighbourhoods. These conditions often hold for the tour suggestion module presented, where the user selects few POI categories and relatively low values for l_p. Moreover, it assumes low values of φ.

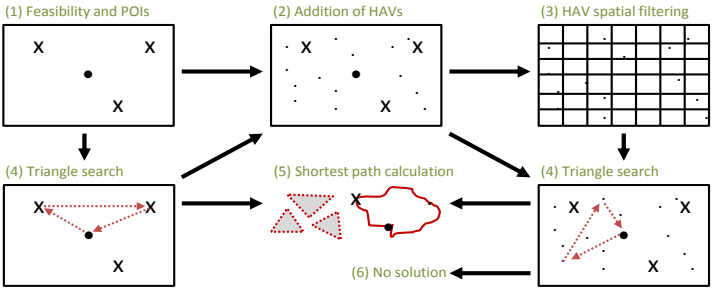


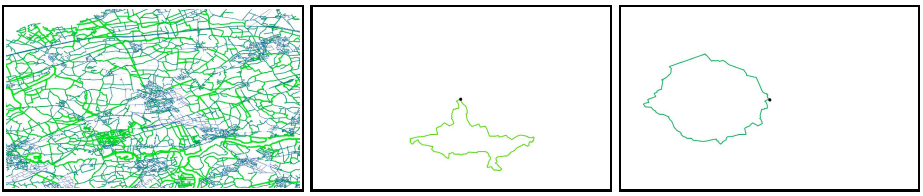
Fig. 1. State diagram of the fast heuristic algorithm for the OATSP

Fig. 1 shows the state diagram of the main algorithm. In state (1), the FW is generated in the geographic coordinate plane, such that v_s is in the center and the projected width and height measured through v_s, equals l_p/e_1. If the FW contains more than n_1 feasible POIs ($p_v > 0$), the algorithm continues with triangle search. Otherwise, the window is enriched by highly attractive vertices (HAVs), in state (2). HAVs are auxiliary POIs representing the most attractive arcs in the FW. This phase looks for the maximal value a_{min} for which there exist at least n_3 arcs r with $a_r \geq a_{min}$. Any vertex that is connected by an arc r

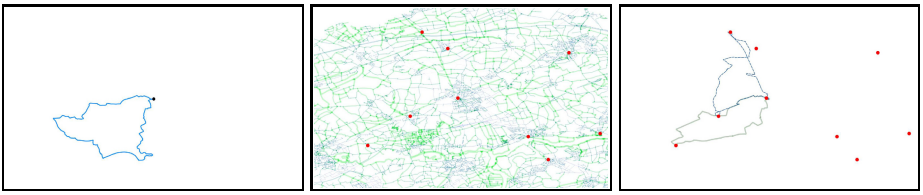
with $a_r \geq a_{min}$, is an HAV. The maximal attractiveness of the connecting arcs of an HAV v is denoted a_v. Next, all HAVs are promoted as POIs, with p_v set to $\varphi a_v / \#HAVs$. If the total number of POIs does not exceed n_2, the algorithm continues with triangle search. Otherwise, the HAVs are subjected to spatial filtering (state (3)). This involves that all points are categorized into one of the boxes of the $g_1 \times g_1$ grid constructed over the window. Only one auxiliary POI per box is kept, absorbing the sum of p_v-values of the other points in the box. Triangle search in state (4) involves the brute-force evaluation of any directed triangle made up by v_s and any other two POIs within the window. This evaluation consists of two steps. First, a triangle undergoes a fast feasibility check. A triangle is feasible if its direction is clockwise (in case of right-hand traffic) and if its perimeter is between $(1 \pm t) \cdot l_p / e_2$. Next, an evaluation function computes a score for a feasible triangle in this simplified window representation. This function returns a weighted sum of the prizes p_v for any of the POIs involved. Prizes of POIs located at one of the triangle vertices are given double weights. POIs located on one of the triangle edges generate p_v, which decreases as the elliptical distance to the closest triangle edge increases. If all angles are greater than x_1, a global bonus ($*x_2$) is granted. When no feasible triangle after HAV promotion, the algorithm does not return any solution (state (6)). Otherwise, the triangle of highest score is passed to the SP calculation state. Given a triangle abc, this fifth state entails finding the concatenation of paths of lowest cost between (a, b), (b, c) and (c, a) in a graph with arc weights equal to l_r / a_r. In order to avoid U-turns and recurring subpaths, the weights of both the forward and available reverse arcs of the resulting subpath are drastically increased, after each subrouting. If the concatenated path does not meet the length window constraint, the paths between (a, b), (b, c) and (c, a) are calculated in a changed order and again concatenated. If it still does not meet the constraint, it is rejected after which the second best triangle is processed, and so on.

The multiple suggestion extension is realized by the integration of a simple *competitive learning* algorithm in the triangle search. It relies on a triangle distance function which is defined as follows. The two variable vertices of a triangle are categorized into one of the boxes of the $g_2 \times g_2$ grid constructed over the window. The *distance* between two (clockwise) triangles is the sum of the Manhattan distance (MD) between the first vertices and the MD between the second vertices in the grid. Two triangles are called *resemblant* if one of the MDs is lower than 2. During the brute-force triangle evaluation, the algorithm manages a store of maximally m prototype solutions. Suppose the current solution does not resemble any prototype solution. If there are less than m prototype solutions, the current solution enters the store as a new prototype. If the store is full and the current solution is better than the worst prototype, it replaces the prototype. Suppose the current solution resembles one or more prototype solutions. If it is better than the closest prototype, it replaces the prototype. In the end, the m prototype solutions are passed to the SP calculation state.

Parameter Overview. The parameters e_1 and e_2 should be set to the minimal and average ratio of a SP length to the Euclidean distance in a graph with arc weights l_r / a_r. The lower limit n_1 and upper limit n_2 determine the number



(a) SN-1: arc attractiveness (b) SN-1: solution #1 (c) SN-1: solution #2



(d) SN-1: solution #3 (e) SN-2: arc attractiveness (f) SN-2: solution #1/#2

Fig. 2. Arc attractiveness map and the three/two best tours in the solution store for SN-1/SN-2. Each subfigure is shown in the FW, centered around v_s. Highly attractive arcs are depicted by thick green lines, and less attractive arcs by thin blue lines. The red dots indicate the POIs of the type 'interesting church'. The best solution of SN-2 is indicated by a solid line.

of POIs used for triangle search. Too low numbers can result in a shortage of feasible triangles, whereas high numbers increase the computational time. n_3 determines the number of HAVs used for triangle search, but depends on the dispersion of arcs of equal attractiveness in the FW. x_1 and x_2 should be experimentally determined in order to reward less acute-angled triangles, since very acute angles result in poor tours.

5 Results

The approach introduced in the previous section has been tested for the outdoor activity mode 'attractive cycling'. The transportation network graph for this mode contains both paved and unpaved roads. The OATSP settings are $\varphi = 1$ and $t = 0.30$. The remaining parameters are set as follows: $e_1 = 1.41$, $e_2 = 1.6$, $n_1 = 5$, $n_2 = 60$, $n_3 = 20$, $x_1 = 40°$, $x_2 = 1.5$, $g_1 = g_2 = 10$, and $m = 10$.

A first experiment, called SN-1, assigns the centre of a medium-sized city in Belgium as starting vertex v_s and sets $l_p = 30km$. No POI category of interest is specified, so initially any vertex prize $p_v = 0$. Fig. 2 shows the arc attractiveness map within the FW and the 3 out of 10 tours of highest scores in the solution store. Most of the arcs within the city are substantially less attractive than the arcs in the neighbourhood. The arcs in the south of the window, either along the rivers or within a woody region, have the highest attractiveness. Each of the top-3 tours visits this region. Any of the 10 tours in the store were assessed by amateur cyclists as attractive tours. They valued the solution diversity in the store highly. Experiment SN-2 adds the POI category 'interesting church' to the

Table 1. Top-5 result characteristics for both experiments

SN-1	Δeval.f.	tour length	OATSP obj.f.
# 1	0.4477	37.6 km	0.8126
# 2	0.3937	34.9 km	0.8316
# 3	0.2998	31.8 km	0.7800
# 4	0.2619	28.6 km	0.7297
# 5	0.2280	30.4 km	0.6826

SN-2	Δeval.f.	tour length	OATSP obj.f.
# 1	4.9709	25.1 km	3.7618
# 2	4.9073	30.1 km	3.7265
# 3	4.3079	34.1 km	2.7078
# 4	4.0000	32.3 km	2.7393
# 5	4.0000	25.6 km	2.7660

specifications of SN-1. Each closest vertex to a POI of this category gets $p_v = 1$. Fig. 2 shows these POIs, and the 2 out of 5 tours of highest scores in the solution store. One of the POIs is located very close to and to the west of v_s. This gives tours in the west containing this POI, priority over the other tours.

Experiment Statistics. In SN-1, 764 HAVs were added to the FW and reduced to a set of 84. It took 752 ms to evaluate[1] $84 \cdot 83$ triangles (out of which 1256 were feasible). SN-2 only went through states (1), (4) and (5), so no HAVs were added to a set of 9 POIs. It took only 2 ms to evaluate $9 \cdot 8$ triangles. Only 5 out of 13 feasible triangles were considered sufficiently diverse by the competitive learning algorithm and kept in the solution store. The triangle evaluation scores, tour lengths and OATSP objective function scores of the 5 best SN-1 and SN-2 results in the solution store are given in Table 1. Specifically in SN-1, the OATSP objective function equals the attractiveness average of arcs visited by the tour. The triangle evaluation score correlates with the objective function. Only the solution of rank 1 has been overestimated. The average arc attractiveness along the (a, b) subpath of this solution is remarkably lower than the average along the other subpaths. In the case of SN-2, the OATSP objective function takes into account both arc and node attractiveness. Only POIs were assigned to the triangle vertices, and therefore each tour scores at least two. Since no HAVs were used to calculate the triangle evaluation score, this function is only effective in predicting the number of POIs contained by a tour.

6 Conclusion

The OATSP, introduced in the present article, involves finding attractive closed paths in a transportation network graph, tailored for a specific outdoor activity mode. An experiment showed that the presented algorithm is able to generate a set of heuristic solutions to the OATSP, satisfying the constraints. The core of this algorithm is a brute-force triangle evaluation in the FW, containing the POIs with $p_v > 0$ and a set of auxiliary POIs, constrained to the number of g_1^2. Although a quadratic number of triangles is evaluated, the evaluation runs in low computational time. The triangle that receives the best evaluation is passed to an SP calculation module, prioritizing paths along attractive arcs. The triangle evaluation function has been found an efficient heuristic for the OATSP objective value of the resulting path. A simple competitive learning algorithm

[1] Using PHP (CLI) 5.3.6 in Ubuntu 11.10, with an Intel Core i7-920 Processor.

enables generating m tours of high objective value, which are spatially different. This algorithm of low computational impact showed to be effective.

Further work includes improving the introduced algorithm and a comparison with the route quality and computational time of the results generated by a local search method. Potential improvements are the introduction of precalculated paths in the triangle evaluation, the replacement of the brute-force technique by a heuristic selection of triangles, and, the evaluation of polygons instead of triangles.

Acknowledgment. The research has been carried out as part of the industrial PhD project "Structural heuristics for personalized routes" funded by the IWT (090726) and the company RouteYou.

References

1. Cheverst, K., Davies, N., Mitchell, K., Friday, A., Efstratiou, C.: Developing a context-aware electronic tourist guide: some issues and experiences. In: Proceedings of the Conference on Human Factors in Computing Systems, pp. 17–24. ACM (2000)
2. Deitch, R., Ladany, S.: The one-period bus touring problem: Solved by an effective heuristic for the orienteering tour problem and improvement algorithm. European Journal of Operational Research 127(1), 69–77 (2000)
3. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
4. Fischetti, M., Salazar-González, J.J., Toth, P.: The generalized traveling salesman and orienteering problems. In: The Traveling Salesman Problem and Its Variations, Combinatorial Optimization, vol. 12, pp. 609–662. Springer US (2004)
5. Godart, J.: Combinatorial optimisation based decision support system for trip planning. In: Buhalis, D., Schertler, W. (eds.) Information and Communication Technologies in Tourism, pp. 318–327. Springer (1999)
6. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybernetics 4(2), 100–107 (1968)
7. Hochmair, H.H., Navratil, G.: Computation of scenic routes in street networks. In: Car, A., Griesebner, G., Strobl, J. (eds.) Proceedings of the Geoinformatics Forum, Salzburg, Austria, pp. 124–133. Wichmann Verlag, Heidelberg (2008)
8. Li, F., Cheng, D., Hadjieleftheriou, M., Kollios, G., Teng, S.-H.: On Trip Planning Queries in Spatial Databases. In: Medeiros, C.B., Egenhofer, M., Bertino, E. (eds.) SSTD 2005. LNCS, vol. 3633, pp. 273–290. Springer, Heidelberg (2005)
9. Malaka, R., Zipf, A.: Deep map - challenging IT research in the framework of a tourist information system. In: Information and Communication Technologies in Tourism, pp. 15–27 (2000)
10. Maruyama, A., Shibata, N., Murata, Y., Yasumoto, K., Ito, M.: A personal tourism navigation system to support traveling multiple destinations with time restrictions. In: Proceedings of AINA 2004, pp. 18–22. IEEE Computer Society (2004)
11. Niaraki, A.S., Kim, K.: Ontology based personalized route planning system using a multi-criteria decision making approach. ESWA 36(2), 2250–2259 (2009)
12. Rogers, S., Langley, P.: Personalized driving route recommendations. In: Proceedings of the AAAI Workshop on Recommender Systems, pp. 96–100. Madison (1998)

13. Sanders, P., Schultes, D.: Engineering Fast Route Planning Algorithms. In: Demetrescu, C. (ed.) WEA 2007. LNCS, vol. 4525, pp. 23–36. Springer, Heidelberg (2007)
14. Sharifzadeh, M., Shahabi, C.: Processing optimal sequenced route queries using voronoi diagrams. Geoinformatica 12(4), 411–433 (2008)
15. Shcherbina, O., Shembeleva, E.: Modeling recreational systems using optimization techniques and information technologies. Annals of OR, 1–21 (2011)
16. Souffiau, W., Maervoet, J., Vansteenwegen, P., Vanden Berghe, G., Van Oudheusden, D.: A Mobile Tourist Decision Support System for Small Footprint Devices. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 1248–1255. Springer, Heidelberg (2009)
17. Souffriau, W., Vansteenwegen, P.: Tourist Trip Planning Functionalities: State–of–the–Art and Future. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010. LNCS, vol. 6385, pp. 474–485. Springer, Heidelberg (2010)
18. Souffriau, W., Vansteenwegen, P., Vertommen, J., Vanden Berghe, G., Van Oudheusden, D.: A personalised tourist trip design algorithm for mobile tourist guides. Applied Artificial Intelligence 22(10), 964–985 (2008)
19. Sun, Y., Lee, L.: Agent-based personalized tourist route advice system. In: SPRS Congress Istanbul 2004, Proceedings of Commission II, pp. 319–324 (2004)
20. Tarapata, Z.: Selected multicriteria shortest path problems: An analysis of complexity, models and adaptation of standard algorithms. Int. J. Appl. Math. Comput. Sci. 17(2), 269–287 (2007)
21. Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D.: A guided local search metaheuristic for the team orienteering problem. European Journal of Operational Research 196(1), 118–127 (2009)
22. Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D.: Iterated local search for the team orienteering problem with time windows. Comput. Oper. Res. 36(12), 3281–3290 (2009)
23. Vansteenwegen, P., Van Oudheusden, D.: The Mobile Tourist Guide: an OR Opportunity. OR Insight 20(3), 21–27 (2007)
24. Zhan, B.F., Noon, C.E.: Shortest path algorithms: An evaluation using real road networks. Transportation Science 32(1), 65–73 (1998)

Interpreting Pedestrian Behaviour by Visualising and Clustering Movement Data

Gavin McArdle[1], Urška Demšar[2], Stefan van der Spek[3], and Seán McLoone[4]

[1] National Centre for Geocomputation, National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland
gavin.mcardle@nuim.ie
[2] Centre for GeoInformatics, School for Geography and Geosciences, University of St Andrews, Fife, Scotland, UK
urska.demsar@st-andrews.ac.uk
[3] Department of Urbanism, Faculty of Architecture, Delft University of Technology, Delft, the Netherlands
s.c.vanderSpek@tudelft.nl
[4] Department of Electronic Engineering, National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland
sean.mcloone@eeng.nuim.ie

Abstract. Recent technological advances have increased the quantity of movement data being recorded. While valuable knowledge can be gained by analysing such data, its sheer volume creates challenges. Geovisual analytics, which helps the human cognition process by using tools to reason about data, offers powerful techniques to resolve these challenges. This paper introduces such a geovisual analytics environment for exploring movement trajectories, which provides visualisation interfaces, based on the classic space-time cube. Additionally, a new approach, using the mathematical description of motion within a space-time cube, is used to determine the similarity of trajectories and forms the basis for clustering them. These techniques were used to analyse pedestrian movement. The results reveal interesting and useful spatiotemporal patterns and clusters of pedestrians exhibiting similar behaviour.

Keywords: Geovisual Analysis, Clustering, Space-time Cube, Movement Data Analysis.

1 Introduction

Movement of individual objects such as animals, humans, human-operated objects or complex natural phenomena represents one of the main processes in the physical environment. For the purposes of data representation and modeling, such moving objects are considered as entities, whose position and/or geometric attributes change over time [1]. Despite the diversity of sources of movement data, fundamentally the same process is captured: how objects move through the basic framework of the physical world, defined by geographic space and time.

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 64–81, 2013.
© Springer-Verlag Berlin Heidelberg 2013

While the ubiquitous nature of location technologies has created issues for analysing movement data, specifically for pattern recognition and path prediction [2], it also provides an opportunity to understand how moving objects interact with each other and their environment. Due to interest in understanding human spatial behaviour, pedestrian mobility has long been an active research area in location based services [3], emergency management [4], determining infrastructural requirements [5] and urban planning [6]. Within urban planning and design, the vitality of a city is not generally measured using spatial analysis but by quantifying the number of people visiting the city and the economic value that brings. However, understanding the spatial and temporal aspects of the city, for example, the time spent and routes used by pedestrians can enhance a city's vitality. Current techniques in urban analysis, map the city based on its morphology, but questions about how the city network is used still remain. By analysing the activity patterns of individual pedestrians, repetitive behaviour and similar goals can be identified. This information can be translated into designs which address the requirements of the population, which is a key goal in urban design [6], [7].

This paper introduces a geovisual analytics tool for trajectory exploration, which contains novel techniques to help analysts comprehend movement datasets by visually representing the temporal and spatial aspects of movement simultaneously. The tool is applicable to a wide array of end users interested in interpreting movement data from disparate sources. Details such as principal routes and frequent stopping locations can be determined using the Space-time Cube visualisation which forms a central element of the tool. A new approach for identifying clusters of similar behaviour based on their speed and acceleration in orthogonal directions, also forms part of this research. A case study, using pedestrian data collected in the city of Delft in the Netherlands demonstrates these techniques.

This paper is structured as follows; the next section provides some related work in the area of movement data visualisation and analysis. Section 3 presents the novel visualisation and analysis techniques used to cluster movement data. A case study of pedestrian data, which reveals interesting spatiotemporal patterns and classifies pedestrians based on behaviour, is presented in Section 4. The paper concludes with a discussion and future work in Sections 5 and 6.

2 Visualisation and Analysis of Trajectory Data

Driven by the increase in the quantity of movement data, this section examines questions regarding techniques to represent, interpret, analyse and compare movement data.

2.1 Trajectory Representation and Visualisation

The path of a moving object is usually represented as a trajectory which is a sequence of positions in a two-dimensional (2D) geographic environment with

time stamps [8], i.e. $T = (x_1, y_1, t_1), ..., (x_n, y_n, t_n)$ for some n, such that (x_i, y_i) is the measured geographic location of the moving object at time t_i. Often trajectories have associated attribute data, which can be static, if the attribute has the same value for the object regardless of its position, (e.g. object type) or dynamic, if the attribute changes over time (e.g. attributes which describe physical properties of movement, such as velocity and acceleration). Other interpretations conceptualise movement as a sequence of moves and stops [9], or activity chains [10]. A taxonomy for describing common movement patterns such as convergence, divergence and concurrence in space and time as well as behavioural patterns including pursuit, play and flocking activities which can be used to classify trajectories, has also been proposed [11].

For visual analysis, the complete trajectory can be represented on a 2D map, however, this removes the temporal aspect, as the sequence in which locations were visited is removed. While animation can be included, it is not possible to interpret the entire spatiotemporal extent of trajectories using this technique. A series of separate maps representing distinct time periods can show the progression of moving entities [12]. This so-called situation-oriented point-of-view offers a discrete rather than a continuous view of the movement and contrasts with a trajectory-oriented point of view which shows each trajectory throughout its life cycle. Trajectory-oriented visualisations are valuable when studying movement patterns, however in order to visually analyse the temporal patterns of trajectories, a technique capable of representing time is required.

The Space-time Cube (STC), which originates from Time Geography, is a visual technique for representing a trajectory-oriented point of view, incorporating the temporal elements of trajectories [13], [14]. In a STC, the geographic coordinates appear on the x-axis and y-axis, while time is represented along the orthogonal z-axis. Time increases along this axis so that more recently occurring events appear above older events. Using this approach, a trajectory can be plotted as a three-dimensional (3D) polyline in the STC. While the STC is not a new concept, due to technological advances it has recently gained interest among the geovisual analytics research community for visual exploration of trajectories [15], [16]. A STC is suitable for simultaneously visualising a relatively low number of trajectories. When that number increases, the STC suffers from occlusion and cluttering, making it difficult to identify trends [17]. Usually aggregation, for example, using kernel densities [18] or generating clusters of the most salient patterns [19] can resolve this. We extend this idea with a new approach for clustering trajectories based on the projection of trajectories on the sides of the STC, described in Section 3.

2.2 Trajectory Similarity

One of the most important tasks for trajectory analysis is the identification of similar trajectories to reveal significant patterns. For example, it is used in animal ecology to estimate migration routes [20] and in meteorology to predict hurricane movement [21]. In urban planning, planners are interested in spatiotemporal

patterns of pedestrians which requires using a large set of individual movement data over a long period.

Methods for identifying groups of similar trajectories are mostly based on their geometric similarity in 2D geographic space. Examples use the Euclidean distance [22], the Hausdorff Metric [23] and the Fréchet Distance Metric [24]. Dynamic Time Warping [25] considers the temporal aspect by stretching the time axis in order to identify similarities in trajectory shape. This allows a comparison of trajectories which span different periods and spatial regions. Spatial transformations can also realign trajectories for better comparison. The Longest Common Substring (LCSS) approach does not consider the entire trajectory but instead finds similarities between substrings [26]. Edit Distance on Real Sequence (EDR), which measures the number of operations (insert, delete or replace) required to transform one trajectory to another, extends this by assigning penalties to the gaps between two matched sub-trajectories [27]. Similarity is also measured by examining the places and locations where stops occur [9]. Additional attributes of movement (speed, acceleration, duration and direction) are also used to determine similarity [1].

The research presented in this article follows this trend. We augment a geovisual environment, based on the STC, with a new and effective technique for identifying salient clusters of trajectories. This novel approach relies on the physical decomposition of movement in orthogonal directions. Each trajectory is then described in terms of speed and acceleration at each point in each direction before similarity measures and clustering are applied. These techniques, as well as a description of the geovisual analysis environment are discussed in the next section.

3 Trajectory Exploration Environment

In the environment introduced here, two visualisation components are used to analyse movement data. The first element visualises trajectories in a STC where a virtual globe represents the spatial aspect and the second component plots trajectory paths decomposed into latitude and longitude to enable comparison. Additionally, powerful analysis tools cluster trajectories based on a novel similarity metric described in Section 3.3.

3.1 Web-Based Space-Time Cube

The principal visualisation component is a STC draped over a virtual globe (Google Earth), which provides a pseudo 3D view of the Earth. The third dimension represents the terrain of the underlying landscape. As demonstrated in figure 1, we manipulate the z-axis to represent time by draping a STC over the elevation surface (while the z-axis is a sum of elevation and trajectory time stamp, this can be changed). The trajectories therefore appear as lines floating above the surface of the Earth. The further from the surface a point on the trajectory appears, the more recent the observation. Time can be represented

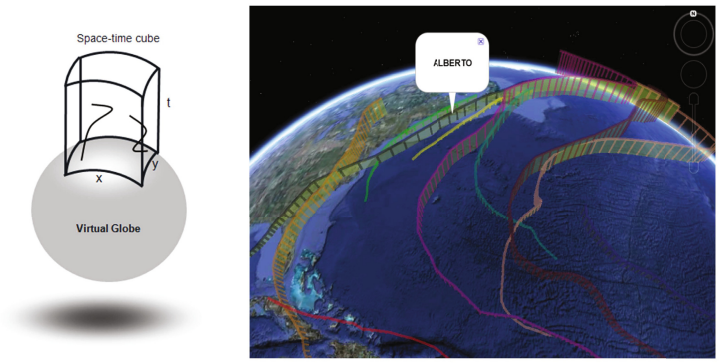


Fig. 1. (a) A STC containing trajectories draped over a virtual globe. (b) A STC containing hurricane trajectories shown in Google Earth

using different paradigms: each trajectory can start at a position relative to its time stamp in the overall dataset or each trajectory can begin their path at the Earth's surface.

Figure 1 also shows the visualisation interface. The bridge-like structures visible on the surface of the Earth represent trajectories. The points, connecting to the ground, along the trajectory are the points where the entity's location was recorded. Colours distinguish individual trajectories. The user can change the viewing angle, zoom level, interact with and identify spatial content. Using Google, the analyst can take advantage of Google Street View and Google Maps to obtain additional spatial information about the study area.

Through visual exploration, the analyst can obtain new information about the trajectories being studied. For example, it is possible to gauge the speed of entities (using the slope of the 3D polylines representing trajectories), the sequence in which locations are visited (by examining the position of the polyline of the z-axis relative to the Earth's surface) and compare the progress of entities using the same route (by examining the distance from the Earth's surface of the polylines representing the entities). Additionally, stopping positions can be determined by identifying areas with a large number of recordings in which the z-axis position of the polyline changes, indicating time has progressed but the entity's spatial position has not altered.

3.2 Physical Decomposition of Movement in Space-Time Cube

As described above, the path of a moving object in the STC is approximated as a series of straight-line segments between sampled points (x_i, y_i, t_i) and $(x_{i+1}, y_{i+1}, t_{i+1})$ [8]. Each trajectory is therefore represented by a continuous 3D polyline. This is combined with a principle stemming from classical mechanics: the

physical description of motion in an inertial frame of reference. In our context, we are interested in the mathematical description of this motion, which is normally done via vector analysis in the inertial frame of reference [28]. In classical mechanics, the inertial frame of reference is represented by a 3D Cartesian coordinate system located at an arbitrary starting point in space (denoted by 0) in which the object, that is not under effect of any force, either appears to be at rest or in a state of uniform motion in a straight line. The position of the object is described with a position vector (x, y, z) stretching from the starting point 0 to the object. This vector is a function of time, i.e. $(x(t), y(t), z(t))$. This vector and corresponding vectors of velocity and acceleration are decomposed into perpendicular directions of the coordinate system axes by projections onto relevant planes. Calculations describing various properties of movement are then completed separately for each plane defined by two of the axes.

We borrow this concept of decomposition and adapt it for our STC. The inertial frame of reference is replaced with the STC. We project each trajectory onto the x-t and y-t planes. These two projections are represented as two separate graphs that show how each object moves in one of the two geographic directions (figure 2). This approach is similar to the projection of time series data on to arbitrary planes developed within the GeoTime community [15]. These projected graphs can be used to visually investigate similarities of movement in each coordinate direction before applying the clustering techniques described below.

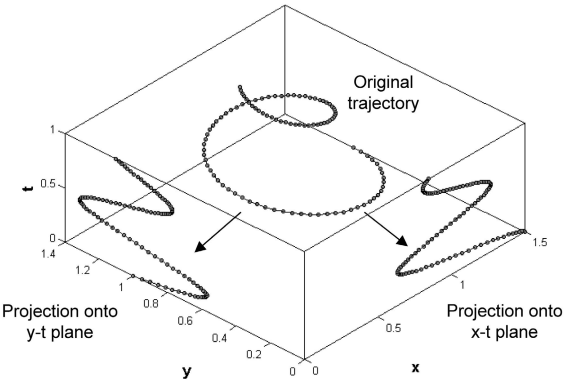


Fig. 2. Trajectories in a STC decomposed and projected on the x-t and y-t planes

3.3 Trajectory Comparison

To investigate similarity of trajectories we adopt a mathematical approach that looks at the shape of each projected curve. This can be done separately for each projection or by considering both projections simultaneously. Shape similarity is calculated based on the slope and rate of change of slope of a mathematical

function in a graph. Similar approaches are used for time series [29], however there, only the slope of time series is utilised, while we also add the rate of change in slope for a more detailed approach.

The first step in obtaining the required dataset is to invert the axes in each projection graph. This is done because the projected curves, as they are in a STC do not necessarily fulfil the criteria to form mathematical functions of a single variable, where at each value of the independent variable there should be at most one value of the dependent variable. Time cannot be given as a function of a single x or y coordinate as an object can visit the same location at different times. The inversion resolves this and is also logical from the conceptual sense of movement through time, since every object is always moving in a linear and unidirectional motion along this dimension. Similarity of two curves $x = f_1(t)$ and $x = f_2(t)$ in the t-x plane is then based on their slopes and rate of change in slope. Mathematically, the slope of a function $x = f(t)$ at a certain point t is represented as the slope of the tangent to f at that point, which is calculated as the first derivative at that point, i.e. slope $S = df(t)/dt$. Since we derive over time, in physical terms this is equal to velocity of movement in x direction. Rate of change in slope is calculated as the second derivative of the function, i.e. change in slope $C = df^2(t)/dt^2$. In physical terms, this is acceleration in x direction.

The shapes of two curves are more similar the more the slopes of the two curves (the derivatives) and the rate of change in slopes of the curves (the second derivatives) are similar to each other at each point t. Based on this principle, curves can be grouped into groups of similarly-shaped curves using the dataset of the first and second derivatives over time at all sampling times. Note, it is possible to compare either direction only (x or y) or both (x and y) in terms of first and second derivatives. For example, if the original trajectories are sampled from entities moving in a block-like city centre (e.g. Manhattan), then looking at each coordinate axis makes sense. If however objects move in a more unconstrained manner (e.g. hurricanes), then a joint dataset of derivatives in both x and y directions is a better option.

This principle does not consider the actual values on the curves (i.e. particular locations). The result is independent of the location of the moving objects. Furthermore, as the approach normalises time over each trajectory, it is not necessary for the trajectories being compared to be from the same period. The technique is effective with timeless trajectories from disjoint spatial regions. The results of the procedure are therefore groups of trajectories that show similar behaviour, regardless of where and when the trajectories are positioned. This allows the technique to be applied to both constrained and free movement.

3.4 Spectral Clustering

Clustering, which involves identifying similarities between data points and using this as a basis to group them [30] is performed on the decomposed trajectories. Spectral Clustering [31], specifically the technique developed by Chen et al. [32]

is used in the approach described here. Spectral Clustering has been effective for trajectory analysis [33], [34] however; the similarity measure was based on trajectory location, rather than movement behaviour, as in our case.

Spectral Clustering is a graph-based clustering technique. The graph is constructed by calculating a similarity matrix $S = s[ij]$ between all the data points in attribute space. We determine similarity using Euclidean distance between the first and second derivatives, as described in the previous section. Although trajectories on the surface of the earth are a non-Euclidean phenomenon, our approach compares the similarity between the values of speed and acceleration rather than geogrpahic positions. An adjacency matrix of the similarity graph, $G = (V, E)$ is then produced, where V is the set of all data points, $x_1 ... x_n$. Two vertices in the graph are connected only if the similarity between them (s_{ij}), is larger than a given threshold value. With this approach, the complexity of original similarity matrix is reduced (making it sparse), as only significant similarities are considered. Clustering now involves partitioning the similarity graph into k groups which can be determined through domain knowledge or analysis. Clusters are formed so that the edges between groups have a low similarity score while the edges within groups have a high similarity score. With Spectral Clustering, clusters are based on local similarity in the attribute space making this method faster. Additionally it can detect clusters that simple k-means would not recognise.

The groups resulting from the clustering process are then visualised. Firstly, the trajectories, which form each group, are plotted. The resulting graphs can explain the groupings and assist the analyst to apply expertise to classify entities based on the spatiotemporal movement patterns. The information is linked to the STC visualisation where individual clusters can be displayed for further visual analysis. The next section presents a case study which effectively applies these techniques to a real dataset.

4 Case Study

This section presents how the above approach is used to analyse pedestrian movement in the city of Delft, the Netherlands. Emphasis is placed on how the STC, summary plots, and associated analytics are used to infer knowledge and interpret the data. Data were collected over a 4 day period by distributing GPS receivers to pedestrians at car-park facilities in Delft. The receivers recorded the location of participants every 5 seconds. From the origin location, each participant made one journey on foot, represented by a single trajectory, from the origin, including stops, back to the origin which marks the end of a trip. Therefore, a trip can have multiple destinations and activities. The researchers who collected the data are interested in routes, destinations and sequences. This section demonstrates how temporal geovisualisation and clustering can answer these questions.

4.1 Route Analysis

A cleaned and validated set of trajectories representing the paths of 113 pedestrians was used to create the initial visualisation (figure 3) which highlights the spatiotemporal extent of the pedestrians. Each trip starts at ground level (time 0).

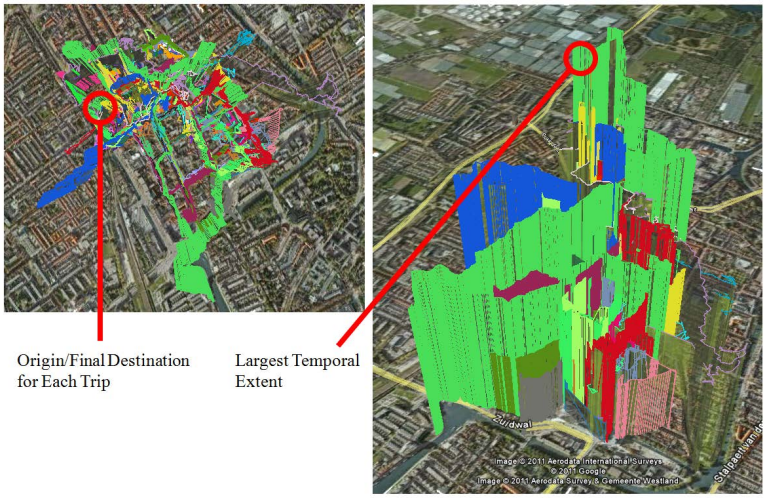


Fig. 3. All trajectories viewed from above and from the side in the STC showing the temporal and spatial elements of trajectories simultaneously

Through interaction, common spatial routes can be identified, for example, the majority of pedestrians travel east, while very few cross the barrier caused by railway tracks, west of the origin. Temporal analysis, achieved by examining the height of the trajectories on the time axis, shows that pedestrians generally travel in an anti-clockwise direction, visiting areas to the south of the origin first before regions, which are parallel with, or north of the origin. Domain knowledge indicates that pedestrians therefore visit chain stores before visiting boutiques.

The inclusion of temporal aspects of trajectories in the visualisation environment helps identify temporal co-existence which represents pedestrians who were travelling together. For example, trajectories which follow similar spatial and temporal routes are likely to be related. Examining figure 4, we can see two trajectories which initially follow this pattern. Midway through the trip, there is a split; one pedestrian continues walking while the other pedestrian stops at a single location. By examining the spatial content at the stop location, also shown in figure 4, it appears to be an office; a long stop here could indicate a meeting or an appointment. The ability to detect the spatial feature in this way demonstrates the power of the approach using Google Earth. Later, both

trajectories exhibit a similar pattern again and return to the origin at a same time. Such behaviour seems to suggest that these pedestrians are travelling together and gives an indication of the social connections in the city.

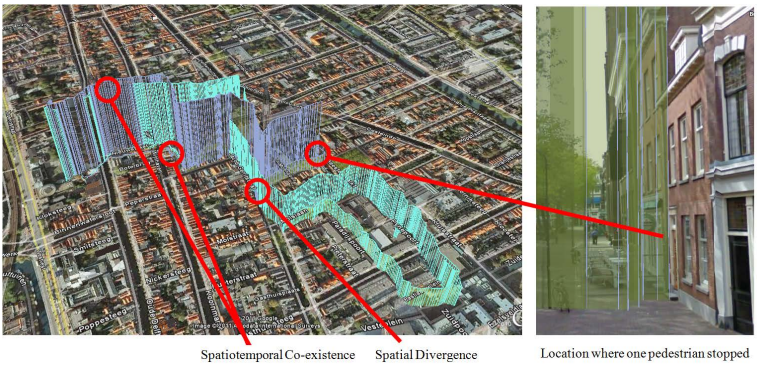


Fig. 4. The STC showing two trajectories exhibiting spatiotemporal coexistence for a portion of their trips

Common locations where people stop provide evidence of salient areas [35]. Stopping locations are determined by the frequency of observations at the same location (based on a temporal threshold). Figure 5 shows all the stopping locations of pedestrians in this case study. Notable stopping locations are the two town squares where there are cafés and restaurants. The height of the rectangular prism indicates the duration of a stop which can distinguish between a stop at a restaurant and a visit to a shop perhaps. Multiple stopping locations on a particular route are significant. Figure 5 identifies the main shopping district based on the frequency and duration of stops which are indicative of people entering and leaving stores.

4.2 Cluster Analysis

Interactive graphs show the decomposed trajectories and are used to identify similarity among trajectories in each direction. As described in section 3.4, clustering is applied to the trajectories described in terms of speed and acceleration at each observation point, in each direction. Visual analysis suggested that 6 clusters produce descriptive results without overlap or gaps.

Figures 6 - 11 show the graphs describing these clusters. Each graph has the same temporal extent to exemplify the differences between shorter and longer trips in each cluster. Time is represented on the x-axis while the location is shown on the y-axis. Each cluster is represented by two graphs (latitude and longitude), however for the purpose of presentation; only one arbitrary graph for each cluster is shown here. Examining the graphs, it is clear to see how the temporal aspects of

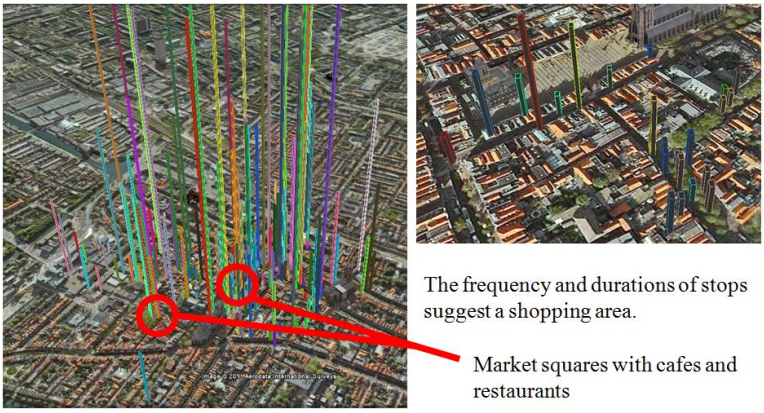


Fig. 5. The STC showing all identified stopping locations and the identification of a shopping area based on the duration and frequency of stops

movement (duration, speed and acceleration) effect the clustering. The number of stops detected also plays an important role in classifying trajectories.

When interpreting the graphs, the straight lines and high oscillating areas on a trajectory represent stop locations (oscillation occurs when a GPS receiver is stationary or the signal is lost when entering a building), while the smooth areas represent movement (figure 6). Using this knowledge, it is possible to describe the clusters in terms of movement patterns in Table 1. The trajectories, which form each cluster, can be visualised independently in the STC in order to gain further insight into pedestrian behaviour and determine if a spatial correlation also exists. For example, when the trajectories and stop locations for cluster 4, which contains a mix of medium and short length stops, is visualised using the STC, we can see pedestrians generally have their longest stop midway through their trip. While the trajectories within a cluster can be aggregated to find the most common trend or patterns, the geographic context is not considered during clustering and so visualising this aggregation in the STC is not necessarily beneficial.

5 Discussion

Although the approach is an effective visual analysis tool, there are several limitations which should be considered when interpreting the results. This section describes these limitations and indicates how future work can resolve many of them. One of the powers of Google Earth is the digital terrain model which shows the topography of the landscape. When analysing trajectories using the approach described above, analysts must be cognisant of the fact that the height of the trajectories is relative to the topography of the landscape. Analysts must examine the distance between the terrain (the Earth's surface) and the height of the trajectory at a given location to understand its temporal aspects. To

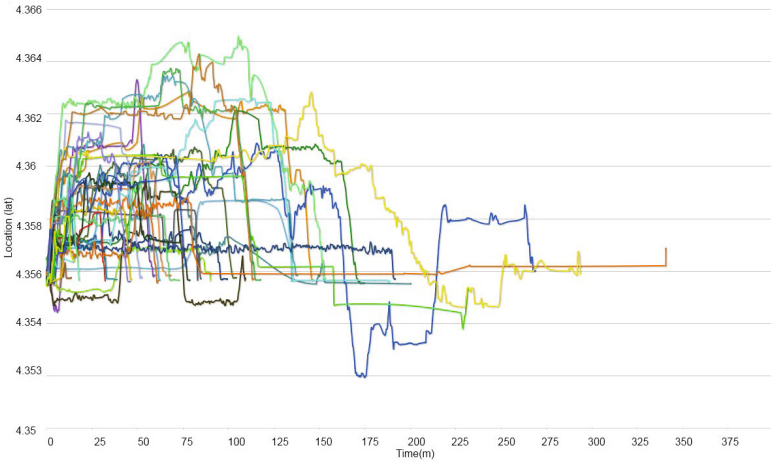


Fig. 6. Graph representing the trajectories in the longitude direction of cluster 1. The cluster represents long trips with few stops.

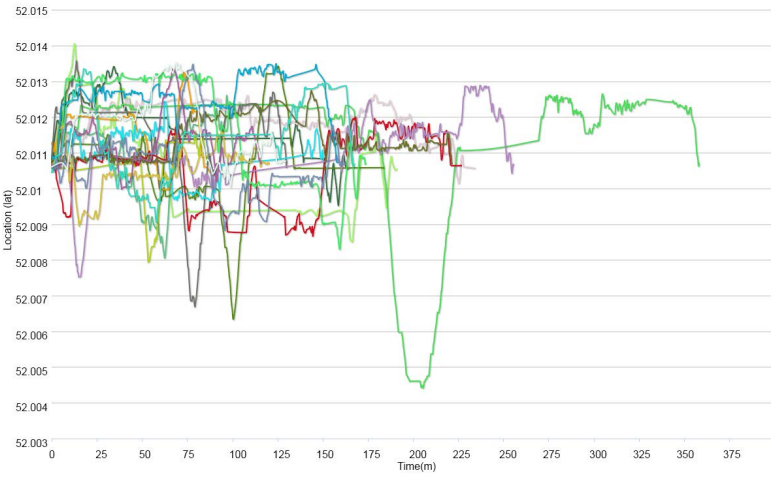


Fig. 7. Graph representing the trajectories in the latitude direction of cluster 2. The cluster represents long trips with many stops.

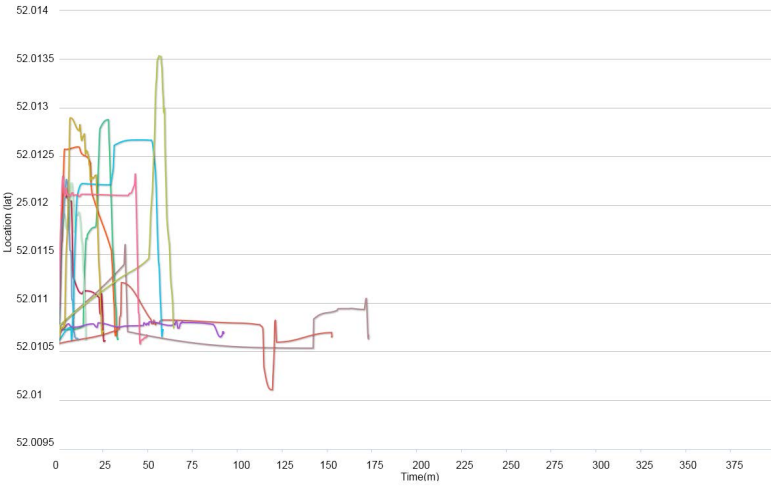


Fig. 8. Graph representing the trajectories in the latitude direction of cluster 3. The cluster represents short trips with no stops.

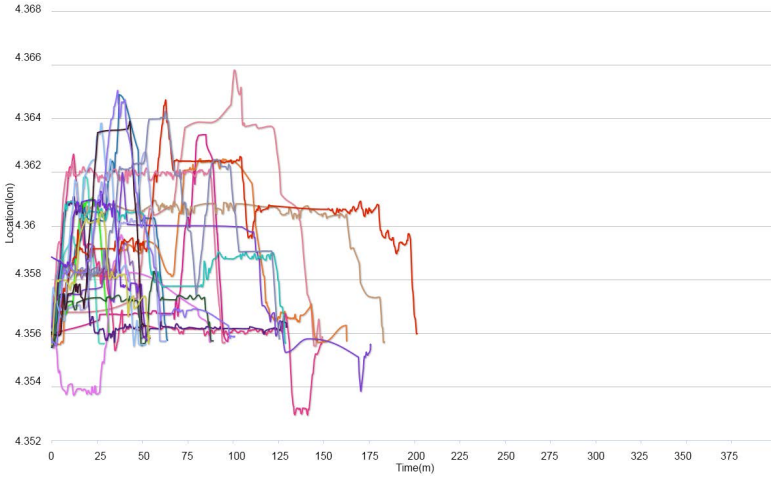


Fig. 9. Graph representing the trajectories in the longitude direction of cluster 4. The cluster represents medium length trips.

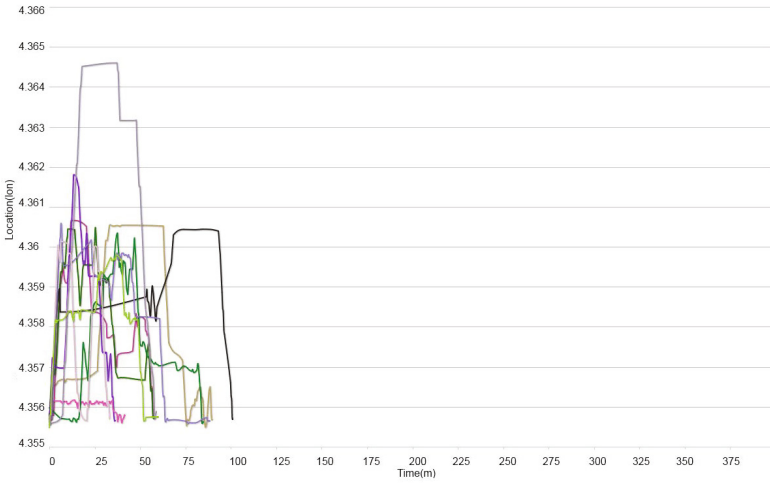


Fig. 10. Graph representing the trajectories in the longitude direction of cluster 5. The cluster represents short trips with one stop.

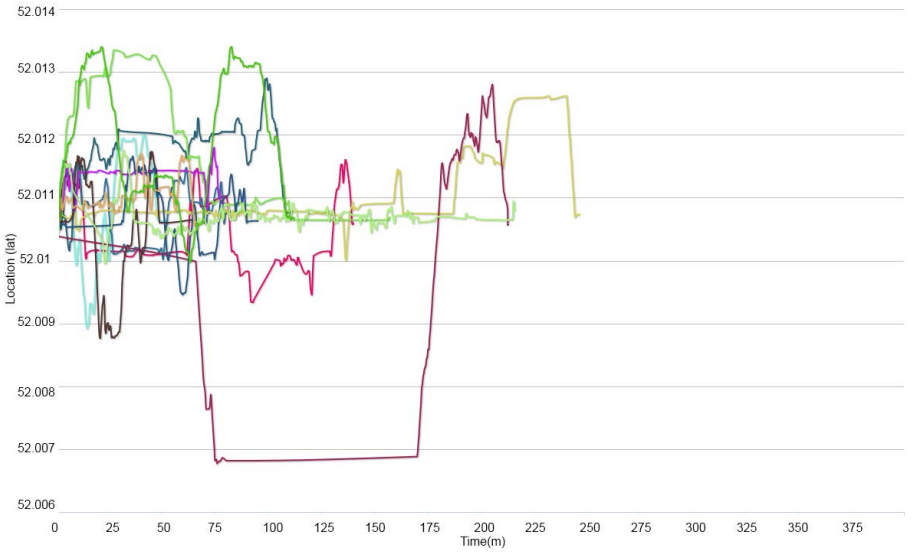


Fig. 11. Graph representing the trajectories in the latitude direction of cluster 6. The cluster represents trips with one long stop.

Table 1. Summary of the clusters produced, along with a potential explanation of the behaviour detected

Figure No.	Cluster No.	Trip Duration	Number of Stops/ Stop Duration	Potential Explanation
6	1	Long	Few/Long Stops	Having lunch, a coffee or attending a meeting
7	2	Medium-Long	Many/Short Stops	Shopping
8	3	Short	No Stops	Wandering around the city for brief window shopping
9	4	Medium	Many/Mix of short and medium stops	Shopping combined with lunch/coffee
10	5	Short	One/Short	Specific purpose for trip - shopping
11	6	Medium	One/Long	Specific purpose for trip - meeting or lunch

aid with this, a shaded area between the Earth's surface and the trajectory has been included. The path of the trajectory may also be rendered along the Earth's contours to help interpret its relative height. In the case study described in this paper, we consider trajectories in the city of Delft in the Netherlands which is a flat geographical region and so this issue is not apparent.

Geovisual Analysis relies on effective Human-Computer Interaction (HCI) and so usability testing, with instruments such as the IBM Computer System Usability Questionnaire [36] or the After Scenario Questionnaire [37], is required. Such analysis ensures that the STC and accompanying graphs are interpreted correctly by users. As the tool is applicable to a wide range of end-users (e.g. urban planners in the case study presented here), some initial training may be required for users to become proficient in interpreting the STC which contains certain nuances as outlined above.

The clustering approach described here projects each trajectory into its latitude and longitude components to visually identify similarity in movement patterns. While this may be considered an arbitrary choice, it is based on the fact that this is how movement is generally recorded and described. While trajectories can be described by many different attributes, in this study, we are interested in the behaviour of moving objects which may not be in the same spatial area. Velocity and acceleration are therefore good proxies for this form of behaviour as stops and moves can be used to gauge similarity in spatially disjoint regions.

The clustering technique produced six clusters. Assessing the validity of the cluster assignments using external validation such as a domain expert is favourable. While the pedestrians in the case study were asked if they were going shopping or visiting tourist locations, this information did not translate to the richer data revealed in the clusters. In future studies, questions can be tailored to match cluster outputs. The results can also be assessed with internal validation metrics, in which there should be high similarity within clusters and low similarity between clusters. It would be beneficial to compare the cluster

results with more established approaches for clustering trajectories discussed in Section 2.2.

6 Conclusion

Movement is a fundamental aspect of the physical world. Studying movement data and recognising patterns can be utilised in many domains to predict future movement, assess how spaces are used and identify unusual behaviour by identifying trajectories not adhering to particular clusters. This article introduces a powerful geovisual environment for exploring movement data. The concept of a classic STC is extended and used in conjunction with Google Earth so that the temporal and spatial aspects of trajectories can be analysed simultaneously. The approach adheres to the classic visual information seeking mantra by providing an 'Overview first', followed by 'Zoom and filter', and 'Details-on-demand' paradigm [38].

Similarity among trajectories is determined by describing them in terms of the speed and acceleration at each observation point. Euclidean distance in the attribute space of first and second derivatives then determines similarity before Spectral Clustering is applied to produce groups. The techniques are particularly valuable for detecting trajectories which have a similar duration and contain a similar number of stops but are not necessarily in the same geographic location.

The approach was tested by exploring pedestrian trajectories in a city. While pedestrian movement and city life have been investigated before, existing approaches are generally based on counting and local observations without insight into complete activity patterns [39]. Knowing more about how people use the city, not only as individuals in a specific situation, but also translated to general situations based on the clusters of pedestrians produced, interventions to improve the city can be proposed, for example by identifying barriers that cause an imbalance between frequently visited and neglected places.

In the case study, the clustering of pedestrian trajectories results in 6 groups of pedestrians. While the descriptions of the clusters are indicative of activities undertaken by pedestrians in a city centre, the ground truth regarding the actual activities and intentions of those pedestrians would be beneficial to validate the clusters. This will be carried out in a future study and the results will be used to enhance the clustering technique.

Acknowledgements. Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) by Science Foundation Ireland under the National Development Plan. We gratefully acknowledge this support.

References

1. Dodge, S., Weibel, R., Forootan, E.: Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. Computers, Environment and Urban Systems 33(6), 419–434 (2009)

2. Orlando, S., Orsini, R., Raffaetà, A., Roncato, A., Silvestri, C.: Trajectory data warehouses: design and implementation issues. Journal of Computing Science and Engineering 1(2), 240–261 (2007)
3. Millonig, A., Gartner, G.: Identifying motion and interest patterns of shoppers for developing personalised wayfinding tools. Journal of Location Based Services 5(1), 3–21 (2011)
4. Zheng, X., Zhong, T., Liu, M.: Modeling crowd evacuation of a building based on seven methodological approaches. Building and Environment 44(3), 437–445 (2009)
5. Hoogendoorn, S.P., Bovy, P.H.L.: Pedestrian travel behavior modeling. Networks and Spatial Economics 5(2), 193–216 (2005)
6. Spek, S.: Tracking tourists in historic city centres. In: Information and Communication Technologies in Tourism 2010, pp. 185–196 (2010)
7. Van Schaick, J.: Timespace Matters-Exploring the Gap Between Knowing About Activity Patterns of People and Knowing How to Design and Plan Urban Areas and Regions (2011)
8. Laube, P., Imfeld, S., Weibel, R.: Discovering relative motion patterns in groups of moving point objects. International Journal of Geographical Information Science 19(6), 639–668 (2005)
9. Ashbrook, D., Starner, T.: Using gps to learn significant locations and predict movement across multiple users. Personal and Ubiquitous Computing 7(5), 275–286 (2003)
10. Wilson, C.: Activity patterns in space and time: calculating representative hagerstrand trajectories. Transportation 35(4), 485–499 (2008)
11. Dodge, S., Weibel, R., Lautenschütz, A.K.: Towards a taxonomy of movement patterns. Information Visualization 7(3-4), 240–252 (2008)
12. Andrienko, G., Andrienko, N.: Spatio-temporal aggregation for visual analysis of movements. In: IEEE Symposium on Visual Analytics Science and Technology, VAST 2008, pp. 51–58. IEEE (2008)
13. Hägerstrand, T.: What about people in regional science? Papers in Regional Science 24(1), 6–21 (1970)
14. Kraak, M.J.: The space-time cube revisited from a geovisualization perspective. In: Proc. 21st International Cartographic Conference, pp. 1988–1996 (2003)
15. Kapler, T., Wright, W.: Geotime information visualization. Information Visualization 4(2), 136–146 (2005)
16. Kraak, M.J.: Geovisualization and time–new opportunities for the space–time cube. In: Geographic Visualization: Concepts, Tools and Applications, pp. 293–306 (2008)
17. Andrienko, G., Andrienko, N., Wrobel, S.: Visual analytics tools for analysis of movement data. ACM SIGKDD Explorations Newsletter 9(2), 38–46 (2007)
18. Demšar, U., Virrantaus, K.: Space–time density of trajectories: exploring spatio-temporal patterns in movement data. International Journal of Geographical Information Science 24(10), 1527–1542 (2010)
19. Andrienko, G., Andrienko, N.: Poster: Dynamic time transformation for interpreting clusters of trajectories with space-time cube. In: Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST) 2010, pp. 213–214 (2010)
20. Horne, J.S., Garton, E.O., Krone, S.M., Lewis, J.S.: Analyzing animal movements using brownian bridges. Ecology 88(9), 2354–2363 (2007)
21. Slingsby, A., Strachan, J., Vidale, P.L., Dykes, J., Wood, J.: Discovery exhibition: Making hurricane track data accessible. Discovery Exhibition Entry (2010)

22. Frentzos, E., Gratsias, K., Pelekis, N., Theodoridis, Y.: Algorithms for nearest neighbor search on moving object trajectories. Geoinformatica 11(2), 159–193 (2007)
23. Zhang, Z., Huang, K., Tan, T.: Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In: 18th International Conference on Pattern Recognition, ICPR 2006, vol. 3, pp. 1135–1138. IEEE (2006)
24. Buchin, K., Buchin, M., Wenk, C.: Computing the fréchet distance between simple polygons in polynomial time. In: Proceedings of the Twenty-Second Annual Symposium on Computational Geometry, pp. 80–87. ACM (2006)
25. Sakurai, Y., Yoshikawa, M., Faloutsos, C.: Ftw: fast similarity search under the time warping distance. In: Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 326–337. ACM (2005)
26. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In: Proceedings of the 18th International Conference on Data Engineering, pp. 673–684. IEEE (2002)
27. Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 491–502. ACM (2005)
28. Halliday, D., Resnick, R., Walker, J.: Fundamentals of Physics. Wiley (1997)
29. Altiparmak, F., Ferhatosmanoglu, H., Erdal, S., Trost, D.C.: Information mining over heterogeneous and high-dimensional time-series data in clinical trials databases. IEEE Transactions on Information Technology in Biomedicine 10(2), 254–263 (2006)
30. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM computing surveys (CSUR) 31(3), 264–323 (1999)
31. Song, Y., Chen, W.-Y., Bai, H., Lin, C.-J., Chang, E.Y.: Parallel Spectral Clustering. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 374–389. Springer, Heidelberg (2008)
32. Chen, W.Y., Song, Y., Bai, H., Lin, C.J., Chang, E.Y.: Parallel spectral clustering in distributed systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(3), 568–586 (2011)
33. Atev, S., Miller, G., Papanikolopoulos, N.P.: Clustering of vehicle trajectories. IEEE Transactions on Intelligent Transportation Systems 11(3), 647–657 (2010)
34. Morris, B., Trivedi, M.: Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 312–319. IEEE (2009)
35. Palma, A.T., Bogorny, V., Kuijpers, B., Alvares, L.O.: A clustering-based approach for discovering interesting places in trajectories. In: Proceedings of the 2008 ACM Symposium on Applied Computing, pp. 863–868. ACM (2008)
36. Lewis, J.R.: Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. International Journal of Human-Computer Interaction 7(1), 57–78 (1995)
37. Lewis, J.R.: Psychometric evaluation of an after-scenario questionnaire for computer usability studies: the asq. ACM SIGCHI Bulletin 23(1), 78–81 (1991)
38. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: Proceedings of the IEEE Symposium on Visual Languages, pp. 336–343. IEEE (1996)
39. Van Schaick, J., Van Der Spek, S.C.: Urbanism on Track: Application of Tracking Technologies in Urbanism, vol. 1. Ios Press Inc. (2008)

A Multi-modal Communication Approach to Describing the Surroundings to Mobile Users

Janne Kovanen, Tapani Sarjakoski, and L. Tiina Sarjakoski

Finnish Geodetic Institute, Department of Geoinformatics and Cartography,
P.O. Box 15, 02431 Masala, Finland
{janne.kovanen,tapani.sarjakoski,tiina.sarjakoski}@fgi.fi

Abstract. Mobile users frequently pass non-obvious features that could be represented to the user in a multi-modal manner. This type of information can be used to affect the decision making of the user or to complement his or her navigation experience. However, data providers do not have a common data interchange schema for describing geographical features multi-modally. This paper presents a multi-modal approach by extending the GeoJSON, GML, and KML formats to describe the surroundings of a mobile user in a Location-Based Service. In addition, the paper discusses how the approach can be implemented on a mobile client. Finally, the paper demonstrates how the proposal has been implemented with a functional prototype for a hiking use case.

Keywords: Multi-Modal Communication, Location-Based Service, Data Interchange Format, Mobile User.

1 Introduction

Traditionally, a user communicates with a map through a visual channel. The visual communication can incorporate varying elements, like textual content, textures, pictorial symbols, and other abstractions. Such information is well-suited to the visual sense, but people use other senses at the same time. A user might use one sense for the primary task, and others for secondary tasks. Hence, it is a natural trend for applications to provide information directed at multiple senses.

Multi-modally described data complements information extracted from the local surroundings, such as a visual overview of the environment, which a map application can provide. Other means, such as speech and pictures, can sometimes better explain natural phenomena or landforms. They also supplement information coming from human-made sources, like signposts or information boards. Similarly as with a printed map, this type of information can be used to affect the decision-making of the user or to simply cultivate his or her user experience.

Multi-modal descriptions can even work concurrently with routing guidance if the auditory or haptic communication does not overlap. They can either be delivered independently. or multi- modal descriptions can be incorporated within

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 82–99, 2013.
© Springer-Verlag Berlin Heidelberg 2013

the routing guidance. Consequently, if the route guidance advises a user to go around a dangerous area, the reasoning for such guidance can be explained to the user. Hence, the user is assured that the routing is based on logical deduction.

One barrier for multi-modal descriptions is the absence of a generic way to define how applications can distribute and represent data in a multi-modal manner. In this paper, we present a user-centered solution that complements the navigation experience in an outdoor environment. The features may include interesting phenomena, recreation alternatives, or dangers, all of which need to be communicated to the user in a multi-lingual and multi-modal way. First, the paper describes the use case of our study and related research. Next, we propose a common solution for how Web services can distribute multi-modally describable geographical data to other media, including mobile applications. The paper continues by presenting the phases relevant to the mobile applications that provide descriptions. Finally, following a discussion of the general phases, we provide an overview of how the approach has been realized in our prototype.

1.1 Use Case

This study was performed so that hikers were able to perceive information in several modalities. The requirements are based on a specific area – Nuuksio National Park, near Helsinki, the capital of Finland. The park is frequently visited by locals and tourists searching for a moment of peace and relaxation outside the urban area. The park contains several types of forest, lakes, and small hills typical of nature at this latitude. In addition, the park provides marked circle trails, footpaths for hikers, and recreational structures such as camping sites and fireplaces. The features that need to be described to the hiker include natural phenomena, recreational activities, opening hours, warnings, and geographical names, as well as other features that are difficult to interpret without help. The descriptions need to be multilingual to serve international visitors.

The natural environment is challenging for navigation performed using mobile devices because the terrain is sometimes hilly and the wooded areas are dense, which weakens the positioning accuracy of the devices. Similarly, mobile data connections are typically non-existent. Visitors with smartphones will most likely download applications and data before entering the park or at a visitor center. However, changing data may require that they will need to download information during the course of their hike.

2 Related Research

During the last decade, an increasing number of studies have focused on smartphones as a consequence of the tremendous increase in their usage and capabilities. Smartphones can be used to deliver information with several channels. For instance, a navigation application might point out the direction to follow using text, voice, visual arrows or a vibration pattern. One field of study where

the surrounding is described is pedestrian navigation. A part of research performed in this field has studied how different modalities affect spatial knowledge acquisition (e.g. [24], [2], and [28]).

Other navigation-related studies have concentrated on how multiple modalities: (1) help people with disabilities, (2) improve navigation performance, or (3) can be used together. For instance, Baus et al. [3] prepared textual route instructions containing auditory perceptible landmarks that they converted into synthesized audio records before their trial. They validated that auditory perceptible landmarks are helpful and may be used by visually impaired persons in a similar fashion as graphically represented landmarks are used by normally sighted users. Kainulainen et al. [21] also applied synthesized speech in their study by comparing it to non-spoken recognizable soundscapes (soundmarks) that are objectively unique to a place. In addition, they tested a mixture of the two by using one type of audio to one type of information. The main conclusion of their study was that users mostly preferred speech as such. However, soundscapes are not the only audio available. Other types of auditory cues representing real-world objects include, for example, earcons [4] and auditory icons [13].

Modalities helping navigation are not restricted to audio. For example, Chittaro and Burigat [8] have shown that photographs and abstract direction arrows are meaningful method of lowering the amount of navigation errors, and Pielot, Henze, and Boll [31] have reported that a tactile belt lowers the disorientation of the users and helps them to find shorter routes. Additional information accompanying the actual route guidance is also a way raise the comfort level of impaired users [17].

Route instructions, however, are typically not meant to contain any supplemental information regarding the surrounding that is not necessary in the guiding process. Instead, route instructions should be as short as possible to reduce the cognitive processing load required from a user; however, in some situations users are open to spontaneous suggestions about nearby interesting locations that lead them to deviate from their original route [32].

Another field where the surrounding is described to the user is festival, fair and museum tour guides which typically not only guide the user in the particular area, but include a disclosure of the environment. Hippie [29], HyperAudio [30], and GUIDE [7] are examples of the first wireless adapting museum guides that made use of the user's location and direction to give sound bites of the relevant objects in a similar fashion as a traditional curator might do to and guide to the next object. A couple of years later Ciavarella and Paternò [9] found out that foreign visitors especially appreciated even richer multimedia, such as video clips of the relevant objects, which traditional curators did not provide. Since then, smartphone museum guides have evolved even further and audio through headphones is not just meant to free the eyes to look at attractions; instead, multi-modality especially helps people with vision or hearing disabilities to visit museums or fairs autonomously. The study of Ghiani et al. [14], for example, shows that synthesized speech and vibrations are valid methods to guide blind visitors in a museum.

Similarly, video clips have been made to contain information nuggets in the form of sign language to improve the user experience of deaf visitors [38].

Most of the studies presented in this chapter or elsewhere do not mention how the descriptions are delivered to the mobile devices. A sophisticated guess is that most systems developed for research use an in-built test set or an application-specific data interchange format. Nevertheless, Richter [33] presents, as part of his paper, XML-fragments that contain context-specific route instructions based on landmarks, which can be externalized into meaningful spoken or textually represented route instructions. Similarly, some progress towards a common and open data interchange is available through the Open Geospatial Consortium (OGC) OpenLS [26], which can be used to deliver instructions for maneuvers that incorporate textual descriptions of surrounding features with a preferred language.

3 Interchanging Descriptions

A suitable open data transfer format for vector data constitutes a relevant part of our solution for enabling cross-modal descriptions. In this section, we describe how data interchange formats can be extended to support cross-modal descriptions of features. At first, we discuss some of the most common open formats. Then we describe the additions that we made to the original schemas.

3.1 Alternative Data Interchange Format Bases

For a feasible solution, we required from the format a high general acceptance at either the present moment or in the near future. In addition, we required that the following criteria be met:

1. that it support geometries such as polygons, linestrings, and points and the multiple instances in which they occur;
2. that the data size and serialization speed are suitable and that it is easy to use [35]; and
3. that it provide extendibility and scalability for the interoperability of Web services and mobile devices.

Good interoperability requires not just that a format can be implemented, but also that the context restrictions are taken into account. For example, deserializing a complex format in comparison to a simpler format generally consumes more battery power on the mobile device, and data with a large overhead consumes more data bandwidth – thereby, they typically take longer to download or upload. To be easy to use, a particular format should be human-readable.

We found that KML [39] and Geography Markup Language (GML) Simple Features Profile [36] are the best alternatives from Extensible Markup Language (XML) grammars; both of these alternatives are standardized by the OGC and defined by their own schema documents. The main difference between the formats is that GML is a grammar more directed to modeling and interchanging

geographical content, whereas KML focuses on representing geographical features from both visualization and viewing perspectives. As a result of the huge range of possibilities related to modeling, GML is not used as such; instead, an application schema that defines the grammar for a specific domain is used in its place. The application schema extends GML and makes sure that the data provider and user have a common understanding of the problem area.

GML-encoded data can also be rendered based on the rules provided by a styling language, such as the Styled Layer Descriptor (SLD) defined by the OGC. Alternatively, because both grammars use a similar geometry model, the content of the GML may be converted into KML for cartographic visualization purposes, even though some information might be lost in the process. The conversion can be implemented by first converting the GML-encoded coordinates into the WGS84 coordinate reference system. Next, the GML-encoded data is translated into KML using a particular solution, such as members of the Extensible Stylesheet Language (XSL) family. Finally, the symbology for each feature is included in the output.

The benefits of using XML include the number of data processing tools that have been developed for it and the possibilities that the structure provides. Nevertheless, XML-encoded data has, in theory, two main problems: it is time-consuming to parse, and it takes a relatively large amount of space, even if it is compressed. For example, a single KML document and the data referenced from that document, such as images or audio files, are typically transferred as a zip archive. In comparison to other formats, XML-related problems have even been empirically validated in contemporary mobile environments, like on the Android platform [34][35]. A formidable solution is to use a binary XML, which has been demonstrated to be significantly smaller in size and faster to parse [23]; however, it is neither human-readable nor widely accepted.

Another solution for overcoming the performance and size problems is to use GeoJSON [5], which is based on the JavaScript Object Notation (JSON). According to Crockford [10], "JSON's design goals were for it to be minimal, portable, textual, and a subset of JavaScript"; consequently, it is no surprise that JSON is typically more compact than XML, faster to parse, and recommended for mobile applications (e.g. [18]). However, JSON does not have an official schema associated with it, which has resulted in a situation where clients are forced to implement specific parsers for every domain. GeoJSON as such is a partial solution for overcoming the situation, but apart from its features and their geometries, it does not define rules for any additional characteristics, like cartographic visualization.

All three formats share a similar set of properties, such as being text-based and human-readable. The main difference between the formats is their audience and usage context. GML is typically used by professionals for Web service communication, like Web Feature Service (WFS) responses. KML, for its part, is used by professionals from diverse fields, but it is also favored by ordinary non-professional users mainly because the format is used by Google Earth and Google Maps API. In comparison to the two XML-based grammars, in its earlier

stages GeoJSON was mostly used in AJAX communication by Web applications because it can easily be serialized into JavaScript. Lately, however, the grammar has also become popular in mobile communications between Web services and mobile applications.

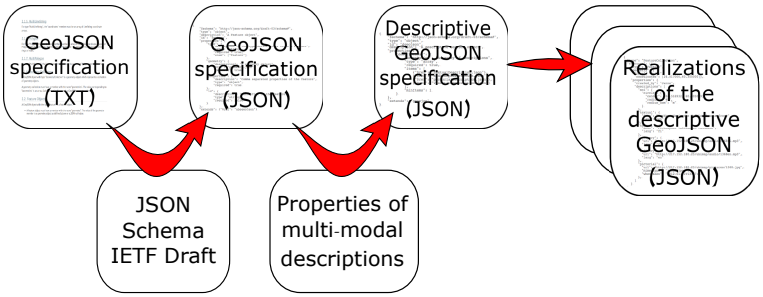


Fig. 1. The process of creating a JSON-based schema for spatial features that can be multi-modally described

3.2 Descriptive Data Interchange Schemas

Based on the properties of the data interchange formats, we came to the conclusion that we needed to create a schema for the three formats mentioned above: namely, we needed to create a schema for GeoJSON that could be used for lightweight mobile communication, a schema for KML aimed at non-professionals who could then visually see the features described on the map, and a schema for GML that would form the basis for domain-specific schemas on the server-side. We extended the KML and GML Simple Feature Profile schemas with XML Schema Definition Language (XSD), version 1.1. Similarly, we defined the extension to GeoJSON by utilizing a draft version of the JSON Schema [40], which is being supported by actively developed JSON validators. The three-format solution allows systems, for instance, to store features with vast properties on the server, to interchange data between the different Web services in GML, and, at some point, to convert the data into simpler KML and GeoJSON formats supported by thin clients. The main advantage of the schemas is the possibility to validate the structure of the data by an automated procedure and to share the data with a common understanding of the content.

We extended the GML and KML schemas with new application profiles. In GML, we used the *AbstractFeatureType* as the base, and in the case of KML, we decided to employ the *ExtendedData* element of KML. The ExtendedData element provides three options for domain-specific extensions. The first option is to use untyped textual name-value pairs for any number of attributes. The second option is to define types for the values by referencing a local or external schema definition, which defines an alphanumerical data type related to each

name. KML-reading clients, such as Google Earth, can use both of these options to represent the name-value pairs for the user. The third option was chosen because it is more flexible. It allows us to define our own schema, one which might use additional primitive data types. The content of the third option is ignored by applications not supporting our schema, which is sensible because the content is not meant to be represented as textual information for the end user. However, the symbology of the features can still be rendered on KML viewers.

The process of defining the GeoJSON extension is presented in Figure 1. It required that we first convert the textually defined GeoJSON format into a JSON Schema written in JSON. The JSON Schema makes it possible to a large extent to validate the syntax and structural integrity of the GeoJSON instances. However, all textually defined restrictions cannot be validated. For example, not even the use of regular expressions is of benefit to validate the amount or range of coordinate values, because that requires an extensive knowledge of the coordinate reference system being used, which the format does not provide. Next, we created a new JSON Schema document for our extension that references the JSON Schema for GeoJSON by using the *extends* property provided by the JSON Schema. The extension inherits all properties of the GeoJSON, and any instance of our extension also has to be valid GeoJSON. Consequently, the descriptions are stored as a property of a GeoJSON feature.

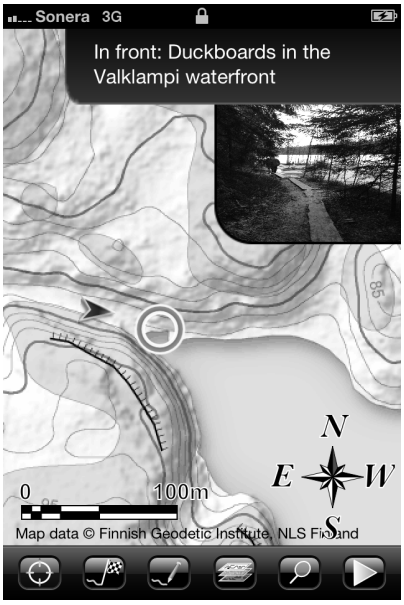


Fig. 2. An application shows what is ahead of the user. The area is visualized with a resizable picture and is described in both textual and auditory form. The blue circle shows the location of the feature.

The descriptive schemas contain the possibility to describe a feature using text, auditory data, haptic vibration patterns, and graphics. The graphics might include a photograph or a sketch. For example, navigating between the map and photographs helps users to find to their destination significantly faster than by using the map alone [8]. An auditory description might include speech or an abstract non-speech sound pattern, such as a soundscape or an auditory icon. A vibration pattern is defined by the length of vibrations and breaks between each vibration.

Area of Influence. The feature described with different modalities needs an Area of Influence (AoI). The AoI is the area where a feature can automatically be described with speech or other modalities. Typically, this is an area where the user can make visual observations about the feature or a decision concerning the feature, like not going too close to it. The AoI may vary between modalities describing the same feature. The simplest way to define an area affected by a point-wise feature is by defining the radius of a feature-centered circle. Traditionally, the non-linearly interpolated geometrical shape has not been included in data interchange formats; instead, a circle has been approximated with a linearly interpolated polygon. In the latest GML versions, this absence has been corrected by adding a *CircleByCenterPoint* type. For the KML and GeoJSON schemas, we had to define such a type. Figure 3 represents the inclusion of a circle in the extension of KML.

In addition to defining an AoI by using circles, an AoI can be defined by using a simple polygon with holes. A hole can be used to denote the area inside of which a particular feature cannot be seen or to create a halo around a particular feature. A halo can be used, for example, when a user needs to make a decision before coming too close to a feature.

The AoI is only used to initiate the description. The original geometry of the feature, or its center point of gravity, can be used to determine the direction and distance to the feature, and, in some cases, its size and shape. This information can be signaled directly or indirectly. In case of non-speech audio or vibrations, a distance can be directly represented by managing the number and density of sequential sound pulses or vibrations [20]. In the case of speech, indirect reporting can be based on stereophonic sounds [22] or on managing the volume [19]. Earlier studies related to audio have even employed features that are equivalent to the AoI; Heuten, Wichmann, and Boll [19] refer to the radius of a circular AoI as a radiation radius, whereas McGookin, Brewster, and Priege [27] define it as being the size of an Audio Bubble.

Relevance from the Time Perspective. Pictures taken by cameraphones are in general only relevant for short periods of time [37]. The same applies to other modalities characterizing the changing environment. Hence, we defined validity as a common, but optional, property for all descriptions. We specified that validity should be defined by a starting moment and duration. Alternatively, the validity may be defined as recurring. One reason for doing this is to warn hikers during winter months about thin ice over treacherous streams; however, the in-built types of XML or JSON cannot be used to define weekdays. The absence of weekdays required that we define our own type, with which a hiker might be told between Monday and Friday that a nearby shop is open.

Multi-Modal Data. For textual and auditory data, we added the option of defining a language using a language tag [1], but this is not mandatory for non-spoken sounds. For auditory descriptions, we included the option to define a type that tells if the audio contains speech or an abstract non-speech sound. In case

```
<xs:complexType name="RadiusType">
  <xs:annotation>
    <xs:documentation>
      The radius of an object. The radius involves the Unit
          of Measure.
    </xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:double">
      <xs:attribute name="uom" type="dkml:UnitOfLengthEnum"
          use="optional" default="m" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="CircleType">
  <xs:annotation>
    <xs:documentation>
      The CircleType defines the centre and radius of a
      circle where the description is given.
      The radius type includes the Unit of Measure.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Centre" type="kml:PointType" />
    <xs:element name="Radius" type="dkml:RadiusType" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AreaOfInfluenceType">
  <xs:annotation>
    <xs:documentation>
      The AreaOfInfluenceType defines the geometrical area
          and temporal time frame when a
      description is valid and is to be given. The geometry
          may be a circle or polygon.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Valid" type="dkml:TemporalValidityType"
        minOccurs="0" />
    <xs:choice>
      <xs:element name="Circle" type="dkml:CircleType" />
      <xs:element name="Polygon" type="kml:PolygonType" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Fig. 3. Fragment from the XSD document used to extend KML. The fragment defines the options for defining the Area of Influence.

of speech, we allowed the user to also include in the file spoken directions (in front, on the right, etc.) that can be used when the description is presented in an egocentric reference frame. The reasoning behind this solution was that Gong and Lai [15] found out in their study that users performed worse with mixed real human and synthesized speech in comparison to only synthesized speech.

For pictures, we allowed the format to inform when and where a picture has been taken and the dimension of the picture in pixels. A client can use this information to visualize the direction from which the picture has been taken and to choose between pre-scaled image alternatives. A client application may choose to only visually render textual descriptions, although it might also represent this information audio-visually by converting the text into speech with a text-to-speech (TTS) synthesizer. However, some information given by way of text may not be adequate for the auditory channel, especially if the text contains acronyms or abbreviations, because TTS engines do not typically decipher such information. Hence, auditory information may include a statement about whether or not the text can suitably be converted into speech by a TTS engine.

In the case of audio and images, we first thought about the possibility of both referencing and embedding binary data. However, embedding the binary data proved to be unsuitable because of the bandwidth restrictions. In addition, encoding images in Base64 leads to an unnecessary overhead. Consequently, the multimedia needs to be downloaded separately, and clients that typically use such media in areas with weak network connections should download the data in advance when they still have a good connection. For URL-referenced audio and video input, we made it possible to include multiple formats; the client can then choose the most suitable format. A MIME type and the length of the data in bytes can optionally accompany an URL. The MIME type is especially useful if the URL does not contain a filename extension (for example http://hostname/service/resource?id=1234). Similarly, the length is useful for a client that cannot stream media, because in the case of a large size, the client can decide to not even start the download process. The client can just ignore the reference.

4 The Portrayal Process on the Client

In this section, we present a generic way to multi-modally describe the surroundings around a mobile user. The generic process is composed of the following steps: 1) downloading, 2) storing, 3) representation, and 4) repeating the descriptions.

4.1 A Generic Approach to Gear Up

A mobile client implementation should asynchronously query the descriptive data while letting the application continue its normal operations. Successfully downloaded data needs to be de-serialized by a data parser. For instance, in the case of extended KML, an XML-related SAX (Simple API for XML) or DOM (Document Object Model) parser could be used to read in the data. The

parsing process could convert each KML placemark into a application-bound feature whose model includes an extension for the descriptions. Alternatively, the parser could create key-value encoded properties that contain application-perceived keys for each descriptive property, such as *textual-desc-en-gb* for a textual description in British English. Next, the parsed objects need to be stored in a persistent, spatially indexed database. At this stage, indexing can be based on the minimum bounding rectangles calculated based on the geometries of the spatial features. Later on, the datasets changing on the server side have to be systematically synchronized with the database.

It is possible to simultaneously store features, download referenced images and audio files, and store them in the local database. However, the relevance of the downloading procedure depends on the usage context, that is, on the availability of network connections during use. Before storing referenced data, a client can make some optimizations, such as resizing the downloaded images so that they can suitably be viewed by the client. In this way, the data takes less space in the database and it is faster to read and represent.

When the data is loaded from the database into the cache of the application, one alternative for increasing the effectiveness is to apply a second spatial indexing. This index should be based on the AoIs of the features. The index does not inhibit the parallel use of an index based on the geometries because both indexes have different purposes. The AoI index can be used when the descriptions are represented automatically by the application, and the geometry-based index can be used when the user manually performs an action on a particular feature.

4.2 A Generic Approach to Representing the Descriptions

A description in different modalities is presented when the user arrives at the AoI. Subsequent repetitions of the same feature may be necessary, for example when the user

- does not notice the first description,
- arrives at the same feature, but does not interpret it as being the same, or
- changes to another user

Playback can be manual or automatic. Manual repeating requires an application that visually shows the map symbols that can be selected, or provides a method, such as playback controls.

The time upon which a description of a feature is automatically represented is stored as a temporal property value affiliated with the feature. The time value is used to manage the repetition intervals. Without a timestamp, the feature might be described continuously until the user leaves the AoI. One option is to use a single Boolean mark, which makes it possible to represent a feature only once while the feature is still loaded into the cache. However, the latter approach is impractical because some spatial features are global or very large; thus, after a while, the user might not be able to recognize a particular feature as the same one he or she experienced earlier. The reference time to which the timestamp is

compared needs to be based on variables affecting the experience. Such variables are, for example, the speed of the user, the distance to the feature, and the type of the feature.

Descriptions should only be provided when the estimated accuracy of the positioning is high enough, because the accuracy affects not only the location of the feature and its AoI but also the reliability of heading towards the feature. In automated use, the AoI index is always searched when the location of the user changes and the accuracy threshold is passed. The results of the query are organized according to the distance from the actual feature to the user. From the ordered list, each feature is processed one by one until a suitable candidate is found.

A feature is unsuitable if it is invalid from a time standpoint or if its actual location is behind the user. The direction where the feature is located is calculated based on previous positioning estimates. An alternative for using positioning estimates would be to use the magnetic compass for bearings, but that is often impractical because it requires the user to hold the mobile device and point it in the direction in which he or she is moving. When the suitable feature is found, its timestamp is set to infinity. The actual updating of the timestamp is performed after the description has been represented. Just before representation the user may be informed with a vibration or sound signal.

4.3 Playback

To continue the process, the following options are available:

1. The multi-modal description of another feature is immediately terminated, and the newly found feature is described
2. The description of another feature is allowed to finish, after which the newly found feature is described
3. The features that need to be described are put into a (first-in, first-out) queue
4. The features that need to be described are put into a (last-in, first-out) stack

The first alternative is especially suitable for important notices, such as warnings that are of current interest; but otherwise, the behavior can be annoying to the user. In rest of the alternatives, a description that is already running will be allowed to finish. Consequently, the three alternatives are more user-friendly because they allow the user to understand the descriptions in the form in which they were intended. The downside is that when the time comes to describe a feature, the description might not be topical anymore. To overcome this problem, the feature has to be revalidated when the time comes to represent it. In other words, the validity of the heading, whether or not the user is in the AoI, and so forth, needs to be recalculated.

The benefit of the second and fourth alternative is that the user always gets the most recent descriptions next. What is notable in the second case is that there is no waiting queue. Hence, only one feature is available at a time, and it is

always replaced by a newer one, which may lead to a situation in which several features compete for the next place each time the user location changes until all of them have been described.

An intermediate system between the first and second option is a system where some descriptions are interrupted and some are not. This type of system could contain prioritized features or feature types that can be used to decide when the context requires that a description that is already running be interrupted.

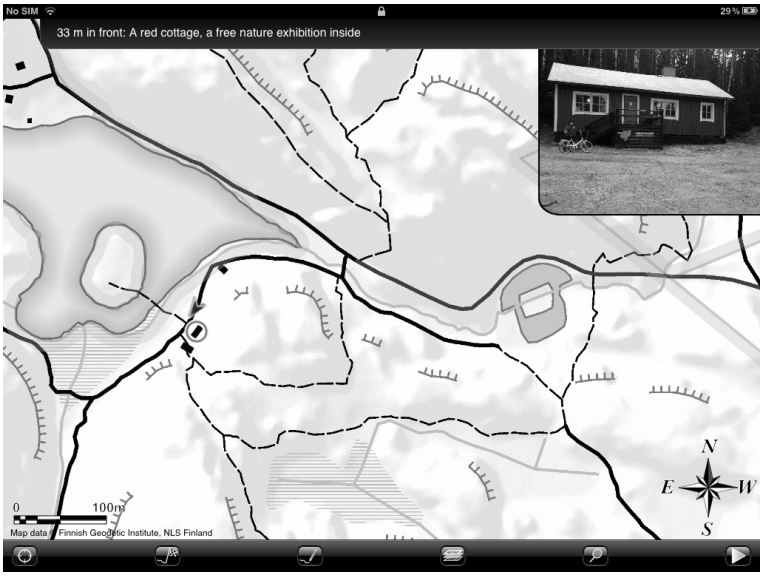


Fig. 4. The prototype implementation on a tablet suggests a non-obvious free nature exhibition. The distance and direction to the feature are appended in front of the description.

5 The Implementation

We implemented the client-side for the iOS operating systems that run on the iPhone phone (Figure 2) and iPad tablet (Figure 4) models. The client-side application visualizes map data in several layers. Raster-based background maps constitute the bottom layer, whereas vector-based thematic data forms the middle layer and user-added data forms the uppermost layer. Thematic data can be regarded as static or dynamic in nature. The client can regularly update dynamic data. The descriptive data is either static or dynamic thematic data, and every descriptive data collection forms its own thematic map layer, which can be turned on similarly as any other map layer. A user can add a new map layer by typing the URL of a resource, or by taking an image of an QRCode or Data Matrix containing a URL with an in-built function. The application supports GeoJSON, KML, and their descriptive extension.

When a map layer is turned on for the first time, the data for that particular layer is downloaded. Downloading and parsing is performed using a background thread. The downloaded descriptions are stored on the client side in a SpatiaLite [12] database. Indexing the database is based on the minimum bounding boxes, which are calculated based on the geometries of the features. For our case study, data connections are typically not available while on a hike, so the client can also instantly download any data that has been referenced to the client after the features have been parsed.

When a map layer containing descriptions is turned on, the data is loaded from the database into the local cache based on the extent of the buffered map view. When the map is being navigated, the cache is refreshed so that it corresponds to the new extent. For the cache, we used a second spatial indexing that is based on the AoI of the features. The index is searched whenever the location of the user changes and whenever the estimated accuracy of the positioning reveals that the A-GPS has a fix.

First, the AoI cache is searched, as described in the previous section. The implementation will determine that a feature lies in the background if the feature is not inside a sector of ± 120 degrees, which has been calculated based on the heading vector. The language to be used is based on the language set on the operating system level. Thus, the language used for the descriptions can only be changed by changing the global language on the device. For the temporal playback threshold, we assigned a time limit of 30 minutes for global features and 5 minutes for local features when the users' speed is less or equal to five kilometers per hour. The corresponding thresholds decrease linearly to 10 minutes and 2 minutes when the speed increases, until a speed of 50 kilometers per hours is reached. Currently, the prototype does not use the type or the distance to the feature.

Onscreen information is given in the top edge of the screen. Textual data is located uppermost, followed by resizable pictures. The data is shown for 15 seconds. Images need to be converted into a size suitable for a particular device model. Vibrations and pre-recorded audio are delivered simultaneously using standard frameworks.

Based on the findings of Kainulainen et al. [21], presenting both speech and abstract audio (such as an auditory icon) at the same time may be hard to understand by users; hence, the system regards only one auditory description. If the feature is not associated with an auditory description, the same text is converted to an audio file using an open source TTS engine called eSpeak [11]. The TTS engine supports the languages that we need for our use case, which are English, Swedish, and Finnish.

Before the text or audio can be represented, the textual or spoken direction (which is calculated based on the heading to the feature) and distance must be concatenated to the record. In case of pre-recorded audio, the system uses pre-recorded directions if such are given. But in case of synthesized speech, only synthesized directions are concatenated to keep the output consistent and to lower the cognitive processing demand of the user.

At the same time, the feature is highlighted if it is visible, otherwise the location is pointed out by a short animation. If a description is already playing, we put the new description in a queue. Visible features can also be selected by the user at any point. Unlike an automatically given description, a user-initiated description terminates any ongoing description.

The feasibility of the presented format was affirmed from the technical perspective by first collecting the descriptions of non-obvious features that are of general interest, found in the National Park, into the proposed data interchange formats. The forms of descriptions included multi-lingual text, recorded speech, synthesized speech, and pictures. Next, the data was loaded to the mobile client with varying mechanisms, including the use of QRCodes that can be printed at the sites where National Park is entered. Finally, the descriptions were confirmed to be given when a user entered a feature-bound Area of Influence.

6 Discussion and Conclusions

This paper has presented the technical background and implementation of a Location-Based Service to extend the user experience that a visitor of a natural park may experience while using a mobile device. However, in this paper we do not address the aspects of usability from the end user perspective, that is, what the users would prefer as the AoI, or how the end users feel about the different modalities. Nevertheless, from related multi-modal turn-by-turn navigation experiments, we know that users prefer different modalities in different situations. For instance, according to the study of Liljedahl et al. [25], users mostly rely on the visual channel, but prefer auditory direction notification.

The solution does not take into account particular vibration patterns, because we are unaware of a coding scheme that would be generally accepted. If feature types are encoded into patterns these need to be known by a client and taught to the user. Neither does our proposal define audio-related properties for textual information that can be converted to speech by a TTS engine. Such properties could include a wish to represent the information using a certain voice (that is based on some characteristics, e.g., gender or age), tone (e.g. personal or enthusiastic), or volume. For example, Caquard et al. [6] use a second voice, in addition to the authoritative "voice-of-god" to present non-objective information from a personal perspective, like opinions and doubts. Graham and Cheverst [16] go even beyond two voices by presenting five personified interaction paradigms for guiding applications.

There are still many challenges to overcome. A question is how the descriptions of different kinds of modalities are created, extended and kept up-to-date. One solution may be collaborative data maintenance by the park visitors. A more futuristic vision is that the users surrounding is not only described human-to-human. Instead, locally embedded sensors may provide additional information, such as images of the latest animals passing the area or popularity of paths. Similarly, technical development may extend the usability of modalities. For example, in a couple of years, off-the-shelf smartphones might have touch screens incorporating haptic Braille support.

Acknowledgements. This survey is a part of the UbiMap project which is funded by the Academy of Finland, Motive program, and is carried out in co-operation with the FGI, Department of Geoinformatics and Cartography, and the University of Helsinki, Department of Cognitive Science. The authors are also grateful to the European Commission which co-funded the IP HaptiMap (FP7-ICT-224675).

References

1. Alvestrand, H.: Tags for the Identification of Languages, RFC 1766, Network Working Group, Internet Engineering Task Force (1995)
2. Aslan, I., Schwalm, M., Baus, J., Krüger, A., Schwartz, T.: Acquisition of Spatial Knowledge in Location Aware Mobile Pedestrian Navigation Systems. In: Proceedings of the 8th Conference on Human-Computer Interaction With Mobile Devices and Services, MobileHCI 2006, Helsinki, Finland, pp. 105–108. ACM, New York (2006)
3. Baus, J., Wasinger, R., Aslan, I., Krüger, A., Maier, A., Schwartz, T.: Auditory perceptible landmarks in mobile navigation. In: Proceedings of the 12th International Conference on Intelligent User Interfaces, IUI 2007, Honolulu, Hawaii, USA, pp. 302–304. ACM, New York (2007)
4. Blattner, M.M., Sumikawa, D.A., Greenberg, R.M.: Earcons and icons: Their structure and common design principles. Human-Computer Interaction 4, 11–44 (1989)
5. Butler, H., Daly, M., Doyle, A., Gillies, S., Schaub, T., Schmidt, C.: The GeoJSON Format Specification, Revision: 1.0 (2008)
6. Caquard, S., Brauen, G., Wright, B.: Exploring sound design in cybercartography. In: Proceedings of the 22nd International Cartographic Conference (ICC), July 9-16 (2005)
7. Cheverst, K., Davies, N., Mitchell, K., Friday, A., Efstratiou, C.: Developing a context-aware electronic tourist guide: some issues and experiences. In: Proceedings of the International Conference on Human-Computer Interaction, CHI 2000, The Hague, The Netherlands, pp. 17–24. ACM, New York (2000)
8. Chittaro, L., Burigat, S.: Augmenting audio messages with visual directions in mobile guides: an evaluation of three approaches. In: Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services, MobileHCI 2005, Salzburg, Austria, pp. 107–114. ACM, New York (2005)
9. Ciavarella, C., Paternò, F.: The design of a handheld, location-aware guide for indoor environments. Personal and Ubiquitous Computing 8(2), 82–91 (2004)
10. Crockford, D.: The application/json Media Type for JavaScript Object Notation (JSON), RFC 4627, Network Working Group, The Internet Engineering Task Force (2006)
11. eSpeak: Speech Synthesizer, version 1.46.27 (2012), `http://espeak.sourceforge.net`
12. Furieri, A.: SpatiaLite, version 3.0.0 (2011), `http://www.gaia-gis.it/fossil/libspatialite/index`
13. Gaver, W.W.: Auditory icons: Using sound in computer interfaces. Human-Computer Interaction 2, 167–177 (1986)
14. Ghiani, G., Leporini, B., Paternò, F.: Supporting orientation for blind people using museum guides. In: CHI 2008 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2008, Florence, Italy, pp. 3417–3422. ACM, New York (2008)

15. Gong, L., Lai, J.: To Mix or Not to Mix Synthetic Speech and Human Speech? Contrasting Impact on Judge-Rated Task Performance versus Self-Rated Performance and Attitudinal Responses. International Journal of Speech Technology 6(2), 123–131 (2003)
16. Graham, C., Cheverst, K.: Guides, locals, chaperones, buddies and captains: managing trust through interaction paradigms. In: Workshop on Mobile Guides at the Fifth International Symposium on Human-Computer Interaction with Mobile Devices and Services, MobileHCI 2004, Glasgow, Scotland (2004)
17. Guy, R., Truong, K.: Crossingguard: exploring information content in navigation aids for visually impaired pedestrians. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2012, Austin, Texas, USA, pp. 405–414. ACM, New York (2012)
18. Hameseder, K., Fowler, S., Peterson, A.: Performance analysis of ubiquitous web systems for SmartPhones. In: International Symposium on Performance Evaluation of Computer Telecommunication Systems, SPECTS 2011, pp. 84–89 (2011)
19. Heuten, W., Wichmann, D., Boll, S.: Interactive 3d sonification for the exploration of city maps. In: Proceedings of the 4th Nordic Conference on Human-Computer Interaction: Changing Roles, NordiCHI 2006, Oslo, Norway, pp. 155–164. ACM, New York (2006)
20. Holland, S., Morse, D.R., Gedenryd, H.: AudioGPS: Spatial audio navigation with a minimal attention interface. Personal and Ubiquitous Computing 6(4), 253–259 (2002)
21. Kainulainen, A., Turunen, M., Hakulinen, J., Melto, A.: Soundmarks in spoken route guidance. In: Proceedings of the 13th International Conference on Auditory Display, Montreal, Canada (2007)
22. Kan, A., Pope, G., Jin, C., van Schaik, A.: Mobile Spatial Audio Communication System. In: Proceedings of ICAD 2004-Tenth Meeting of the International Conference on Auditory Display (July 2004)
23. Kangasharju, J., Tarkoma, S.: Benefits of alternate XML serialization formats in scientific computing. In: Proceedings of the 2007 Workshop on Service-Oriented Computing Performance: Aspects, Issues, and Approaches, SOCP 2007, Monterey, California, USA, pp. 23–30. ACM, New York (2007)
24. Krüger, A., Aslan, I., Zimmer, H.D.: The Effects of Mobile Pedestrian Navigation Systems on the Concurrent Acquisition of Route and Survey Knowledge. In: Brewster, S., Dunlop, M.D. (eds.) Mobile HCI 2004. LNCS, vol. 3160, pp. 446–450. Springer, Heidelberg (2004)
25. Liljedahl, M., Lindberg, S., Delsing, K., Polojärvi, M., Saloranta, T., Alakärppä, I.: Testing Two Tools for Multimodal Navigation. In: Advances in Human-Computer Interaction (2012)
26. Mabrouk, M.: OpenGIS® Location Services (OpenLS): Core Services, Document OGC 07-074, Version 1.2, Open Geospatial Consortium Inc. (2008)
27. McGookin, D., Brewster, S., Priego, P.: Audio Bubbles: Employing Non-speech Audio to Support Tourist Wayfinding. In: Altinsoy, M.E., Jekosch, U., Brewster, S. (eds.) HAID 2009. LNCS, vol. 5763, pp. 41–50. Springer, Heidelberg (2009)
28. Münzer, S., Zimmer, H.D., Schwalm, M., Baus, J., Aslan, I.: Computer-assisted navigation and the acquisition of route and survey knowledge. Journal of Environmental Psychology 26(4), 300–308 (2006)
29. Oppermann, R., Specht, M.: A nomadic information system for adaptive exhibition guidance. In: Proc. of the International Conference on Hypermedia and Interactivity in Museums, pp. 103–109 (1999)

30. Petrelli, D., Not, E., Sarini, M., Stock, O., Strapparava, C., Zancanaro, M.: Hyperaudio: location-awareness + adaptivity. In: CHI 1999 Extended Abstracts on Human Factors in Computing Systems, CHI EA 1999, pp. 21–22. ACM, New York (1999)
31. Pielot, M., Henze, N., Boll, S.: Supporting Map-Based Wayfinding with Tactile Cues. In: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI 2009, Bonn, Germany, pp. 1–10. ACM, New York (2009)
32. Pielot, M., Poppinga, B., Boll, S.: Understanding tourists on a bicycle trip "in the wild". In: Mobile Living Labs Workshop in Conjunction with MobileHCI 2009 (2009)
33. Richter, K.-F.: A Uniform Handling of Different Landmark Types in Route Directions. In: Winter, S., Duckham, M., Kulik, L., Kuipers, B. (eds.) COSIT 2007. LNCS, vol. 4736, pp. 373–389. Springer, Heidelberg (2007)
34. Rodrigues, C., Afonso, J., Tomé, P.: Mobile Application Webservice Performance Analysis: Restful Services with JSON and XML. In: Cruz-Cunha, M.M., Varajão, J., Powell, P., Martinho, R. (eds.) CENTERIS 2011, Part II. CCIS, vol. 220, pp. 162–169. Springer, Heidelberg (2011)
35. Sumaray, A., Makki, S.K.: A comparison of data serialization formats for optimal efficiency on a mobile platform. In: Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, ICUIMC 2012, Kuala Lumpur, Malaysia, pp. 48:1–48:6. ACM, New York (2012)
36. van den Brink, L., Portele, C., Vretanos, P.A.: Geography Markup Language (GML) simple features profile (with Corrigendum), Version 2.0, OGC® 10-100r3, Open Geospatial Consortium (2012)
37. Van House, N., Davis, M.: The Social Life of Cameraphone Images. In: Proceedings of the Pervasive Image Capture and Sharing: New Social Practices and Implications for Technology Workshop (PICS 2005) at the Seventh International Conference on Ubiquitous Computing (UbiComp 2005), Tokyo, Japan (2005)
38. Wilson, G.: Multimedia tour programme at tate modern. In: Bearman, D., Trant, J. (eds.) Proceedings of the Museums and the Web 2004. Archives & Museum Informatics, Toronto (2004)
39. Wilson, T.: OGC®KML, Version: 2.2.0, Document 07-147r2, Open Geospatial Consortium (2008)
40. Zyp, K.: A JSON Media Type for Describing the Structure and Meaning of JSON Documents, Version 03, Internet-Draft, Internet Engineering Task Force (2010)

An Adaptive Context Acquisition Framework to Support Mobile Spatial and Context-Aware Applications

André Sales Fonteles*, Benedito J.A. Neto**, Marcio Maia, Windson Viana, and Rossana M.C. Andrade***

Group of Computer Networks, Software Engineering and Systems (GREat), Federal University of Ceará (UFC), Campus do Pici, Bloco 910, Zip Code 60455-760, Fortaleza, CE, Brazil
{andrefonteles,beneditoneto,marcio,windson,rossana}@great.ufc.br
http://www.great.ufc.br

Abstract. The increasing number of mobile devices allows users to access applications anytime and anywhere. In such applications, location is a key information to improve the interaction between user and services. Existing applications combine location with other context information, such as weather, user's activity, temperature, among others. However, developing context-aware applications is still a non-trivial task due to the complexity to implement context management. Additionally, existing context management infrastructures are too brittle to handle changes in the underlying execution infrastructure. In this scenario, this work proposes a context acquisition framework, which tries to reduce the development complexity of mobile spatial and context-aware applications. The framework uses tuples space and OSGi to promote uncoupling and to adapt itself according to application requirements. A proof of concept was developed in order to show how spatial and context filters can be easily implemented during the development of a tracking application.

Keywords: GIS, Context-Aware, Adaptation, Mobility, Android.

1 Introduction

The increasing number of mobile devices, such as smart phones and tablets, allows users to access the Internet and a wide range of applications anytime and anywhere. This scenario converges to the concept of Ubiquitous Computing predicted by Mark Weiser [20], where the most profound technologies are those that disappear, becoming part of everyday life.

Once one application can be accessed anytime and anywhere, the user's location can be relevant to improve the interaction among user, application and

* Master Scholarship (MDCC/DC/UFC) sponsored by CNPQ.
** Master Scholarship (MDCC/DC/UFC) sponsored by CAPES.
*** Researcher scholarship - DT Level 2, sponsored by CNPq.

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 100–116, 2013.
© Springer-Verlag Berlin Heidelberg 2013

services provided. For example, if a user is hungry and decides to eat something, an application can provide a list of nearby restaurants without any need of explicit asking the user for his/her location. Many mobile applications have been already developed using spatial data in order to provide personalized services and content, such as Google Places[1], Localscope[2] and Wikicrimes Mobile[3], from the WikiCrimes [7] project.

Moreover, some studies in mobile applications have combined user's location with a larger data set concerning the user's situation, which is considered as user's context. Taking into account that Dey and Abowd [1] defines contextual information as any information that can be used to characterize the situation of entities considered relevant to the interaction between a user and an application, including the user and the application themselves.

The development of mobile applications that integrate spatial information with context data is an important research subject in many domains: from multimedia [3][17] to mobile learning [9]. These applications use contextual information (e.g., weather forecast, temporal data, user's activity, presence of nearby friends or devices) to provide relevant services, to adapt user interface or to produce meta data about multimedia documents.

Despite the benefits of using contextual information to improve mobile applications, to implement context acquisition, management and exploitation methods can be a difficult task. The more contextual information is considered by the application, the greater is the challenge.

First of all, context data is often provided by physical sensors (e.g., GPS, thermometers and accelerometers) as low level information (e.g., latitude, longitude or temperature in degrees). Applications, otherwise, expect high level context data to adapt their behaviour according (e.g., user's current city and if a day is warm or cold). Context-aware applications developers should implement code to provide this high level context information.

Second, to deal with physical sensors may require developer's expertise in some specific hardware. To develop a logical sensor with a context inference mechanism need even more specific knowledge. For example, to develop a logical sensor that given the data from an accelerometer, it infers if the user is walking, running or still is not a trivial task. In addition, mechanisms to detect user's situation and react according are frequently integrated to the code responsible for the application business logic, which makes code reuse a problematical activity.

In this scenario, we propose a context acquisition framework implemented in the Android platform, which tries to reduce the complexity of the development of mobile spatial and context-aware applications. The framework provides context information as tuples in a shared memory space (i.e., tuple space). Mobile applications can connect to the tuple space to query contextual information or to subscribe for being notified when the user's situation changes. Spatial filters can be specified and combined with other context filters to compose this

[1] `http://www.google.com/places/`
[2] `http://www.cynapse.com/localscope`
[3] `http://www.wikicrimes.org`

notification rule. For the application developers, no knowledge concerning the physical layer is required. As a consequence, the framework reduces coupling among applications and the context acquisition layer. This clear separation between context acquisition and context consumption allows the framework to adapt the way context is acquired without the perception of the applications above. This adaptation mechanism tries to reduce resource consumption and provides transparency when sensors are switched.

The remainder of this paper is divided as follows: Section 2 presents the theoretical foundations of our work, which includes the description of the SySSU middleware. Section 3 describes the proposed framework, how it extends the SySSU middleware for including spatial and context-aware filters. The fourth section presents a proof of concept we have developed to illustrate the notification and adaptation mechanisms. Finally, Section 5 contains conclusions and future works.

2 Theoretical Foundations

The development of mobile and context-aware applications is not a novel domain. Many technologies have been proposed by researchers and companies aiming to assist developers and software engineers in the task of designing, implementing and testing these applications. In this section, we briefly present achievements and limitations of these approaches. Some of them have been reused, extended or inspired our framework. In particular, we describe SysSU, a middleware proposed by our research group, which is adapted and extended in this work.

2.1 Context-Aware Supporting Infrastructures

In an attempt to deal with the complexity of context-aware systems, many frameworks and middleware platforms have been proposed by researchers, as can be seen in [2], [8] and [5]. Baldauf et al. [2] and Marinho et al. [13] identify recurrent architecture layers and their roles in existing frameworks, as follows:

- The Sensors Layer consists of a collection of sensors, varying from physical and logical to virtual. Virtual sensors gather context information from software application or services (e.g., accessing a weather Web Service).
- The Raw Data Retrieval Layer is responsible for retrieving context data. Usually, this layer is composed of wrapper components, which encapsulate sensors, making low-level details, such as hardware access, transparent to applications. Another frequently feature is interface standardization, focused on maximize component reuse. It is possible, for instance, to replace a component that uses GPS by one that uses cellular antenna triangulation to provide location without major modifications in the system.
- The Preprocessing Layer is responsible for translating raw context information into semantic-enriched information. This layer is uncommon in mobile context-aware systems since few inference mechanisms are available on mobile platforms.

- The Storage/Management Layer organizes context information and offers them via a public interface to applications.
- The Application Layer is where the applications using the framework are. These applications are responsible for implementing methods to react and adapt to changes in context.

One important characteristic of existing frameworks and middleware platforms is that they provide means for applications to adapt to context changes. These frameworks, however, do not adapt themselves to the context [5][14]. Furthermore, many of them are monolithic systems, in a sense that their components cannot be deployed separately [14]. Their lack of adaptation capabilities plus their inflexibility to add new components make it difficult to use them in mobile devices with limited resources, inserted in dynamic environments with continual changing contexts. Additionally, analyzing the previous cited surveys on context awareness, there are no references to any work supporting geographic data management and operations.

There are existing frameworks and middleware platforms somehow similar to our approach. In [16] is proposed a context framework for mobile devices running Android called ContextDroid. Its main characteristics are: efficiency, extensibility and portability. Although ContextDroid is unable to adapt its context acquisition mechanism, it proposes a context acquisition infrastructure based on the component oriented programming (COP) model of the Android platform. Our proposed framework uses a similar approach, but the life cycle management of the components is achieved using OSGi[4].

Kramer et al. [10] proposes an infrastructure for context acquisition that monitors context changes independently of application. This infrastructure does not adapt itself as well, but it uses the strategy that a single instance of the infrastructure is shared by many applications in a device, optimizing resource utilization. The same strategy of a single instance is adopted by our proposed framework.

In [14] it is presented a framework capable of performing deployment and runtime adaptation of its components. This framework aims to achieve a better resource utilization in mobile devices. Our adaptation mechanism is inspired by this work. At last, authors in [11] present a context-monitoring framework called MobiCon. Our proposed framework is similar to MobiCon from many perspectives. First of all, both architectures allow different applications to share a single framework instance. Furthermore, MobiCon's approach for context acquisition adapts itself, minimizing the number of working components based on the interest of the applications (expressed by queries). Despite these similarities, MobiCon is event oriented and does not provide contextual information in a synchronous way.

Among the studied context-aware middlewares and frameworks, location data is treated as high-level abstractions (e.g., location=home, theater). They rarely provide mechanisms to deal with spatial relations and operations, such as distance among points and regions, and topology relations (e.g., inside, outside).

[4] `http://www.osgi.org`

2.2 SysSU

SysSU (System Support for Ubiquity) [12] is an infrastructure that aims to provide mechanisms to implement the main requirements of ubiquitous systems, such as coordination and service discovery/description. It implements a coordination model formed by the composition of tuple space [4] and event based [6] approaches. The SysSU permits the execution of operations like writing and reading tuples, synchronously or asynchronously.

A tuple is composed of a set of key/value fields. For example, {(user,"John"), (age,10),(gender,"M")}. There are two ways for an application to access a tuple published at SysSU. The first one is to query it. Doing so, the information should be available before it is required by the application. The second one is event based and is used when the information is not yet available or an application wants to be notified every time a new information is published. Both methods make use of templates and filters to select the required tuples. A template is a collection of fields representing a set or a subset of fields composing the required tuple. For example, to find tuples containing a field with a key "user" it would be used the following template: {(user,?)}. A template can also be used to match values. For example, the template {(user,?),(age,10)} returns every user with 10 years old. The templates can be also used to select tuples with same fields, or field with a specific value. To improve the expressiveness of a tuple selection, a filter should be used. A selection filter is created using Java and is only limited by the program language expressiveness. Fig. 1 shows an example of filter that select tuples where a field age is greater than 16. To use a filter it is necessary to create

```
public class AgeFilter implements IFilter {
    public boolean filter(Tuple tuple) {
        for (int i = 0; i < tuple.size(); i++) {
            if(tuple.getField(i).getName().equals("age")) {
                int age = (Integer)tuple.getField(i).getValue();
                if(age > 16)
                    return true;
            }
        }
        return false;
    }
}
```

Fig. 1. Example of a filter implementation

a class that implements a Java interface called IFilter. The IFilter is composed by a single method called *filter*. Everytime a query is performed, a template is formerly used to select a subset of all the tuples in SysSU, then every lasting tuple will pass one by one through the *filter* method of the IFilter instance. This method is responsible to check if these tuples meet the given requirements of the filter.

SysSU has showed its efficiency in improve the uncoupling among applications and services that share the tuple space to coordinate their operations. It play a central role in our framework providing uncoupling among Android applications and the context acquisition layer. Originally, SysSU was proposed as a client-server architecture where the server hosted the tuple space to share information among thin clients. In order to reuse SysSU in our framework, we had to adapt its architecture to fully embed it in a single mobile device where applications can simultaneously access an internal tuple space. Its filters mechanisms are extended to support spatial and context-aware notifications.

3 Proposed Framework

We propose a framework for Android that acquires context in an adaptive manner designed to be used embedded in mobile devices, which has limited resources and are inserted in dynamic environment. This framework allows deployment and dynamic (i.e., runtime) adaptation of its context acquisition components (CAC) to achieve a better resource usage [14]. Besides, dynamic adaptation enables a system to fulfill its requirements in response to changes in context at runtime [15]. Another characteristic of this framework for better resource utilization is that just one single instance runs per device, while many applications can use it.

Fig. 2 shows the architecture of the proposed framework. The main entities of the architecture are Context Acquisition Component (CAC), CAC Manager, Adaptation Reasoner, SysSU, SysSU Filters and Application.

A CAC is a component type that wraps physical, logical and virtual sensors and provides context information to the framework. These components can be started, stopped, installed and removed through the CAC Manager even during runtime. To provide such flexibility the CAC Manager makes use of OSGi framework. The manager is not responsible for reasoning about what configuration of CACs the framework will use. This is a role of the Adaptation Reasoner. The Adaptation Reasoner decides what CACs should be used at runtime according to the needs of the Applications.

Our framework provides a low coupled interaction with the application using an infrastructure called SysSU. SysSU is proposed by Lima et al.[12] based on tuple space that provides means of coordination and interaction that are decoupled for the development of ubiquitous systems. All the communication between the application and the framework is done by using SysSU. Applications perform queries for context information through SySU. The Adaptation Reasoner then determines a valid configuration of CACs that fulfills the requirement of the queries and use the CAC Manager to achieve this configuration. Finally, the running CACs publish their context information at SysSU where the applications are able to access it.

The SysSU Filters is an optional API intended to provide a collection of filters (see Session 2.2) used to query or to subscribe to events based on contextual information. This API also should provides spatial filters to manage geographic

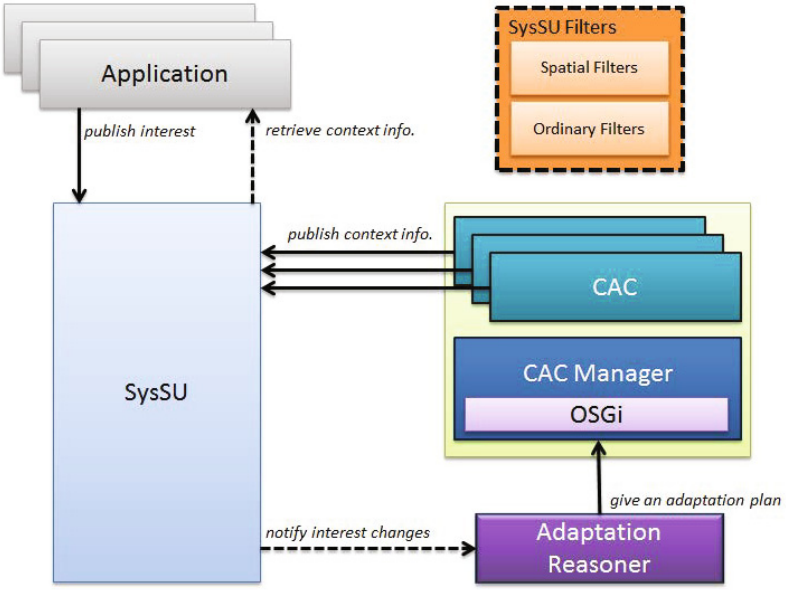


Fig. 2. Architecture of the proposed framework

contextual information. At the moment, we only provide two of there filter as a proof of concept.

3.1 Context Representation

In order to allow communication between components and applications, it is necessary to use a shared vocabulary to represent types of context information. Each type of context information (e.g., temperature, location and weather) requires a unique key or identifier that should be used by CACs, to determine what kind of information it provides to framework and to publish information, and by applications to query this same kind of information.

We defined, based on the Management Information Base (MIB)[5] model, a hierarchical scheme to generate unique keys, called context keys (CK), to types of context information. Using this schema, a contextual information is referenced using a sequence of names separated by points. For instance, a key context.device.location references to a device's location information. Thus, any CAC that publishes a contextual information of a device's location at SysSU should use this CK. In order for an application to find that information at SysSU, the same CK should be used.

Fig. 3 shows a subset of the hierarchy already defined to compose CKs. As soon as new CACs providing new kinds of context are developed, this hierarchy can be extended from any node at the tree.

[5] `http://www.ieee802.org/1/pages/MIBS.html`

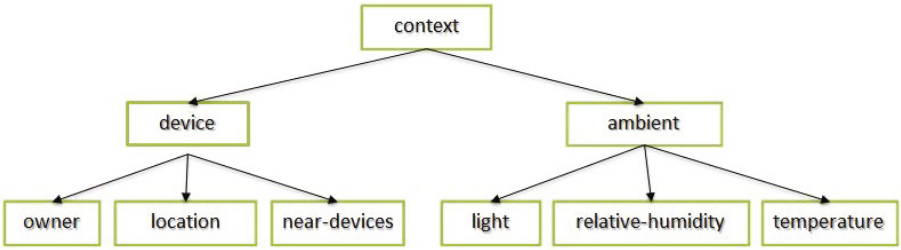


Fig. 3. Subset of the hierarchy already defined to compose CKs

Besides the need of CKs to identify kinds of context, it is also necessary to specify how can such information be represented on a tuple space. Based on a context metamodel proposed by Vieira et al. [19], we define that a tuple with context information (called context tuple) must have at least four fields:

- ContextKey: In this field, the CK representing the kind of context information at the context tuple should be inserted. Applications should use this field to match tuples with their required kinds of context.
- Source: It informs the source of the information. Contextual information can be originally provided by physical, logical and virtual sensors.
- Values: This field contains an array with the actual values of the context. For example, in a context tuple representing an ambient temperature in Celsius, this array would have one index with a value such as 28.
- Timestamp: This field contains the time, in milliseconds, which a contextual information was read.

Others fields can be added according to the needs of contextual representation. For example, in some cases, it is added an *Accuracy* field where the accuracy (precision) of the reading is set (e.g., user's location with a accuracy of 5 meters). Fig. 4 shows examples of context tuples.

Using the CK definition and the context tuple aforementioned, applications may access the tuple space and execute queries or subscribe themselves to be notified when an specific context information is generated, as described in section 2.2. For instance, let an existing CAC be responsible for the address of the user, in which its CK is context.device.location.address. Applications may access the address of the user using the {(contextkey,"context.device.location.address")} template.

3.2 Spatial Filters

In most part of the mobile and context-aware applications, user's location information plays an important role. However, the manipulation of geographic data in mobile applications is a difficult task, especially, for inexperienced developers in the domain. In some cases, developers will be unable to use a remote geographic

ContextKey	Source	Values	Timestamp	Accuracy
context.device.location	Physical Sensor	38.68551, -9.103271	1337821714638	5
context.ambient.temperature	Physical Sensor	28	1337829018765	3

ContextKey	Source	Values	Timestamp
context.device.owner	LogicalSensor	"John"	1337829056475
context.device.near-devices	LogicalSensor	"Billy", "Jin"	1337829018721
context.user.activity	LogicalSensor	"running"	1337829018721

Fig. 4. Examples of context tuples

database to answer simple spatial queries such as "what is the minimum distance in meters between the user and a given spatial point?" or "is the user inside a given region?". For instance, when there is no available internet connection or when the application runs locally in the mobile device.

In addition, analyzing surveys on frameworks and middleware platforms for context management as [2], [8] and [5], one can conclude that these infrastructures provide few services or mechanisms for dealing with geographic data.

To reduce this inconvenience, the proposed framework enables the development and reuse of code for managing this type of data, by extending the notification filters of SysSU with spatial operations. As part of this work, we developed two filters for creating queries and events based on geographical information: DistanceFilter and PlaceFilter.

The DistanceFilter is used when an application wants to know if the user is at a minimum distance of a specific coordinate. It was developed making use of the class Location of the Android Framework. The Location class has a method called *distanceTo* that, according to the Android documentation[6], returns the approximate distance in meters between two given locations.

The PlaceFilter is used to determine if the user is within one or more geographic areas determined by a set of coordinates, as shown in Fig. 5. For example, a PlaceFilter can be used by a mobile learning application, which requires a notification when a user is inside or near to a historical monument in order to play video or audio containing related information. This filter also makes use of a third-party API, called JST[7], to test intersections between a polygon and a point. The JTS is an API of 2D spatial predicates and function. We did not use map projections to transform latitude and longitude to points in a cartesian plan, so this filter is an approximation.

[6] `http://developer.android.com/reference/android/location/Location.html`
[7] `http://www.vividsolutions.com/jts`

Fig. 5. Example of an area used by PlacesFilter

Like the two spatial filters aforementioned, our framework enables developers to create new filters from the domain specific needs of their applications and share it as a Java class for reuse purposes.

3.3 Adopted Context Definition

As we discussed before, Dey and Abowd defined context as any information that can be used to characterize the situation of entities considered relevant to the interaction between a user and an application. In this work, we adopted another definition of context made by Viana [18]. Viana extends the former definition but focusing at context acquisition, and affirms that context can be characterized as an intersection between two dynamic and evolutionary sets, as shown in Fig. 6.

A set is named Interest Zone (IZ) and is contains all the information related to environment and user that the system would like to know. The other set is named as Observation Zone (OZ), where there are all contextual information that the system can obtain. The intersection between what is interesting to the system and what the system can observe in a certain time is considered as context. For example, suppose a smartphone application that, according to the ambient sound, adjust the bell volume. In this case, the ambient sound is an element from IZ, because the system wants this information. Through the device's microphone is possible to capture the ambient sound, then, it is also in the OZ. Finally, for being in both sets, the ambient sound is considered as context to the application.

The elements that are in IZ and OZ may change over time. Following the previous example, as the time pass, the user can configure his/her smartphone to silent mode. If this occurs, his device will not emit any sound alert. Thus, it does not matter to the system to have knowledge about the ambient sound to adjust the bell. This condition removes the ambient sound from IZ, although it

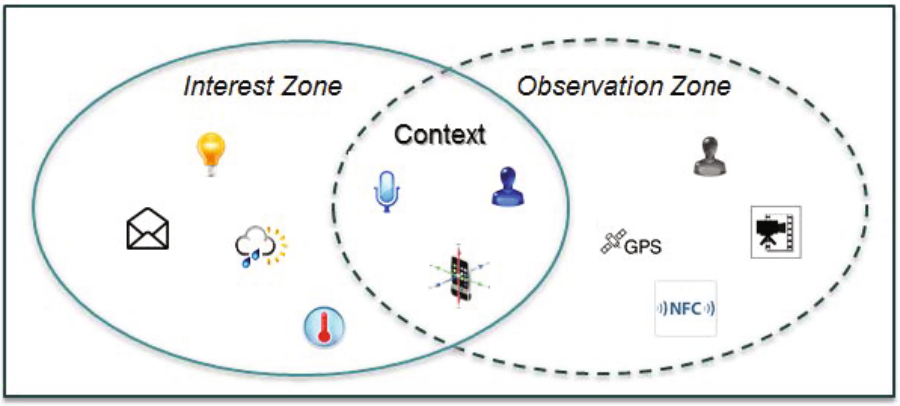


Fig. 6. Context as an intersection between interest zone and observation zone. Adapted from [18]

is still in OZ. Once the ambient sound is not present in both sets at the same time anymore, it is not considered as context. Likewise, even if the device was not in silent mode, its microphone might be damaged, and this would be an impediment to the capture of the ambient sound. Therefore, it would not be in OZ and would not be part of the context, despite the fact is still present in IZ. The behaviour previously described is known as the context evolutionary feature. Formally speaking, we can say that the observed elements that compose the context in a certain time t are not necessarily the same in another time t'.

Each application that uses the proposed framework must send an updated CK list that represents its interests. This list, along with its possible interests (dependencies) of running CACs, represents the IZ for the framework. On the other hand, the set of all context CKs provided by running CACs represents the OZ.

3.4 Adaptation Mechanism

The adaptation of the proposed context acquisition framework takes place in two different moments: deployment time and runtime. Deployment time adaptation permits the developer to install strictly those CACs with CKs required by the applications.

Additionally, runtime adaptation is performed based on the context of the running applications. Context, as previously described, is the intersection between Interest Zone and Observation Zone. Thus, the CAC lifecycle management allows the framework to stop and remove CACs in order to optimize resource consumption based on the current context of all applications. The main goal of the runtime adaptation is to reach a state where IZ contains the OZ, as shown in

Fig. 7. Therefore, all unnecessary CACs in a given time are stopped. When the IZ changes, the adaptation mechanism is responsible for detecting this change and searches for compatible CACs to handle the new IZ.

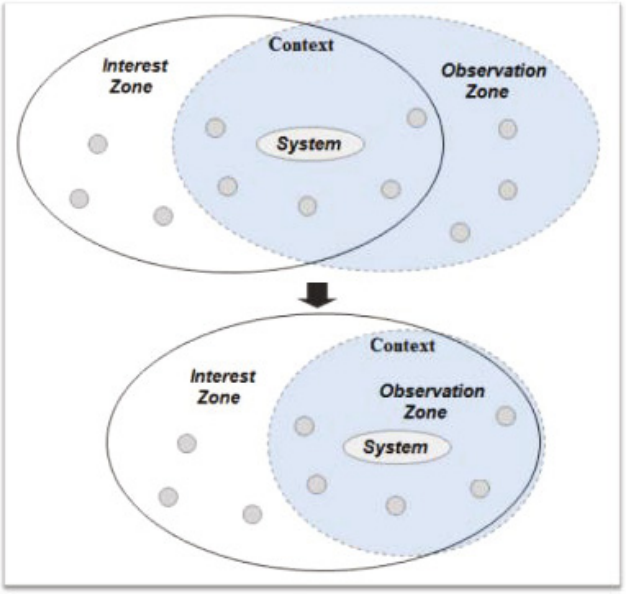


Fig. 7. Context acquisition adaptation behaviour

4 Validation

As a proof of concept, in this section we describe how an application can be implemented using our framework to incorporate a context-aware behaviour and easily react for location and context changes. We define a scenario where an application tracks a user trajectory. Every coordinates pair of the trajectory is annotated with more contextual information (e.g., ambient light, temperature and relative humidity). These kind of contextual "log" is common on many multimedia applications, such as Photomap [17] and Captain [3]. We also define a region on map (e.g., an area around a square), in a certain point of the trajectory, where an application would desire to be notified to perform some action when a user get inside or nearby (depending on the filter). This feature is very common in mobile location based applications. For instance, a mobile learning application that proposes multimedia content about a historical monument when the user is inside or in the neighbourhood, can use the same filter mechanism.

The track application was implemented in Android using our framework. Fig. 8 shows a visual representation of some tracked points acquired during an

real application usage test, in which the user walked through some street blocks. Fig. 8 also presents a region of interest, where our application successfully get warned.

Fig. 8. Visual representation of some tracked points acquired

To use our framework in the track application, three steps were necessary to follow:

1. publish IZ in the tuple space;
2. filter subscription; and
3. tuple space query

The first step in the application development was to publish its IZ at SysSU tuple space. This step was necessary to allow the framework to adapt its OZ enabling the necessary CACs. Fig. 9 shows how we did it. SysSUAndroid is the hotspot of our framework,

```
sysSUAndroid = new SySSUAndroid();
sysSUAndroid.publishInterests("appName", "context.device.location");
sysSUAndroid.publishInterests("appName", "context.ambient.light");
sysSUAndroid.publishInterests("appName", "context.ambient.temperature");
sysSUAndroid.publishInterests("appName", "context.ambient.relative-humidity");
```

Fig. 9. Publishing application's interest

Once the IZ was published at the tuple space, the application has subscribed to listen new context tuples when user's location was published by CACs. Subscriptions need three phases. First, we created a tuple template as following {(ContextKey, "context.device.location")}. Second, we created a class called

LocationReaction that implements an interface called IReacion and uses the template. This class will be notified every time a new tuple is published if this tuple matches with our template. Finally, in order to use the reaction, the track application subscribed through a method called subscribe, as shown in Fig. 10.

```
sysSUAndroid.subscribe(new LocationReaction(), "put", null);
sysSUAndroid.subscribe(new PolygonReaction(), "put", null);
```

Fig. 10. Subscribing to receive events

In our IReaction implementation, everytime the application received an event of a new context tuple with user's location, the application read synchronously the state of the ambient light, temperature and relative humidity. All these metadata was associated to the location information in a log file. Fig. 11 shows how one can query a context information from tuple space synchronously with our framework.

```
Tuple lightTuple = sysSUAndroid.read("context.ambient.light");
Tuple temperatureTuple = sysSUAndroid.read("context.ambient.temperature");
Tuple humidityTuple = sysSUAndroid.read("context.ambient.temperature");
```

Fig. 11. Querying a context information from tuple space synchronously

Finally, we subscribed again to the framework, this time to receive events when the user entered in a defined region (an area around a square). We created another class implementing IReaction that was responsible to log when this situation occurs. To perform this task, the same template of the previous IReaction were reused. In this case, it was necessary to use the PlaceFilter described in Section 3. To use a filter together with a IReaction, we should request a method called getRestriction, which return this filter. Fig. 12 shows how we created the PlaceFilter instance.

```
Coordinate[] squareCoordinates = new Coordinate[5];
squareCoordinates[0] = new Coordinate(-3.730942, -38.518051);
squareCoordinates[1] = new Coordinate(-3.730674, -38.517778);
squareCoordinates[2] = new Coordinate(-3.73084, -38.517225);
squareCoordinates[3] = new Coordinate(-3.731204, -38.517332);
squareCoordinates[4] = new Coordinate(-3.730942, -38.518051);
PlaceFilter placeFilter = new PlaceFilter(squareCoordinates);
```

Fig. 12. Creating a PlaceFilter instance

This proof of concept shows how the context acquisition layer is isolated from the application code. As seen, the developer of a mobile application should only be concerned about creating filters and query the contextual properties to which the application is interested. No code for accessing the physical and logical sensors is required to be implemented by third application developers. This allows the framework to change how the context acquisition is performed without the application worry about these changes (e.g., sensor errors, sensors reconfiguration to reduce resource consumption). In addition, the framework facilitates the development of this kind of application by hiding the complexity of context acquisition and matching.

5 Conclusion and Future Work

This paper presented an adaptive context acquisition framework to support mobile spatial and context-aware applications. Its main benefits are the flexibility to adapt the context acquisition infrastructure both at deployment time and at runtime, minimizing resource consumption and improving overall uncoupling between the application and the context acquisition infrastructure.

The second benefit is the ease with spatial filters that are implemented and reused, facilitating the use of mechanisms to deal with spatial relations and operations, such as distance among points and regions, and topology relations. Then, spatial and other context information is aggregated, providing mechanisms for applications to perform more elaborate queries.

As a future work, a online CAC repository is going to be developed, permitting CACs to be accessed at runtime. This repository would be used in case of an application interest can not be satisfied. Also, we would like to increase the number of filters to be used with SysSU to facilitate the management of contextual data. Finally, information regarding quality of service (QoS) is going to be annotated on each CAC, permitting applications to consider QoS when deciding which CAC is more appropriate in a given situation.

Acknowledgements. This work is a partial result of the UbiStructure project supported by CNPq (MCT/CNPq 14/2011 - Universal) under grant number 481417/2011-7, and FUNCAP project number PJP-0072-00091.01.00/12. It was also partially funded by the Program of Scientific Cooperation called STIC-AmSud. The sponsored project is entitled Learning While Moving (LWM).

References

1. Abowd, G.D., Dey, A.K.: Towards a Better Understanding of Context and Context-Awareness. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 304–307. Springer, Heidelberg (1999), http://dx.doi.org/10.1007/3-540-48157-5_29
2. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing 2(4), 263–277 (2007)

3. Braga, R.B., de Moraes Medeiros da Costa, S., de Carvalho, W.V., de Castro Andrade, R.M., Martin, H.: A Context-Aware Web Content Generator Based on Personal Tracking. In: Di Martino, S., Peron, A., Tezuka, T. (eds.) W2GIS 2012. LNCS, vol. 7236, pp. 134–150. Springer, Heidelberg (2012), http://www.springerlink.com/index/0U2052W26025Q545.pdf
4. Carriero, N., Gelernter, D.: Linda in context. Commun. ACM 32(4), 444–458 (1989), http://doi.acm.org/10.1145/63334.63337
5. Da, K., Dalmau, M., Roose, P., et al.: A survey of adaptation systems. International Journal on Internet and Distributed Computing Systems 2(1), 1–18 (2011)
6. Eugster, P., Felber, P., Guerraou, R., Kermarrec, A.M.: The many faces of publish/subscribe. ACM Computing Surveys 35(2), 114–131 (2003), http://portal.acm.org/citation.cfm?doid=857076.857078, http://dl.acm.org/citation.cfm?id=857078
7. Furtado, V., Ayres, L., de Oliveira, M., Vasconcelos, E., Caminha, C., D'Orleans, J., Belchior, M.: Collective intelligence in law enforcement - the wikicrimes system. Inf. Sci. 180(1), 4–17 (2010), http://dx.doi.org/10.1016/j.ins.2009.08.004
8. Hong, J., Suh, E., Kim, S.: Context-aware systems: A literature review and classification. Expert Systems with Applications 36(4), 8509–8522 (2009)
9. Hwang, G., Tsai, C., Yang, S., et al.: Criteria, strategies and research issues of context-aware ubiquitous learning. Educational Technology & Society 11(2), 81–91 (2008)
10. Kramer, D., Kocurova, A., Oussena, S., Clark, T., Komisarczuk, P.: An extensible, self contained, layered approach to context acquisition. In: Proceedings of the Third International Workshop on Middleware for Pervasive Mobile and Embedded Computing, p. 6. ACM (2011)
11. Lee, Y., Iyengar, S., Min, C., Ju, Y., Kang, S., Park, T., Lee, J., Rhee, Y., Song, J.: Mobicon: a mobile context-monitoring platform. Communications of the ACM 55(3), 54–65 (2012)
12. Lima, F., Rocha, L., Maia, P., Andrade, R.: A decoupled and interoperable architecture for coordination in ubiquitous systems. In: 2011 Fifth Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS), pp. 31–40. IEEE (2011)
13. Marinho, F.G., Andrade, R.M., Werner, C., Viana, W., Maia, M.E., Rocha, L.S., Teixeira, E., Filho, J.B.F., Dantas, V.L., Lima, F., Aguiar, S.: Mobiline: A nested software product line for the domain of mobile and context-aware applications. Science of Computer Programming (2012), http://www.sciencedirect.com/science/article/pii/S0167642312000871
14. Preuveneers, D., Berbers, Y.: Towards context-aware and resource-driven self-adaptation for mobile handheld applications. In: Proceedings of the 2007 ACM Symposium on Applied Computing, vol. 11, pp. 1165–1170 (2007)
15. Salehie, M., Tahvildari, L.: Self-adaptive software: Landscape and research challenges. ACM Transactions on Autonomous and Adaptive Systems (TAAS) 4(2), 14 (2009)
16. Van Wissen, B., Palmer, N., Kemp, R., Kielmann, T., Bal, H.: Contextdroid: an expression-based context framework for android. In: PhoneSense 2010: International Workshop on Sensing for App Phones, pp. 6–10 (2010)
17. Viana, W., Filho, J.B., Gensel, J., Villanova Oliver, M., Martin, H.: PhotoMap – Automatic Spatiotemporal Annotation for Mobile Photos. In: Ware, J.M., Taylor, G.E. (eds.) W2GIS 2007. LNCS, vol. 4857, pp. 187–201. Springer, Heidelberg (2007)

18. Viana, W.C.: Mobilité et sensibilité au contexte pour la gestion de documments multimédias personnels: CoMMediA. Ph.D. thesis, Université Joseph-Fourier - Grenoble (2010), http://hal.archives-ouvertes.fr/tel-00499550/
19. Vieira, V., Tedesco, P., Salgado, A.C.: Designing context-sensitive systems: An integrated approach. Expert Systems with Applications 38(2), 1119–1138 (2011), http://www.sciencedirect.com/science/article/pii/S0957417410004173; Intelligent Collaboration and Design
20. Weiser, M.: The computer for the 21st century. Scientific American 265(3), 94–104 (1991)

Comparing Close Destination and Route-Based Similarity Metrics for the Analysis of Map User Trajectories

Ali Tahir[1], Gavin McArdle[2], and Michela Bertolotto[1]

[1] School of Computer Science and Informatics, University College Dublin (UCD)
Dublin, Ireland
(ali.tahir, michela.bertolotto)@ucd.ie
[2] National Centre for Geocomputation, National University of Ireland Maynooth (NUIM)
Maynooth, Ireland
gavin.mcardle@nuim.ie

Abstract. Movement is a ubiquitous phenomenon in the physical and virtual world. Analysing movement can reveal interesting trends and patterns. In the Human-Computer Interaction (HCI) domain, eye and mouse movements reveal the interests and intentions of users. By identifying common HCI patterns in the spatial domain, profiles containing the spatial interests of users can be generated. These profiles can be used to address the spatial information overload problem through map personalisation. This paper presents the analysis and findings of a case study of users performing spatial tasks on a campus map. Mouse movement was recorded and analysed as users performed specific spatial tasks. The tasks correspond to the mouse trajectories produced while interacting with the Web map. When multiple users conduct similar and dissimilar spatial tasks, it becomes interesting to observe the behaviour patterns of these users. Clustering and geovisual analysis help to understand large movement datasets such as mouse movements. The knowledge gained through this analysis can be used to strengthen map personalisation techniques. In this work, we apply OPTICS clustering algorithm to a set of map user trajectories. We focus on two similarity measures and compare the results obtained with both when applied to particular saptial tasks carried out by multiple users. In particular, we show how route-based similarity, an advanced distance measure, performs better for spatial tasks involving scanning of the map area.

1 Introduction

Recently there have been huge advances in spatial technologies. The use of Web maps and positioning technologies are now ubiquitous. The drive to generate spatial content has led to a major information overload problem [1], in which finding timely and relevant spatial information becomes a challenge. This makes it difficult for the users to choose and filter content to match their current needs.

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 117–128, 2013.
© Springer-Verlag Berlin Heidelberg 2013

Therefore, there is a need to adapt Web map contents into personalised maps by understanding the user interests. Analysing spatial interaction patterns offers an approach to resolve this.

The research presented in this paper focuses on data generated by HCI in a Web map environment, in particular, the paths generated by users through computer mouse movements. These paths, called trajectories, represent a series of mouse cursors locations. Mouse trajectories reflect usage patterns and activities, which help to predict future movements and interests. In order to analyse our trajectory data, we use Visual Analytics (VA) [2], which promotes cognition and knowledge discovery through data visualisation in a geospatial environment combined with analysis tools such as machine learning, data mining and clustering.

In our approach, clustering of mouse trajectories (obtained as 2D mouse movements) is an integral component in order to determine the usage patterns. We apply the OPTICS clustering algorithm [3], a density-based clustering approach. Such an algorithm requires a metric upon which to determine similarity between trajectories. We compare two similarity metrics, namely close destination and route-based [4], and assess when it is advantageous to use them.

In our previous work [5] we performed a preliminary evaluation based on a small experiment (involving 12 users) using the close destination measure. In this paper we present a larger and more comprehensive user trial in which 27 users participated. Each user was asked to perform ten spatial tasks. The total trajectories collected from all users were analysed in order to see common patterns with the help of visual analysis and clustering techniques. As we are interested in identifying types of user, we analysed multiple users over individual tasks using both route-based and close destination simialrity measures to identify common patterns. Maps of tasks and users are generated to validate the clustering results. The results show that clustering and geovisual analysis is an effective technique to understand user patterns and behaviours. They also show how route-based measures perform better for certain types of tasks.

The goal of this paper is to demonstrate the use of a geovisual analysis tool to analyse user mouse trajectories. In particular, the work assesses the benefits of different distance measures for clustering such trajectories. The results highlight how different users approach particular tasks and will provide input into map personalisation algorithms to resolve the spatial information overload problem.

The remainder of this article is organised as follows. To provide the background for our work, some related literature is presented in Section 2. The overall approach which includes a description of the clustering algorithm, distance measures and spatial tasks is described in Section 3. Section 4 presents experimental results and the evaluation we have conducted. Finally, Section 5 outlines the conclusions and directions for future work.

2 Related Work

Activity recognition from movement data (in particular, spatial trajectories) is challenging due to the huge amount of datasets available. Zhu et al. [6] define this

activity recognition as "the process to extract high-level activity and goal related information from low-level sensor readings through machine learning and data mining techniques". The authors represent trajectory-based activity recognition into three levels. The first level inputs the location from a sensor (for example, GPS, WiFi, cameras). At the next level, an activity is recognised, such as transportation modes. As an example of the final (highest) level, Microsoft GeoLife [1] a Location-based Social Network (LBSN) application, has a transportation detection system which categorises a GPS trajectory into various activities (such as walking, biking, driving and onBus). These transportation modes facilitate activity recognition. GeoLife aims to provide social connectivity between people using their trajectories.

The advances in location-based positioning technologies such as GPS as well as the growth of mobile and wireless technologies have enabled movement data,in the form of spatial trajectories, to be collected. These trajectories can be collected in both outdoor and indoor [7] environments and correspond to free [8, 9] and constrained movements [10, 11]. For example, a road network corresponds to a constrained movement, while the movement of an animal can be termed as free movement. Zheng and Zhou [12] provide several movement data examples. They categorise movement of people into active and passive recordings. For example, active recording occurs when travellers share their travel routes with their friends to strengthen social connectivity which is evident in the context of Location-based Social Networks (LBSN) applications. On the other hand, a user carrying a mobile phone unintentionally generates spatial trajectories. These trajectories correspond to a sequence of cell tower IDs. Other categories include the mobility of vehicles (which can be used for traffic analysis and resource allocation) and the mobility of animals and natural phenomenons (such as migratory birds research, hurricanes and tornadoes).

While most researchers focus on physical trajectories, mouse, eye and touch gestures on a computer screen are also forms of movement which generate specific trajectories. When interacting with a spatial application, such as a Web map, these trajectories can be analysed to examine Human-Computer Interaction (HCI), the study of interaction between humans and machines [13]. We are interested in analysing trajectories generated by HCI in our research, particularly in the spatial interaction domain.

Typical activity recognition involves monitoring a single as well as multiple user activities. Zhu et al. [6] present a detailed account of single and multiple users activity recognition from trajectory data. The study shows that much work has been done on single users while less attention has been given to multiple user activity analysis. Single user activity recognition corresponds to the analysis of an individual's user history in order to predict future trends. In general, single user activity can be recognised through supervised methods, unsupervised methods and frequent pattern mining approach [6]. Supervised learning methods typically input trajectory data as well as activity labels. The intention is to use a suitable classification model such as Bayesian networks, Hidden Markov Model

[1] `http://research.microsoft.com/en-us/projects/geolife/`

(HMM) and decision trees, in order to predict the activity of the trajectory in question. These methods attempt to identify activities within a trajectory, for example, significant places, stop rate, velocity change, etc. Unsupervised methods on the other hand do not take activity labels into account, but instead discover patterns directly from the trajectory data [6], for instance, by applying clustering methods. For this reason, in the case of mouse trajectories, unsupervised methods are found to be more relevant. To the best of our knowledge, mouse trajectories on a map interface have not been studied in order to understand user's intentions and behaviours.

In our previous work, we have applied spatial clustering to mouse trajectories in order to identify usage patterns [5]. The approach successfully identified spatial tasks with the help of clustering while some outliers were also detected. As an extension to spatial clustering, we presented clustering based on temporal information which considers speed and acceleration at each location in a trajectory to describe behaviour [14]. A detailed discussion on clustering algorithms suitable for trajectories as well several distance measures is presented in [5, 14]. The research presented here is an extension to this. We adapt an advanced similarity measure (route similarity) [4] and use this as a spatial distance measure in OPTICS clustering. In previous work, we used close destination distance measures to find the similarity between trajectories. This function considers the end points of each trajectory while computing the trajectory similarity. Now we compare the distance between each point of each trajectories in order to produce a new overall similarity score.

3 Approach

In this section we present our approach. Firstly, we conducted a user trial in which users performed several spatial tasks. The trials were conducted in an unsupervised manner, similar to a previous set of trials described in [5]. The participants had to register in order to start the spatial tasks. The Web interface, designed for this purpose, was deemed to be as user friendly as possible as shown in Figure 1. The interface contains different components on a Web page including the spatial task description, an input area for answers, a mapping component and a map legend. The geographic area selected was that of the University College Dublin (UCD) Belfield Campus.

The 10 spatial tasks presented to users are listed below, while the Web interface is shown in Figure 1.

- Task 1: As a new student, you are required to open up a student account with the bank located at the campus. Find the bank on the campus map. On your way back, you would also like to get some food. Which is the nearest restaurant to the bank?
- Task 2: You finish a lecture at the John Henry Newman Building (Arts) and need to catch a bus to the city centre (all bus stops connect to the city). Find the bus stop most convenient to you on campus. How many bus stops are there on campus?

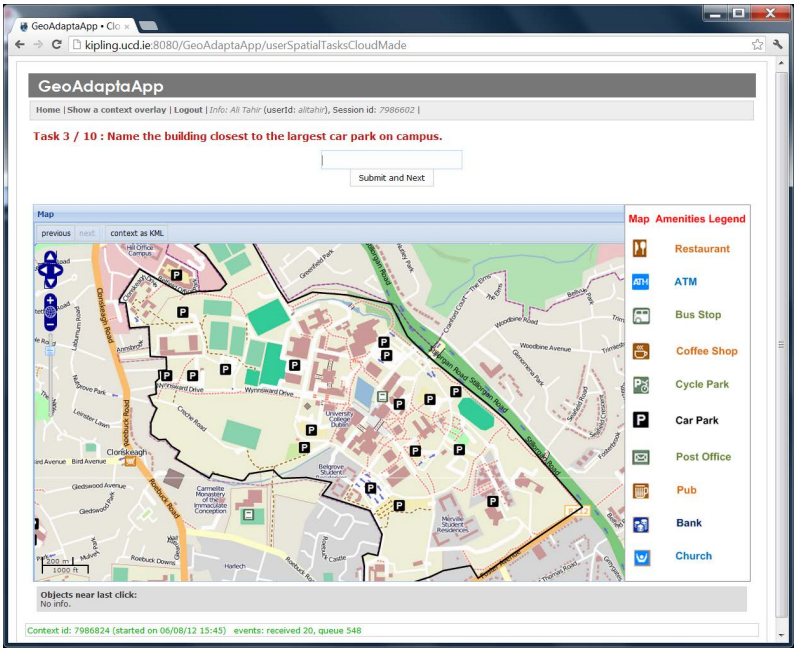


Fig. 1. Web interface for experiments

- Task 3: Name the building closest to the largest car park on campus.
- Task 4: In order to attend a service at the church located on campus, you need to find the closest car park to the church. Find your way to this car park if you are driving from the Stillorgan Road/N11 entrance.
- Task 5: You need to attend a graduation ceremony at UCD. You will be driving to UCD and enter from the Stillorgan Road/N11 entrance. Locate the reception and O'Reilly Hall and find the shortest route between the two.
- Task 6: You need to drive to the student bar in the evening. You are required to enter the UCD campus from the Wynsward Drive entrance. Follow the path to find the closest car park to the student bar. How many pubs are shown on the map?
- Task 7: You finish a lecture in Computer Science and Informatics (CSI) building. You need to meet your friend in front of the Health Science Centre and go together to the James Joyce Library (close to the central largest lake - coloured in blue) in order to return a book. Plan the route.
- Task 8: In order to post a letter to your friend, plan the route to cycle from Charles Institute located at North West of the central lake (coloured in blue) to Belfield Post Office. Find the nearest bicycle parking stand to the post office.

– Task 9: You need to meet your friends at the sports centre building (coloured in green). How would you get to the Glenomena student residence (south east of the central lake) from the sports centre?
– Task 10: Count the number of roads crossing the UCD boundary (as outlined by the black line).

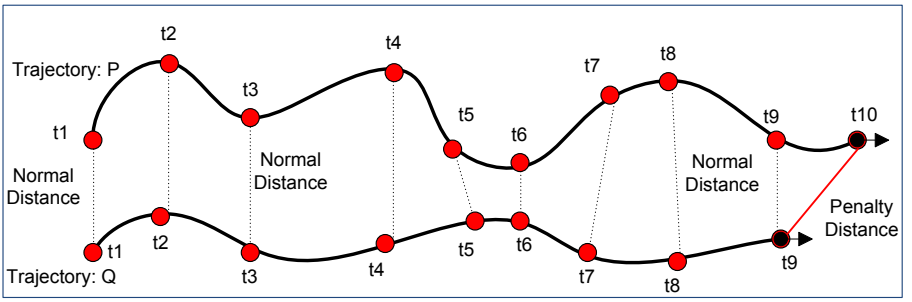


Fig. 2. Normal distance and penalty distance between two trajectories. In case of non-corresponding points between two trajectories, penalty distance is calculated (shown in red), and added to the overall distance between two trajectories.

The trajectories produced by the user mouse movements were analysed using the OPTICS clustering algorithm. OPTICS [3] produces an ordering of a dataset while it searches for a core distance and a reachability distance of each trajectory with respect to its predecessor. OPTICS outputs a reachability plot. From the output plot, groupings can be obtained by choosing an appropriate threshold value of reachability distance. Let ρ = object from a dataset D, ε = distance threshold, $N\varepsilon$ (ρ) = ε-neighborhood of object ρ, *minPts* = natural number, *minPts-distance*(ρ) = distance from ρ to its *minPts* neighbor. The core distance (CD) is defined as:

$$CD = \begin{cases} Undefined, & \text{if } Card(N\varepsilon(\rho)) < minPts \\ minPts\text{-}distance(\rho), & \text{otherwise} \end{cases}$$

The *core distance* is the smallest distance ε between ρ and an object in its ε-neighborhood such that ρ would be a core object. The *core distance* is *Undefined*, otherwise. For reachability distance, let ρ and o = objects from a dataset D, $N\varepsilon$ (o) = ε-neighborhood of object o, *minPts* = natural number. The reachability distance (RD) of ρ with respect to o is defined as:

$$RD = \begin{cases} Undefined, & \text{if } Card(N\varepsilon(o)) < minPts \\ max(core\text{-}distance(o), distance(o, \rho)), & \text{otherwise} \end{cases}$$

Thus, the reachability distance of ρ is the smallest distance such that ρ is directly density-reachable from a core object o. Otherwise, if o is not a core object, even

at the generating distance ε, the reachability distance of ρ with respect to o is *Undefined*.

In order to apply this clustering algorithm, an appropriate similarity measure is required. The similarity measure is used with the clustering algorithm along with a distance threshold and the minimum number of neighbours required to form a cluster. Route similarity [4] is a complex (computationally expensive) distance measure that computes the geographical distance between two trajectories. Moreover, this distance function deals with incomplete trajectories and with more significant positioning errors. Although mouse trajectories do not have positioning errors, they are uncertain and hold frequent sequences of movements. This function is also more suited to unequal time intervals between records (the case of mouse trajectories). The function repeatedly searches for the closest pair of positions in two trajectories. It computes two derivative distances: mean distance between the corresponding positions and a penalty distance for the unmatched positions. The penalty distance is increased if a position is skipped and decreased if the corresponding position is found. The final distance is the sum of two derivative distances. We have opted to use this measure in order to compute the distance between two trajectories. Figure 2 illustrates the concept of normal distance and the penalty distance between two trajectories. The next section presents results and evaluation we have conducted.

4 Evaluation

In order to demonstrate the effectiveness of route similarity with OPTICS, we carried out a series of analysis tasks. Although our main interest lies with route similarity applied to individual tasks, initially we show the effects of applying close destination clustering to all trajectories produced from the user trial as the results provide an example of a situation where this is not an effective metric. Additionally, the approach highlights how to use and interpret the output from the OPTICS algorithm which we also used to compare the route-based and close destination clustering of trajectories from specific tasks.

Figure 3 shows the results of applying OPTICS clustering with close destination similarity measure to all meaningful trajectories collected during the user trial. While a 100 percent completion rate by all participants would yield *270* trajectories, not all users interacted with the map during trials, or completed all tasks and so *258* trajectories were recorded. Close destination clustering requires a distance parameter and a minimum number of neighbours to be specified, these were set to 1000m and 7 neighbours respectfully. The graph produced reveals the clusters present in the trajectories. Clusters can be identified using the rules provided by the authors of [2]: "the first point of a cluster (called the start of the cluster) is the last point with a high reachability value, and the last point of a cluster (called the end of the cluster) is the last point with a low reachability value". Using this rule a cluster starting at position *21* and ending at position *40* can be extracted. Similarly, another cluster starting at position *97* and ending at position *113* can be identified.

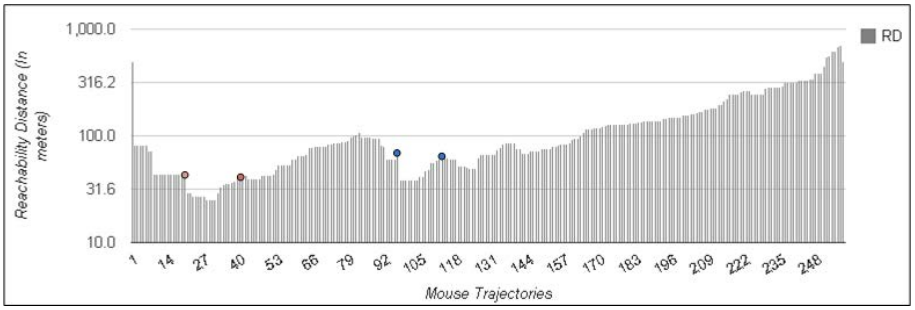


Fig. 3. Similarity measure: close destination, distance: 1000 meters, neighbours: 7, Trajectories: 258

Once extracted, the trajectories can be visually analysed, for example, in Figure 4 all trajectories forming a cluster are assigned the same colour. The black symbol indicates the end of each trajectory. Overall, this clustering approach did not produce clusters corresponding to specific tasks or specific users. In this approach trajectories which terminate in close proximity to each other are deemed similar and so should cluster trajectories based on the task they are part of (i.e. terminating at the specific spatial feature of interest in a given task). However, due to the small study area, the trajectories terminate in close proximity for many tasks and so few distinct clusters were identified. Importantly, several of the tasks do not require the identification of a specific spatial feature but require the user to scan the map, count features or identify routes. As a result, similarity based on close destination is not always effective and is task specific.

As our interest is in individual tasks, we applied route-based similarity measures, with OPTICS clustering on a task-by-task basis and compared it to the results obtained using close destination. Initially, we considered task 1 (described in Section 3). This task required users to locate a bank and a nearby restaurant on the campus map. Task 1 was performed by *25* users which included *15* familiar and *10* non-familiar (with the map area) users. Figure 5 (a) shows the outcome when OPTICS was applied to the *25* user trajectories using a close destination similarity measure along with the chosen input parameters (distance: 300m, neighbours: 5). It can be seen that the reachability chart in this case does not produce a definitive set of clusters. This is due to the nature of the task involving scanning the map and identifying *2* locations. Figure 5 (b) shows the results using a route similarity measure (distance parameter: 2500m, neighbours: 5). Here, three distinct clusters are identified. These clusters are shown as an overlay on the study area in Figure 6. The clusters clearly identify the approach users took to complete the tasks and can be broadly broken down into users who were familiar and unfamiliar with the campus.

To further demonstrate the effectiveness of using a route similarity metric we also analyse the results for task 10. In task 10, users were required to count the number of roads entering the UCD campus. This is an example of a situation in which close destination similarity is not effective because users are free to

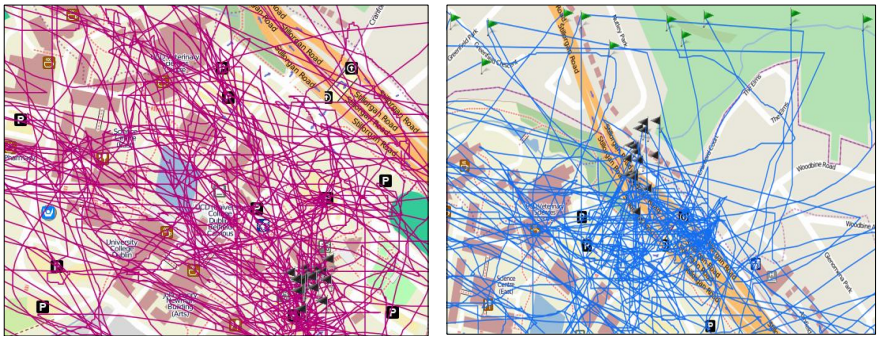


Fig. 4. (a) Cluster showing 22 trajectories starting from location 21 and ending at 40. (b) Cluster showing 22 trajectories starting from location 97 and ending at 113.

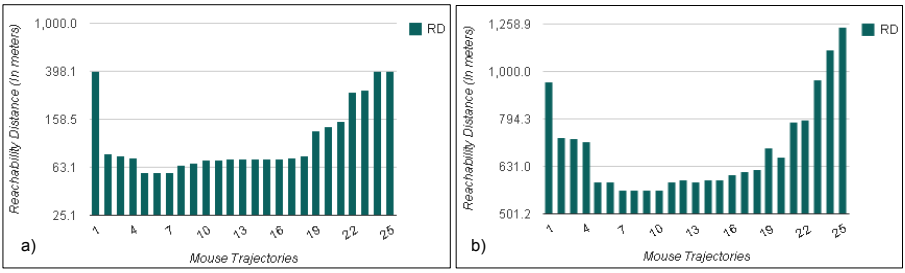


Fig. 5. Task 1- a) Similarity measure: close destination, distance: 300 meters, neighbors: 5, trajectories: all b) Similarity measure: route similarity, distance: 2500 meters, neighbors: 5, Trajectories: all

start and end their trajectories at arbitrary locations. This task involves scanning the map to identify roads crossing the campus boundary. This task was performed by all *27* users. Both close destination and route similarity measures were used in conjunction with OPTICS and the results are shown in Figure 7. The parameters used were a distance of 2500m and 5 neighbours. The close destination metric did not identify a distinct set of clusters whereas route similarity clearly identified two predominant groups of trajectories. These two clusters can be visually analysed in Figure 8. The first cluster (Figure 8 (a)) is found to be very neat where users precisely followed the path along the boundary line of the campus map. These trajectories correspond to users who were familiar with the campus and are in contrast to the trajectories in Figure 8 (b) where there is seemingly random mouse movement and trajectory shape. These represent users who indicated they were unfamiliar with the campus.

In conclusion, the use of a close destination similarity metric is not effective for all tasks. This is especially true for tasks which do not have a specific goal or target and tasks taking place in a small geographic area. For scanning tasks,

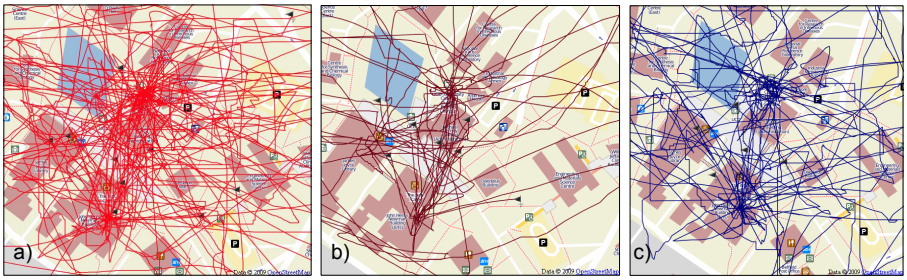


Fig. 6. Task 1- a) First cluster (1-11) b) Second cluster (12-18) c) Third cluster (19-25)

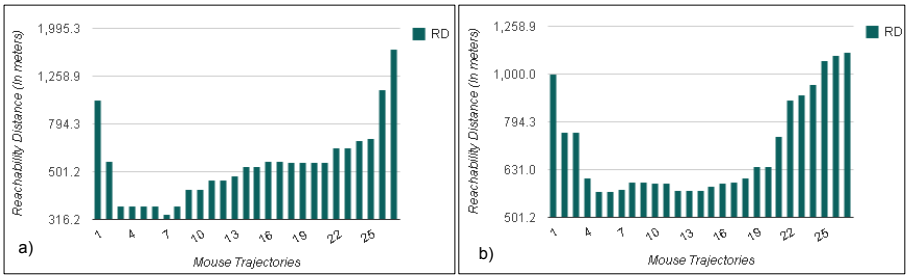


Fig. 7. Task 10- a) Similarity measure: close destination, distance: 2500 meters, neighbors: 5, trajectories: all b) Similarity measure: route similarity, distance: 2500 meters, neighbors: 5, trajectories: all

it is more advantageous to use a similarity metric which looks at the complete path of a trajectory such as route-based similarity. The results here show that this is particularly useful for examining the behaviour of users and determining their familiarity with the study area and the task at hand.

5 Conclusion and Future Work

This paper presents clustering and geovisual analysis of mouse trajectories collected from users performing spatial tasks. By analysing users as they perform such tasks, parameters which can be used in map personalisation can be collected. In this paper we describe a tool we have developed which applies OPTICS clustering and geovisualisation to analyse HCI data. The paper focuses on the use of similarity measures for comparing and clustering trajectories of users completing specific tasks. In particular we examine close destination and route-based similarity measures. Close destination similarity identifies trajectories which terminate in the same region whereas route similarity measures consider the complete trajectory path. We apply these techniques to *258* mouse trajectories collected during a user trial in which participants had to complete *10*

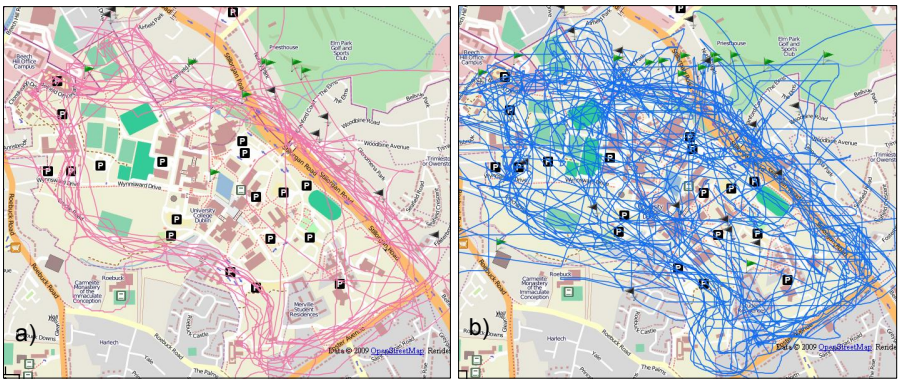


Fig. 8. Task 10- a) Small cluster (1-10) b) Large cluster (11-27)

specific spatial tasks using a Web map. The results highlight the benefits of using the route-based approach for tasks which involve scanning the map or identifying more than one spatial feature, and show it is effective at highlighting users' familiarity with the study area. As route-based clustering considers the trajectory shape, it also shows the approach users take to complete a task. This was validated through visual analysis of cluster assignment and via a questionnaire presented to users.

The results reinforce the importance of choosing the correct clustering approach based on the task at hand. Individual user trajectories for several spatial tasks can be clustered to reveal usage patterns for specific users or participants. The participants can also be distinguished based on their map usage experience and the area familiarity. Similarly mouse trajectories can be clustered by taking map scale factor into account which at the moment is part of the visual analysis. To further strengthen and validate the clustering techniques, heat maps and speed maps can be combined with the results [15]. Overall, the paper has demonstrated an approach for the analysis of map based HCI. The next stage is to apply the results to map personalisation. This can be achieved by adapting map content based on the users' interests, ability and approach as determined through the cluster analysis presented here.

Acknowledgements. Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support.

References

[1] Yang, Y., Claramunt, C.: A Hybrid Approach for Spatial Web Personalization. In: Li, K.-J., Vangenot, C. (eds.) W2GIS 2005. LNCS, vol. 3833, pp. 206–221. Springer, Heidelberg (2005)

[2] Thomas, J., Cook, K.: A visual analytics agenda. IEEE Computer Graphics and Applications 26, 10–13 (2006)
[3] Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: Ordering Points to Identify the Clustering Structure. SIGMOD Rec. 28, 49–60 (1999)
[4] Andrienko, G., Andrienko, N., Wrobel, S.: Visual analytics tools for analysis of movement data. SIGKDD Explorations Newsletter - Special Issue on Visual Analytics 9, 38–46 (2007)
[5] Tahir, A., McArdle, G., Bertolotto, M.: Identifying specific spatial tasks through clustering and geovisual analysis. In: 2012 20th International Conference on Geoinformatics (GEOINFORMATICS), pp. 1–6. IEEE (2012)
[6] Zhu, Y., Zheng, V.W., Yang, Q.: Activity recognition from trajectory data. In: Zheng, Y., Zhou, X. (eds.) Computing with Spatial Trajectories, p. 180. Springer, New York (2011)
[7] Jensen, C.S., Lu, H., Yang, B.: Indexing the Trajectories of Moving Objects in Symbolic Indoor Space. In: Mamoulis, N., Seidl, T., Pedersen, T.B., Torp, K., Assent, I. (eds.) SSTD 2009. LNCS, vol. 5644, pp. 208–227. Springer, Heidelberg (2009)
[8] Güting, R., Böhlen, M., Erwig, M., Jensen, C., Lorentzos, N., Schneider, M., Vazirgiannis, M.: A foundation for representing and querying moving objects. ACM Transactions on Database Systems (TODS) 25, 1–42 (2000)
[9] Forlizzi, L., Güting, R., Nardelli, E., Schneider, M.: A data model and data structures for moving objects databases, vol. 29. ACM (2000)
[10] Güting, H., de Almeida, T., Ding, Z.: Modeling and querying moving objects in networks. The VLDB Journal, The International Journal on Very Large Data Bases 15, 165–190 (2006)
[11] Speičvcys, L., Jensen, C., Kligys, A.: Computational data modeling for network-constrained moving objects. In: Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems, pp. 118–125. ACM (2003)
[12] Zheng, Y., Zhou, X.: Computing with Spatial Trajectories. Springer-Verlag New York Inc. (2011)
[13] Dix, A., Finlay, J., Abowd, G.: Human-Computer Interaction. Prentice Hall (2004)
[14] McArdle, G., Tahir, A., Bertolotto, M.: Spatio-temporal clustering of movement data: An application to trajectories generated by human-computer interaction. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences I-2, 147–152 (2012)
[15] Tahir, A., McArdle, G., Bertolotto, M.: A geovisual analytics approach for mouse movement analysis. International Journal of Data Mining, Modeling and Management. Special Issue on Spatial Information Mining, Modeling and Management (2012) (in press)

A Sensor Data Mediator Bridging the OGC Sensor Observation Service (SOS) and the OASIS Open Data Protocol (OData)

Chih-Yuan Huang[1], Steve Liang[1], and Yan Xu[2]

[1] Department of Geomatics Engineering, University of Calgary, Alberta, Canada
{huangcy,steve.liang}@ucalgary.ca
[2] Microsoft Research, Redmond, Washington, USA
yanxu@microsoft.com

Abstract. The World-Wide Sensor Web is generating tremendous amount of real-time sensor data streams, and will enable scientists to observe phenomena that are previously unobservable. As the concept of sensor web is to connect all the sensors and their data to achieve shared goals, improving the openness and accessibility of sensor data is important. Open Geospatial Consortium Sensor Observation Service (SOS) defines standard web service protocols for sharing sensor data online in an interoperable manner. However, the SOS has a relatively weak ecosystem, which makes it difficult to build and consume; and it only supports predefined queries. On the other hand, the OASIS[1] Open Data Protocol (OData) has a strong ecosystem and flexible query functions. But the soft-typing approach of OData requires it to have a commonly agreed data model to be interoperable. As we find that the two standards can benefit from each other, we propose a sensor data mediator solution and define an SOS entity data model for OData (SOS-OData) to bridge these two standards. Our prototype demonstrates that the proposed system can convert between existing SOS services and SOS-OData services. As a result, we can not only consume SOS data with the flexible OData protocol, but can also easily build an SOS-compliant service with the strong OData ecosystem. We argue that the bridge between these two standards would lead us to the vision of *open data for sensor web*.

Keywords: Open Geospatial Consortium (OGC), Sensor Observation Service (SOS), Open Data Protocol (OData), Sensor Data, Mediator, Open Data.

1 Introduction

The World-Wide Sensor Web [1] is generating tremendous volumes of real-time sensor data streams ranging from video camera networks monitoring real-time traffic to matchbox-sized wireless sensor networks embedded in the environment to monitor habitats. As these data streams enable scientists to observe phenomena that were previously unobservable, the sensor web is increasingly attracting interests from a wide

[1] OASIS: Organization for the Advancement of Structured Information Standards.

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 129–146, 2013.
© Springer-Verlag Berlin Heidelberg 2013

range of applications, including: habitats monitoring [2], environment observation [3], structure health monitoring [4], human health applications [5], fire emergency response [6], etc.

As the concept of sensor web is to connect all the sensors in the world and their data together to achieve shared goals [7], improving the openness and accessibility of sensor data, which we refer to as the *open data for sensor web*, is one of the major objectives. However, according to the *long tail* phenomenon in the sensor web that we have previously observed [8], this sensor web long tail could cause issues on the open data for sensor web vision. As shown in the Fig.1, we divide the long tail into three parts, namely the *head*, the *middle*, and the *tail*. The head mainly consists of large scale sensor arrays operated by national organizations, such as NOAA, NASA, and NRCan. Although the number of these large organizations is small but they collect a vast amount of sensor data. The middle contains medium size sensor arrays, which are usually maintained by provincial organizations. Comparing to the head, the middle has more operating organizations but smaller amount of sensor data. The sensor data in the tail are collected by small organizations or individuals, such as small research groups and scientists. However, unlike the head and the middle, the small organizations and individuals in the tail usually do not maintain sensor arrays for a long period of time; instead, they collect data based on the need of their short term projects. Therefore, although there are a large number of participants in the tail, the amount of sensor data they collect is small.

Based on the definition of the long tail [9], we know that the summation of the sensor data in the tail has a similar size with that in other parts. However, the sensor data in the tail are usually not accessible to the public due to the lack of *easy* and *interoperable* ways to share the data online. Hence, we call the sensors in the tail as *missing sensors* or *dark sensors*. In order to address the interoperability issue on sharing sensor data online, the Open Geospatial Consortium (OGC) Sensor Web Enablement (SWE) working group defines open standard protocols and data models for sensor devices and sensor data [10]. It is similar to the World Wide Web Consortium (W3C) that defines open standards for the World Wide Web (WWW) to provide an interoperable way for people to communicate on the Internet. Therefore, by following the

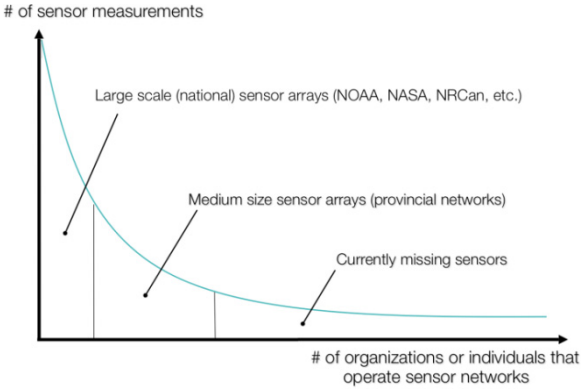


Fig. 1. The sensor web long tail

OGC SWE standards, organizations or individuals can share sensor data in an interoperable manner, and it helps us to *capture* the long tail and achieve the vision of open data for sensor web.

Among the OGC SWE specifications, the Sensor Observation Service (SOS) defines a web service protocol for accessing sensor observations in a standard way [11]; the Observation and Measurement (O&M) defines the standard models and XML schema for observations and measurements collected by sensors [12]; and the Sensor Model Language (SensorML) specification provides the standard models and XML schema for representing the metadata of sensor systems and processes [13]. These three standards will be further discussed in this paper, especially the SOS standard.

As SWE standards provide an interoperable way to access sensors and their data, there have been some sensor web applications built with the SWE standards. For example, the Groundwater Information Network (GIN)[2] uses the OGC Web Mapping Service (WMS) [14] to present a map of sensor locations at small scale and uses the OGC Web Feature Service (WFS) [15] to show the actual sensor observations at large scale. The Geospatial Cyberinfrastructure for Environment Sensing (GeoCENS)[3] built an online sensor web browser that can retrieve and display sensor data from WMS and SOS services.

In this paper, we focus on the SOS standard. Like most standards in the OGC SWE, the SOS is a hard-typing approach specifically designed for the sensor web. As other specifications define a comprehensive and concrete conceptual model for sensor web components (e.g., the O&M and SensorML specifications), the SOS defines a standard web service protocol of sharing sensor metadata and sensor observations.

However, we find that there are three major issues in the real-world SOS services. The first issue is that the real-world SOS services have heterogeneities in terms of syntactic, semantic, and implementation. The syntactic heterogeneity here refers to the various ways of encoding and representing the same concept. For example, “windspeed”, “WIND_SPEED”, “ogc:urn:ucalgary:geomatics/Ontology.owl#Wind** Speed” are used in some existing SOS services to represent the concept of “wind speed”. The semantic heterogeneity means that the semantic information between observed properties (i.e., URIs representing phenomena) is missing. An example would be the relationship between “precipitation” and “rain”. The implementation heterogeneity means that some real-world SOS services do not fully conform to the SOS specification, which directly makes SOS-compliant clients simply insufficient for data access. For example, some services swap the positions of latitude and longitude in XML responses, and some services use time string formats that do not conform to the ISO 8601 standard. We argue that the implementation heterogeneity comes from human negligence or misunderstandings about the SOS specification, which makes it difficult to discover and could vary largely from service to service. As the semantic and syntactic heterogeneities are out of the scope of this paper, the implementation heterogeneity directly damages the interoperability of SOS. The second issue is the relatively weak ecosystem of SOS. Since the SOS is specifically related to

[2] http://analysis.gw-info.net/gin/public.aspx
[3] http://www.geocens.ca/

the sensor web and the geoweb domain, it unavoidably limits the number of participants and developers in this community. This relatively weak ecosystem causes difficulties on finding resources (e.g., libraries, implementations, and documentations), which consequently make the implementation heterogeneity even worse. Finally, the third issue is that the SOS only supports some defined query operations, such as queries by sensor identifiers and spatio-temporal cubes (i.e., combinations of bounding boxes and time periods). For example, the SOS does not support any filtering functions (e.g., =, ≠, >, <, etc.) or aggregate functions (e.g., average, count, summation, etc.) on sensor measurement values. However, it is foreseeable that as the number of sensor observations grows rapidly, flexible query operations become necessary to efficiently discover the sensor data of interest.

On the other hand, Microsoft presents the Open Data Protocol (OData)[4], which is a web protocol for querying and updating data based on HTTP, Atom publishing protocol (AtomPub), and JSON. In June 2012, OData has been proposed to OASIS, an international open standard consortium. Comparing to the SOS, the OData protocol has a much stronger ecosystem. That is because OData is designed for the general Web, and SOS is designed specifically for the geoweb and sensor web community. The OData ecosystem has abundant resources including consumers (i.e., applications to process OData), applications (i.e., applications exposing OData), producers (i.e., live OData services), sample services, sample code, and libraries in various programming languages and for different platforms. As a result, the implementation heterogeneity would be less severe than that in the SOS. Hence, it is easier for developers and data providers to publish and use OData instances than SOS instances. In addition, the OData protocol supports flexible query and filtering functionalities on any types of properties. As we mentioned earlier, flexible query functions are important for data in large size. However, as the OData aims on supporting various kinds of applications, it adopts a soft-typing approach without defining any domain specific information model. Therefore, we argue the major issue of OData is that it requires a commonly agreed conceptual model and schema to be actually interoperable.

In summary, the SOS and the OData standards both have their strength and weakness. For example, as shown in Table 1, we compare these two in terms of the conceptual model, query function, and ecosystem. As the SOS has a comprehensive and complete conceptual model for the sensor web, the SOS supports only some defined query functions and is difficult to implement because of the relatively weak ecosystem. On the other hand, the OData supports query functions on any types of properties and easy to construct and consume with the strong ecosystem; but it needs a commonly agreed conceptual model and schema to be interoperable. Therefore, through the comparison, we found that these two standards can be complementary to each other. For example, we can apply the SOS conceptual model on the OData, and the OData can benefit the SOS with its strong ecosystem and flexible query functionalities. As a result, the major objective in this paper is to design and implement a sensor data mediator that bridges the SOS and the OData and harvests the benefits from both of them.

[4] http://www.odata.org/

Table 1. A comparison between the SOS and the OData

	The SOS	The OData
Conceptual model	Comprehensive and concrete conceptual model defined for the sensor web components.	Requires users to define a conceptual model and schema.
Query function	Only supports defined query functions.	Flexible query and filtering functions on any types of properties.
Ecosystem	Relatively weak ecosystem.	Strong ecosystem.

In order to achieve the objective, the main idea is to propose an SOS profile for OData (i.e., SOS-OData), which is an OData Entity Data Model (EDM) applying the SOS conceptual model and schema. With the SOS-OData, we can transform the data in an SOS service to an OData service. In this case, we can consume the SOS data with the standard OData protocol, which provides flexible query and filtering functions. On the other hand, with the strong OData ecosystem, we can easily construct an SOS-OData service. And since the data model matches the SOS conceptual model, the SOS-OData service can be converted to an SOS service, which consequently allows us to consume the SOS-OData service with the standard SOS protocol.

This paper is organized as follows. Section 2 introduces the detail background of the SOS and the OData standards. In Section 3, we present the proposed solution, including the system design and the SOS-OData data model. Section 4 presents the current implementation and prototype. Finally, Section 5 discusses conclusions and future work.

2 The SOS and OData Standards

In order to clearly convey the proposed solution, it is necessary to introduce the SOS and the OData standards in detail.

2.1 The Sensor Observation Service (SOS)

The SOS is one of the standards that make up the OGC Sensor Web Enablement (SWE) [10]. Besides the SOS, there are other standards in SWE, such as the Sensor Model Language (SensorML) [13] and the Observations & Measurements (O&M) [12] specifications. As the O&M standard defines the conceptual model and data encodings for sensor observations and the SensorML models the sensor metadata, the SOS defines a standard protocol that allows clients to retrieve sensor observations and metadata from SOS services. The first version of SOS was published in 2008 [11]; and the second version of SOS was approved in 2012 [16]. Although there are some differences between these two versions, the parts that we are going to introduce are very similar and this paper focuses on the first version of SOS.

In the SOS, a sensor observation is modeled as an event producing an estimate of a *property* (i.e., a characteristic of one or more feature types, also called *phenomenon*) of a *feature of interest* (i.e., features or feature collections that represent the

identifiable object(s) on which the sensor systems are making observations). This definition is presented in the O&M specification, and the model is summarized in Fig. 2. In order to provide sensor observations in an aggregate manner, an SOS organizes collections of related sensor observations into *observation offerings*.

The SOS provides three profiles of operations for different purposes, namely (1) core operations profile, (2) transaction operations profile, and (3) enhanced operations profile. In this paper, we only discuss the core operations. The core operations are defined for clients to retrieve observations from an SOS service. There are three operations in core profile, which also are the only mandatory operations in the SOS. First, the *GetCapabilities* operation allows clients to retrieve the service metadata including information about the SOS version, the service provider, allowed values on querying sensor observations, and the metadata of observation offerings. With the information in a *GetCapabilities* response, clients can use the other two core operations to get sensor metadata and observations. The *DescribeSensor* operation allows clients to obtain the metadata describing the characteristics of an observation procedure (i.e., a sensor or a sensor constellation) with a Uniform Resource Identifier (URI) of procedure. The metadata of sensors is usually encoded in SensorML format. In addition, clients can use the *GetObservation* operation to retrieve sensor observations in O&M format by specifying an observation offering and one or many observed properties. In addition, clients can set spatio-temporal constrains (i.e., bounding boxes and time periods) in the *GetObservation* request to further filter the returned observations.

In addition, based on the SOS specification, we can know the relationships between entities in the SOS. The relationships are summarized in Table 2. There are five entities in the Table 2 (i.e., the Capability, Observation Offering, Observed Property, Procedure, and Feature of Interest); and the relationships between them are shown as subject-relation-object triples. One additional entity type not shown in Table 2 is the Observation. Based on our understanding, the multiplicity relationships between Observation and other entities are not clearly stated in the specification. In this paper, we consider that an Observation Offering holds a "zero or more" relation with the Observation entity.

2.2 The Open Data Protocol (OData)

Currently, the OData has few versions in place. In this paper, we discuss the OData version 2 [17] simply because we use an OData version 2 library. As the detail explanation about the OData version 2 can be found online[5], this subsection summarizes the necessary information for this paper. The OData defines a HTTP protocol to publish, query, and update data in data services through web clients sending simple HTTP requests. The cores of OData are *feeds*, which are *collections* of *entities*. As each entity represents a structured record with a key and a list of *properties* of primitive or complex types, entities can have related entities and related feeds through *links*. As an OData service may contain one or more feeds, OData services provide a service document that lists all the top-level feeds so users can access them. Also, through the

[5] `http://www.odata.org/documentation/IndexV2`

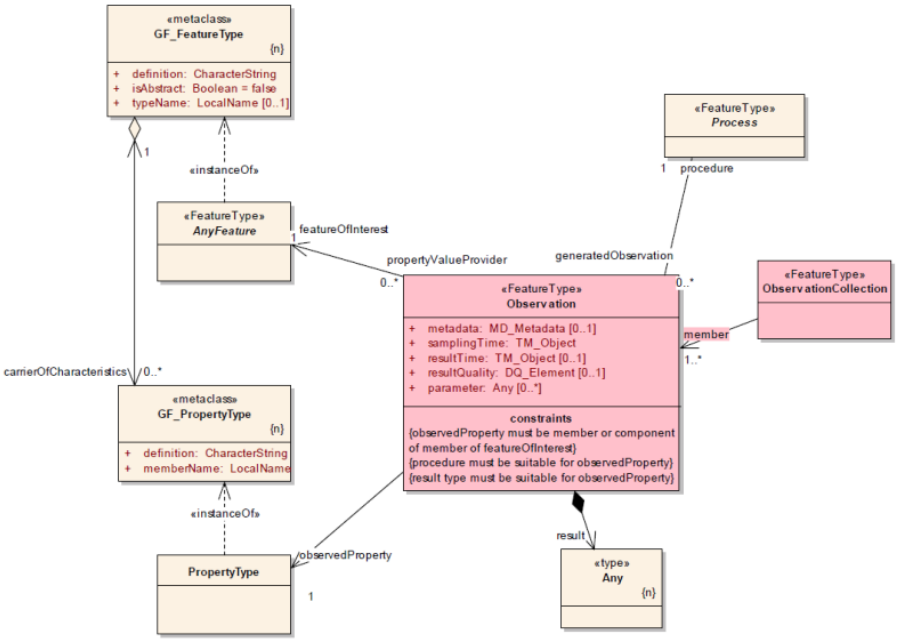


Fig. 2. The O&M model [11]

Table 2. Relationships between SOS entities

Subject Entity	Multiplicity Relation	Object Entity
Capability	Has one or more	Observation Offering
Observation Offering	Has one or more	Observed Property
Observation Offering	Has one or more	Procedure
Observation Offering	Has one or more	Feature of Interest

"$metadata" request, each OData service provides a service metadata document describing its entity data model (EDM). Table 3 further explains the resources in OData.

The OData also defines the URI conventions, a set of rules for constructing URIs to identify the data and metadata served by an OData service. In this paper, we only explain the two kinds of basic rules, namely the resource path and the query string options. As the example shown in Fig. 3, there are three major URI components, which are the service root URI, the resource path, and the query options located after the question symbol "?".

The resource path is mainly for addressing a collection, an entity in a collection, or a property of an entity. To address a collection, the resource path is simply the name of that collection. To address an entity in a collection, the resource path is the name of that collection with a key predicate located in brace marks. In order to address a property of an entity, we need to first locate the entity and then identify the name of the property after it. In addition, the resource path can be used to resolve the navigation links. Also, the resource path supports two additional operations called "$value" and

Table 3. Descriptions of OData resources

OData Resource	Description
Entity	Entities are instances of *entity types* which are structured records consisting of named and typed properties and with a key.
Property	Primitive or complex type property of an entity.
Complex Type	Complex types are structured types consisting of a list of properties but with no key.
Link	Link is a navigation property defining the relationship between two or more entity types, which is bound to a specific relationship and can be used to refer to associations of an entity.
Collection	Entities with the same entity type are grouped in collections.
Service Operation	OData service can serve service-specific functions that accept input parameters and return entities or complex/primitive values.

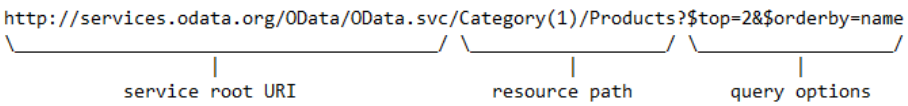


Fig. 3. An example of OData URI components

"$count". The "$value" operation return the value of the selected property. The "$counts" operation can be used on a collection of entities to identify the number of entities. Fig.4 shows the examples of addressing different kinds of resources.

For the query string options, the OData provides eight system query options, which are "$orderby", "$top", "$skip", "$filter", "$expand", "$format", "$select", and "$inlinecount". In Table 4, we briefly introduce the "$filter" and the "$select" options as they are used in this paper.

However, we found that there is one issue in the OData. Unlike many database management systems (DBMSs) that can join tables when performing queries, the

```
http://services.odata.org/OData/OData.svc/Categories
```
(a)
```
http://services.odata.org/OData/OData.svc/Categories(1)
```
(b)
```
http://services.odata.org/OData/OData.svc/Categories(1)/Name
```
(c)
```
http://services.odata.org/OData/OData.svc/Categories(1)/Products
```
(d)
```
http://services.odata.org/OData/OData.svc/Categories(1)/Name/$value
```
(e)
```
http://services.odata.org/OData/OData.svc/Categories(1)/Products/$count
```
(f)

Fig. 4. Examples of OData resource path

Table 4. OData system query options

Query Option	Description	Example
$filter	Used to only select the entities that satisfy the filter predicate. The filter predicate can be represented with defined logical operators, arithmetic operators, grouping operator, string functions, date functions, math functions, and type functions.	$filter=Price le 200 and Price gt 3.5
$select	Used to request a subset of the properties on the entity or collection identified by the resource path. Can be a comma-separated list.	$select=Price,Name

OData protocol can only query on a single collection of entities (either from the existing collections or navigation links). Without the join functionality, some properties need to be stored in multiple collections in order to support complete query functions. As this issue happens in our proposed solution, we will further explain it in the next section.

3 Methodology

In this section, we present the proposed solution of bridging the SOS and the OData and harvest the benefits from both of them. We first explain the system design, followed by the SOS profile for OData (SOS-OData), and finally the query convertor component that translates SOS requests to OData requests.

3.1 System Design

There are multiple approaches to bridge the SOS and the OData. For example, one solution is to directly implement both SOS and OData interfaces on a web server and both SOS and OData services can directly connect to the backend database. Since both services can directly communicate with the database, this solution would have the best performance. However, it depends on the service providers to decide whether or not to support both SOS and OData services. Therefore, in order to maximize the benefit of bridging the SOS and the OData, we choose the mediator approach. We propose a web service called *sensor data mediator* that allows users to convert the specified service on-the-fly. In this case, instead of depending on the service providers, users can convert any existing SOS or OData (with the proposed SOS profile) services to the other type of service.

Fig. 5 shows our system architecture with three tiers, namely the data tier, the business logic tier, and the presentation tier. The data tier is basically the existing SOS or SOS-OData services. The business logic tier is the proposed sensor data mediator providing two major functionalities, which are SOS-to-OData conversion and OData-to-SOS conversion. Finally, the presentation tier contains the OData consumers and SOS consumers that act as the clients. As the converted services from the

business logic tier are compatible with the OData or SOS standard, any standard-compatible client side consumer can be used in the presentation tier.

There are two workflows in the proposed system. We first explain the SOS-to-OData conversion. For users who want to consume an existing SOS service with the OData protocol, they can use an OData consumer to send OData requests to the sensor data mediator with the information about which SOS service they want to access. The sensor data mediator first confirms if the specified service is online and is an SOS service. Then it uses the *SOS data crawler* to retrieve all the data (including the service metadata, sensor metadata, and sensor observations) from the SOS service. As we can imagine that this is a time and resource consuming task, unfortunately, it is a necessary step in order to support the complete OData protocol. This is because many query functions in the OData protocol require a global knowledge about the data. Two intuitive examples are the "$count" and "$orderby" functions. An OData service needs to know the total number of entities to answer the "$count" request, and needs to know all the values of properties to support the "$orderby" function. Therefore, in order to be OData-compliant and support all the operations, it is necessary to retrieve all the data from the SOS. After retrieving the data from the SOS service, the sensor data mediator creates an SOS-OData service to serve those data in its local machine. And as long as the data remain cached in the sensor data mediator, requests for this SOS will be directly handled by the SOS-OData service and no SOS data crawling task will be triggered.

Similarly, for users who want to consume an existing SOS-OData with SOS protocol, they can use an SOS consumer to send SOS requests to the sensor data mediator with the information about which SOS-OData service they want to access. The sensor data mediator will first confirm if the specified service is online and is an SOS-OData service. Then it uses the *query convertor* to convert the SOS requests into one or more corresponding OData requests, and send these requests to the specified SOS-OData service. Unlike that in the SOS-to-OData conversion, it does not need to crawl all the data from the SOS-OData service. It is due to the fact that OData services support more flexible query functions, so we can only retrieve the required information to

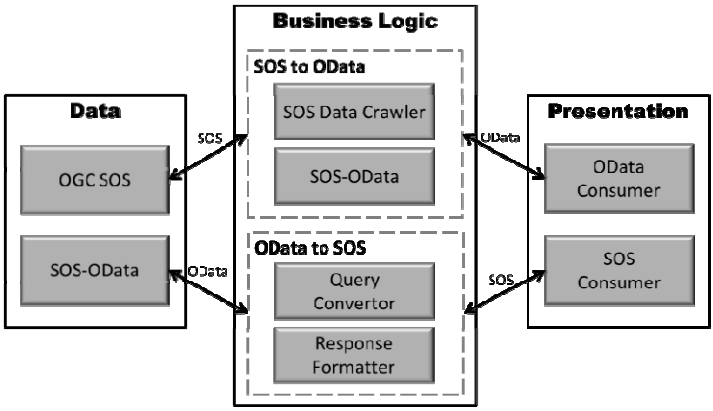


Fig. 5. System architecture

respond to the user queries. In section 3.3, we further explain the details of how the query convertor translates SOS requests to SOS-OData requests. After retrieving the required information from the SOS-OData service, the sensor data mediator parses the OData responses, *reformats* the information into an SOS response, and finally returns the SOS response back to the SOS consumer.

This proposed solution would work only if there is a predefined and commonly agreed SOS profile for OData. As this paper is not proposing a new standard SOS entity data model for OData, the SOS profile for OData we proposed here is an example showing that it is possible to bridge these two standards.

3.2 The SOS Profile for OData (SOS-OData)

In order to convert between SOS and OData service, an SOS entity data model for OData is required. In this paper, we propose one simple example of the SOS profile for OData (SOS-OData) to demonstrate that it is feasible to bridge these two standards. Fig. 6 shows the class diagram of the SOS-OData. The SOS-OData has six collections, namely the Capabilities, ObservationOfferings, ObservedProperties, Procedures, FeaturesOfInterest, and Observations. The only entity in the Capabilities collection contains properties about the service metadata and the service provider information, which are included in the SOS capabilities document. Based on the relationships between SOS entities in Table 2, we know that a Capability has one or more Observation Offerings.

Similar to that in the SOS capabilities document, each Observation Offering in SOS-OData contains an offering identifier, an offering name, and a spatio-temporal extent (i.e., a time period and a bounding box) that bounds its observations. Also from Table 2, we know that an Observation Offering has one or more Observed Properties, Procedures, and Features of Interest. Although these relationships can be represented by the OData navigation properties (i.e., links), we have encountered issues in the OData library we chose when creating the navigation property. We will provide more explanations in the implementation section. Therefore, without using the navigation property, we simply put the offering identifier as one property in the Observed Property entities. However, if the same Observed Property (i.e., the same phenomenon) happens in multiple Observation Offerings, there will be multiple Observed Property entities with only a difference on the offering identifier property, which would cause many redundancies. This issue would be worse on the Procedures and Features of Interest as the number of them could be vast. Therefore, in the SOS-OData, we only put the offering identifier in the Observed Properties to distinguish its relationship with Observation Offerings. But for Procedures and Features of Interest, we simply treat them as "related to all Observation Offerings".

In the Observed Properties, besides the offering identifier, we have the observed property URI that identifies the phenomenon. Also, as the OData requires each entity to have a key to be identifiable, each Observed Property entity has an observed property ID, which is simply the combination of the offering identifier and the observed property URI. For the Procedures, in order to support the SOS *DescirbeSensor* operation, each Procedure entity contains a procedure identifier and the corresponding

SensorML document. For the Features of Interest (FOIs), besides the FOI identifier and the FOI name, this paper simply demonstrates with the *point* geometry (i.e., latitude and longitude). However, it can be easily extended to support more complex geometries by using the complex type property.

The final collection in the SOS-OData is the Observations. Based on the O&M model (i.e., Fig. 2), we know that an observation is related to an Observation Offering (as a grouping), an Observed Property (as a phenomenon), a Procedure (as a sensor or process), and a Feature of Interest (as the identifiable object). As we mentioned that the OData does not support the DBMS-like join operation, in order to support complete query functions on Observations (e.g., spatio-temporal queries), we put the necessary information as properties in the Observation entities, such as the offering identifier, observed property identifier, observed property URI, procedure identifier, FOI identifier, FOI name, latitude, and longitude. Although it is possible to keep these information in other entities and handle the join function on the client side, a large amount of unnecessary Internet transmission becomes another issue. Therefore, we consider that although the proposed solution has the data redundancy issue, it allows us to handle queries on the server side. Besides the aforementioned properties, an Observation entity has an event time (i.e., the time the observation was made), a text result (is *null* if has a numeric result), a numeric result (is *null* if has a text result), a unit of measurement, and an observation identifier (i.e., a combination between offering identifier, observed property URI, procedure identifier, and event time).

3.3 Query Convertor

With the proposed SOS profile for OData, we can now map the data from an SOS

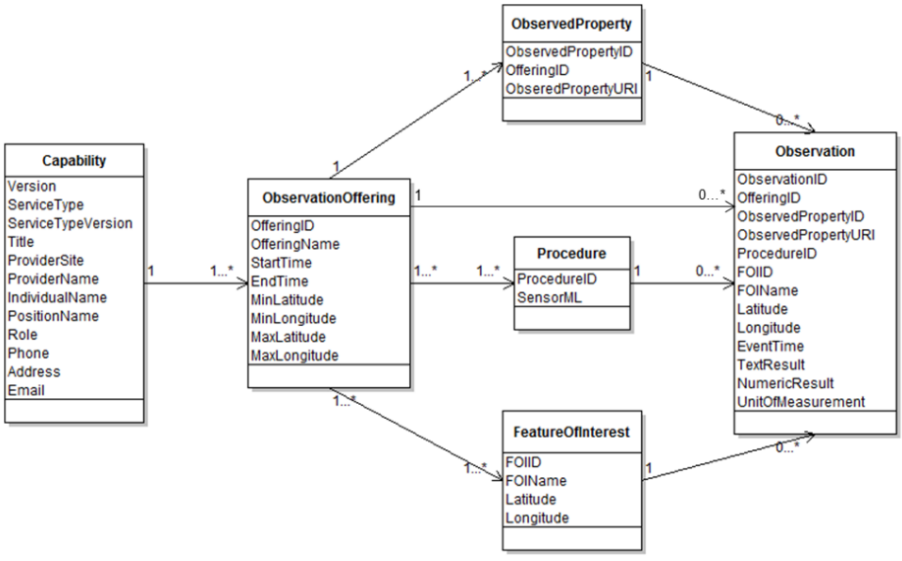


Fig. 6. The SOS-OData class diagram

service to an SOS-OData service. However, in order to make an SOS-OData service fully SOS-compliant, we need to translate SOS requests to SOS-OData requests. In the proposed sensor data mediator, we use the query convertor to perform these translations.

In the current implementation, we only support the three SOS core operations. Table 5 shows the translations between SOS operations and the corresponding OData requests, where the input parameters are in bold and italic font. First, for the *GetCapabilities* operation, it returns a capabilities document that contains the service metadata, service provider information, and allowed values in each observation offering (including a time period, a bounding box, a list of observed property URIs, a list of procedures, and a list of features of interest). As shown in Fig. 6, the required information can be found in Capabilities, Observation Offerings, Observed Properties, Procedures, and Features of Interest collections in the SOS-OData. To be more specific, as shown in Table 5, the *GetCapabilities* operations maps to five different kinds of OData requests. As all the information in the Capabilities and Observation Offerings are required, only the observed property URI is required for the capabilities response (while using the OfferingID to relate). For the Procedures and Features of Interest, we can simply put the URIs to indicate the list of procedure identifiers and the list of FOI identifiers in the capabilities document, instead of putting the entire lists inline.

Second, the *DescribeSensor* operation allows users to request sensor metadata with procedure identifier. In this case, we can directly query the Procedures collection with a "$filter" function on procedure identifier and a "$select" function to return the SensorML document only. Third, the *GetObservations* operation allows users to request sensor observations with two mandatory parameters (i.e., an offering identifier and an observed property URI) and optional parameters on the spatial and temporal criteria. As the last example shown in Table 5, we can simply use the "$filter" function on the Observations collection to limit the SOS-OData responses with specified criteria.

Table 5. Translations between SOS operations and OData requests (input parameters are in bold and italic font)

SOS Operation	OData Requests (without service URI)
GetCapabilities	/Capabilities
	/ObservationOfferings
	/ObservedProperties?$select=OfferingID,ObservedPropertyURI
	/Procedures?$select=ProcedureID
	/FeaturesOfInterest?$select=FOIID
DescribeSensor (***procedureID***)	/Procedures?$filter=ProcedureID eq '***procedureID***' &$select=SensorML
GetObservations (***offeringID***, ***observedPropertyURI***, ***minLat***, ***minLon***, ***maxLat***, ***maxLon***, ***startTime***, ***endTime***)	/Observations?$filter=OfferingID eq '***offeringID***' and ObservedPropertyURI eq '***observedPropertyURI***' and Latitude ge '***minLat***' and Latitude le '***maxLat***' and Longitude ge '***minLon***' and Longitude le '***maxLon***' and EventTime ge datetime'***startTime***' and EventTime le datetime'***endTime***'

There is one important thing to note for the implementation. As many OData service uses *pagination* mechanism to avoid sending a lot of entities in a single response, in order to get the complete list of entities, it is necessary to resolve all the "next" navigation links (i.e., the next pages) specified in the OData responses.

4 Implementation and Prototype

In this section, we present our current implementation and prototype. We first introduce the libraries we use for the SOS and the OData. Then it is followed by the actual workflow of the proposed system. Finally, we demonstrate our prototype with a real-world dataset.

4.1 SOS and OData Library Choices

For the implementation of the proposed system, we choose our own SOS implementation, which is the GeoCENS SOS [18], as the SOS library. As the GeoCENS SOS implementation is based on Java, it is much more convenient for us to choose a Java-based OData library. Therefore, we choose the odata4j implementation[6] as the OData library. However, as we mentioned in section 3.2 that the odata4j does not complete support all the OData functions. For example, we have discovered that the odata4j library does not support the navigation property (which leads to the modifications in our proposed SOS-OData) and the "$select" function. Although the odata4j is incomplete, we believe that it does not affect much on our proposed solution and does not prevent us from demonstrating the feasibility and benefits of bridging the two standards. Therefore, we still use it in our current implementation.

4.2 Workflow of the Proposed System

Before we present the workflow of the proposed system, we need to first discuss about how users are going to specify the data service they want to access. Some naïve approaches are to put the service URI into query string, HTTP header, or HTTP parameter. However, these approaches require modifications on the client side, which does not fit to our objective that the converted services can be accessed by any standard-compliant clients. Therefore, here we propose to include the data service URI into the converted service root URI, so there will be no modifications needed on the client side. In order to do that, we first allow users to encrypt the data service URI to be URL-safe, and then we append the encrypted service URI to the URI of the sensor data mediator to compose the new converted service URI. The encryption approach we use here is the Base64 encoding [19] and replace character "+" with "-" and "/" with "_" to be URL-safe. Fig. 7 shows an example of creating a new service URI, where the "http://alec.geocens.ca:9182/SOS-OData" is the sensor data mediator

[6] `https://code.google.com/p/odata4j/`

location and the followed "/odata/" indicates that the new service is an OData service. If replacing the "/odata/" with "/sos/", it means that the new service is an SOS service.

Therefore, the first step in the workflow is to encrypt the data service URI, and the second step is to create the new service URI. The overall workflow is shown in Fig. 8. second step is to create the new service URI. The overall workflow is shown in Fig. 8. With the new service URI, users can send requests to the URI with the corresponding protocol. When the sensor data mediator receives requests from the new service URI, it first decrypts the data service URI. Then if the data service is an SOS and there is no corresponding SOS-OData service in the sensor data mediator, the sensor data mediator tasks the SOS data crawler to retrieve all the data from the SOS. Once the crawling is complete, the sensor data mediator creates an SOS-OData service in its local machine. Finally, the sensor data mediator forwards the user requests to the created SOS-OData service and returns SOS-OData responses.

On the other hand, if the data service is an SOS-OData service, the sensor data mediator uses the query convertor to translate SOS requests to SOS-OData requests (i.e., Table 5) and retrieves the required information from the SOS-OData service. After receiving the SOS-OData responses, the sensor data mediator parses the responses, reformats the information into SOS responses, and finally returns the SOS responses to the client.

4.3 Prototype Demonstration

In the prototype demonstration, we use an existing validated SOS service (i.e., http://app.geocens.ca:8221/sos) as a reference SOS data service. The SOS is a real-time service containing sensor observations since July, 2011. It contains one observation offering, five observed properties, and about 380,000 sensor observations.

In order to demonstrate that our prototype can convert an SOS service to an SOS-OData service and vice versa, we first convert the testing SOS service to an SOS-OData service, and then convert the created SOS-OData service to a new SOS service. Then we compare the SOS responses from the original SOS service and the converted SOS service. If the responses are both OGC SOS compatible, it proves that our proposed system can successfully convert between SOS and OData services.

Based on the comparison between the capabilities documents we retrieved from the two SOS service, the two documents are basically the same besides (1) the operation links, (2) the time string formats, and (3) the lists for procedures and features of interest. We have also compared the *DescribeSensor* and *GetObservations* responses from the two SOS services. The SensorML documents from both services are identical. And for the *GetObservations* responses, only some differences occur on the time string formats and some XML element identifiers. We believe that these differences do not cause any inconsistency from the client's perspective, and the converted service is the same as the original service. Therefore, this test demonstrates that the proposed system can successfully convert between SOS and SOS-OData services.

In addition, in order to further demonstrate the benefits of the proposed system, we test the converted SOS-OData and SOS services with existing OData and SOS clients.

Fig. 7. An example of composing the new service URI

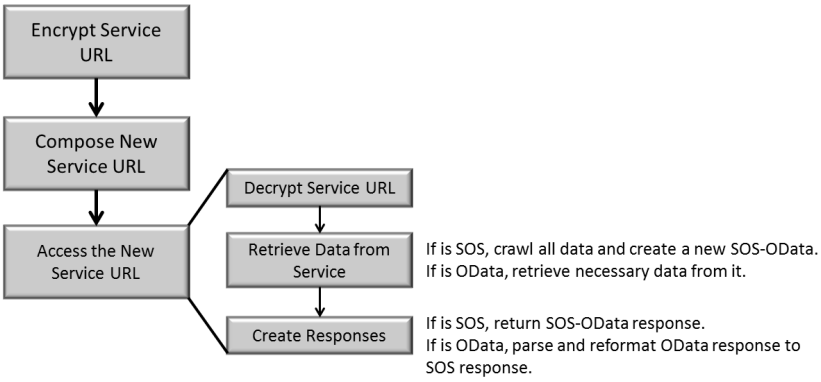


Fig. 8. The overall workflow of the proposed system

For the OData clients, we choose the Sesame Data Browser[7] and Microsoft PowerPivot Excel Add-In[8]. We can successfully use the existing OData consumers to access the converted SOS-OData service, which means that users can use the flexible OData query and browsing functionalities to access SOS data.

For the SOS clients, we choose the 52°North SOS Test Client version 2[9] and the GeoCENS WorldWind Client[10]. We are able to access the converted SOS service with the existing SOS consumers. This test shows that with the strong OData ecosystem, we can easily create a SOS-compliant service by building an SOS-OData service.

5 Conclusions and Future Work

We have presented the sensor data mediator that can bridge the SOS and the OData. As shown in the prototype demonstration, the proposed system can successfully convert between SOS services and SOS-OData services. We also demonstrated that the converted services can be directly consumed by the standard-compliant SOS and OData clients.

[7] http://metasapiens.com/sesame/data-browser/
[8] http://www.microsoft.com/en-us/bi/powerpivot.aspx
[9] http://v-swe.uni-muenster.de:8080/WeatherSOS/
[10] http://dev.geocens.ca/gswclient

The major contributions of the proposed system are mainly on the application side. First, as the OData requires a commonly agreed conceptual model to be actually interoperable, we proposed an SOS entity data model based on the SOS conceptual model. The SOS-OData services become interoperable in the context of OGC SOS. Second, for the SOS-OData services, we also proposed the query interfaces that are compliant with the SOS core operations. This allows users to consume SOS-OData with standard SOS operations. And in this case, since OData services are easy to build with its strong ecosystem, we can easily build an SOS-compliant service with the SOS conceptual model and query interfaces. Third, with the proposed SOS-OData, we can convert any existing SOS services into SOS-OData services, which allow users to consume SOS data with the more flexible data query and browsing functions provided by the OData. Finally, as we are not proposing a standard SOS entity data model for OData and there are certainly other ways to incorporate the SOS conceptual model into the OData, this paper simply presents an example of bridging these two standards. And the outcome shows that the SOS and the OData can benefit from each other and the bridging would consequently lead us to the vision of open data for sensor web. Based on our best knowledge, we believe the bridging of the SOS and the OData is new and unique.

There are a few remaining issues remain to be solved. First, as the conversion from SOS to OData requires the system to crawl all the data from SOS, this time and resource consuming task would cause many issues, such as the out-of-memory exceptions or it may take days to finish the crawling task. In order to address this issue, one possible solution is to utilize cloud/distributed computing to crawl with more communication threads and resources. The second issue is that there are some improvements can be done on the current implementations. For example, we are currently using the InMemory OData service provided by the odata4j library, which stores all the information in memory. This may cause out-of-memory exceptions when handling large datasets. One simple solution is to store the data in the database. In addition, as we mentioned earlier, the odata4j does not support navigation links and "$select" function. To address this issue, we can either implement these functions by ourselves or switch to a more complete OData library (e.g., .Net library). Third, we will analyze the system performance in terms of the required time of crawling SOS data (which depends on the improvements we will make on the SOS data crawler) and the latencies of SOS operations from SOS-OData services (which would related to the OData pagination setting). Finally, when converting SOS to OData, besides crawl the existing SOS data, we can retrieve the real-time sensor data from SOS. This can be done with one of our previous researches: the adaptive feeder [20].

Acknowledgements. The authors would like to thank CANARIE, Cybera, Alberta Innovates Technology Futures, and Microsoft Research for their supports.

References

1. Liang, S.H.L., Croitoru, A., Tao, C.V.: A Distributed Geospatial Infrastructure for Sensor Web. Computers and Geosciences 31(2), 221–231 (2005)

2. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., Anderson, J.: Wireless Sensor Networks for Habitat Monitoring. In: 2002 ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, United States (2002)
3. Hart, J.K., Martinez, K.: Environmental Sensor Networks: A revolution in the earth system science? Earth Science Reviews 78, 177–191 (2006)
4. Hsieh, T.T.: Using Sensor Networks for Highway and Traffic Applications. IEEE Potentials 23(2), 13–16 (2004)
5. Xu, X., Han, J., Lu, W.: RT-Tree: An Improved R-Tree Indexing Structure for Temporal Spatial Databases. In: The International Symposium on Spatial Data Handling, pp. 1040–1049 (1990)
6. Kassab, A., Liang, S., Gao, Y.: Real-Time Notification and Improved Situational Awareness in Fire Emergencies using Geospatial-based Publish/Subscribe. J. Applied Earth Observation and Geoinformation 12(6), 431–438 (2010)
7. Botts, M., Robin, A.: Bringing the Sensor Web Together. Geosciences, 46–53 (2007)
8. Liang, H.L.S., Chang, D., Badger, J., Rezel, R., Chen, S., Huang, C.Y., Li, R.Y.: Capturing the Long Tail of Sensor Web. In: International Workshop on Role of Volunteered Geographic Information in Advancing Science, Zurich, Switzerland (September 14, 2010)
9. Anderson, C.: The Long Tail: Why the Future of Business is Selling Less of More. Hyperion (2006)
10. Botts, M., Percivall, G., Reed, C., Davidson, J.: OGC Sensor Web Enablement: Overview and High Level Architecture (OGC 07-165). Open Geospatial Consortium white paper (December 28, 2007)
11. Na, A., Priest, M.: Sensor Observation Service (OGC 06-009r6). Open Geospatial Consortium Implementation Standard (October 26, 2007)
12. Cox, S.: Observations and Measurements - XML Implementation (OGC 10-025r1). Open Geospatial Consortium Implementation (March 22, 2011)
13. Botts, M., Robin, A.:OpenGIS Sensor Model Language (SensorML) Implementation Specification (OGC 07-000). Open Geospatial Consortium Implementation Specification (July 17, 2007)
14. Beaujardiere, J.:OpenGIS Web Map Server Implementation Specification (OGC 06-042). Open Geospatial Consortium Implementation (March 15, 2006)
15. Vretanos, P.:OpenGIS Web Feature Service 2.0 Interface Standard (OGC 09-025r1). Open Geospatial Consortium Implementation (November 02, 2010)
16. Bröring, A., Stasch, C., Echterhoff, J.: OGC Sensor Observation Service Interface Standard (OGC 12-006). Open Geospatial Consortium Implementation Standard (April 16, 2012)
17. Microsoft Corporation: Open Data Protocol (OData) Specification. Open Specifications Documentation (June 10, 2011)
18. Liang, H.L.S., Chang, D., Badger, J., Rezel, R., Chen, S., Huang, C.Y., Li, R.Y.: GeoCENS: Geospatial Cyberinfrastructure for Environmental Sensing. In: GIScience 2010 Conference, Zurich, Switzerland (2010)
19. Internet Engineering Task Force (IETF): The Base16, Base32, and Base64 Data Encodings (RFC 4648) (October 2006)
20. Huang, C.Y., Liang, S.: A Hybrid Pull-Push System for Near Real-Time Notification on Sensor Web. In: The XXII Congress of the International Society for Photogrammetry and Remote Sensing, Melbourne, Australia, August 25-September 1 (2012)

Dynamic Objects Effect on Visibility Analysis in 3D Urban Environments

Oren Gal and Yerach Doytsher

Mapping and Geo-information Engineering
Technion - Israel Institute of Technology
Haifa, Israel
{orengal,doytsher}@technion.ac.il

Abstract. This paper presents a unique formulation and concept of the dynamic object effect on constant objects, such as buildings, dealing with visibility problems in 3D urban environments. Dynamic objects in a 3D urban environment, such as cars, pedestrians and trees, are usually omitted from visibility analysis. In order to challenge this problem, we focus on modeling predicting and estimating dynamic objects' future location in the environment. We integrate all these factors into our visibility analysis and create a probabilistic visibility analysis, changed over time due to the dynamic character of these objects.

Our probabilistic visibility concept takes into account 3D boxes and cylinders, generating a fast and exact analytic solution to dynamic and static objects; to illustrate our concept, we use web-cameras located at constant points in the treated environments in order to update our model in each time period from web source data and to analyze the environment. Dynamic objects prediction is based on validated models for driver behavior; pedestrians' walking routes, trees' displacement and wind effect. A real urban environment with dynamic objects approximated by 3D boxes and cylinders demonstrates our approach.

Keywords: Visibility, Spatial Analysis, Urban Environments, 3D.

1 Introduction and Related Work

The visibility problem has been extensively studied over the last twenty years, due to the importance of visibility in GIS and Geomatics, computer graphics and computer vision, and robotics. Visibility analysis in open terrains has been extensively studied, mostly by using already-known Digital Terrain Models (DTM). Only a few research projects have treated visibility analysis in 3D dense environments such as urban scenes, and as of now, have still not been targeted as a major research field.

Accurate visibility computation in 3D environments is a very complicated task demanding a high computational effort, which can hardly be carried out in a very short time using traditional well-known visibility methods [15]. The exact visibility methods are highly complex, and cannot be used for fast applications due to their long computation time. As mentioned above, previous research in visibility computation

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 147–163, 2013.
© Springer-Verlag Berlin Heidelberg 2013

has been devoted to open environments using DEM models, representing raster data in 2.5D (Polyhedral model), and do not address, or suggest solutions for, dense built-up areas. Most of these published papers have focused on approximate visibility computation, enabling fast results using interpolations of visibility values between points, calculating point visibility with the Line of Sight (LOS) method [2]. Other fast algorithms are based on the conservative Potentially Visible Set (PVS) [3]. These methods are not always completely accurate, as they may render hidden objects' parts as visible due to various simplifications and heuristics.

A vast number of algorithms have been suggested for speeding up the process and reducing computation time. Franklin [7] evaluates and approximates visibility for each cell in a DEM model based on greedy algorithms. Wang et al. [22] introduced a Grid-based DEM method using viewshed horizon, saving computation time based on relations between surfaces and the line of sight (LOS) method. Later on, an extended method for viewshed computation was presented, using reference planes rather than sightlines [23].

One of the most efficient methods for DEM visibility computation is based on shadow-casting routine. The method casts shadowed volumes in the DEM, like a light bubble [17]. Extensive research has treated Digital Terrain Models (DTM) in open terrains, mainly Triangulated Irregular Network (TIN) and Regular Square Grid (RSG) structures. Visibility analysis in terrain was classified into point, line and region visibility, and several algorithms have been introduced, based on horizon computation describing a visibility boundary [5].

Only a few works have treated visibility analysis in urban environments. A mathematical model of an urban scene, calculating probabilistic visibility for a given object from a specific viewcell in the scene, has been presented by [11]. This is a very interesting concept, which extends the traditional deterministic visibility concept. Nevertheless, the buildings are modeled as cylinders, and the main challenges of spatial analysis and building model were not tackled. Other methods have been developed, subject to computer graphics and vision fields, dealing with exact visibility in 3D scenes, without considering environmental constraints. Plantinga and Dyer [15] used the aspect graph – a graph with all the different views of an object.

Lately, several spatial analysis methods for urban environments have been presented, using constant known 3D models tailor-made for spatial aspects [24]. An urban environment can be divided into static objects such as buildings, roads, etc., and dynamic objects such as moving cars, tree branches, pedestrians etc.

Static objects which are modeled by 3D GIS models are not frequently changed. On the other hand, dynamic objects cannot be modeled efficiently by a constant 3D model, and must be updated from on-line sources, such as web cameras, in a certain time period.

Visibility analysis from point A to B is mostly captured as Boolean values of 'visible' and 'invisible' ("1" and "0" values). The complex modeling of these dynamic objects, and the uncertainties during the time update gap using on-line data sources, such as web cameras, in the environments, to update the urban environment model, lead to the probabilistic visibility analysis concept. This concept sets probability value for object visibility.

In this paper we present the probabilistic visibility concept in urban environments, taking into account dynamic objects and their effect on visibility analysis of constant objects such as buildings. In order to address this problem, we focus on modeling, predicting and estimating their future location in the environment. We integrate all these factors into our visibility analysis and create a probabilistic visibility analysis which changes over time, due to the dynamic character of these objects.

Our probabilistic visibility concept is related only to dynamic objects; static objects, such as buildings, are analyzed by using an efficient visibility analysis method presented by [8].

To illustrate our concept, we use web-cameras located at constant points in the environments in order to update our model in each time period (as is already being done in Google Maps application).

We start by mapping dynamic objects in an urban environment, modeling them for efficient visibility analysis, and defining a probabilistic visibility model for each of them.

2 Dynamic Objects – Modeling and Probabilistic Visibility

Dynamic objects such as moving cars, tree branches, pedestrians etc., directly affect visibility in urban environments.

Due to modeling limitations, these entities are usually neglected in spatial analysis aspects. We focus on three major dynamic objects in an urban case: moving cars, tree branches and pedestrians. Each object is modeled with 3D boxes or 3D cylinders, which allow us to extend the use of our previous visibility analysis in urban environments presented for static objects [8].

2.1 Moving Car

2.1.1. 3D Modeling

As we mentioned earlier, web-cameras in urban environments can record the moving cars at any specific time. Image sources such as web cameras, like other similar sensors sources, demand an additional stage of Automatic Target Detection (ATD) algorithms to extract these objects from the image [18]. In this research we do not focus on ATD, which must be implemented when shifting from the research described in the paper toward an applicable system.

The common car structure can be easily modeled by two 3D boxes, as can be seen in Fig. 1(b), which is similar to the original car structure presented in Fig 1(a).

Fig. 1. Car Modeling Using 3D Boxes: (a) the Original Car, (b) the Modeled Car

We define the Car Boundary Points (CBP) as the set of visible surfaces' boundary points of 3D boxes modeling the car presented in Fig 1(b). Each box is modeled as 3D cubic $C_{car}(x, y, z)$ as presented extensively in [8] for a building model case:

$$C_{car}(x,y,z) = \begin{pmatrix} x = t \\ y = \begin{pmatrix} x^n - 1 \\ 1 - x^n \end{pmatrix} \\ z = c \end{pmatrix} \quad (1)$$

$$-1 \le t \le 1$$
$$n = 350$$
$$c = c + 1$$

Car Boundary Points (CBP) - we define CBP of the object i as a set of boundary points $j = 1..N_{CBP\_bound}$ of the visible surfaces of the car object, from viewpoint $V(x_0, y_{0,} z_0)$, where the maximum surface's number is six and each surface defined by four points, $N_{CBP\_bound} \le 24$.

In Fig. 2, car is modeled by using two 3D boxes. Visible surfaces colored in red, CBP marked with yellow points.

$$CBP_{i=1..N_{CBP\_bound}}(x_0, y_0, z_0) = \begin{bmatrix} x_1, y_1, z_1 \\ x_2, y_2, z_2 \\ .. \\ x_{N_{CBP\_bound}}, y_{N_{CBP\_bound}}, z_{N_{CBP\_bound}} \end{bmatrix} \quad (2)$$

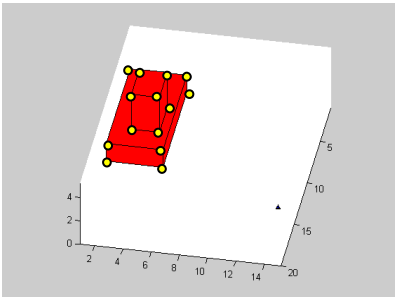


Fig. 2. Modeling Car Using 3D Boxes (CBP Marked with Yellow Points)

2.1.2. Probabilistic Visibility Analysis

Visibility has been treated as Boolean values. Due to incomplete information and the uncertainties of predicting the car's location at future times, visibility becomes much more complicated.

As it is well known from basic kinematics, CBP can be estimated in future time $t + \Delta t$ as:

$$CBP_i(t + \Delta t) = CBP_i(t) + V(t)\Delta t + \frac{A(t)\Delta t^2}{2} \quad (3)$$

Where $V(t)$ is the car velocity vector $V(t) = (v_x v_y)^T$, and the acceleration vector $A(t) = (a_x a_y)^T$. Estimation of a car's location in the future based on a web camera is not a simple task. Driver behavior generates multi-decision modeling, such as car-following behavior, gap acceptance behavior, or lane-change cases including traffic flow, speed etc.[1].

Our probabilistic car model is based on microscopic simulation models that were properly calibrated and validated using VISSIM simulation. VISSIM is a time-based microscopic simulation tool that uses various driver behaviors and vehicle performances to accurately represent an urban traffic model. The VISSIM simulation model has been validated when compared to the data from various real-world situations [4] and used for the test-bed network by [14],[16], and on driver behavior research defining average speed and acceleration [1].

VISSIM simulation environment is a very robust one, characterizing each driver/vehicle unit include: behavior of driver/vehicle unit (desired speed, desired acceleration and deceleration, sensitivity thresholds and parameters that determine behavior); technical specifications of the vehicle (e.g. length, weight, engine power, maximum speed, acceleration and deceleration potential); relative positions of preceding and following vehicles on different lanes, and the positions of objects.

The average speed in urban environments is about 45 [km/hr], from a minimum of 40 [km/hr] up to a maximum of 50 [km/hr]. In the situation of a free driving case, which is the common mode in urban environments [21], the acceleration of family car can change between $1 - 3.5 \left[{}^{m}/_{\sec^2}\right]$, and on average $2.5 \left[{}^{m}/_{\sec^2}\right]$, as seen in Fig. 3.

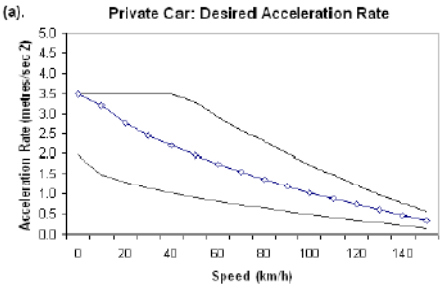


Fig. 3. Average Acceleration Rate of a Family Car in an Urban Environment [1]

As can be seen from several validations of car and driver estimation, velocity and acceleration are distributed as normal ones, and lead to normal location distribution:

$$\begin{gathered} V(t) \sim N(\mu = 45, \sigma^2 = 10) \\ A(t) \sim N(\mu = 2.5, \sigma^2 = 1) \\ CBP(t + \Delta t) \sim \sum N \end{gathered} \tag{4}$$

In time step t, where the car's location is taken from a web-camera, visibility analysis from $CBP(t)$is an exact one, based on our previous visibility analysis [8], as seen

in Fig. 2. Visibility analysis becomes probabilistic for future time $t + \Delta t$, applying the same visibility analysis for $CBP(t + \Delta t)$ presented in Fig 4:

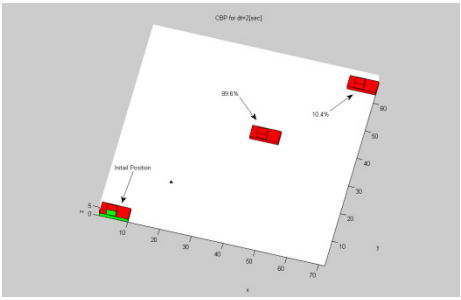


Fig. 4. Probabilistic Visibility Analysis for CBP: 3D View of Car Modeling Location from Web-oriented Source with Estimated Probabilistic Location; CBP is the Boundary for Visible Surfaces (Colored in Red)

In Fig. 4, the car's location from a web-camera appears in the bottom left side. For $\Delta t = 2[sec]$, the car's location is marked by two 3D boxes, where CBP for each of them is the boundary of visible surfaces marked in red. The probability that the visible surfaces, which are bounded by CBP, will be visible in future time is based on the last update taken from the web application (depicted with arrows in Fig. 4), computed by using two different random normal PDF values for V and A based on eq. (4).

2.2 Pedestrians

2.2.1. 3D Modeling

Pedestrian modeling can be done in high resolution, but due to Auto Target Detection (ATD) algorithms capabilities, pedestrians are usually bounded by a 3D cylinder and not as an exact detailed model [18]. For this reason, we model pedestrians as 3D cylinders, which is somewhat conservative but still applicable.

Pedestrian can be easily modeled by 3D cylinder, as seen in Fig. 5 (marked in red), which is similar to the output from ATD methods tested on a web-camera output recognizing walkers in urban environments.

We extend our previous visibility analysis concept [8] and include new objects modeled as cylinders as continuous curves parameterization, $C_{Peds}(x, y, z)$. Cylinder parameterization can be described as:

$$C_{Peds}(x, y, z) = \begin{pmatrix} r\sin(\theta) \\ r\cos(\theta) \\ c \end{pmatrix} \tag{5}$$

$$0 \le \theta \le 2\pi$$
$$c = c + 1$$
$$0 \le c \le h_{peds\_\max}$$

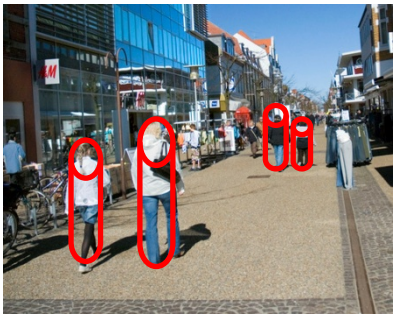


Fig. 5. Modeling Pedestrians in Urban Scene Using Cylinders (Colored in Red)

We define the visibility problem in a 3D environment for more complex objects as:

$$C'(x,y)_{z_{const}} \times (C(x,y)_{z_{const}} - V(x_0, y_0, z_0)) = 0 \tag{6}$$

where 3D model parameterization is $C(x,y)_{z=const}$, and the viewpoint is given as $V(x_0, y_0, z_0)$. Extending the 3D cubic parameterization, we also consider the cylinder case. Integrating eq. (5) to (6) yields:

$$\begin{pmatrix} r\cos\theta \\ -r\sin\theta \\ 0 \end{pmatrix} \times \begin{pmatrix} r\sin\theta - V_x \\ r\cos\theta - V_y \\ c - V_z \end{pmatrix} = 0 \tag{7}$$

$$\theta = \arctan\left(-\frac{-r - \frac{\left(-vy\, r + \sqrt{vx^4 - vx^2 r^2 + vy^2 vx^2}\right) vy}{vx^2 + vy^2}}{vx}, -\frac{-vy\, r + \sqrt{vx^4 - vx^2 r^2 + vy^2 vx^2}}{vx^2 + vy^2} \right) \tag{8}$$

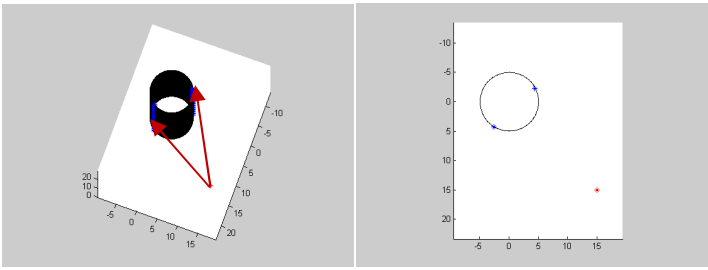
As can be noted, these equations are not related to Z axis, and the visibility boundary points are the same for each x-y cylinder profile.

The visibility statement leads to complex equation, which does not appear to be a simple computational task. This equation can be efficiently solved by finding where the equation changes its sign and crosses zero value; we used analytic solution to speed up computation time and to avoid numeric approximations. We generate two values of θ generating two silhouette points in a very short time computation. Based

on an analytic solution to the cylinder case, a fast and exact analytic solution can be found for the visibility problem from a viewpoint.

We define the solution presented in eq. (8) as x-y-z coordinates values for the cylinder case as **Pedestrian Boundary Points (PBP)**. PBP are the set of visible silhouette points for a 3D cylinder modeling the pedestrian, as presented in Fig 6:

$$PBP_{i=1..N_{PBP\_bound}=2}(x_0, y_0, z_0) = \begin{bmatrix} x_1, y_1, z_1 \\ x_{N_{PBP\_bound}}, y_{N_{PBP\_bound}}, z_{N_{PBP\_bound}} \end{bmatrix} \tag{9}$$



(a) (b)

Fig. 6. Pedestrian Boundary Points (PBP) for a Cylinder using Analytic Solution marked as blue points, Viewpoint Marked in Red: (a) 3D View (Visible Boundaries Marked with Red Arrows); (b) Topside View

2.2.2. Probabilistic Visibility Analysis

The kinematic model of a pedestrian is only a part of the estimation and prediction of his movement in an urban environment. For simplicity, we use only a kinematic model for pedestrian's future location, since decision-making in this field is very complicated.

A well-known and common kinematic model for pedestrians is presented by Hoogendoorn et al. [9]. Based on this model, pedestrian location can be predicted as:

$$PBP(t + \Delta t) = PBP(t) + V(t)\Delta t + w \tag{10}$$

where w is a white noise of a standard Wiener Process which reflects the uncertainty in the expected traffic condition, described as Gaussian distribution.
Pedestrian speed V can be divided into three major groups:

1. Fast: 1.8 meter per second
2. Standard: 1.3 meter per second
3. Slow: 0.85 meter per second

$$\begin{gathered} V(t) \sim N(\mu = 1.3, \sigma^2 = 0.5) \\ w \sim \sqrt{\Delta t} N(0,1) \\ PBP(t + \Delta t) \sim \sum (N + \sqrt{\Delta t} N) \end{gathered} \tag{11}$$

In time step t, where pedestrian location is taken from a web-camera, visibility analysis from $PBP(t)$ can be computed.

Visibility analysis becomes probabilistic for a future time $t + \Delta t$. Applying the same visibility analysis for $PBP(t + \Delta t)$ and basing ourselves on the probabilistic model, leads to a probabilistic visibility definition. In Fig. 7, pedestrian location from a web-camera is marked as a black cylinder, where PBP are marked with blue points. For $\Delta t = 3[sec]$, pedestrian location is marked with green cylinders, where PBP for each pedestrian is marked in black. The probability that the surface bounded by PBP will be visible in future time is presented from the last update taken from the web application, presented with arrows in Fig. 7, using four different random normal PDF values for *V* and *w* based on eq. (11).

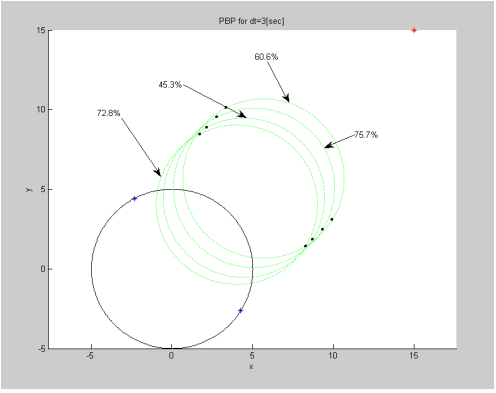


Fig. 7. Probabilistic Visibility Analysis for PBP: Topside View of Pedestrian Location from Web-oriented Source Marked with Black Cylinder, Estimated Probabilistic Location Marked in Green, PBP Marked with Black Points

2.3 Trees

2.3.1. 3D Modeling

Unlike the two previous objects, trees do not move. However, tree branches, which are very common in urban environments, are greatly affected by wind.

Modeling and analyzing the capability to 'see' through trees is not a trivial undertaking, and is related to many factors such as tree type, age, number and shape of leaves, wind profile etc.

We model tree structure as two cylinders, one for the tree stem and a second, larger one, above the first for the upper part of the tree, as seen in Fig 8.

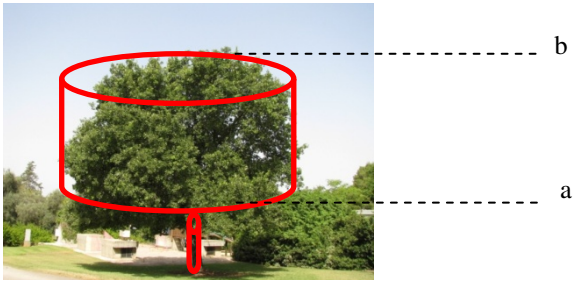
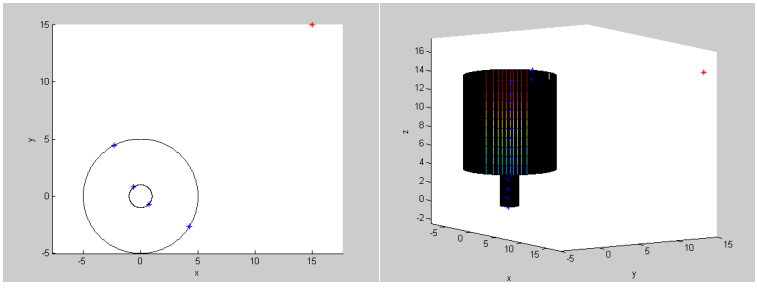


Fig. 8. Modeling Tree by Two Cylinders colored in Red: Minimum and Maximum Higher Cylinder Boundaries [a, b] marked with Dotted Lines

Tree branches are taken into account by the probabilistic model and wind effect, as detailed in the next sub-section.

We define the **Tree Boundary Points (TBP)** as the set of visible silhouette points for two 3D cylinders, modeling as presented in Fig 9, with TBP marked as blue points (TBP is also the solution to eq. (6) for two cases of cylinder parameterization):

$$TBP_{i=1..N_{TBP\_bound}=4}(x_0, y_0, z_0) = \begin{bmatrix} x_1, y_1, z_1 \\ x_2, y_2, z_2 \\ \dots \\ x_{N_{NBP\_bound}}, y_{N_{NBP\_bound}}, z_{N_{NBP\_bound}} \end{bmatrix} \tag{12}$$



(a) (b)

Fig. 9. TBP for Tree Object Modeled by Two Cylinders colored in Black: (a) Topside View (TBP Marked in Blue Point); (b) 3D View

2.3.2. Probabilistic Visibility Analysis

In the case of the tree, we focus on branches' movements in the presence of a wind. Modeling a tree as a dynamic system is a very complex task; such a system is affected by an unstable phenomenon – the wind.

We use Finite Element (FE) analysis solution to predict tree motion in the presence of the common wind profile introduced by [10].

As for natural wind profile, wind speed $v_w(z)$ increases as the distance from the ground increases. In our simulation, wind profile is approximated according to the following logarithmic law:

$$v_w(z) = \ln\left[1 + z(e-1)/(b-a)\right] \tag{13}$$

The outcome of the wind profile - described in eq. (13) – and its effect on tree branches' displacement can be simulated as sinusoidal function with a frequency of f=0.27 Hz (y-axis displacements can be neglected relative to x-axis displacements) [10]. Based on validated models [10], tree branches' movements in the presence of wind can be simulated as:

$$TBP(t+\Delta t)_{i=3}^{4} \sim TBP(t)_{i=3}^{4} \cdot \sin(f(t+\Delta t)) \cdot U(a,b) \tag{14}$$

where a and b are the minimum and the maximum highs of the second 3D cylinder presented in Fig.8.

It should be mentioned that $TBP(t+\Delta t)_{i=1}^{2}$ (the first 3D cylinder modeling tree stem) is a static object which is rarely affected by the wind. The probabilistic tree modeling in time,$TBP(t+\Delta t)_{i=3}^{4}$ leads directly to a probabilistic visibility solution.

In Fig. 10, tree modeling from a web-camera is marked as a black cylinder, where TBP are marked with blue points. For $\Delta t = 15[sec]$, in the presence of wind, estimation of tree branches movement is marked with green cylinders, where TBP for each of them is marked in red. The probability that the surface, which is bounded by TBP, will be visible at a future time is based on the last update taken from a web application (as presented with arrows in Fig. 10), and is computed by using TBP future time prediction, presented in eq. (14). It can be noticed that $TBP(t+\Delta t)_{i=1}^{2}$ stays constant.

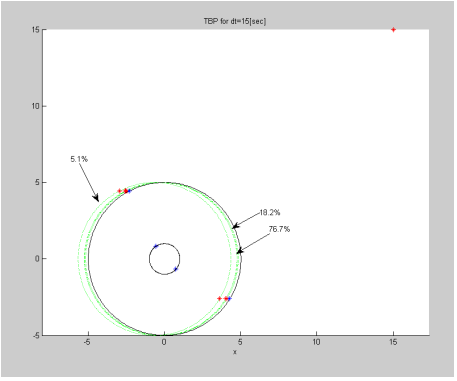


Fig. 10. Probabilistic Visibility Analysis for TBP: Topside View of the Tree Modeling with Probabilistic Visibility Analysis for Tree Branches in the Presence of Wind (Estimated Probabilistic Location Marked in Green Cylinders, TBP Marked in Red Points)

3 Problem Statement

We consider visibility problem in a 3D urban environment, consisting of static constant objects and dynamic objects.

Given:

- Static objects:
  - 3D buildings modeled as 3D cubic parameterization $\sum_{i=1}^{N_{of\_build}} C_i(x, y, z = {}^{h_{\max}}_{h_{\min}})$
- Dynamic objects:
  - Moving cars modeled as 3D cubic parameterization, $C_{car}(x, y, z)$
  - Pedestrian modeled as cylinder parameterization , $C_{peds}(x, y, z)$
  - Trees modeled with two cylinder parameterization, $C_{tree}(x, y, z)$
- Wind profile $v_w(z)$ and Viewpoint $V(x_0, y_{0,} z_0)$, in 3D coordinates.

Computes:

Set of all visible points in $\sum_{i=1}^{N}[C_{building_i}, C_{car_i}, C_{tree_i}, C_{peds_i}]$ from $V(x_0, y_{0,} z_0)$,.

We extend our previous work [8], developed for a fast and efficient visibility analysis for buildings in urban environments, and consider also a basic structure of cylinders, which allows us to model pedestrians and trees. Based on our probabilistic visibility computation of dynamic objects, we test the effect of these by using data gathered from web-oriented GIS sources to update our estimation and prediction on these entities.

4 Simulations

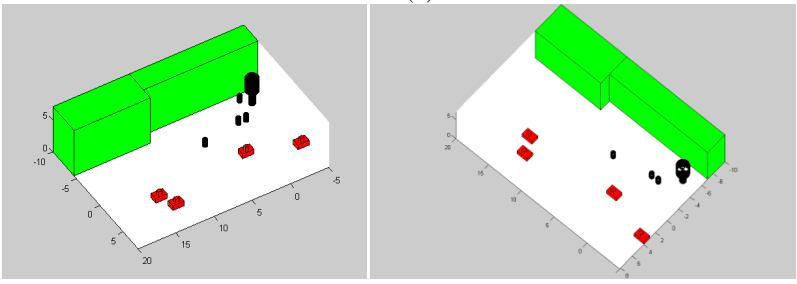
We demonstrate dynamic objects' effect on visibility analysis in a 3D urban environment using source data from web cameras located at constant points.

Based on the availability of web cameras on the internet, we approximated the images collected from the web cameras, and tested and analyzed the environment using 3D box and cylinder modeling. The tests were carried out on a 1.8GHz Intel Core CPU and the algorithms were developed with Matlab.

We used a web-camera data source located at 14th St, East Village NYC, U.S. [available at `http://joemaller.com/webcam/`]. The map showing the approximate area visible from the webcam is presented in Fig. 11. The Web camera image with moving cars, pedestrians and trees is presented in Fig. 12(a), and the approximated modeling of these features by using 3D boxes and cylinders can be seen in Fig 12 (b),(c).

Fig. 11. Approximate Area Visible from the Webcam Located at 14th St, East Village NYC, U.S. [available at `http://joemaller.com/webcam/`]

(a)



(b) (c)

Fig. 12. A Real Webcam-based Scene: (a) 14th St, East Village NYC Webcam Image with Moving Cars, Pedestrians and Trees; (b) Approximated 3D Model using 3D Boxes and Cylinders (Moving Cars Colored in Red, Buildings in Green, Pedestrians and Tree in Black); (c) Approximated 3D model from other 3D direction

Prediction from the current image of the web camera is the basic data source. Based on that, we estimate and predict the dynamic object's location at a future time, as described in Fig. 13(a) and Fig. 13(b).

In Fig. 13(c) we present the probabilistic visibility analysis where dynamic objects affect the visibility from the viewpoint to the buildings. We predict the dynamic

objects' location for a future time, t+Δt, where $\Delta t = 1[\text{sec}]$. The visible and hidden parts of the surfaces of the constant object, i.e. in this case, buildings, are determined by values between [0,1]. The probability that the dynamic object will be located at a specific point in a future time sets the visibility value, based on the web-camera image generated in the past. In Fig. 13(d), we present the same analysis from other viewpoint.

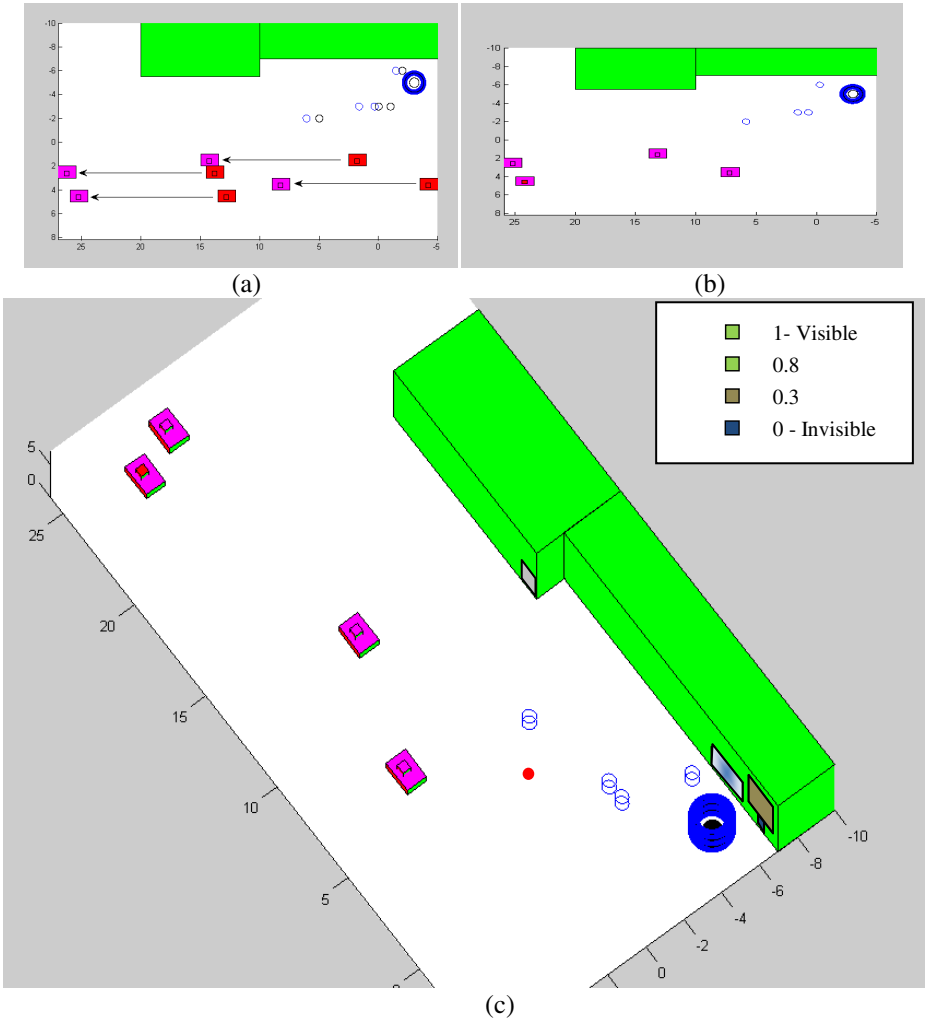
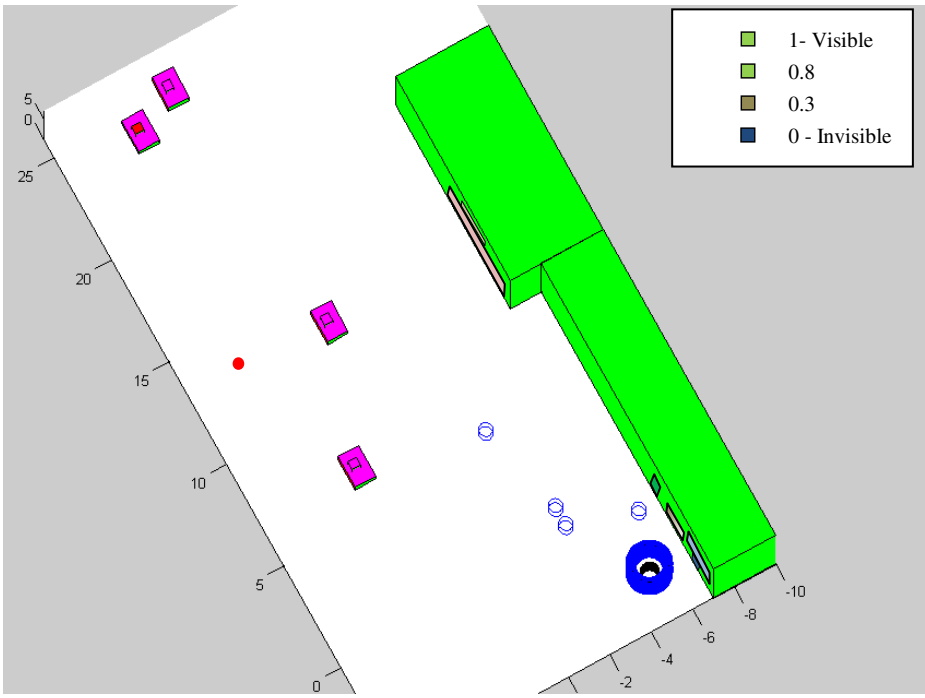


Fig. 13. Visibility Analysis of Scene no. 1: (a) Image from the Webcam at Time *t* with Estimated Car Location Marked with Arrows; (b) Scene in a Future Time; (c) Probabilistic Visibility Analysis, Viewpoint Marked in Red, Visible and Invisible Parts Marked in Different Colors on the Static Objects; (d) Probabilistic Visibility Analysis from another Viewpoint marked in red.



(d)

Fig. 13. (*continued*)

4.1 3D Model Construction From 2D Image

Modeling dynamic objects by using web-based data, such as cameras, is not a simple task. Creating a 3D scene from a 2D image is becoming more and more viable due to advanced hardware and image-processing computer vision algorithms [12]. Saxena et al. [19], presented a 3D scene structure algorithm from a single image using Markov Random Field (MRF) method. The algorithm was extensively tested and successfully applied on large-scale 3D models.

Nowadays, several products can be used to build a 3D model in urban scenes from a 2D image, such as [12],[13],[20].

Our simulations are based on the assumption that the images which are taken from web servers, such as Google Map web cameras, are reconstructed to 3D models using the current product and algorithms in this field. For simplicity, we approximated the dynamic and static objects from the scenes, as can be seen in the following simulations.

4.2 Efficiency

Efficient LOS-based visibility methods for DEM models, such as Xdraw, have been introduced in order to generate approximate solutions [6]. However, the computation

time of these methods is at least $O(n(n-1)$, and, above all, the solution is an approximate one. Complexity analysis for simple boxes is detailed in [8]. Extending a visibility algorithm, while also considering 3D cylinders with an analytic solution for visibility boundaries, is meaningless from the complexity aspect.

5 Conclusions and Future Work

We have presented an efficient algorithm for visibility computation in urban environments, which considers dynamic objects integrated to the scene. The urban environment is divided into static objects such as basic buildings, and to dynamic objects like moving cars, pedestrians and trees.

Due to the inherent character of dynamic objects, we must predict and estimate the future location of these objects in the environment. We present a probabilistic visibility analysis which is related to the probability of an object's presence in the environment and the effect on visibility analysis from validated models of transporting and human behavior fields. To illustrate our concept, we used web-cameras located at constant points in the environments, in order to update our model at each time period (as is already being done in the Google Maps application).

Moving car and basic building structures are modeled with mathematical approximating, for presentation of buildings' corners and car boundary modeling, by using 3D boxes. Pedestrians and trees are modeled by cylinders parameterization and extending the basic model structure. We formulated our visibility algorithm based on a fast visibility boundary challenging fast and exact hidden surfaces computation.

We tested dynamic objects' effect on urban environments by using web-cameras updating dynamic objects' location at certain time periods in the 3D model, as can be seen in simulations. The efficiency and complexity which has been presented in our previous work is still valid for 3D cylinders with probabilistic visibility.

The main contribution of the presented method in this paper is challenging dynamic objects' effect on visibility aspects in urban environments using a probabilistic concept, defining and formulating cylinder entity using fast visibility computation, all this without the need for special hardware.

Further research will focus on modeling and exploring other tree profiles with methods other than cylinders, and also consider other dynamic entities such as bicycle riders and non-rectangular building shapes.

References

1. Archer, J.: Methods for the Assessment and Prediction of Traffic Safety at Urban Intersections and their Application in Micro-simulation Modeling, Centre for Traffic Simulation Research, CTR, Sweden. Technical Report (2010)
2. Doytsher, Y., Shmutter, B.: Digital Elevation Model of Dead Ground. In: Symposium on Mapping and Geographic Information Systems (Commission IV of the International Society for Photogrammetry and Remote Sensing), Athens, Georgia, USA (1994)
3. Durand, F.: 3D Visibility: Analytical Study and Applications. PhD thesis, Universite Joseph Fourier, Grenoble, France (1999)

4. Fellendorf, M., Vortisch, P.: Validation of the Microscopic Traffic Flow Model VISSIM in Different Real world Situations. In: 79th Annual Meeting of Transportation Research Board, UK (2001)
5. De Floriani, L., Magillo, P.: Visibility Algorithms on Triangulated Terrain Models. International Journal of Geographic Information Systems 8, 13–41 (1994)
6. Franklin, W.R., Ray, C.: Higher isn't Necessarily Better: Visibility Algorithms and Experiments. In: Waugh, T.C., Healey, R.G. (eds.) Advances in GIS Research: Sixth International Symposium on Spatial Data Handling, pp. 751–770. Taylor & Francis, Edinburgh (1994)
7. Franklin, W.R.: Siting Observers on Terrain. In: Proc. of 10th International Symposium on Spatial Data Handling, pp. 109–120. Springer (2002)
8. Gal, O., Doytsher, Y.: Fast and Accurate Visibility Computation in a 3D Urban Environment. In: Proc. of the Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services, Valencia, pp. 105–110 (2012)
9. Hoogendoorn, S.P., Bovy, P.H.L.: Pedestrian route-choice and activity scheduling theory and models. Transportation Research Part B 38, 169–190 (2004)
10. Hu, X., Tao, W., Guo, Y.: Using FEM to predict tree motion in a wind field. Journal of Zhejiang University, Science A 9(7), 907–915 (2008)
11. Nadler, B., Fibich, G., Lev-Yehudi, S., Cohen-Or, D.: A Qualitative and Quantitative Visibility Analysis in Urban Scenes. Computers & Graphics 5, 655–666 (1999)
12. Make3D, http://make3d.cs.cornell.edu
13. PictoGraph, http://palentier.blogsopt.co.il/2011/11/3d-prehistoric-pictograph-printing.html
14. Park, B., Schneeberger, J.D.: Microscopic Simulation Model Calibration and Validation: Case Study of VISSIM Simulation Model for a Coordinated Actuated Signal System, Transportation Research Record 1856 , Paper No. 03-2531
15. Plantinga, H., Dyer, R.: Visibility, Occlusion, and Aspect Graph. The International Journal of Computer Vision 5, 137–160 (1990)
16. Parker, D., Lajunen, T.: Are Aggressive People Aggressive Drivers? A Study of the Relationship between Self-Reported General Aggressiveness Driver Anger and Aggressive Driving. Accident Analysis and Prevention 33(2), 243–255 (2001)
17. Ratti, C.: The Lineage of Line: Space Syntax Parameters from the Analysis of Ur-ban DEMs. Environment and Planning B: Planning and Design 32, 547–566 (2005)
18. Song, Y.: The research of a new Auto Target Recognition directed Image compression. In: 3rd Int. Congress on Image and Signal Processing (CISP), China, October 16-18 (2010)
19. Saxena, A., Sun, M., Ng, A.Y.: Make3D: Learning 3D Scene Structure from a Single Still Image. IEEE Transaction on Pattern Analysis and Machine Intelligence, PAMI (2008)
20. 3D Studio Max, http://usa.autodesk.com/3ds-max
21. Wiedemann, R., Reiter, U.: Microscopic Traffic Simulation: The Simulation System MISSION, Background and Actual State. Project ICARUS (V1052), Final Report, Brussels CEC.2: Appendix A (1992)
22. Wang, J., Robinson, G.J., White, K.: A Fast Solution to Local Viewshed Computation Using Grid-based Digital Elevation Models. Photogrammetric Engineering & Remote Sensing 62, 1157–1164 (1996)
23. Wang, J., Robinson, G.J., White, K.: Generating Viewsheds without Using Sightlines. Photogrammetric Engineering & Remote Sensing 66, 87–90 (2000)
24. Zlatanova, S., Rahman, A., Wenzhong, S.: Topology for 3D Spatial Objects. In: International Symposium and Exhibition on Geoinformation, pp. 22–24 (2002)

Exploring Spatial Business Data: A ROA Based eCampus Application

Thanh Thoa Pham Thi, Linh Truong-Hong, Junjun Yin, and James D. Carswell

Digital Media Centre, Dublin Institute of Technology, Ireland
{thoa.pham,linh.hongtruong,jcarswell}@dit.ie,
yinjunjun@gmail.com

Abstract. In "Smart" environments development, providing users with search utilities for interacting efficiently with web and wireless devices is a key goal. At smaller scales, Google Maps and Google Earth with satellite and street views have helped users for querying general information at specific locations. However, at larger local scales, where detailed 3D geometries linked to business data are needed, there is a recognized lack of related information and functionality for in depth exploration of an area. Linking spatial data and business data helps to enrich the user experience by fulfilling more task specific user needs. This paper presents an eCampus Demonstrator for the National University of Ireland, Maynooth (NUIM) based on a Resource Oriented Architecture (ROA), in which various *RESTful* web-services have been developed and deployed for querying both spatial data and associated business data. The benefits and drawbacks of the chosen technologies are also discussed. This work can be considered as a platform that can be applied to similar application domains such as exploring business parks, hospitals, museums, etc.

Keywords: RESTful web-services, Resource Oriented Architecture, Mobile Spatial Interaction, eCampus Information Systems, 3D modeling.

1 Introduction

Utilities for location-dependent queries are readily provided today by Google Maps and Google Earth with satellite and street views. However, at local scales, where detailed 3D geometries and associated business data are needed, there is a general lack of related information and search functionality for in depth exploration of an area. For instance, the following types of questions cannot be answered when interacting with Google Maps/Earth on a typical university campus: what classes are scheduled in that room over there? Whose office window is that up there? What specific buildings/classrooms/labs can I actually see around me while standing at a particular location on campus? Or in a hospital environment, by spatial querying hospital datasets we can ask what the surgery schedule is in that operating room? What medical equipment is installed in that room? What route should I take to move a bed or medical equipment from this room to another?

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 164–179, 2013.
© Springer-Verlag Berlin Heidelberg 2013

In order to answer these types of specific questions, Location Based Services (LBS) need to link spatial data with non-spatial business data. Spatial data deals with detailed topology and geometry or coordinates of objects while business data can describe the business semantic aspect of a related object in some business domain, e.g. a university campus.

In general, geospatial data does not change that much over time but business data seems to change faster depending on the nature of the business. Furthermore, conventional business data is often managed and produced by enterprise information systems independently of any geospatial applications. Thus linking spatial data and business data in one application helps to enrich the user experience by fulfilling more specific user needs. In particular for task specific decision making applications that need access to detailed local scale data typically found in Zoos, Museums, Hospitals, a University Campus or Business Park settings. Business data is often indirectly or virtually associated to spatial data via its location given by a generic address, a room number, a building name, or even lat/long coordinators of the business location. Such business data can provide further detailed information to users about the objects in question and be tailored or application domain dependent. For instance, in a university campus environment, the business data involved may be in the form of class schedules for a specific room, lists of equipment installed in a lab, office hours or contact details for a lecturer, today's special meal deal in a cafeteria, etc..

Typically, 2D footprint data provides just flat geometry representation of physical objects (e.g. buildings) in the horizontal plane. But for our application we need spatial and non-spatial attribute details in the vertical dimension as well for in-depth 3D BIM (Building Information Modeling) data query operations. Depending on how much 3D detail data is captured and how available any related business data is, more relevant task specific answers can be provided to the user. For instance, 3D BIM data of a building can include detailed digital representations of physical and functional characteristics for its different floors, rooms, windows, and doors where all aspects are potentially available for interrogation. In contrast to our approach of exploring such detailed 3D data, in [11] the authors also reconstruct 3D models of a campus but in this case it stays at the building level, where the room details have not been fully developed, and the implementation does not provide utilities to further question the area beyond its physical, spatial nature.

Within the framework of the StratAG project (http://stratag.ie/), we developed an eCampus Demonstrator for the National University of Ireland, Maynooth (NUIM). This web and wireless based GIS application aims to assist users in exploring and analyzing their surroundings within a detailed data environment; in our case, domain specific business data is provided together with 3D spatial data to answer more specific user queries. The application addresses two types of users: public users (e.g. visitors) and local users (e.g. students and staff). Access privileges and query levels depend on user type. For example, visitors are presented with a campus map for general information querying of buildings and rooms, and utilities for general navigation to various buildings or departments. Meanwhile, in addition to these functionalities, staff and students are able to visualize their class schedules together

with personalized navigation plans and recommendations based on their academic and social interests.

Each partner in the StratAG cluster is responsible for developing different functionality in the Demonstrator such as utilities for directional querying the area, for path navigation assistance, for user interests recommendation, or for 3D modeling of the campus itself. *RESTful* web-services [7] were chosen for the deployment technology of these distributed components thanks to its simplicity when applied particularly to the geospatial domain [8].

In this paper, we present the architecture of our eCampus Demonstrator based on a Resource Oriented Architecture (ROA) model as it consumes and integrates the *RESTful* web services mentioned above. We focus on 2D and 3D web services for querying various distributed university datasets and for 3D modeling of campus buildings. The remainder of the paper is organized as follows: Section 2 gives an overview of the *RESTful* web-service concept and the design principles of ROA; Section 3 follows with a description of our eCampus Demonstrator application based on the ROA where different layers of the architecture are explained in more detail; Section 4 discusses some strengths and weaknesses of today's available technology for building such applications; and Section 5 draws conclusions and gives some direction for future work.

2 Resource Oriented Architecture (ROA)

ROA is "a paradigm for organizing and utilizing distributed *resources* that may be under the control of different ownership domains" [8]. Those distributed *resources* are called *RESTful* web-services. *REST* stands for the *Representational State Transfer* style that is "an abstraction of the architectural elements within a distributed hypermedia system" [4, 7]. It relies on basic web protocols HTTP and MIME (Multipurpose Internet Mail Extensions) [9]. Some main concepts of *REST* can be found in [7] as follows: *Resource,* which is the central concept, is "the intended conceptual target of a hypertext reference". Anything can be a resource such as a document, an image, physical object, or a collection of other resources, etc.; The *resource identifier* is the hypertext link Uniform Resource Locator (URL) or Uniform Resource Identifier (URI). When the link is selected, the resource will be moved from where it is stored to where it is used by HTTP.; The *representation* in *REST* is a sequence of bytes, plus *representation metadata* to describe those bytes such as a text file or an image; In addition, the *REST* implementation requests some constraints such as *Uniform Interface*, *Self-Describing Messages*, *Hypermedia Driving Application State* and *Stateless Interactions* which are discussed in detail in [1].

A *RESTful* web-service is any web-service that implements the *REST* architecture style and is identified by a URI or URL link on the web. Calling a *RESTful* web-service is done simply by accessing its resource identifier through the HTTP protocol. There are typically five HTTP operations available to *RESTful* web-services [4]: PUT (create a new resource), GET (retrieve the representation of the resource), DELETE (delete the resource), POST (modify the resource) and HEAD (obtain meta

information about the resource). For instance, when calling a *RESTful* web-service with the GET operation, the service returns (gets) the results (resource representation) in popular formats such as XML, JSON (JavaScript Object Notation), or KML (Keyhole Markup Language) format. The following example illustrates calling a *RESTful* web-service with Ajax-Javascript and retrieving the resource in JSON format:

```
$.ajax({
  type: "GET",
  url: URIstring, // URI of the RESTful web-service
    with its parameters
  dataType: "json",
  success: function(js){
    // process js
  },
  error: function(e){
    alert(' Error ' + e.status + ' : ' +e.statusText);
  }
});
```

In the scope of our demonstrator, the system aims at providing answers to user queries that demand retrieval of both spatial data and business data. Therefore only the GET operation is used.

3 ROA Based eCampus Application

Our NUIM eCampus Demonstrator is a web-based application accessible to both desktop and mobile devices. It aims to help users explore in more detail the university campus by providing them maps and utilities for both 2D and 3D querying. Different query functionality is provided so that users can ask questions by interacting with the map itself. For instance, they can ask: What is that building over there by pointing at it with their mobile device; What is the class schedule of this classroom by clicking on it; What can I actually see around me when standing in a particular location on campus, etc. The query results are displayed on the map afterwards together with links to further business data where available. This process is repeated and illustrated in Figure 1 in UML (Unified Modelling Language) activity diagram notation as follows:

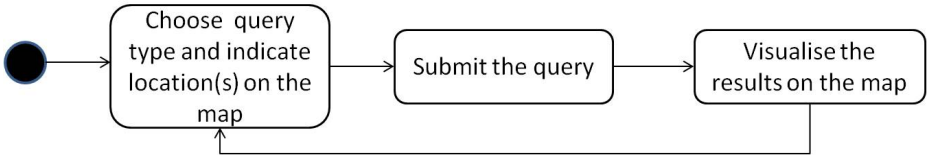


Fig. 1. The workflow of user interactions

The application invokes various *RESTful* web-services to access the various query functionality. A catalogue of the services is provided in the form of a list of URI services with a description of their functionality and required parameters. The application architecture is depicted in Figure 2. It includes 3 main layers: Database Layer; Web-Services Layer; and the Interface Layer.

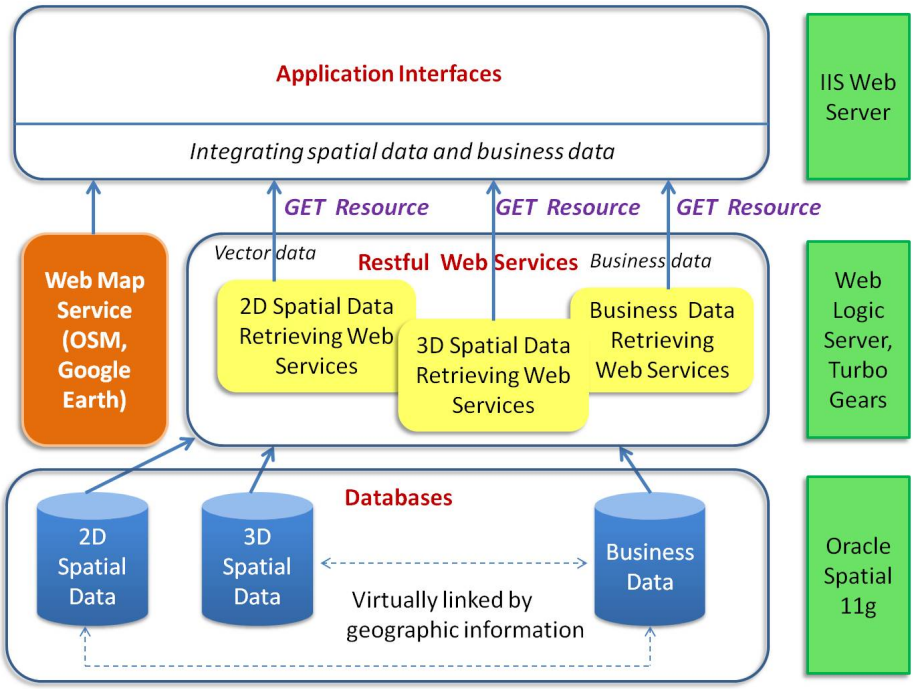


Fig. 2. eCampus Demonstrator architecture based on ROA

3.1 Database Layer

The 2D footprint and detailed 3D models of NUIM campus are hosted in Oracle Spatial 11g databases that also serve the spatial data retrieving web-services. For visualizing 3D buildings in Google Earth for querying, corresponding KML files are prepared. This preparation process will be discussed in more detail at the end of this section.

2D Database Preparation.
Many online maps such as Google Maps and Bing Maps have considerably poor data coverage over many areas especially in those less populated, like Maynooth Ireland. As NUIM's campus is located in such a rural area, OpenStreetMap (OSM) shows the advantage of collective efforts for Volunteered Geographic Information (VGI), where

anonymous users contribute to OSM by uploading geographic features such as buildings, streets, point-of-interests (PoIs), etc. to complete the map coverage. In particular, students and others from NUIM have already done this, which proved very beneficial in our case [15]. Therefore, many 2D campus footprints were first downloaded from OSM and then uploaded to our local Oracle Spatial DBMS where geometry data is stored in a single column data type of SDO_GEOMETRY which defines the geometry type (e.g. points, lines, polygons, solids, etc.), the dimension, and an array of x, y (and z for 3D) coordinates comprising points or vertices of campus objects.

3D Database Preparation.

According to [16], volumetric objects like 3D buildings can be represented by polyhedrons, triangulated polyhedrons or tetrahedrons of which the polyhedron is the most appropriate representation for buildings. As such, the polyhedron that bounds a solid comprised of multi-polygons [17] used to construct a 3D building model is adopted in our work. Meanwhile, a simple polygon is used to represent the 2D footprint of a room. Figure 3 describes the process of capturing 3D building data and transferring the data into the Oracle Spatial databases used in our application.

Firstly, the 3D building models were manually reconstructed from terrestrial laser scanning data by using proprietary programs of the scanner and CAD environments where point cloud data gets geo-referenced before post-processing. 3D solid objects represent the various building components and polygons are created from the borders of windows and doors in order to support spatial queries of objects at a detailed level. Further room attribution derived from architectural drawings is also assigned to the extracted room level data. In summary, 3D solids and 3D polygons respectively represent the campus building models and their associated rooms linked to any available metadata information. The developed database schema consists of two tables named NUIMBUILDING and NUIMBUILDINGOBJECT, where the first table contains the geometry of the buildings, while the second stores the geometry of the associated windows/doors (Figure 3).

Additionally, for exporting 3D objects in an AutoCAD environment, the Feature Manipulation Engine (FME) workbench utility was used. A workflow for exporting data from a DWG file of AutoCAD into Oracle Spatial was developed and is shown in Figure 4. The upper right window depicts the workflow to transfer the external objects (exterior walls, roofs, balconies and window frames) of the buildings into the table NUIMBUILDING. These objects were stored in a row of that table, which are underlying multisurface (SDO_GTYPE = 3007). The lower right window illustrates the workflow to transfer the polygons on peripheries of all windows of each room to a row in the NUIMBUILDINGOBJECT table as heterogeneous (SDO_GTYPE = 3004).

Note that at this phase of the project, objects inside buildings are not modelled and captured. A room is represented by its windows, selecting a room corresponds to selecting one of its exterior windows (or doors). Therefore our spatial database contains mainly building objects and associated window/door objects.

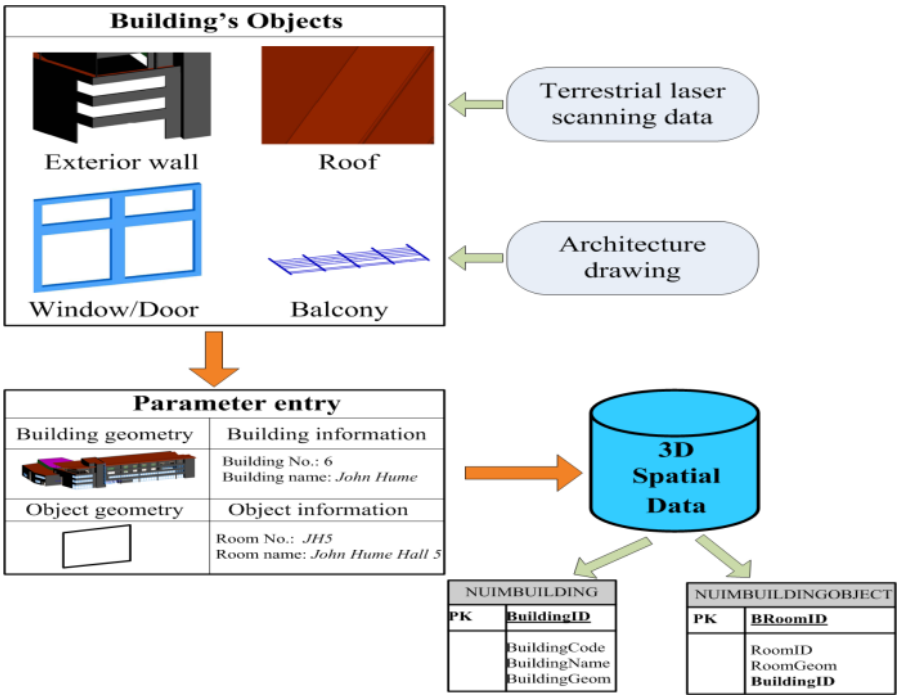


Fig. 3. Process of 3D data development in Oracle Spatial 11g

KML File Preparation.

The process of converting data utilizing FME workbench is shown in Figure 5, where 3D geometries are made readable by Google Earth by first transforming them to a KML/KMZ file as 3D polygons through two schemes. The first scheme (upper middle window in Fig 5) deals with the creation of all external objects of a building (e.g. exterior walls, roof, doors, balcony and window frames) associated with general information (e.g. building name) through the AttributionCreator Function of FME workbench. In the second scheme (lower middle window), polygons of window/door peripheries and associated "room" attribution were loaded into a KML file, in which the room name was directly extracted from CAD drawings. In this work, information about the building and its related rooms were defined through a transformer KMLPropertySetter while rendering was carried out by manually assigning colors for each geometry object via a transformer KMLStyle in the FME workbench utility. Loading general information about buildings and its rooms (windows/doors) into a KML/KMZ file is used for display when users click on those objects in Google Earth.

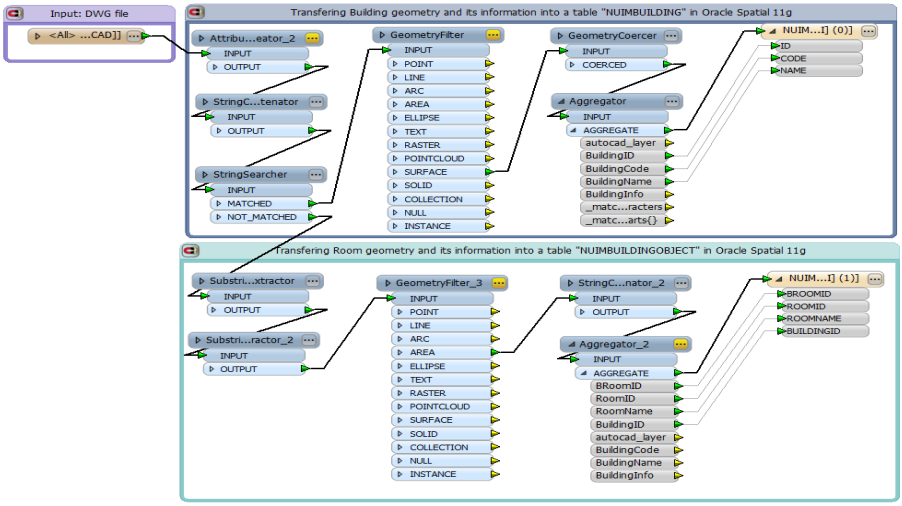


Fig. 4. Process of transferring geometries of buildings and associated windows/doors together with their metadata information into Oracle Spatial 11g

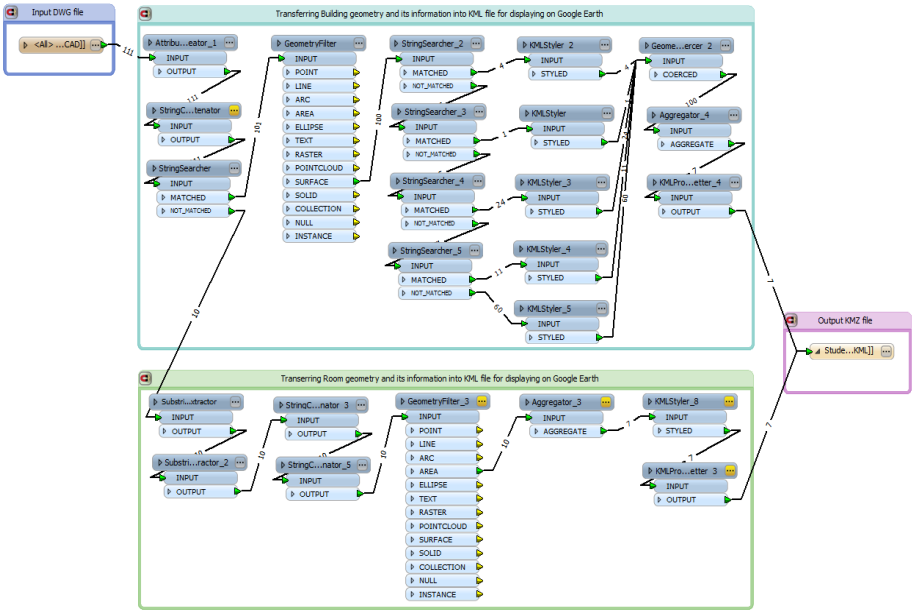


Fig. 5. Process of transferring geometry of a building and associated windows/doors into a KML file suitable for upload to Google Earth

3.2 Web-Services Layer

This layer relates to the development of spatial data retrieving web-services and business data retrieving web-services.

Spatial Data Retrieval Web-Services.
The collection of spatial functions for performing 2D and 3D visibility based directional querying are deployed in the form of *RESTful* web-services. In relation to the different spatial data formats found in the database, the web-services are divided into two sub-groups, one applied to 2D campus footprints and the other applied to 3D campus models. A diagram illustrating detailed components of the various web-services is shown in Figure 6. All neighbors, Field-of-view, Line-of-Sight and Threat dome are just a few of the different types of spatial queries available to the user where different search algorithms and web-services were developed and deployed for each query type [13, 14].

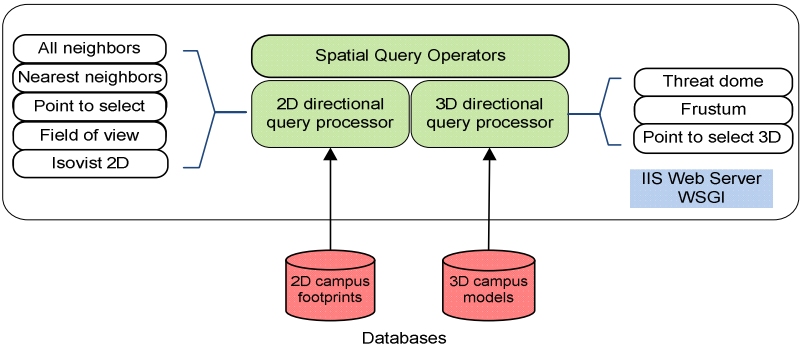


Fig. 6. System architecture for delivering 2D/3D spatial query services

More specifically, an IIS (Internet Information Services) server is appointed to host the web-services. Query requests are constructed using standard HTTP calls containing a valid URL filled with the required query parameters, like the location of a mobile device. A Web Server Gateway Interface (WSGI) was used for Python programming, which defines a universal interface for communications between the web server and web applications. Below are two example URLs constructed for requesting a particular web-service:

URL1:
http://ecampus.dyndns.org/fov/?tg_format=JSON&lat=53.3387&lng=6.2675&heading=44.0&angle=30.0

URL2:
http://ecampus.dyndns.org/point2select3d/?tg_format=JSON&lat=53.383522033691406&lng=-6.60053253173828&heading=44.0&tilt=30.0

URL1 is for the "field-of-view" query request, where (fov) in the string is the function name, the specified response format is (JSON) whilst the required parameters are: the user's coordinates in the form of latitude (lat) and longitude (lng), their heading or pointing direction (heading) and the size of the view angle (angle).

URL2 is for the "point-to-select" or "line-of-sight" query request in a 3D environment. The format for a 3D query URL is almost the same as in 2D. However, because it involves the vertical dimension, the last parameter in this function is the tilting angle (tilt) of the mobile device (taken from the accelerometer) at a particular location (taken from the GPS receiver) while maintaining the heading direction (taken from the compass). For the desktop version of this 3D query, the above parameters are input by the user directly on the Google Earth interface.

The response from the service is wrapped in JSON format which is essentially a text based key-value set. For example, the result returned from a "field-of-view" 2D web-service call is presented below. This returns all objects intersecting the user's viewshed polygon at the current location, view angle, and heading with its attribute information listed as name, latitude and longitude respectively. It is up to the application developer to further parse this information and visualize accordingly depending on end user requirements.

```
{"Result":{"items":
[["name" : "John Hume Building",
"coordinate" : { "lat": 53.3833658533,"lng":
-6.5994806592},"shape": [-6.6004514, 53.3841367,
-6.60037709999995, 53.3840353000001,...,
-6.600309499999979, 53.3842268000001, -6.6004514,
53.3841367]], ["name" : "Saint Catherines Building",
"coordinate" : { "lat": 53.383704272, "lng":
- 6.5991442043}, "shape": [-6.5966732, 53.3821449, -
6.5966932, 53.3816629,..., -6.5961998, 53.3821441, -
6.5966732, 53.3821449]]]}}
```

Business Data Retrieval Web-Services.

The business data retrieval web-service is used for querying schedules of class rooms, retrieving equipment lists of labs, or for retrieving timetables of students from separate databases. In our application, such business data also happens to be stored in Oracle 11g but maintained by NUIM independently. The web-services for retrieving business data are also *RESTful*, and are hosted on an Oracle Web Logic server. For example, the following web-service call returns the current class schedule for room 1_02JH (1st floor, room number 2, in John Hume Building) in JSON format indicating date, time, and class modules taking place in that room for the week:

URL:
http://147.252.87.151:7101/getSchedulecontextroot/jersey/Schedule/getS?roomID =1_02JH

Returns:

```
{"schedule" : {"items":[["Monday",9,"MN304"],["Monday",
11, "MT211S"], ["Tuesday",10.30, "HY210"], ["Thursday",
13, "EC201"]]}}
```

In our approach, the integration of web services is performed at the interface level. This is presented in more detail in the following section.

3.3 Interface Layer

The integration of spatial data and business data is performed at the client side when information about buildings and associated rooms is found and returned. The retrieved spatial data is visualized on a map as additional layers for 2D queries (using OpenLayers API). Meanwhile for 3D queries, the results are returned in JSON format and drawn as Placemarks added to a Google Earth view.

Once the room(s) detail, including room number, is extracted from the result set, it is filled into the URI for calling the getSchedule web-service, then this response is added to a *Balloon feature* of the Placemark so that the corresponding business data gets displayed whenever the room is highlighted as showed in the following code:

```
var balloon = ge.createHtmlStringBalloon('');
balloon.setFeature(polygonPlacemark);
  // polygonPlacemark is the window viewed
balloon.setContentString(content);
  // content is business data
ge.setBalloon(balloon);
```

Figure 7 illustrates a directional query from the user's current position and pointing to a specific room in a building using the Point-to-Select web-service. The yellow line illustrates the user's 3D pointing direction, the coloured window represents the room. When the users click on that window, the class schedule of the corresponding room is displayed. In case of a Frustum query, it depends on the provided horizontal angle and vertical angle so that any query response may involve many rooms.

Figure 8 shows the result of a 360˚ Isovist query in 2D. Users can either click on the map while interacting with the desktop version of the application or use their mobile GPS reading when interacting with the wireless version to indicate a location, the red polygon represents the viewshed of the user in 360 degrees out to a user defined radius. The blue polygons represent the objects that intersect the Isovist (i.e. what users can actually see from their current location in all directions).

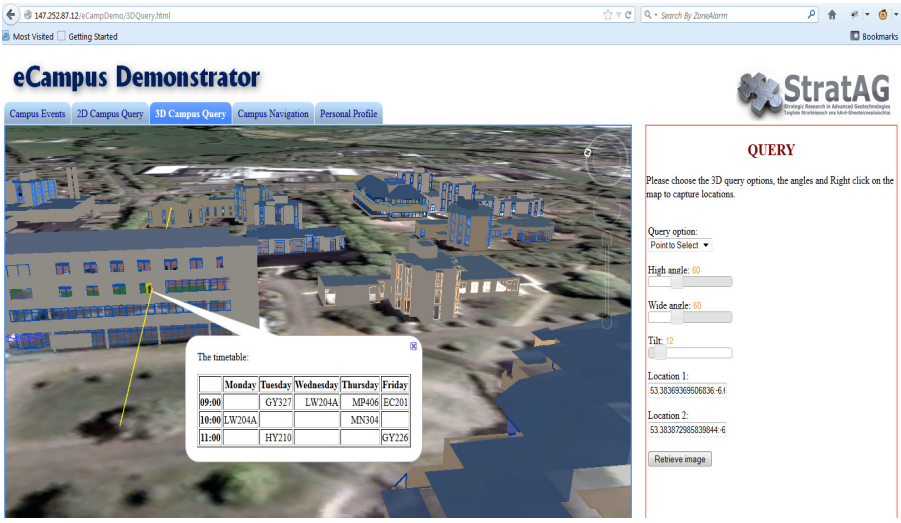


Fig. 7. The result of a Point-to-Select 3D query showing linked business data (class schedules) retrieved for that room in a *Balloon feature* within Google Earth

Fig. 8. The result of a 360˚ Isovist 2D query

4 Discussion

During the development of this application, some important limiting conditions emerged regarding the technology employed. After analyzing the capabilities of the chosen technologies for implementation, the following discussion presents some advantages and disadvantages of our approach.

Integration Level of Spatial Data and Business Data.
As mentioned previously, the integration of spatial data and business data is performed at the client side, i.e. at the visualisation level. We considered three options to provide spatial data and related business data to users as shown in Figure 9.

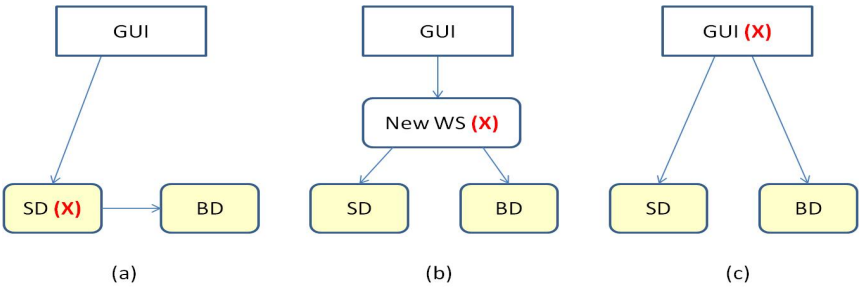


Fig. 9. Different spatial data and business data integration approaches. The red (X) represents the integration point. The arrow describes the calling direction. WS: Web-Service; SD: spatial data WS; BD: business data WS; GUI: Graphical User Interface

In Figure 9 above:

(a) Early-integration: In this approach, retrieving business data is performed from inside the spatial data retrieving WS. The final results sent back to GUI include spatial data and business data.

(b) Aggregated web service: A new web-service is developed to compose the results returned by the spatial data web-service and the business data web-service.

(c) Integration at the visualisation level: The results of the spatial data web-service and business data web-service are overlaid at the visualisation level. We have chosen the third approach: integration at the visualisation level, as this approach provides some advantages compared to the other two approaches (Table 1).

OGC Services.
The Open Geospatial Consortium (OGC) has developed some standards for geospatial processing technologies to enable applications from different commercial vendors to interoperate. However, the locationing services developed within OGC focus mainly on tracking and location-based applications for mobile devices [10]. However, these services are far from what we require in this application, where all *RESTful* web-services for location-dependent directional and Vista space querying have been developed in our own algorithms.

Table 1. Spatial data and business data integration options

(a) Early-integration	There is a dependency of SD retrieving WS and a specific BD retrieving WS. That means SD WS cannot be reused for other purposes
(b) Aggregated web-service	SD and BD WS are independent. It depends on the user needs that the aggregated WS (AggWS) will integrate suitably SD and BD WS. Suppose *fA(WS)* is the cost of analyzing the result of a WS, *fC(WS)* is the cost of calling a web-service, then the cost of this approach to display the result is: $fC(AggWS)+ fA(Agg\ WS) + fC(SD)+ fA(SD)+ fC(BD)$ Note that the results of AggWS needs to be analyzed to draw the geometry shapes and the business data is then added to the feature data of the geometry object. In case the results of AggWS is in KML format, there is no need of analysing AggWS (so no cost), all spatial data and business data can be visualized. However, in that case the visualisation is fixed according to the API provided.
(c) Integration at visualisation level	SD and BD WS are independent. It depends on user needs that suitable SD and BD WS are consumed at the visualisation (client side code source). The cost to display the result to the users is: $fC(SD)+ fA(SD)+ fC(BD)$ Visualisation of the results is flexible according to the users' needs.

ROA Instead of SOA.

For the last decade, Service Oriented Architectures (SOA) have been widely used for distributed applications, particularly on the Web. In Geographic Information Systems (GIS), there is no exception here. For instance in [2], a GIS web-service architecture was proposed based on SOA technology. However, while SOA is a proven approach, in some cases it can be overly complicated thus processor heavy. For example, when handling a SOAP message, the client (desktop or mobile) needs to send a request with parameters constructed and wrapped in XML format with special headers and other elements. It also has to parse any response from the server in the same effusive XML format [6]. But in the case where the client is a mobile device, this approach contains far too much processing overhead in terms of the volume of data, most of it quite unnecessary, that must be sent/received on mobile devices having relatively limited wireless connection speeds and often a data transmission cost [12]. In this respect, JSON is a much lighter data format in terms of processing and transmitting wirelessly. Furthermore, we agree with the general statement that the *REST* architecture provides a "scalable and simple deployment of web services and particularly appealing for Earth and Space Science" [8] as *RESTful* web services have been much used in geo-information sharing.

Dependency of 3D Query Performance and 3D Data Details.
In our application, users carry out spatial queries from outside buildings. Therefore only the geometries of exterior structural components of the building (e.g. facades, roofs, windows, doors, balcony, canopy and so on) associated with room level attribution (e.g. room name and function) were loaded into the database. In this way, it helps to reduce the complexity of the 3D models and thus improve 3D spatial query performance.

Limitations of the Length of URI Link.
In theory, the HTTP protocol does not limit the length of a URI (according to RFC 2616 - Hypertext Transfer Protocol - HTTP/1.1). However some browsers may limit extremely long URI text strings, therefore the application should take into account this issue when designing *RESTful* web-services.

5 Conclusions and Future Work

Providing users with business context data in location dependent queries helps to fulfil more users needs within detailed data environments. The flexibility, interoperability and heterogeneity of this kind of linked search application demands a suitable software architecture. In particular to this geospatial application, a Resource Oriented Architecture (ROA) was chosen for the implementation.

The eCampus application allows users to explore the NUIM campus in 2D and 3D by interacting with Open Street Map and Google Earth interfaces. We also discussed the benefits and drawbacks of the chosen technologies. This work can be considered as a platform for developers and researchers when developing for similar application domains, such as when exploring a business park, hospital, or a shopping centre, etc.

In the next version of the Demonstrator, we address more advanced query types to support more specific user needs when exploring the campus in 3D mode. For example, the system could answer queries about a specific building such as, "Show me all class rooms that are available (i.e. not booked) from 9-10am?". These kinds of queries need more business web-services to retrieve and analyze the required business data. Consequently, the user interfaces also need to be improved to allow for asking these more detailed questions.

In the near future we aim to test the application in the hands of real students/staff/visitors to NUIM Campus on mobile devices which, apart from interacting with the map to indicate specific locations and ad hoc directional queries as usual, real-time user location and personal preferences can also be gathered and exploited as input query parameters to other StratAG partner web-services included in the complete eCampus Demonstrator application. Enhancements to graphical object interaction functionality within the desktop Google Earth interface will also be investigated.

Acknowledgements. Research presented in this paper was funded by a Strategic Research Cluster Grant (07/SRC/I1168) by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support.

References

1. Guinard, D., Trifa, V., Wilde, E.: A Resource Oriented Architecture for the Web of things. In: IEEE International Conference on Internet of Things, Tokyo (December 2010)
2. Alameh, N.: Chaining Geographic Information Web Services. IEEE Internet Computing (September-October 2003)
3. Lucchi, R., Millot, M., Elfers, C.: Resource Oriented Architecture and REST, JRC Scientific and Technical Report (2008)
4. Fielding, R.T., Taylor, R.N.: Principal Design of Modern Web architecture. ACM Transactions on Internet Technology 2(2), 115–150 (2002)
5. OpenGIS Abstract Specification Topic 12: OpenGIS Service Architecture (2002)
6. Snell, J., Tidwell, D., Kulchenko, P.: Programming Web Services with SOAP. O'Reilly Publisher (December 2001)
7. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. PhD dissertation. University of California, Irvine (2000)
8. Mazzetti, P., Nativi, S., Caron, J.: RESTful Implementation of geospatial services for Earth and Space Science applications. International Journal of Digital Earth 2 (suppl. 1), 40–61 (2009)
9. Kurtagic, H., Birch, J., Zeiss, G.: An Open Architecture for RESTful Geospatial Web Services. In: FOSS4G, Sydney (2009)
10. OGC Report- Summary of OGC Web Services, Phase 4, Interoperability Testbed (2007)
11. Kaharaman, I., Karas, I.R., Rahman, A.A.: Developing Web-Based 3D Campus Information System. In: ISG & ISPRS (2011)
12. Yin, J., Carswell, J.D.: Touch2Query enabled mobile devices: a case study using OpenStreetMap and iPhone. In: Web & Wireless Geographical Information System (W2GIS 2011), pp. 203–218. Springer, Kyoto (2011)
13. Carswell, J.D.: 3DQ: Threat Dome Visibility Querying on Mobile Devices. GIM International 24(8), 24 (2010)
14. Carswell, J., Gardiner, K., Yin, J.: Mobile Visibility Querying for LBS. Transactions in GIS Journal 14(6), 791–809 (2010)
15. Jacob, R., Zheng, J., Ciepłuch, B., Mooney, P., Winstanley, A.C.: Campus Guidance System for International Conferences Based on OpenStreetMap. In: Carswell, J.D., Fotheringham, A.S., McArdle, G. (eds.) W2GIS 2009. LNCS, vol. 5886, pp. 187–198. Springer, Heidelberg (2009)
16. Zlatanova, S.: 3D geometries in spatial DBMS. In: 3D-GIS, pp. 1–14 (2006)
17. Musliman, I.A., Abdul-Rahman, A., Coors, V.: Incorporating 3D-GIS Spatial Operator with Building Information Models in Construction Management using Geo-DBMS. In: 5th International Conference on 3D GeoInformation, Berlin, Germany, November 3-4, vol. XXXVIII-4/W1, pp. 147–154 (2010)

ISOGA: A System for Geographical Reachability Analysis*

Markus Innerebner[1], Michael Böhlen[2], and Johann Gamper[1]

[1] Free University of Bozen-Bolzano, Italy
[2] University of Zurich, Switzerland

Abstract. In this paper, we present a web-based system, termed ISOGA, that uses isochrones to perform geographical reachability analysis. An isochrone in a spatial network covers all space points from where a query point is reachable within given time constraints. The core of the system builds an efficient algorithm for the computation of isochrones in multimodal spatial networks. By joining isochrones with other databases, various kinds of geospatial reachability analysis can be performed, such as how well is a city covered by public services or where to look for an apartment at moderate prices that is close to the working place. ISOGA adopts a service-oriented three-tier architecture and uses technologies that are compliant with OGC standards. We describe several application scenarios in urban and extra-urban areas, which show the applicability of the tool.

Keywords: Spatial networks, isochrones, geospatial reachability analysis, WebGIS.

1 Introduction

Geospatial analysis covers various approaches to perform analysis on data with a geographical dimension and provides an important tool in many application areas, including environmental sciences, social sciences, emergency management, or city planning.

In this paper, we describe ISOGA, a system for geographical reachability analysis using isochrones in multimodal spatial networks. An *isochrone* in a spatial network is a possibly disconnected subgraph that covers all space points from where a query point q is reachable within a given time span and by a given arrival time at q. As an example, consider a person looking for an apartment in a specific price range, from where his/her working place is reachable in less than 15 minutes using the public transportation system. Figure 1 illustrates this query for Bozen-Bolzano. The '*' indicates the working place (query point), the gray area represents the isochrone, and white circles represent buildings that satisfy the search criteria. The popup shows additional information about one of the qualifying apartments, such as the actual distance and details on how (bus numbers and departure times) to reach the working place.

In ISOGA, the user inputs first the parameters for an isochrone, i.e., one or more query points, walking speed, arrival time, and a maximal timespan. At the core of the system is an efficient algorithm for the computation of isochrones in multimodal spatial

* This work is partially funded by the Province of Bozen-Bolzano as part of the AQUIST project.

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 180–189, 2013.
© Springer-Verlag Berlin Heidelberg 2013

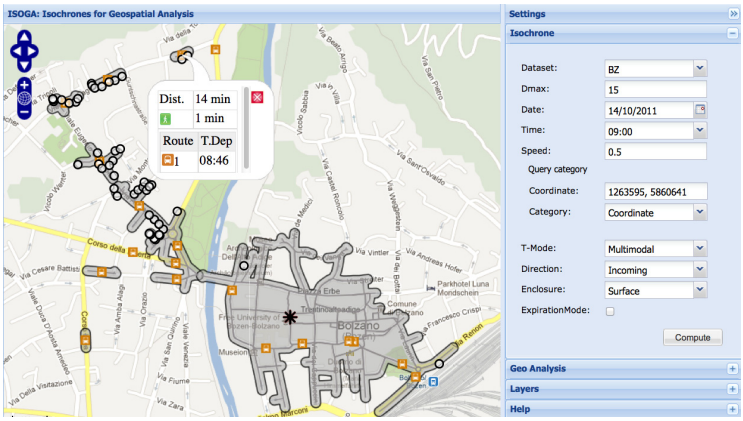


Fig. 1. A 15 minute Isochrone in Bolzano-Bozen

networks together with the possibility to join an isochrone with an arbitrary relation of geo-referenced objects, e.g., people, houses, or hotels. First, the disk-based algorithm MINEX computes an isochrone as a subgraph of the multimodal network. Second, the surface of the isochrone subgraph is determined. Third, the isochrone surface is joined with a relation of geo-referenced objects, which can be specified by the user as an arbitrary SQL expression. As the result of a query, a simple summary statistics is shown together with a list of all objects that are located within the isochrone. The objects can be visualized on the interactive map as well, and by clicking on an object a popup shows additional information. ISOGA adopts a service-oriented three-tier architecture and uses standardized OGC services for exchanging spatial data. The system can be accessed at `www.isochrones.inf.unibz.it/isoga`. To summarize, the main contributions of this paper are as follows:

- We present the ISOGA system for geographical reachability analysis in multimodal networks, which uses isochrones to efficiently determine geo-referenced objects that are reachable within given time constraints.
- We describe the three-tier architecture of the system with an interactive WebGIS client that uses OGC standards for the communication between client and server.
- We discuss three application scenarios using real-world data that illustrate the applicability of ISOGA for various kinds of geographical reachability analysis.

The rest of the paper is structured as follows. Section 2 discusses related work. In Section 3 we describe the scientific background of the system. The system architecture is presented in Section 4. Section 5 describes three application scenarios.

2 Related Work

Previous work on isochrones, upon which this paper is based, has been presented in [1, 4, 5, 7]. Bauer et al. [1] introduce a main memory algorithm that suffers from a high initial loading cost and is limited by the available memory. To overcome these limitations, Gamper et al. [4] propose a disk-based algorithm that loads the network

incrementally and is independent of the network size. This work is extended in [5] with network expiration, which allows to keep in memory only the expansion frontier to avoid cyclic expansions. Marciuska and Gamper [7] present two approaches to transform an isochrone subgraph into a spatial area that is needed for joins with other spatial objects. The architecture of the ISOGA system follows the WebGIS framework in [6].

Different query types have been studied for spatial network databases, e.g., [2, 3, 8]. Isochrones are closest to range queries which return all objects that are within a given distance, whereas an isochrone query returns all *space points* within a given distance. Isochrone queries are more flexible than range queries. Once an isochrone is computed, it can be reused to retrieve any kind of geo-referenced objects that are located within the isochrone without the need to compute the actual distance to these objects.

The project pgRouting[1] extends the spatial DBMS PostGIS with geospatial routing functionalities. The driving distance (isoline) function supports range queries with a dynamic cost parameter, but it does not support schedule-based networks. Mapnificient[2] uses a simple heuristic based on the Euclidean distance to approximate isochrones. Network expansion examines only the transportation network, whereas the reachable areas in the pedestrian network are approximated by drawing a circle around bus stops with a radius that is determined by the available time. A similar approach is adopted in Mapumental[3], which uses isochrones for house hunter services. The company Hacon[4] offers the product Hafas that computes range queries in transportation networks. The trip planner OpenTripPlanner[5] provides an extension for isochrones to measure the accessibility to or from specific locations as well as to perform aggregate search analysis.

The ISOGA system described in this paper integrates previous work on isochrones into a tool for geographical reachability analysis. Different from other systems, ISOGA works for multimodal networks that represent different transportation modes, and it efficiently computes exact isochrones rather than approximate solutions. The geo-referenced objects used in the analysis can be specified by an arbitrary SQL query.

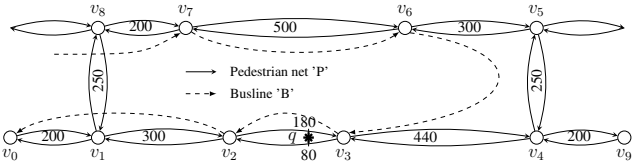
3 Scientific Background

3.1 Isochrones in Multimodal Networks

A *multimodal network* is defined as a seven-tuple $\mathsf{N} = (G, R, S, \rho, \mu, \lambda, \tau)$. G is a directed multigraph with a set V of vertices and a multiset E of edges. Vertices represent crossroads of the street network, stops of the public transport system, etc. Edges represent street segments, transport routes, moving walkways, etc. R is a set of transport systems, such as the pedestrian network or the public transport system (buses, trains, etc.). Function μ assigns to each transport system a transport mode, e.g., continuous space and time mode ('*csct*') for the pedestrian network or discrete space and time mode ('*dsdt*') for the public transport system. The functions ρ and λ assign to each edge transport system and edge length, respectively. Finally, function $\tau(e, t)$ computes the time-dependent transfer time that is required to traverse edge $e = (u, v)$ when starting at u as late as possible yet arriving at v no later than time t.

[1] www.pgrouting.org
[2] www.mapnificent.net
[3] http://mapumental.com/
[4] www.hacon.de
[5] www.opentripplanner.org

Figure 2 shows a multimodal network with two transportation systems, $R = \{\text{'P'}, \text{'B'}\}$, representing the pedestrian network with mode $\mu(\text{'P'}) = \text{'}csct\text{'}$ and bus line B with mode $\mu(\text{'B'}) = \text{'}dsdt\text{'}$, respectively. Solid lines are street segments of the pedestrian network, e.g., edge $e = (v_1, v_2)$ with $\rho(e) = \text{'P'}$. Pedestrian edges are annotated with the edge length, which is the same in both directions, e.g., $\lambda((v_1, v_2)) = \lambda((v_2, v_1)) = 300$. We assume a constant walking speed of 2 m/s, yielding a transfer time $\tau(e, t) = \frac{\lambda(e)}{2\,\text{m/s}}$ for a pedestrian edge e. Dashed lines represent bus line B. An excerpt of the schedule is shown in Figure 2(b). The transfer time of a bus edge $e = (u, v)$ is computed as $\tau(e, t) = t - t'$, where t' is the latest departure time at u in order to reach v before or at time t. This might include a waiting time at v.



(a) Network

TID	Stop	Arrival	Departure
1	v_7	05:31:30	05:32:00
1	v_6	05:33:00	05:33:00
1	v_6	05:35:00	05:35:00
⋮	⋮	⋮	⋮

(b) Schedule

Fig. 2. Multimodal Network

A *location* in N is any point on an edge $e = (u, v) \in E$ that is accessible. We represent it as $l = (e, o)$, where $0 \leq o \leq \lambda(e)$ is an offset that determines the relative position of l from u on e, e.g., the location of q in Figure 2 is $l_q = ((v_2, v_3), 180) = ((v_3, v_2), 80)$. In continuous space networks all points on the edges are accessible. In discrete space networks only vertices are accessible.

The *network distance*, $d_N(l_s, l_d, t)$, from a source location l_s to a destination location l_d with arrival time t at l_d is defined as the minimum cost of any path from l_s to l_d with arrival time t at l_d if such a path exists, and ∞ otherwise. The network distance is time-dependent. For instance, $d_N(v_6, v_3, \text{05:33:00}) = 120\,\text{s}$ because the bus with trip id 1 departs from v_6 at 05:33:00 and arrives at v_3 at 05:35:00. In contrast, $d_N(v_6, v_3, \text{05:34:00}) = 495\,\text{s}$ since the shortest path passes through the pedestrian network, traversing the edges (v_6, v_5), (v_5, v_4), and (v_4, v_3).

An *isochrone*, $N^{iso} = (V^{iso}, E^{iso})$, is defined as the minimal and possibly disconnected subgraph of G that covers exactly those locations that have a network distance to q smaller or equal than a user-defined timespan d_{max}. In Figure 3, the boldface edges (edge segments) represent an isochrone, which is formally represented by the vertices $V^{iso} = \{v_3, v_2, v_3, v_6, v_1, v_7, v_4\}$ and the edges (edge segments) $E^{iso} = \{((v_0, v_1), 80, 200), ((v_8, v_1), 130, 250), ((v_2, v_1), 180, 300), ((v_1, v_2), 0, 300), ((v_3, v_2), 0, 260), \ldots\}$.

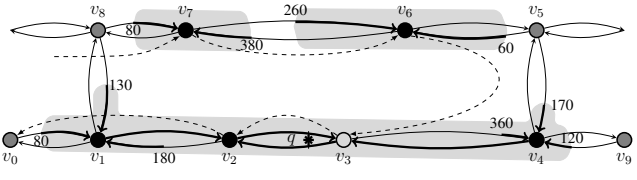


Fig. 3. Isochrone with d_{max} = 5 min, s = 2 m/s, and t = 06:06:00

For the computation of isochrones, we use the MINEX algorithm [5], which adopts an incremental network expansion strategy in combination with vertex expiration. The expansion starts from q and propagates backwards along the incoming edges in all directions. The multimodal network is stored in the database, and – as the expansion proceeds – the portions of the network that eventually will form the isochrone are incrementally retrieved. By loading only those network portions that eventually will be part of the isochrone and by keeping in memory only a minimal set of vertices to avoid cyclic network expansion, MINEX is independent of the network size and the memory requirements are only a small fraction of the isochrone size.

3.2 Creating the Surface of an Isochrone Subgraph

In order to determine objects that are located within isochrones, we have to construct an area around an isochrone. Marciuska and Gamper [7] propose two algorithms for this. The surface-based approach (SBA) computes a minimum bounding polygon around the outermost edges in the isochrone. However, it is limited to isochrones in pedestrian networks, where an isochrone consists of a single subgraph.

Algorithm SBA^* in Figure 4 extends SBA by allowing isochrones that are composed of disconnected subgraphs, which is typically the case if public transport systems are present in addition to the pedestrian network. The input parameters are a set of edges E^{iso} that represents an isochrone and a parameter *size* that represents the margin of the buffer that is created around the outermost edges. SBA^* iterates over set E^{iso} and computes in each iteration an area around a single subgraph of the isochrone. The algorithm identifies first the leftmost edge e in E^{iso} and calls then function DFS, which in a recursive way traverses the outermost edges of the subgraph that contains e. Parameter P returns the ordered list of points that represent the geometry of the

Algorithm: $SBA^*(E^{iso}, size)$

while $E^{iso} \neq \emptyset$ **do**
- $P \leftarrow \emptyset$;
- $e = (u, v) \leftarrow$ leftmost edge in E^{iso};
- $DFS(e, e, E^{iso}, P, 0)$;
- $g \leftarrow ST\_Buffer(ST\_MakePolygon(P), size)$;
- $E^{iso} \leftarrow E^{iso} \setminus ST\_Within(E^{iso}.geo, g)$;
- Output $ST\_Multi(g)$;

Function: $DFS(root, e, E^{iso}, P, l)$

if $e = root \wedge l > 0$ **then**
- $P \leftarrow P \cup geom(e)^{-1}$;
- **return** $true$;

else
- $(u, v) \leftarrow e$;
- **if** $l = 0$ **then**
  - $P \leftarrow P \cup geom(e)$;
  - **foreach** $(u', v) \in E^{iso}$ *sorted by angle* $\alpha((u, v), (u', v))$ **do**
    - **if** $DFS(root, (u, v), E^{iso}, P, l + 1)$ **then return** $true$;
    - **else** $P \leftarrow P \cup geom(e)$;
  - **return** $false$;
- **else**
  - **if** $o(e) + o(e^{-1}) \leq \lambda(e)$ **then**
    - $P \leftarrow P \cup geom(e)^{-1} \cup geom(e^{-1}))$;
    - **foreach** $(t, u) \in E^{iso}$ *sorted by angle* $\alpha((u, v), (t, u))$ **do**
      - **if** $DFS(root, (t, u), E^{iso}, P, l + 1)$ **then return** $true$;
      - **else** $P \leftarrow P \cup geom(e)$;
    - **else**
      - $P \leftarrow P \cup geom(e)^{-1}$;
    - **return** $false$;

Fig. 4. Algorithm SBA^*

outermost edges. P is transformed into a polygon, and a buffer with a margin of size $size$ is created around the polygon. Finally, all edges that are within the isochrone area are removed from E^{iso}.

Function DFS recursively computes the outermost edges of an isolated subgraph and collects the geometry of these edges in an ordered set of points P. DFS has in input the leftmost edge $root$, the currently visited edge e, the isochrone edges E^{iso}, the ordered set of points P, and the recursion level l. DFS examines in a depth-first traversal the subgraph until it returns to the root edge (i.e., the left-most edge) or no other edges are found. If $e = root$ and $l > 0$, the traversal returned to the root edge. The geometry of the current edge is added to P (after converting the order of the points) and the recursion terminates. Otherwise, if the recursion level l is equal to 0, the geometry of the current edge $e = (u, v)$ is added to P. Then, all incoming edges to v, ordered by the angle to e, are considered by recursively calling DFS with a recursion level that is incremented by one. If DFS returns true, the recursion is stopped. Otherwise, the geometry of e is added to P, and the next incoming edge is considered. If $l > 0$, the edges are processed in a similar way. The only difference is that we have to consider the case that an edge might only be partially reachable; $o(e)$ is the offset of the reachable segment from the source vertex of e and $o(e^{-1})$ is the offset of the reachable edge segment from the target vertex of e. If $o(e) + o(e^{-1}) \leq \lambda(e)$, the two segments cover the entire edge, hence the recursive traversal of the edges continues. If this is not the case, only the segment of the current edge is added and the recursion stops.

We illustrate algorithm SBA^* using the isochrone in Figure 3. The first leftmost edge that is identified is (v_0, v_1). It is passed to DFS together with E^{iso}, an empty set P, and recursion level 0. DFS adds the geometry of the edge to P and iterates then through all incoming edges to v_1, sorted by their relative (counterclockwise) angle with respect to (v_0, v_1), i.e., (v_2, v_1), $(v_8, v1)$, and $(v_0, v1)$ in that order. Next, DFS is called with $e = (v_2, v_1)$, the current set P, and recursion level 1. The recursive traversal continues until edge (v_9, v_4) is encountered, which is only partially reachable. DFS returns by one recursion level and examines edge (v_5, v_4), etc. Finally, the root edge is visited again, and DFS returns to SBA^*. The points in P are transformed into a polygon, around which a buffer with a margin of size $size$ is created (light-gray area in Figure 3). After removing all edges from E^{iso} that are enclosed in the new isochrone area, the next leftmost edge (v_8, v_7) is determined, followed by a call to DFS, etc.

4 Architecture

Figure 5 shows the three-tier architecture of the ISOGA system.

Presentation Tier. The presentation tier is a WebGIS client, implemented in JSP and JavaScript. It uses *Comet*[6] as web application model for asynchronous data sending and for managing long polling requests, *Openlayers*[7] as web mapping framework, and *GeoExt*[8] as framework for building interactive web applications. The main tasks of the client is the interaction with the map, the input of the query parameters, and the

[6] www.cometd.org
[7] www.openlayers.org
[8] www.geoext.org

visualization of the results. The client communicates with the server over the HTTP protocol using the following standardized OGC[9] services: Web Map Service (*WMS*) for serving geo-referenced map images and Web Feature Service (*WFS*) for requesting geographical features. The client submits asynchronously three different types of HTTP requests. An *isochrone request* ① invokes MINEX for the computation of an isochrone. A *map request* ② retrieves the isochrone in form of a binary image format and includes it as a separate layer in the map. Similar, the base layer images (e.g., the street network) come from different sources (Google, OpenStreetMap) and are fetched via WMS. The request is triggered during the initialization of the map, when MINEX and SBA^* are terminated, or whenever there is an interaction with the map (zoom, pan, identify). A *feature request* ③ retrieves detailed information about a selected feature in the map (e.g., the pop up in Figure 1). The request is triggered by clicking an object on the map.

Logical Tier. The logical tier (server) accepts requests via a Java Servlet. An isochrone request invokes MINEX to compute an isochrone that is represented as a logical network. This representation is passed to the *GeoBuilder* module, which performs two tasks. First, the isochrone is annotated with geometry information and stored in vector format in a spatial relation in the DB. Second, the network representation of the isochrone is transformed into a spatial area (polygon). In order to enable the client to access the isochrone via WMS and WFS, the three tables (vertices, edges, and areas) are registered as vector layers in the map builder module. As Map Builder we use the rendering engine *Geoserver*[10], which is accessible via standardized OGC services. For a WMS request, Geoserver reads the spatial data from the DB and creates an image that is sent back to the client. For a WFS request, the information is retrieved from the DB and sent to the client as a feature in text format. The Map Builder serves also as proxy for providing base layers from external servers (e.g., Google, OpenStreetMap, Bing, etc.). The second main task of the logical tier is to support geospatial reachability analysis. Once an isochrone is computed, the user can specify an arbitrary SQL query that is sent to the server via a *geoAnalysis request* ④. The result of this SQL query is a table of geo-referenced objects that are joined with the isochrone, e.g., the apartments in Figure 1. The final result is converted from a relational format into a JSON object and sent back to the client.

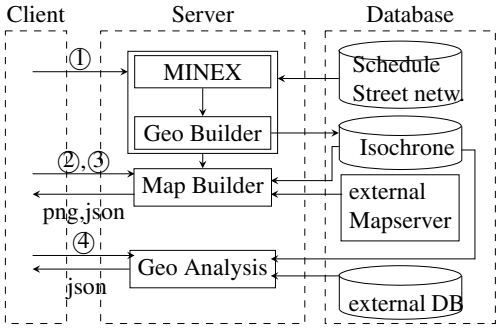


Fig. 5. Architecture

Data Tier. The data tier uses a relational DBMS with a spatial extension to perform spatial operations, such as edge clipping if an edge is only partially reached, locating the query point to the closest edge, area buffering, or spatial intersection. The ISOGA

[9] `http://www.opengeospatial.org/` [10] `http://www.geoserver.org`

system works with PostGIS 2.0 as well with Oracle11g. Only standardized spatial operators (OGC/SQL-MM) are used, which simplifies the migration to other spatial DBMSs.

5 Application Scenarios

In this section we describe two application scenarios that emerged from a collaboration with the local municipality and one example that shows network expiration to illustrate the low memory requirements of MINEX.

Scenario 1. We want to determine how well the primary schools in Bozen-Bolzano are reachable by walking and taking the public transport system (Figure 6). For this we compute an isochrone with multiple query points that represent the schools, a maximal duration of 15 minutes, and an arrival time 9 am at the schools. Next, we specify an SQL query that retrieves from the inhabitants database all kids with an age between 6 and 11 years. The result of the SQL query is joined with the isochrone. The statistics shows the number and percentage of kids that reach the closest school in less than 15 minutes. By selecting the option *Outside Isochrone* for the join, it is possible to identify the number of kids who do not reach the school within the given time constraints.

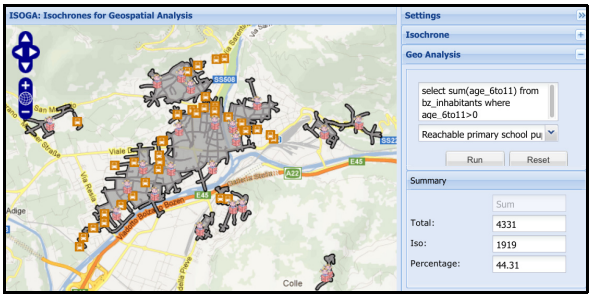


Fig. 6. Reachability of Primary Schools

Scenario 2. We want to determine cheap apartments that are close to the working place (Figure 7). The user specifies a query point on the map that represents his/her working place, a maximal acceptable traveling duration, and a price range he/she is willing to pay. After computing the isochrone, an SQL query retrieves all flats in the specified price range. The result is joined with the isochrone and all available flats that are located in the isochrone are visualized as circles on the map. The example shows additional information for one such apartment, namely that the working place can be reached in 25 min by walking first for 4 min to the closest bus station, where to take bus line 5 at 08:39 am.

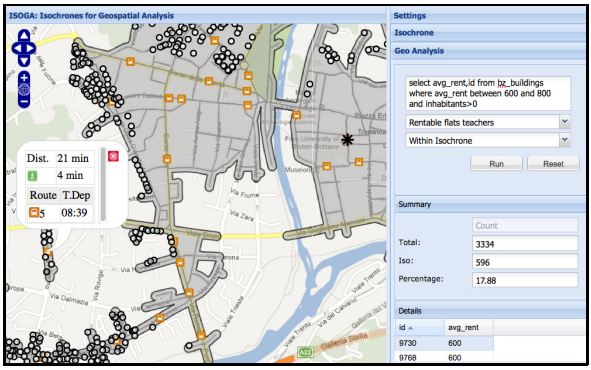


Fig. 7. Flat Search Scenario

Scenario 3. In this scenario we illustrate the low memory consumption of MINEX due to network expiration (Figure 8). The user can select different datasets. Currently, the cities of Bolzano-Bozen, Washington DC, and San Francisco as well as the regional networks of South Tyrol and Italy are available, all having different network topologies and transportation systems. After computing an isochrone, the user can open the Layers panel and activate the visualization of vertex expiration. Black circles represent the vertices that are kept in memory to avoid cyclic expansions. White circles represent expired vertices that are removed from memory to minimize the memory requirements.

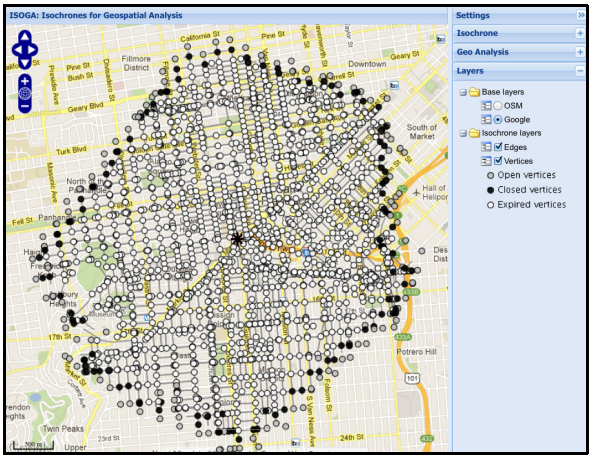


Fig. 8. Vertex Expiration in MINEX

6 Conclusion and Future Work

In this paper, we presented the web-based system ISOGA that uses isochrones to perform geospatial reachability analysis. Core features of the system are the support of multimodal networks and the possibility to join isochrones with the result of general SQL queries, which allows to analyze the reachability of various types of georeferenced objects without the need to compute the distance of each object. The system adopts a service-oriented, three-tier architecture and uses technologies that are compliant with OGC standards. Future work includes the further development of the analysis component as well as the implementation of ISOGA for a mobile client.

References

1. Bauer, V., Gamper, J., Loperfido, R., Profanter, S., Putzer, S., Timko, I.: Computing isochrones in multi-modal, schedule-based transport networks. In: GIS, pp. 1–2. ACM (2008)
2. Deng, K., Zhou, X., Shen, H.T., Sadiq, S.W., Li, X.: Instance optimal query processing in spatial networks. VLDB J. 18(3), 675–693 (2009)
3. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1(1), 269–271 (1959)
4. Gamper, J., Böhlen, M.H., Cometti, W., Innerebner, M.: Defining isochrones in multimodal spatial networks. In: CIKM, pp. 2381–2384 (2011)
5. Gamper, J., Böhlen, M., Innerebner, M.: Scalable Computation of Isochrones with Network Expiration. In: Ailamaki, A., Bowers, S. (eds.) SSDBM 2012. LNCS, vol. 7338, pp. 526–543. Springer, Heidelberg (2012)

6. Innerebner, M., Böhlen, M.H., Timko, I.: A web-enabled extension of a spatio-temporal dbms. In: GIS, pp. 34–41. ACM (2007)
7. Marciuska, S., Gamper, J.: Determining Objects within Isochrones in Spatial Network Databases. In: Catania, B., Ivanović, M., Thalheim, B. (eds.) ADBIS 2010. LNCS, vol. 6295, pp. 392–405. Springer, Heidelberg (2010)
8. Papadias, D., Zhang, J., Mamoulis, N., Tao, Y.: Query processing in spatial network databases. In: VLDB, pp. 802–813 (2003)

A High Performance Web-Based System for Analyzing and Visualizing Spatiotemporal Data for Climate Studies

Zhenlong Li, Chaowei Yang, Min Sun, Jing Li, Chen Xu,
Qunying Huang, and Kai Liu

Center of Intelligent Spatial Computing for Water/Energy Sciences, George Mason University, Fairfax, VA, 22030-4444
{zli1,cyang3,msun,jlih,cxu3,qhuang1,kliu4}@gmu.edu

Abstract. Large amount of data are produced at different spatiotemporal scales by many sensors observing Earth and model simulations. Although advancements of contemporary technologies provide better solutions to access the spatiotemporal data, it is still a big challenge for researchers to easily extract information and knowledge from the data due to the data complexities of high dimensions, heterogeneity, distribution, large amount and frequently updating. This is especially true in climate studies, because climate data with coverage of the entire Earth and a long time period (such as 200 years) are often required to extract useful climate change information and patterns. A well-developed online visual analytical system has the potential to provide an efficient mechanism to bridge this gap. Using performance improving techniques for an online visual analytical system, we researched and developed a high performance Web-based system for spatiotemporal data visual analytics includes the following components: 1) a Spatial Data Registration Center for managing the big spatiotemporal data and enabling researchers to focus on analyses without worrying about data related issues such as format, management and storage; 2) a workflow for pre-generating and caching frequently requested data to reduce the server response time; and 3) a technique of "single data fetch, multiple analyses" to reduce both server response time and client response time; Finally, we demonstrate the effectiveness of the prototype through a few use cases.

Keywords: Big Data, CyberGIS, Online Visual Analytics, Computing Optimization.

1 Introduction

Over the past decades, the advancements of sensor technologies have greatly improved our capabilities to record the spatiotemporal snapshots of a variety of natural or social phenomena as data. The spatiotemporal data collected by these sensors are characterized by high dimensions, heterogeneity, distribution and large amount (Yang et al. 2010). In climate studies, scientists use the spatiotemporal data captured by various sensors to validate and improve climate models. Meanwhile, large amounts of (or big) spatiotemporal data are generated by climate model runs. For example,

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 190–198, 2013.
© Springer-Verlag Berlin Heidelberg 2013

ModelE (Schmidt et al., 2006), one of the climate models, can simulate the climate of Earth system in many scenarios, and will produce approximately 25 gigabytes data with a single run for a 30-year monthly simulation. Generally, climate studies require hundreds, thousands, to millions of model runs with simulations of 200 years, and will generate petabytes of spatiotemporal data. It is an urgent scientific demand to effectively manage and mine the big spatiotemporal data to support scientific research.

With the advancements of the Internet and distributed computing technologies, many visual analytic applications are migrated from standalone computing environment to Web-based computing environment. There are several initiatives of online visual analytic system in climate domain. For example, the TRMM (Tropical Rainfall Measuring Mission) online Visualization and Analysis System (TOVAS) is developed by NASA (Berrick et al. 2009) to analyze TRMM gridded rainfall data. Fetch Climate application (http://fetchclimate.cloudapp.net/) is developed by Microsoft to access mean values of different climate parameters for selected geographic regions of the Earth surface.

Online visual analytical systems make it more convenient for end users in that they do not have to install the entire analytical software package on local computers, nor to download and prepare their own data locally. However, performance becomes an essential issue when developing online visual analytic systems (Yang et al. 2011). An acceptable waiting time for a Web system response is approximately 3-8 seconds (Corner 2010). In a simple Web application, this second rule is easy to comply. However, when manipulating big spatiotemporal data, system performance should be considered systematically similar to the WebGIS performance (Yang et al. 2005).

This paper addresses Web-based system performance in two aspects: 1) how fast the system is? This measures the system response time between sending the requests to receiving the results by end users, 2) how interactive and intuitive the system is? This is hard to quantify but can be evaluated by the user interface complexity (Coskun et al. 2005), which could be reduced by minimizing the number of user clicks and reducing the number of Web pages involved in an operation (Galitz 2007). Focusing on these two performance aspects, we developed a high performance online visual analytical system.

2 Related Work

Online visual analytics of big spatiotemporal data provides a potential solution and is becoming increasingly important (Liu et al. 2007). For example, Sun et al (2012) developed a Web-based visualization platform for climate studies based on Google Earth. Their approach provides a state of the art visualization style, but the installation of Google Earth plug-in significantly reduces the portability and availability. Another limitation for this approach is that it does not support user-defined areas (or Area of Interest, AOI), which poses a major issue when mining knowledge and patterns at a local or regional scale. Liu et al (2009) developed an online analysis system focusing on global satellite precipitation algorithm validation and inter-comparison. While

these initiatives provide good examples for developing online visual analytical applications in climate domain, the critical system performance challenge is barely mentioned considering massive concurrent requests, complex processing, and large amounts of datasets (Yang et al. 2011).

Metadata are descriptive data about the data and have been widely used to manage large amounts of data in data-intensive applications (Yee et al. 2003, Singh et al. 2003, Weibel et al. 1998). A spatiotemporal visual analytical system is a typical data-intensive scientific application for managing and analyzing Terabytes to a Petabyte of data that may span millions of files or data objects. Singh et al (2003) proposed a Metadata catalog service to support general data-intensive applications. We adopt this approach in the SDRC by extending the spatiotemporal support to improve data discovery performance.

Caching data on both the server side and the client side can reduce the load on network transmission and server processing (Rowstron et al. 2001, Li et al. 2011, Yang et al. 2005) and improve the overall system performance. There are two types of caching: 1) caching data based on user requests on-the-fly with dynamic cache architecture and 2) loading at start-up and caching preprocessed data with static cache architecture (Yang et al. 2005). Leveraging the static cache architecture, we propose a workflow to pre-generate and cache frequently requested data to reduce the server response time.

By adopting the approaches discussed above, we developed a high performance Web-based spatiotemporal visual analytical system for climate studies. The system performance is optimized from various aspects, including server side data storage, management and preprocessing, client side data presentation, client/server data transmitting and user interactive interface. This paper elaborates the details of the system: section 3 introduces the system overall architecture; section 4 explains the methods and techniques used in the system; section 5 conducts three experiments to evaluate the performance of the proposed methods and techniques; section 6 demonstrates the system with two usage scenarios to show how it can facilitate climate studies in practice; section 7 concludes and discusses future research.

3 System Architecture

Our high performance online visual analytical system includes the spatiotemporal data registration center (SDRC), intermediate result pre-generator, server side advanced analyzer and client side analyzer (Fig.1):

SDRC is the core component for server side data management. Raw data and pre-generated data from Intermediate Result Pre-generator are registered into SDRC. Metadata is extracted and recorded in the Metadata Database dynamically whenever a new dataset is registered. Metadata and spatiotemporal index techniques are adopted in this component.

The Intermediate Result Pre-generator is responsible for pre-generating intermediate result for frequently accessed data to avoid repeated calculations and reduce server response time. Caching and data pyramid techniques (Yang et al. 2005) are adopted in this component.

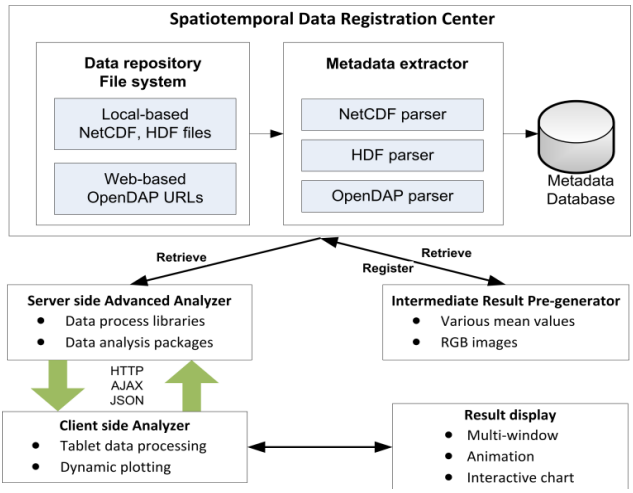


Fig. 1. System Architecture

Server Side Advanced Analyzer is a function-driven component including third party libraries and packages: NetCDF-Java (http://www.unidata.ucar.edu) java-based library and R package (http://www.r-project.org/).Client Side Analyzer is responsible for 1) providing an effective mechanism "single data fetch, multiple analyses" to reduce server response time and user interactive time, 2) processing and plotting the returned table data.

4 Components and Methods

This session elaborates the system components and methods from three aspects: server side data management, client side result presentation and server/client data communication.

4.1 Spatiotemporal Data Registration Center

The major goal of SDRC is to manage all the registered spatiotemporal data in a uniformed manner by dynamically extracting the metadata of the spatiotemporal data into a Metadata Database. Data repository maintains all the registered data in the file system, data may either be manually input to a directory, or in most cases, be uploaded from distributed model runs automatically. Metadata extractor is activated whenever a new dataset is uploaded to the data repository. The plug-and-play parsers of NetCDF (Network Common Data), HDF (Hierarchical Data Format) and OpenDAP (Open-source Project for a Network Data Access Protocol) interpret the data when adding, extract metadata and insert to the Metadata Database. To enhance the security and interoperability, a standard Web service is provided when accessing metadata outside of SDRC (Erl 2004).

The SDRC is flexible, extensible and interoperable in that 1) the metadata is automatically extracted to the database whenever new data are registered, 2) the structure

of Metadata Extractor could support any kinds of data formats by adding corresponding plug-and-play parsers, and 3) the standard Web service based on REST (Representational State Transfer) (Costello 2007) serves a secure method for accessing the metadata though the Web in an interoperable manner.

4.2 Intermediate Result Pre-generator

In climate studies, most analyses are based on averaged (mean) values of monthly, quarterly or yearly data spanning a long time period. These values are either calculated for every data point or for a point-of-interest region with averaged values of all the data points. Since the mean values are the basic inputs for various advanced analyses, it is necessary to do the calculation only one time and store them in a proper manner for further access to avoid repeating calculations. Visualizing the spatiotemporal data as a two dimensional image is an effective way for analysts or general users to visually discover spatial patterns (Keim 2000). Because these images are frequently requested by the users, pre-generating them could dramatically reduce the server response time, especially when animating for a long time period.

Therefore, we proposed a workflow used in Intermediate Result Pre-generator component (Fig.2) to 1) pre-calculate the intermediate mean values, 2) pre-render frequently accessed images, 3) effectively store and manage these pre-generated data for fast access. Three types of intermediate data are pre-generated: temporal-based means stored in NetCDF files, spatial-based means stored in CSV files, and RGB images rendered from both original data and the newly created mean data (NetCDF). Finally, all these newly generated data are registered in SDRC.

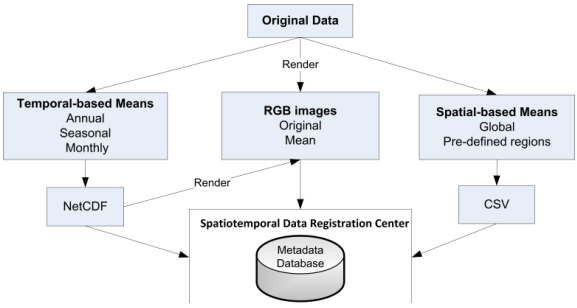


Fig. 2. Workflow for pre-generating frequently requested data

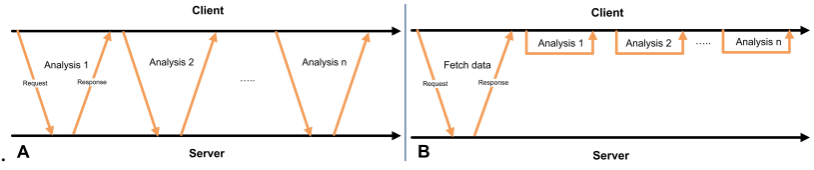


Fig. 3. (A) Traditional online analysis workflow; (B) Proposed online analysis workflow

4.3 Single Data Fetch, Multiple Analyses

A typical spatiotemporal analyses request includes five parameters: dataset(s), variable(s), time period, spatial coverage and analysis type. The analysis type includes time series, correlation, animation, anomaly and outlier to name a few. For the traditional approach (Fig.3A), a request containing the five parameters is built for each analysis. This approach is not efficient or interactive because 1) for each analysis, server side dataset (s) need to be read at least one time, which is time consuming, 2) only statistic images are returned to the client, any modification to the result, for example, adjust a parameter or to change a plot style, needs a new request-wait-response workflow.

To overcome these drawbacks, we use a "single data fetch, multiple analyses" (Fig.3B) method. This method can significantly improve the overall system performance by 1) reducing the number of request-wait-response loops by carefully considering both the request parameters and the response data structure; 2) shifting partial of the visual analytical processes from the server side to the client side. Instead of returning static images, the processed data are returned to the client side for further analysis and visualization.

5 Performance Comparison

Three experiments were conducted to evaluate the performance of the proposed components and methods.

5.1 Performance of SDRC

The data used in this experiment are generated from ModelE runs with 10 years' monthly data from 1951 to 1960. Nine NetCDF datasets with total data volume of 80 Gigabytes are generated. 566 climatic variables are included in these datasets. Figure 4A shows the server response time with and without SDRC when extracting the variables, time period and spatial coverage from the data in a Web-based environment. When using SDRC, variables, time period and spatial coverage are fetched directly from the metadata database instead of being extracted from the nine NetCDF files, which dramatically reduces the server response time.

5.2 Performance of Intermediate Result Pre-generator

This experiment requested the 1 year mean data, 5 years mean data, global mean of 5 years mean data and RGB image of 5 year mean data with or without using pre-generating/caching method. Data used is the same as the previous experiment as described in 5.1. As presented in figure 4B, rendering "1 year mean images" is the most time consuming process. Since "1 year mean" need to be calculated before rendering "1 year mean images", most time is consumed by calculating "1 year mean" (more than 20 seconds). Time consumed by "5 years mean" and "5 years global mean" is relatively shorter, but we can still see a significant performance improvement after using pre-generating/caching method.

5.3 Performance of "Single Data Fetch, Multiple Analyses"

In this experiment, we use three variables and two AOIs from one data to 1) conduct time series analysis for each variable in each study area, 2) for each study area, conduct correlation analysis for any 2 of the 3 variables. We recorded the time consumed on the server side and the number of request-wait-response loops for these analyses when using and without using "single data fetch, multiple analyses" method. The result (Tab.2) shows that "single data fetch, multiple analyses" could dramatically reduce the total server response time by reducing the number of workflow loops.

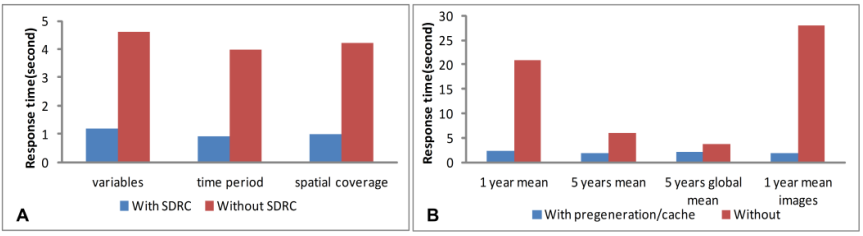


Fig. 4. (A) Server response time with and without SDRC; (B) Server response time with and without pre-generation/cache

Table 1. Request/response loops and consumed time with/without "single data fetch, multiple analyses" method

Analysis type		With	Without
Time series	Request/response loops	1	6
	Total time (second)	4.3	12.6
Correlation	Request/response loops	1	21
	Total time (second)	4.2	41.7

6 System Demonstration

For time series analyses, users can select multiple model simulations, multiple variables and multiple AOIs at the same time, and plot them together for better comparison (Fig.5A).

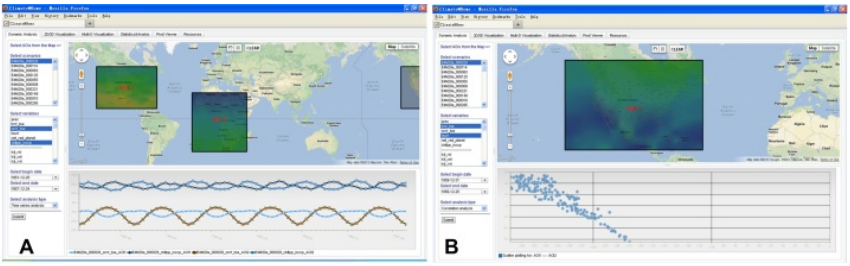


Fig. 5. (A)Time series plotting for two variables and two AOIs ; (B) Correlation analyses for two variables

Correlation analyses enable users to plot the relationships of any two variables or two AOIs. Figure 5B shows the scatter plot for two variables from one simulation at the same area of interest.

7 Discussion and Conclusion

This paper developed a high performance web-based system to analyze and visualize spatiotemporal data to support climate studies by focusing on the performance improvement methods and techniques. The system performance is optimized by leveraging performance improvement techniques of server side data management and data preprocessing, client side result presentation and user interactive interface, and server/client communication.

Three experiments were conducted using ModelE data to evaluate the performance of the proposed methods and techniques. The results show that these methods and techniques could dramatically improve the system performance by reducing the system response time and user interactive time. A prototype is developed based on the proposed methods and techniques. The prototype is served as an initial approach to handle big spatiotemporal data in climate domain.

Further studies are required to continue improving the system through, but not limited, to the following four aspects: 1) improving SDRC by leveraging the techniques used in big data management (Herodotou et al.2011, Bughin et al.2010); 2) enhancing the spatiotemporal data storage and process strategies by leveraging spatial cloud computing (Yang et al. 2011); 3) optimizing the system architecture for collecting, archiving, sharing, analyzing and visualizing spatiotemporal data by adopting geospatial cyberinfrastructures technologies to support various scientific domains (Zhang et al. 2009, Yang et al. 2010); 4) providing more innovative result display methods based on the characteristics of spatiotemporal data and scientific requirements.

Acknowledgements. Research reported is supported by Microsoft Research, NSF (IIP-1160979), and NASA (NNX12AF89G).

References

1. Bailey, B.P., Konstan, J.A., Carlis, J.V.: The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In: Proceedings of INTERACT, vol. 1, pp. 593–601 (2001)
2. Berrick, S., Leptoukh, G., Liu, Z., Pham, L., Rui, H., Shen, S., Teng, W., Zhu, T.: Multisensor distributive on-line processing, visualization and analysis system. In: Proceedings of the 2004 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2004, vol. 3, pp. 2030–2033 (2004)
3. Bughin, J., Chui, M., Manyika, J.: Clouds, big data, and smart assets: Ten tech-enabled business trends to watch. McKinsey Quarterly 56 (2010)
4. Corner, S.: The 8-second rule, `http://www.submitcorner.com/Guide/Bandwidth/001.shtml`
5. Coskun, E., Grabowski, M.: Impacts of User Interface Complexity on User Acceptance and Performance in Safety-Critical Systems. Journal of Homeland Security and Emergency Management 2(1) (2005)

6. Erl, T.: Service-oriented architecture: a field guide to integrating XML and web services. Prentice Hall PTR (2004)
7. Hendler, J.: Web 3.0 Emerging. Computer 42(1), 111–113 (2009)
8. Herodotou, H., Lim, H., Luo, G., Borisov, N., Dong, L., Cetin, F.B., Babu, S.: Starfish: A self-tuning system for big data analytics. In: Proc. of the Fifth CIDR Conf. (2011)
9. Keim, D.A.: Designing pixel-oriented visualization techniques: Theory and applications. IEEE Transactions on Visualization and Computer Graphics 6(1), 59–78 (2000)
10. Li, Z., Yang, C.P., Wu, H., Li, W., Miao, L.: An optimized framework for seamlessly integrating OGC Web Services to support geospatial sciences. International Journal of Geographical Information Science 25(4), 595–613 (2011)
11. Liu, Z., Rui, H., Teng, W., Chiu, L., Leptoukh, G., Kempler, S.: Developing an online information system prototype for global satellite precipitation algorithm validation and intercomparison. Journal of Applied Meteorology and Climatology 48(12), 2581–2589 (2009)
12. Liu, Z., Rui, H., Teng, W., Chiu, L., Leptoukh, G., Vicente, G.: Online visualization and analysis: A new avenue to use satellite data for weather, climate, and interdisciplinary research and applications. In: Measuring Precipitation From Space, pp. 549–558 (2007)
13. Rowstron, A., Druschel, P.: Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. ACM SIGOPS Operating Systems Review 35(5), 188–201 (2001)
14. Schmidt, G.A., Ruedy, R., Hansen, J.E., Aleinov, I., Bell, N., Bauer, M., Yao, M.S.: Present-day atmospheric simulations using GISS ModelE: Comparison to in situ, satellite, and reanalysis data. Journal of Climate 19(2), 153–192 (2006)
15. Singh, G., Shishir, B., Ann, C., Ewa, D., Carl, K., Mary, M., Sonal, P., Laura, P.: A metadata catalog service for data intensive applications. In: 2003 ACM/IEEE Conference on Supercomputing, p. 33. IEEE (2003)
16. Sun, X., Shen, S., Leptoukh, G.G., Wang, P., Di, L., Lu, M.: Development of a Web-based visualization platform for climate research using Google Earth. Computers & Geosciences 47, 160–168 (2012)
17. Galitz, W.O.: The essential guide to user interface design: an introduction to GUI design principles and techniques. Wiley (2007)
18. Taylor, K.E.: Summarizing multiple aspects of model performance in single diagram. Journal of Geophysical Research 106(7), 7183–7192 (2001)
19. Weibel, S., Kunze, J., Lagoze, C., Wolf, M.: Dublin core metadata for resource discovery. In: Internet Engineering Task Force RFC 2413, vol. 222 (1998)
20. Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., Bambacs, M., Fay, D.: Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing? International Journal of Digital Earth 4(4), 305–329 (2011)
21. Yang, C., Wu, H., Huang, Q., Li, Z., Li, J.: Using spatial principles to optimize distributed computing for enabling the physical science discoveries. Proceedings of the National Academy of Sciences 108(14), 5498–5503 (2011)
22. Yang, C., Wu, H., Huang, Q., Li, Z., Li, J., Li, W., Sun, M., Miao, L.: WebGIS performance issues and solutions. In: Advances in Web-Based GIS, Mapping Services and Applications. Taylor & Francis Group, London (2011) ISBN 978-0
23. Yang, C., Raskin, R., Goodchild, M., Gahegan, M.: Geospatial cyberinfrastructure: past, present and future. Computers, Environment and Urban Systems 34(4), 264–277 (2010)
24. Zhang, T., Tsou, M.H.: Developing a grid-enabled spatial Web portal for Internet GIServices and geospatial cyberinfrastructure. International Journal of Geographical Information Science 23(5), 605–630 (2009)

Personalized Accessibility Maps (PAMs) for Communities with Special Needs

Hassan A. Karimi, Lei Zhang, and Jessica G. Benner

Geoinformatics Laboratory, School of Information Sciences,
University of Pittsburgh, Pittsburgh PA, USA
{hkarimi,lez13,jgb14}@pitt.edu

Abstract. Accessibility data/information is necessary to support the everyday mobility of people with special needs. As a means to accommodate mobility of students with special needs, universities and colleges provide maps with accessibility data/information for their campus, where some are static and others are interactive. In this paper, we describe the concept of Personalized Accessibility Maps (PAMs) and discuss the development of a PAM for the University of Pittsburgh's main campus as a representative PAM. As a result of this development, there is a better understanding of the technologies and techniques needed for PAMs along with challenges and future research directions.

Keywords: accessibility map, social navigation network, special needs, pedestrian navigation service.

1 Introduction

Mobility is a common and routine activity performed by all people. As people travel to unfamiliar locations, they commonly rely on online services (e.g., Google Maps, Bing Maps) and/or navigation systems/services (e.g., Garmin, TomTom) to find their way in new locations. For these systems and services, and the maps produced by them, to address mobility of people with special needs (e.g., people with mobility and visual impairments), they need to contain accessibility data/information. However, containing only accessibility data/information, while necessary, is insufficient to address the range of mobility challenges faced by special needs communities. For systems and services to be of value to special needs communities, they must: (a) contain useful accessibility data/information; (b) utilize accessibility data/information in meaningful ways (such as for trip planning and real-time navigation) that address specific special needs; and (c) present accessibility data/information, or their utilization, through easy-to-access and easy-to-use interfaces.

Accessibility is essential for the quality of life of people with disabilities [12]. Aspects of the built environment, public transport, and levels of shopper activity present a range of difficulties restricting the physical mobility of wheelchair users [2]. Provisions for accessible infrastructure are regulated through legislation such as the Americans with Disabilities Act (ADA) in the U.S. and the Disability Discrimination Act (DDA) in the U.K [2], [18]. The term *accessible* is defined by the Uniform Federal

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 199–213, 2013.
© Springer-Verlag Berlin Heidelberg 2013

Accessibility Standards (UFAS) as "[describing] a site, building, facility, or portion thereof that complies with [UFAS] standards and that can be approached, entered, and used by physically disabled people" [19].

In compliance with the ADA standards, universities and colleges aim to provide students with maps that highlight accessibility of their campus taking different approaches where some provide static maps and others support interaction with their maps. These maps are usually hosted by a designated unit in a university or college whose main objective is to provide special resources that aid students with special needs with their learning activities on campus.

In this paper, we present the concept of Personalized Accessibility Maps (PAMs) which feature an interactive map with specific functions suitable for students with special needs. To better understand PAMs, the development of a PAM for the University of Pittsburgh's main campus (PAM-Pitt), as a representative PAM, is described. Users of PAM-Pitt can locate accessible entrances of campus buildings, find shortest paths between campus buildings, request personalized optimal paths between campus buildings (e.g., a path personalized for a wheelchair user) based on their requirements, and find and locate university shuttles for travelling between campus locations.

The contributions of this paper are: (a) recognition of existing services which provide accessibility data/information; (b) conceptualization of a new geo-web system, called Personalized Accessibility Map (PAM), and its components; and (c) realization of theoretical gaps and research challenges in PAMs by developing a representative PAM for the University of Pittsburgh's main campus (PAM-Pitt). For the latter contribution, we have used, as much as possible, existing data, technologies, techniques, and tools (some based on our prior works), to develop a representative PAM.

The rest of the paper is structured as follows. Section 2 gives a background to specialized navigation services and tools and university accessibility maps focused on communities of users with special needs. The architecture and components of PAM-Pitt is detailed in Section 3. Section 4 discusses advanced features and functions in PAMs. Section 5 highlights challenges and future research directions. Section 6 provides a summary of the paper.

2 Background

2.1 Navigation Services for Communities with Special Needs

[10] have developed an algorithm to personalize routes for wheelchair users that employs individual preferences and sidewalk parameters obtained through interviews with wheelchair users, ADA guidelines, and relevant research literature. [14-17] have developed different map matching algorithms suitable for pedestrian and wheelchair navigation.

Other research projects integrate geo-crowdsourcing and routing techniques. OurWay is a prototype system that allows users to rate road segments in terms of accessibility and use the ratings for route planning [4-6]. The system is focused on physically impaired users such as wheelchair users and parents with strollers [5]. The basic road network is from OpenStreetMap and users are provided a map-based interface on

which they can rate the quality of streets, sidewalks, and paths, create new segments for the underlying network, and engage in route planning [5].

[20] have developed algorithms for utilizing multi-modal annotation in personalized multi-criteria routing for blind and wheelchair pedestrians. These algorithms utilize predefined user group profiles that are adapted by users as they rate segments on a 5-point Likert scale. The prototype system, RouteCheckr, includes an underlying geographic network from a local land surveying office and is evaluated using two criteria: safety for blind users and accessibility for wheelchair users [20].

2.2 University Accessibility Maps

Maps of accessibility can be generated for any area. As the focus of this work is university accessibility maps, it is important to understand existing accessibility maps created by universities in the US. A search of accessibility maps provided to students resulted in 20 universities (Table 1). We categorized the map types as static and interactive in which the former is either an image or a PDF map and the latter is a web map that allows users to select information to be displayed.

Of the 20 universities, seven provide students with an *interactive* web map. Of the remaining 13 universities, 11 offer detailed maps of the accessibility of the campus as a *static* PDF and two offer the same type of map as a JPEG image. The universities in which map type is marked with an asterisk offer both map types to students. None of the interactive university maps searched offers any geo-crowdsourcing features that allow students to contribute data to the maps. The data layers displayed on a map are important in determining the usefulness of the data for specific populations of users. We divide the data layers available in each university's map into two categories: basic data layers and accessibility data layers.

One other aspect of these maps that is useful to report is their support for indoor vs. outdoor information. All of the maps include accessibility information useful for outdoor navigation while 13 include both indoor and outdoor accessibility information. While the seven interactive maps searched include more data layers than PAM-Pitt, none offers interactive features that support trip planning and navigation which is at the core of PAMs. Some interactive maps have accessible paths displayed on a map but the maps do not allow any interactivity between the map and the users.

3 Personalized Accessibility Maps

3.1 Architecture and Components

The architecture of PAM-Pitt is used as a representative PAM for any campus. PAM-Pitt is designed based on a web-based client-server architecture. The client is a web browser that communicates with an internal web server and an external web server over the Internet. The client is developed using Javascript and other basic web authoring tools to receive user requests, communicate requests to the two servers and render the map and

Table 1. University accessibility maps characteristics

Campus	Map Type	Data Layers	*Accessibility* Data Layers	Environment
Allegheny College	Static	Street network, Campus buildings, Building number, Parking lot numbers, Emergency call box	Accessible buildings, Non-accessible buildings, Barrier-free routes, Accessible entrances, Automatic entrances, Accessible restrooms, Elevators, Van accessible parking, Accessible parking	Indoor and Outdoor Features
California State University at Fresno	Static	Street network, Sidewalk, Campus buildings, Public transit stop, Emergency phones, Pay phones, Drinking fountains	Accessible Routes, Bus stops, Services for students with disabilities, Telephone devices for the deaf, Blue curb, Rest area, Automatic doors, Elevators, Entrances, Ramp, Accessible restrooms	Indoor and Outdoor Features
Harvard University	Interactive*	Campus Buildings, Street Network, Green space	Accessible entrances, Entrances accessible with assistance, Accessible paths, Intra-University phones, Accessible parking, Construction areas	Indoor and Outdoor Features
MIT	Interactive*	Street network, Buildings, Campus buildings, Waterbodies, Emergency phones, Public Art, Bike racks	Accessible Entrances, Accessible ramps	Outdoor Features
North Carolina State University	Static	Street network, Railways	Accessible buildings, Partially-accessible buildings, Inaccessible buildings, Future accessible building, Access routes, Handrails, Elevation change, Steep grade, Accessible facility entrances, Accessible facility entrances with door opener, Passenger elevator, inaccessible tunnel, Accessible parking, Accessible public transit, Emergency phone	Outdoor Features
Northern Kentucky University	Static	Street network, Campus buildings, Visitor parking, Student parking, Faculty/staff parking, Open parking, Restricted parking, Tank bus stops, Emergency phones	Accessible routes (plaza level), Accessible routes (ground level), Accessible entrance (power door), Accessible parking	Indoor and Outdoor Features
Ohio University	Static	Street network, Campus buildings, Buildings, Waterbodies, Occupied and Vacant campus buildings	Handicap Parking, Accessible entrance, Construction fence, Accessible walkway, Accessible raised walkway, Difficult route, Inaccessible route, Inaccessible campus buildings	Indoor and Outdoor Features

Table 1. (*contiued*)

Purdue University	Static	Street network, Campus buildings	Accessbile buildings, Partially-accessible buildings, Non-accessible buildings, Ramps or grade-level entrances, Ramps or grade-level entrances with a door opener, TTY text telephones, Non-accessible ramps, Passenger elevators, Service elevators, Entrances with a lift, Wheel-chair-accessible restrooms, Accessible tunnels, Inaccessible tunnels	Indoor and Outdoor Features
Rice Uni-versity	Static	Campus Streets, Bus stop, Information center, Campus entrance, Visitor's entrance	Accessible buildings, Automatic accessible entrances, Accessible entrances, Ramps, Curb cuts, Accessible parking, Wheelchair lift, Assistive listening devices, TTY available, Accessible re-strooms	Indoor and Outdoor Features
St. Olaf College	Static	Street network, Campus buildings	Accessible parking, Accessible entrances	Indoor and Outdoor Features
Tufts Uni-versity	Interac-tive*	Street network, Buildings, Public transit stops	Accessible parking, Accessible restrooms, Fully accessible buildings, Buildings accessible with assistance	Indoor and Outdoor Features
University of Missouri	Interactive	Athletic Fields,Campus Buildings, Campus Streets, Parking Lots, Walkways, Non-University Buildings, Construction layers.	Accessible Entrances, Automatic Entrances, Ramps, Elevators, Chair Lifts, CurbCuts, Cross-walks, TTYs, Access Barriers, Barrier-free Sidewalks, Stairs on Walkways, Accessible Parking	Indoor and Outdoor Features
University of Montana	Interactive	Street network, Campus buildings	ADA accessible parking, En-trances with no opener, Entranc-es with automatic opener, Priori-ty snow routes, Curb cuts	Indoor and Outdoor Features
University of North Carolina	Static	Street network, Campus buildings	Ramps, Staff power entrances, Staff non-power entrances, Entrances with power door, Entrances with non-power door, Elevators/lifts, Restrooms	Outdoor Features
University of Rich-mond	Interac-tive*	Street network, Walkways, Visitor parking, Building numbers, Campus information	Accessible buildings, Accessible walkways, Accessible doors, Handicapped parking	Indoor and Outdoor Features
University of Texas at Austin	Static	Street network, Campus boundary	Automatic door, Manual door, Curb cut, Ramp	Outdoor Features
University of Wiscon-sin at Green Bay	Interactive	Google Map layers	Buildings with text description about accessibility and links to floor plans	Outdoor Features

Table 1. (*contiued*)

Wesleyan University	Static	Street network, Campus buildings	Accessible entrances, Accessible walkways, Automatic door openers, Accessible entrance to first floor, Ramps, Accessible classrooms, Elevators, ADA accessible restrooms, ADA parking	Indoor and Outdoor Features
Western Michigan University	Static	Street network, Campus buildings, Campus boundary, Waterbodies, Emergency call boxes	Parking availability, Curb cuts/ramps, Ramp or grade level entrances, Ramp or grade level entrances with door opener, Restrooms, Passenger elevators/lifts, service elevators, wheelchair lifts, Accessible, partially-accessible, and inaccessible buildings	Outdoor Features
Yale University	Static	Street network, Principal building entrance, Restricted parking, Public parking	Accessible routes w/in city blocks, Indoor routes from accessible to otherwise inaccessible buildings, Accessible entrances, Accessible entrances with automatic door, Accessible entrances normally locked, Entrances accessible via wheelchair lift, Elevator, Curb cut or driveway	Outdoor Features

other interface components to the display. The external web map server, which is Google, is accessed using the application programming interface (API) provided for the Google Maps application. The client sends a request for map data to the external server each time the page is loaded into the client. Once the requested data is returned to the client, it is rendered to the display as the background map for PAM-Pitt. The internal web server, maintained by the Geoinformatics Laboratory in the School of Information Sciences, is a web server containing the local PAM-Pitt database and PAM-Pitt's routing and mapping functions. All computations, coded in PHP, are performed in the internal web server and the client is used for map rendering and presentation.

PAM-Pitt's components include a data layer, a control layer, and a view layer (see Figure 1). The data layer, shown in the bottom of Figure 1, represents the database used by PAM-Pitt. The control layer includes the map functions and routing module to support map presentation, routing features and audio presentation modules, a module for communication with the external web map server and a module for managing user information. This layer connects the data layer to the interface layer and provides the basic procedures for data retrieval, presentation and computation. The view layer is the interactive layer designed for user interaction. In summary, the main components of PAM-Pitt are a user interface, a routing module, map functions, an audio module, and a database, each described in the remainder of this section.

User Interface. The user requests accessibility data/information and submits navigational requirements and personalized settings to PAM-Pitt through its user interface. The interface consists of a web map with interactive controls for displaying data layers, selecting origin/destination pairs, and searching for specific locations (see Figure 2).

Routing Module. The routing module in PAM-Pitt provides a user with optimized routes from origin/destination pairs using the underlying sidewalk network. Optimized routes are shortest paths and personalized routes for wheelchair users. Shortest path, which is the default route choice in PAM-Pitt, is a commonly employed route criterion in navigation systems and services and the use of distance as a criterion is motivated by user preference for the shortest possible journey between locations [13]. Personalized routing is based on individual preferences which include type of wheelchair, age, and fitness level and on sidewalk parameters such as segment length, width, slope, sidewalk condition, sidewalk traffic, and steps [10].

The routing module, in addition to the shortest and personalized routes, provides a suitable University shuttle that is close to user's origin and destination locations. In this option, PAM-Pitt finds the nearest shuttle bus stops to the origin and destination locations and links to the University's shuttle tracking system so that a user can view the selected shuttle traveling along their chosen route and the shuttle's current location. The real-time shuttle position updating consists of two parts, shuttle's position through GPS data and schedule information on the web page using Javascript. We use a server push technology called long-polling for obtaining real-time data every three seconds in the browser and reorganize the data for navigational needs. After obtaining the real-time data, the position of the shuttle is displayed and the time a shuttle will pass the stop closest to the origin is estimated. Figure 3 shows a shuttle route between a pair of origin and destination and several views of the shuttle's real-time location over several minutes.

Another important feature of PAM-Pitt's routing module is the directions provided to the users. The first requirement for providing users with turn-by-turn directions is to know the street name for each segment in the underlying network. Since PAM-Pitt uses a sidewalk network to calculate optimal routes, the use of traditional directions based on a road network is inadequate. Figure 4 shows an example sidewalk intersection. In this example, it can easily be identified that segments AI, HL, JD and KE belong to Street Alpha, and that segments BI, CJ, LG and KF belong to Avenue Beta. The remaining segments IJ, LK, IL and JK represent potential crosswalk areas that are less straightforward for identification.

Visually, IJ and LK seem to belong to Street Alpha, but when directing a user to cross Street Alpha neither IJ nor LK is useful. For instance, if a user is walking from point A to point I, the instruction "keep walking on Street Alpha" would be fine, but when the user comes to point I, "go straight on Street Alpha" is the only option. However, "cross Avenue Beta" is a better instruction for this intersection. In addition to using cross streets when providing directions at an intersection, we divide sidewalk segments into 3 different categories: sidewalk along a street, crosswalk along a building, and sidewalk along a building (see Figure 5). These different sidewalk categories have distinct attributes necessary for turn-by-turn instructions for pedestrian navigation. An example of each sidewalk category is shown in Figure 5.

Map Presentation. The map presentation module retrieves and displays the background map data from the external map server using functions from the Google Maps API and the local PAM-Pitt data layers from the internal server. Examples of the presentation of routing results and the internal data layers are shown in Figure 6.

Database. PAM-Pitt uses PostgreSQL database to store map data, data about users, and weights for route calculation. PAM-Pitt's database includes segment and node entities defining the sidewalk network, user entity that defines user characteristics, personalized parameters, ranking parameters and cost entities utilized in the routing module, and buildings and accessible entrances entities. [11] developed the sidewalk network, the backbone for routing in PAMs, for the University of Pittsburgh. A coordinate pair (i.e., latitude and longitude) represents the nodes of the sidewalk network (Figure 6d) in the database. The segments are represented by using a from-node and to-node structure and include several attributes. The attributes include street name, slope, surface type, surface condition, traffic, number of steps, width and distance.

The database also contains the campus buildings, where users can select them as origin and destination locations, and the accessible entrances (Figure 6b) to the buildings whose locations are described in texts and images (the building and the entrance). One additional layer is the University service boundary (Figure 6c) that defines the extent of University services.

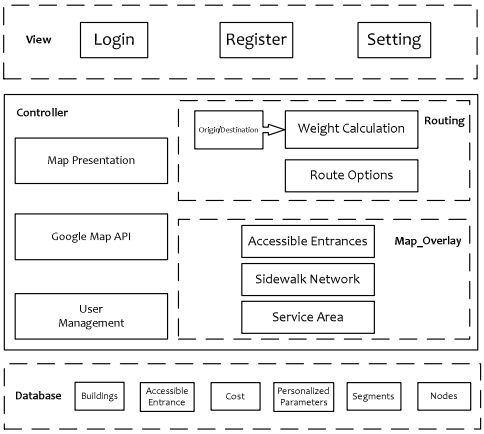


Fig. 1. PAM-Pitt's components

Fig. 2. PAM-Pitt's user interface

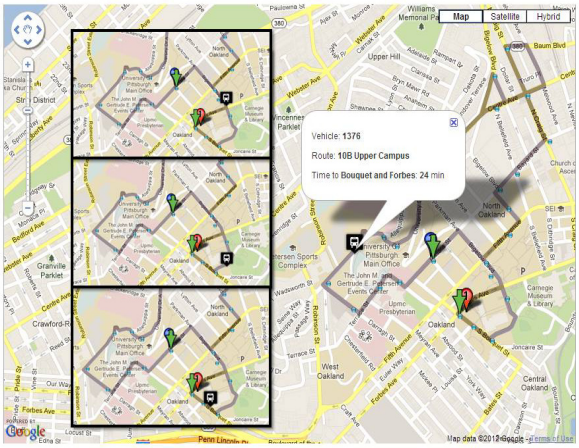


Fig. 3. Real-time shuttle location

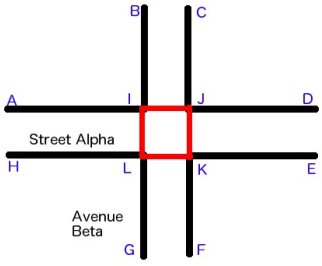


Fig. 4. Example sidewalk intersection

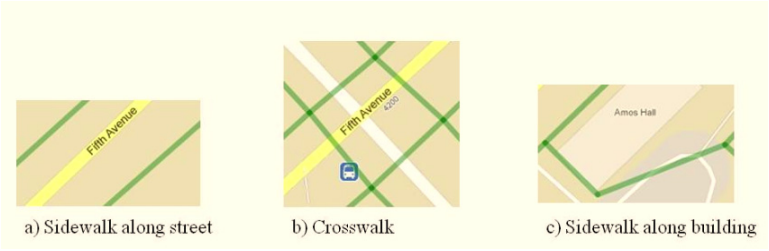


Fig. 5. Sidewalk segment categories

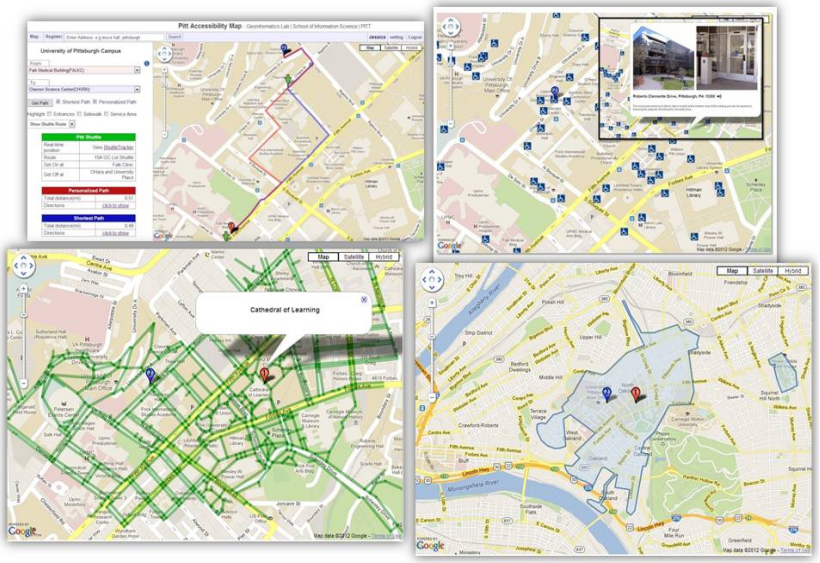


Fig. 6. Accessibility functions and data presentation. Clockwise from top left: (a) routing result presentation, (b) accessible entrances, (c) university's service boundary, (d) sidewalk network.

4 Advanced Features and Functions

Our current plan is to deploy PAM-Pitt for the University of Pittsburgh, described above, as the first version and make it available to the campus community of students, faculty, and staff. Our future plan for PAM-Pitt is to deploy new versions while gathering feedback on the first version. In this section, we describe the platform, features, and functions in future versions of PAM-Pitt.

4.1 Mobile PAM-Pitt and Data Extension

Our immediate next step is to extend the first version in consultation with the Disability Resources and Services office and implement a mobile version of PAM-Pitt, on the mobile web, to enable the widest access to potential users and allow students to access the system through their mobile devices such as iPhone, Android, iPad, among other mobile devices. We also plan to determine the basic set of accessibility data/information that a PAM must contain. We have identified an initial set of new data necessary for PAM-Pitt (see Table 2); however, the type of data the basic data set must contain remains an open question.

Table 2. Additional data for PAM-Pitt

Environment	Data	Community Relevance
Outdoor	Extension of sidewalk network to the extent of the service boundary	All
	Entrances and entrance images for all campus buildings	Mobility
	Power entrance doors	Mobility
	Manual entrance doors	Mobility
	Accessible parking lots	Mobility
	Intersections with assistive technology (APS - accessible pedestrian signals)	Vision
	Curb cuts	Mobility
	Tactile Warnings at Intersections (truncated domes)	Vision
	Alleys	Vision
	Ramps	Mobility
	Landmarks	All
Indoor	Elevator locations	All
	Accessible restrooms	Mobility
	Areas of narrow turning	Mobility
	Building floor plans for campus buildings	All

4.2 Integration of PAM-Pitt with SoNavNet and PNS

We plan to integrate PAM-Pitt with a Social Navigation Network (SoNavNet) and a Pedestrian Navigation System (PNS); both have been developed in the Geoinformatics Laboratory. These integrations will allow the full features of PAMs to be achieved. In the integrated system, students (members of SoNavNet) can share their experiences about POIs/routes/directions with other students who face same/similar mobility challenges. These POIs/routes/directions can then be recommended to other users of SoNavNet. In addition, students can receive real-time guidance from PNS for travelling on computed/recommended as well as on personalized routes. The next two sections briefly describe SoNavNet and PNS.

SoNavNet. SoNavNet is a location-based social navigation network (LBSN) that can address the navigational needs and preferences of any user, anywhere, anytime [7-8]. SoNavNet is a personalized navigation assistance provided through members' experiences. SoNavNet has a high potential for assisting people with disabilities, both indoors and outdoors, by providing a platform for users to share new and personalized data/information. SoNavNet is based on a link between the real world in which navigation occurs and the virtual world of an interactive map. In the real world, there are people who navigate the environment using different modes of transportation. These people become users of a SoNavNet and can participate by sharing and recommending POIs/routes/directions with other users. The vehicle for receiving or sharing recommendations is the interactive map in which users can add POIs, routes, directions, areas and GPS traces.

SoNavNet offers three main functions. The first allows the user to manipulate a map of their surrounding area and to place POIs, routes, directions, and areas on it,

thus letting them trace places they visit and how they get there. The second and third functions are request and recommend services. Members seeking navigation recommendations are able to message their friends with a map on which they can mark off an area and request a POI/route/direction within it. The system searches the recommended POIs/routes/directions stored in its database and sends a match POI/route/direction to the user. While SoNavNet is applicable to all communities, users with special needs have unique navigational needs that can be aided by the contributions of other users and SoNavNet offers an ideal medium for capturing the details of the unique navigational needs of these communities.

Pedestrian Navigation Service (PNS). The PNS is focused on providing navigation guidance to pedestrians on walking paths. A pedestrian network, a topological map that describes the geometric relationship between pedestrian path segments, is the key component in any PNS. Pedestrian network data is currently not available (or partly available) for many areas and in many countries compared to road network data. [9] investigated and developed techniques for automatically constructing pedestrian networks using three approaches: network buffering, collaborative mapping, and image processing. To evaluate the three approaches for automatically constructing pedestrian networks, [11] collected and constructed the pedestrian network of the University of Pittsburgh's main campus as the baseline. Different types of a pedestrian path in this network are sidewalk, crosswalk, walking trail, entrance to POI, pedestrian bridge, and pedestrian tunnel [9].

Other components of the PNS are geocoding, map matching, routing, and direction. The PNS is based on a client/server architecture where the servers provide map data (Google Maps) and render maps, compute map matching, compute routes, and generate directions. The client is an Android mobile phone which performs the following main functions: obtaining GPS data on the Android device, sending the new GPS data to the server, and receiving a map showing a computed route and the current location. The map matching module, which uses GPS data and a pedestrian network to map match newly updated GPS data, is based on a chain-coding technique [14]. The routing module is currently based on shortest distance criterion but the plans are in place to include an option for wheelchair users taking into account such criteria as shortest distance, minimum barriers, fewest slopes, avoiding bad surfaces, using only controlled crossings, and limited road crossing [10]. The direction module presents step-by-step instructions guiding the user to travel on the computed route using both text and audio on the Android phone.

5 Challenges and Future Research Directions

PAM-Pitt is the first of its kind and the first attempt at developing such a system for a university campus. This development, described above, however, has helped us realize several challenges and issues that developers of PAMs must address. Below each of these challenges along with future research directions are described.

Considering the advances and interest in geo-crowdsourcing [1], it is expected that updated and new accessible data/information will increasingly become available

through existing and new geo-crowdsourcing services. An example of data update is information about a new construction on a sidewalk segment which makes the segment impassible for a period of time. An example of new data/information is an accessible entrance to a newly constructed building or previously unmapped entrance. Updating a PAM's database with new data and updates, whenever they become available, requires routinely searching for updates in existing and new geo-crowdsourcing services. This routine checking will become a challenge for the developers of PAMs and if it is not addressed properly may result in outdated data being used with PAMs. To address this challenge, there is a need for development of techniques/tools to continually search existing geo-crowdsouring services and identify new services as they become available, and for procedures to download the required data/information.

Once adequate methods for retrieving new data and updates from geo-crowdsourcing services are determined, the next challenge is to determine an automatic means for updating PAM's database with the new accessible data/information. To address this challenge, there is a need for development of techniques/tools that: (a) convert heterogeneous data/information (current geo-crowdsourcing services do not adhere to a specific agreed upon standard type and format) to the underlying data/information type and format supported by a PAM; (b) integrate data/information using a suitable data schemas supported by a PAM's database; and (c) populate a PAM's database with new data/information.

Considering that currently there are no standards with respect to accessible data/information quality, understanding the quality of the new data and updates downloaded and populated in a PAM's database is another challenge. To address this challenge, development of techniques/tools to automatically validate the quality of accessible data/information in a PAM is needed.

6 Summary

In this paper, we introduced the concept of PAMs and discussed the development of PAM-Pitt. We have shown that PAM-Pitt is novel compared to existing accessibility maps provided at 20 university campuses in the US in that it offers navigation services, such as routing, in addition to displaying a map of accessibility at certain locations. We discussed advanced features and functions for the representative PAM that include development of a mobile PAM-Pitt and integration of PAM-Pitt with SoNavNet (a social navigation network system) and PNS (a pedestrian navigation service). The purpose of SoNavNet in PAM-Pitt is to allow users to share personal POIs/routes/directions experiences with the members of a university community and the purpose of PNS in PAM-Pitt is to facilitate real-time guidance on the computed/recommend routes within the campus area. We also discussed some challenges and future research directions such as development of techniques and tools for automatically finding updated and new accessible data/information, populating PAMs' databases, and determining quality of accessible data/information.

Acknowledgements. The work presented in this article is the result of collaboration between the Geoinformatics Laboratory in the School of Information Sciences and the

Disability Resources and Services office at the University of Pittsburgh. The authors would like to thank Ms. Piyawan Kasemsuppakorn for making the base software for an interactive map available for this work. We also thank the Disability Resources and Services at the University of Pittsburgh for providing data regarding the accessible entrances as well as consultations on the various aspects of the project.

References

1. Benner, J.G., Karimi, H.A.: Geo-Crowdsourcing. In: Karimi, H.A. (ed.) Advanced Location-Based Technologies and Services. Taylor & Francis (2013)
2. Bromley, R.D.F., Matthews, D.L., Thomas, C.J.: City Centre Accessibility for Wheelchair Users: The Consumer Perspective and the Planning Implications. Cities 24(3), 229–241 (2007)
3. Elwood, S.: Geographic Information Science: Emerging Research on the Societal Implications of the Geospatial Web. Progress in Human Geography 34(3), 349–357 (2010); First published on (July 28, 2009)
4. Holone, H., Misund, G.: People Helping Computers Helping People: Navigation for People with Mobility Problems by Sharing Accessibility Annotations. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2008. LNCS, vol. 5105, pp. 1093–1100. Springer, Heidelberg (2008)
5. Holone, H., Misund, G., Holmstedt, H.: Users Are Doing It for Themselves: Pedestrian Navigation with User Generated Content. In: Next Generation Mobile Applications, Services and Technologies (NGMAST 2007), pp. 91–99. IEEE Press, Cardiff (2007)
6. Holone, H., Misund, G., Tolsby, H., Kristoffersen, S.: Aspects of Personal Navigation with Collaborative User Feedback. In: NordiCHI 2008: Using Bridges, pp. 182–191. ACM Press, Lund (2008)
7. Karimi, H.A., Zimmerman, B., Ozcelik, A., Roongpiboonsopit, D.: Sonavnet: A Framework for Social Navigation Networks. In: International Workshop on Location Based Social Networks (LBSN 2009), Seattle, WA (2009a)
8. Karimi, H.A., Nawn, D., Zimmerman, B.: Navigation Assistance Through "Models" or "Experiences"? GIM International 23(12) (2009b)
9. Karimi, H.A., Kasemsuppakorn, P.: Pedestrian Network Construction Approaches and Recommendation. International Journal of Geographical Information Science (2012) (in press)
10. Kasemsuppakorn, P., Karimi, H.A.: Personalized Routing for Wheelchair Navigation. Journal of Location Based Services 3(1), 24–54 (2009)
11. Kasemsuppakorn, P., Karimi, H.A.: Data Requirements and Spatial Database for Personalized Wheelchair Navigation. In: 2nd International Convention on Rehabilitation Engineering & Assistive Technology, Bangkok, Thailand (2008)
12. Matthews, H., Beale, L., Picton, P., Briggs, D.: Modelling Access with Gis in Urban Systems (Magus): Capturing the Experiences of Wheelchair Users. Area 35(1), 34–45 (2003)
13. Pang, G.K.H., Takahashi, K., Yokota, T., Takenaga, H.: Intelligent Route Selection for in-Vehicle Navigation Systems. Transportation Planning and Technology 25, 175–213 (2002)
14. Ren, M., Karimi, H.A.: A Chain-Code-Based Map Matching Algorithm for Wheelchair Navigation. Transactions in GIS 13(2), 197–214 (2009a)
15. Ren, M., Karimi, H.A.: A Hidden Markov Model Map Matching Algorithm for Wheelchair Navigation. The Journal of Navigation 62(3), 383–395 (2009b)

16. Ren, M., Karimi, H.A.: A Fuzzy Logic Map Matching for Wheelchair Navigation. GPS Solutions 16(3), 273–282 (2012a)
17. Ren, M., Karimi, H.A.: Movement Pattern Recognition Assisted Map Matching for Pedestrian/Wheelchair Navigation. Journal of Navigation 65(4), 617–633 (2012b)
18. U.S. Access Board: ADA Guidelines, http://www.access-board.gov/ada/index.htm
19. Uniform Federal Accessibility Standards: 24 C.F.R. § 40 (1984)
20. Völkel, T., Weber, G.: Routecheckr: Personalized Multicriteria Routing for Mobility Impaired Pedestrians. In: ASSETS 2008, pp. 185–192. Halifax, Nova Scotia (2008)

A Probabilistic Model for Road Selection in Mobile Maps

Thomas C. van Dijk* and Jan-Henrik Haunert

Chair for Computer Science I, University of Würzburg,
Am Hubland, D-97074 Würzburg
{thomas.van.dijk,jan.haunert}@uni-wuerzburg.de

Abstract. Mobile devices provide an interesting context for map drawing. This paper presents a novel road-selection algorithm based on PageRank, the algorithm famously used by Google to rank web pages by importance. Underlying the PageRank calculation is a probabilistic model of user behavior. We provide suitable generalizations of this model to road networks. Our implementation of the proposed algorithm handles a sizable map in approximately a tenth of a second on a desktop PC. Therefore, our methods should be feasible on modern mobile devices.

Keywords: mobile maps, generalization, wayfinding, PageRank.

1 Introduction

Maps on smartphones and other mobile devices are widely used for wayfinding, orientation, and exploration tasks. Most maps displayed by commercial software, however, are not well adapted for such applications. Often, they present too much unnecessary detail and thus distract users from comprehending the essential information quickly. Generally, this problem can be approached with map generalization [5], which means that the features in the map may get simplified, aggregated, displaced, or eliminated through selection. In addition to map generalization, map distortion and map schematization can be applied to improve readability [1,10]. In this paper, however, we concentrate on the selection of roads in mobile road network maps.

The selection of a good subset of features from a geographic data set has in fact been studied at least since the 1960s when Töpfer and Pillewizer [13] investigated existing maps to derive guidelines about the number of features a map of a certain scale should contain. For a long time, however, research on map generalization focused on static topographic maps. New challenges appeared with the advent of web mapping and mobile cartography. For example, solutions that also work on small screens and with limited data bandwidths were needed [5], but also new possibilities of tailoring a map to a user's task became possible, for example, by highlighting regions of interest [17].

* Corresponding author.

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 214–222, 2013.
© Springer-Verlag Berlin Heidelberg 2013

(a) a focus-and-context map (b) a route map

Fig. 1. Two maps that show different subsets of edges (black) of a road network (gray)

In order to address different tasks, researchers have proposed methods for producing road network maps of different types. This includes you-are-here maps [11], which allow a user to localize himself, route maps [1], which allow a user to navigate from his current location to a specified destination, and destination maps [7], which allow users coming from different directions to navigate to a common destination. Two techniques that have been proven to be rather generally applicable are the grouping of sequences of almost collinear road segments into strokes [12] and the selection of roads based on betweenness centrality [4,11]. Generally, however, algorithms are missing that are configurable for multiple applications.

In this paper, we address a *range* of applications that we define by its two extreme cases, namely pure exploration and pure wayfinding. In a pure exploration application, the user needs detailed information about his surrounding but does not need to reach a certain destination. Therefore, he might favor the left map in Fig. 1(a), which displays the region around his location s in relatively high detail. With pure wayfinding, on the other hand, we mean that the user is solely interested in getting to a certain destination t and thus only needs a map similar to the right in Fig. 1(b). This map visualizes a single precomputed route from node s to node t plus some information that the user needs to not miss turns. Algorithms for maps similar to those in Fig. 1(a) and (b) have been presented before, for example, by Yamamoto et al. [16] and Agrawala and Stolte [1], respectively. We argue, however, that these maps might be of limited use if, for example, the user is interested in sights that lie close to his route or wants to have some freedom in selecting his route. In these cases, the user needs to be provided with a road network map that contains a sufficiently large choice of routes and relatively highly detailed information about his surroundings. We therefore present an algorithm for road selection that is configurable with respect to how freely the user moves through the road network. Note that this depends on the mode of transportation employed by the user, since, for example, pedestrians are usually less constrained in choosing their routes than cars. In this paper, our primary motivation is pedestrian navigation.

We propose a technique related to PageRank, the algorithm famously used by Google to rank web pages by importance [9]. As a way to rank items in a

network and a way to filter information, we consider how to apply it to road networks. In fact, PageRank has earlier been used by Jiang [3] to predict human movement in road networks, but has not been applied to road selection or map generalization before. PageRank has a probabilistic interpretation. For 'normal' PageRank this is the *random surfer* model, where the score of a page corresponds to the probabilistic behavior of a hypothetical user. In his paper, Jiang shows that his version of PageRank corresponds well to observed traffic amounts. This suggests that his (implied) model of traveler behavior is reasonable. It also suggests that specifically this underlying probabilistic model can be an interesting object of study. This provides proper interpretation for our results. We base our edge selection on the probabilities in a model of a 'random traveler:' if the user is likely to traverse a certain edge, then we should display it.

The rest of the paper is structured as follows. In Sect. 2 we review some definitions, in particular PageRank. In Sect. 3 we develop the random traveler model. We finish in Sect. 4 with some experimental results using an implementation of our proposed algorithm.

2 Preliminaries

In this paper we use the concept of a *line graph* or *linear dual graph* [15]. We use a directed version.

Definition 1 (Line graph $\mathcal{L}(G)$). *Let $G = (V, A)$ be a directed graph. Then G's line graph $\mathcal{L}(G) = (A, B)$ is a directed graph on the arcs of G, where B is defined as follows. For all $a, b, c \in V$, we have $((a, b), (b, c)) \in B$ if and only if both $(a, b) \in A$ and $(b, c) \in A$.*

We interpret the input road network as a (possibly directed) graph. We replace any undirected edges with two arcs, one going each way. Let $G = (V, A)$ be the resulting directed graph and let $n = |V|$ and $m = |A|$. Then we take the line graph $L = \mathcal{L}(G)$. At this point it would be possible to drop some arcs from L, for example to model turn restrictions [15]. We have not done so, since our data sets did not contain turn restrictions for pedestrians.

We run our ranking algorithm on the line graph L. (For correctness of the algorithm it is not required that L is a line graph.) Since the input graph G is a road network, it is reasonable to assume that it has bounded degree. Then the line graph $\mathcal{L}(G)$ has $\mathcal{O}(n)$ nodes and arcs.

Random Walks and PageRank

PageRank is a network ranking algorithm that is famous for its application in web search. It is typically formulated as an eigenvector calculation and as such is closely related to eigenvector centrality. It is well known from literature that PageRank can be interpreted using the *random surfer model* [9]. The high-level interpretation is that important pages are those that a user is likely to view. In this paper we apply a modified random 'surfer' model to road networks:

important roads for a user are those that he is likely to traverse. This involves introducing appropriate generalizations to model user behavior and to handle road networks.

First we recall the random surfer model for PageRank. Consider a hypothetical user viewing a web page. At every time step this user follows a uniformly random link on the page he is viewing. The chosen outgoing link does not depend on the incoming link that the user arrived from, so in this model the location of the user has the Markov property. Then the probability distribution of the user location as time goes to infinity can be calculated using standard methods.

The above process will get stuck on any pages that have no outgoing links. As time goes to infinity, the hypothetical user would end up only in such nodes, which clearly does not correspond to actual user behaviour. Therefore, the model includes the possibility that the user jumps to a uniformly random page. This is called *damping* and happens, with some constant probability $1 - d$, at every time step. Otherwise (with probability d) an outgoing link is followed as before. This is still a Markov process. The parameter d is called the damping factor.

Let W be the directed graph of the n pages and their links. Let M be the *link matrix*, which is the adjacency matrix of W where the columns are each scaled to sum to 1. That is, if j links to i, then $M_{ij} = 1/outdeg(v_j)$ and otherwise $M_{ij} = 0$. Let $\boldsymbol{\Delta}$ be a vector with all entries $1/n$. Then by definition, in a steady state the PageRank vector $\boldsymbol{r}$ satisfies

$$\boldsymbol{r} \;=\; dM\boldsymbol{r} + (1-d)\boldsymbol{\Delta}. \tag{1}$$

Such $\boldsymbol{r}$ exists and a numerical solution can be found efficiently, for example by iterative methods (see for example [8]).

3 Random Traveler Model

We now introduce the *random traveler model*. Like PageRank, it concerns random walks. Conceptually it applies to the line graph of a road network instead of to a web graph. Concretely, it differs from the normal random surfer model in three ways. First, we use non-uniform transition probabilities: to model user behavior, not every outgoing arc in the line graph is equally likely. Secondly, we use non-uniform damping. This allows us to focus the road selection on the user. Lastly, we handle non-uniform transition times: in contrast to links in the world-wide web, the time it takes to traverse a road segment is relevant.

Non-uniform Transition Probability. Let G and $L = \mathcal{L}(G)$ be graphs as defined before. Note that a node in L corresponds to a direction on a road segment in G. We consider the user task of navigating from a node s to a node t. A path in L corresponds to a route in the road network. Every arc on such a path corresponds to transitioning from one road segment to another. In this way, every arc represents a *routing decision*. We want to model that some routing decisions are more likely than others. This can easily be achieved by modifying

M in Equation (1): each non-zero entry in M corresponds to an arc in L and non-uniform transition probabilities can be put in the matrix instead of the uniform ones.

Then the question is of course: what transition probabilities do we use? In Jiang's Weighted PageRank [3], a method for traffic prediction, the transition probabilities are proportional to indegrees (though not specifically in a line graph). As our application is different, we pick different probabilities.

At first we tentatively consider that each arc is equally likely; we assign to every arc odds 1. Then we will increase the odds of certain arcs and afterward normalize to get probability distributions. We propose two factors for determining these odds. First, note that the user is trying to navigate to the node t. We argue that the user is more likely to navigate onto road segments that lie on the shortest path tree towards t, since that is a good decision for getting to t. Therefore, a bonus of β_{SP} is added to the odds of such arcs, where β_{SP} is a tweakable parameter. These arcs can be found using a standard shortest path tree algorithm (for example Dijkstra's algorithm [2]). The set of arcs does not depend on s and therefore only needs to be recomputed when t changes.

Secondly, we propose a bonus for arcs that represent going approximately straight on. This corresponds for example to the *non-turning* concept of Klippel et al. [6]. Taking such non-turning transitions means routes with low turn complexity; people prefer such routes [14]. Additionally, we argue that the user might 'miss a turn' and mistakenly go straight ahead. Therefore, we give a bonus of $\beta_{\|}$ to arcs that correspond to a turn of less than $\alpha_{\|}$. This depends neither on s nor on t so it needs to be computed only once per road network.

With a higher value for this non-turning bonus $\beta_{\|}$, we model a dislike of turns. Note that we could instead have put a cost for turns into the shortest-path calculations. (See for example Winter [15].) That results in prescribing to the user what the exact trade-off is between turn minimization and distance minimization. By putting this trade-off into our random traveler model we allow for a more nuanced calculation that is likely to find multiple reasonable paths. We think this better reflects the nature of pedestrians.

In Sect. 4 we explore how the values for these bonus parameters influence subsequent edge selection. Throughout the paper we use $\alpha_{\|} = 30°$. For now, we mention that on our data, values like $\beta_{\mathrm{SP}} = 5$ and $\beta_{\|} = 5$ give reasonable results.

Non-uniform Damping. Our second modification to PageRank is non-uniform damping. With a motivation different from ours, a modified damping vector is known in web-search literature as a *personalization vector*. The damping in PageRank corresponds to the hypothetical user jumping to a random web page. Instead, we set the damping to always return to node s. In terms of Equation (1), this is accomplished by setting $\boldsymbol{\Delta}$ to 1 at s and 0 elsewhere.

To see why this makes sense, consider a random walk starting from s, influenced by damping. As argued before, the ranking vector $\boldsymbol{r}$ that we calculate is the probability distribution of the location of the traveler as time goes to infinity. Because of our damping vector, at every step of the process there is a probability

$(1-d)$ that the walk returns to s. Consider splitting the walk at these jumps so that each sub-walk starts at s. Sub-walks of length k occur with exponential distribution in k: at every time step, there is a constant probability that the sub-walk terminates. Then $\boldsymbol{r}$ is also the probability distribution of the location of the traveler, taking a random walk of exponentially distributed length, starting at s. This is a reasonable distribution for how far the user's interest deviates from his current location.

The interpretation of the calculated scores is then: starting from s, where is the traveler likely to go soon? This way we can use a real-world property to pick the damping factor d, a parameter for which it is generally not clear how to pick a value. A sub-walk of k steps occurs by taking k normal steps, each at probability d, and then a damping step, at probability $(1-d)$. The relation between the expected length $\mathcal{E}$ and d is then $\mathcal{E} = \sum_{k=1}^{\infty} k \cdot (1-d) \cdot d^k$ which results in $d = \frac{\mathcal{E}}{\mathcal{E}+1}$.

Non-uniform Transition Time. As third and final modification to PageRank, we handle non-uniform transition time. A node in L corresponds to (a direction on) a road segment, so spending time traversing a road segment corresponds to spending time in a node. The amount of time does not depend on where the user came from or where the user is going. The ranking vector $\boldsymbol{r}$ as used up to now is the proportion of events where the user goes to a certain node. To get the probability over time instead of over events, simply weigh by the time spent at each node and renormalize.

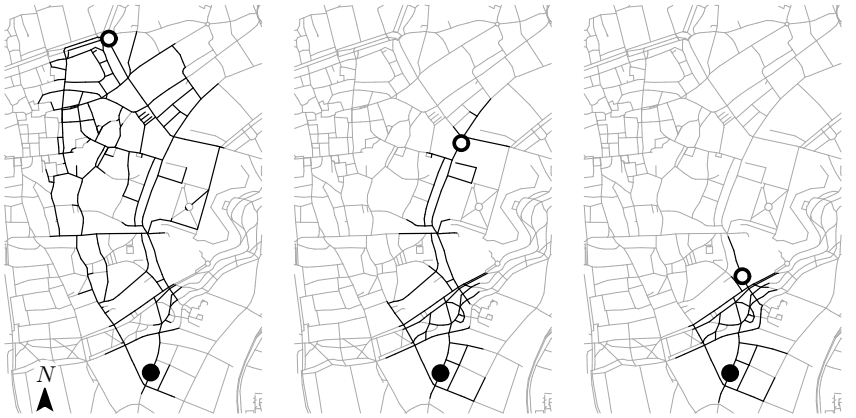
4 Experiments

We have implemented the ranking algorithm proposed in this paper. To get an edge selection from the calculated scores, we select the minimum number of edges that together sum to score at least 90%. According to the random traveler model, this set of edges contains wherever the user is likely to go soon.

In summary, the runtime on reasonably large data sets is good and certainly feasible on a mobile device. In this section, we give details of this claim. All experiments have been run on a desktop PC with an Intel® Core™ i5-2400 CPU at 3.10Ghz. Though that is certainly more powerful than a current smartphone, it is not so by orders of magnitude.

Our examples are run on a road network of the German city of Würzburg and its surroundings, consisting of 3786 nodes and 4987 road segments. This network is a crop of a much larger OpenStreetMap data set of Lower Franconia (approximately 10^5 nodes) that is available online[1] in the ESRI shapefile format. This is generally a reasonable approach for our method: first generously clip an area of a larger map and then run our algorithm on the cropped map. By the nature of the random traveler model, we will not select edges very far from the initial user location. Since our algorithm is fast and space efficient, a very wide

[1] `download.geofabrik.de/osm/`



(a) $T = 0$ (b) $T = 16$ (c) $T = 32$

Fig. 2. Several edge selections while a user travels a shortest path to a destination. The value T indicates the number of road segments traveled. Current user location is marked with ○, destination is marked with ●.

crop can be used so as not to influence the results. The figures in this paper have been clipped further to show only the area where edges are actually selected.

In Fig. 2 we show the edge selections at several times for a user traveling a shortest path toward a destination. The value T indicates the number of road segments traveled. The focus area shifts to match the current user location. Our algorithms are fast enough to do these calculations in realtime on a smartphone.

Note that in the first selection, possibilities for various distinct routes are presented. As time goes on, the selection becomes more concentrated. Also note that more detail is added near the destination as the user approaches. These things follow from the random traveler model.

To calculate the scores in our random-traveler model, we use the typical iterative method, where nodes distribute score to their neighbors. It should be possible to use more sophisticated methods as known from PageRank literature, but we have not investigated this: a simple implementation already gives good runtime on reasonable data sets.

As termination criterion for our calculations we have used a threshold of 10^{-6}: we continue as long as the sum of score changes in an iteration is more than this value. Then it is reasonable to assume convergence. In an experiment with 1000 random source/destination pairs, this procedure took an average of 102ms (719 iterations).

All experiments use damping factor $d = 0.98$. As calculated in Sect. 3, this means the expected length of paths considered is 49 transitions, which is reasonable for queries in this network. As mentioned before, all our experiments use $\alpha_{\|} = 30°$. Figures 3 and 4 show the effect of the shortest-path bonus β_{SP} and the non-turning bonus $\beta_{\|}$. Notice how β_{SP} allows a gradual change from exploration to wayfinding. Some discussion is included in the captions.

(a) $\beta_{\mathrm{SP}} = 0$ (b) $\beta_{\mathrm{SP}} = 1$ (c) $\beta_{\mathrm{SP}} = 5$

Fig. 3. Effect of the shortest-path-bonus parameter β_{SP}. Notice that a higher value directs the edge selection toward wayfinding versus exploration. All of these pictures have $\beta_{\|} = 0$. User location is marked with ○, destination is marked with ●.

(a) $\beta_{\|} = 0$ (b) $\beta_{\|} = 5$ (c) $\beta_{\|} = 15$

Fig. 4. Effect of the non-turning-bonus parameter $\beta_{\|}$. Notice that a higher value leads to the selection of longer non-turning stretches of road. Here more edges are selected if $\beta_{\|}$ is higher: the random traveler distributes over a wider area. All of these pictures have $\beta_{\mathrm{SP}} = 0$, so are uninfluenced by the location of the destination. User location is marked with ○.

5 Conclusion

We have presented a novel algorithm for edge selection. Partially inspired by PageRank, it is based on a 'random traveler model' and has a solid foundation in probability theory. Using an implementation we have shown that the proposed algorithm runs in realtime.

As regards future work, we have currently only concerned ourselves with road selection. It could be interesting to apply schematization to the resulting selection. Additionally, we note that it may be interesting to use data mining to train a traveler model based on real-world data.

References

1. Agrawala, M., Stolte, C.: Rendering effective route maps: improving usability through generalization. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001, pp. 241–249. ACM (2001)
2. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
3. Jiang, B.: Ranking spaces for predicting human movement in an urban environment. International Journal of Geographical Information Science 23(7), 823–837 (2009)
4. Jiang, B., Claramunt, C.: A structural approach to the model generalization of an urban street network. Geoinformatica 8(2), 157–171 (2004)
5. Jones, C.B., Ware, J.M.: Map generalization in the Web age. International Journal of Geographical Information Science 19(8), 859–870 (2005)
6. Klippel, A., Tappe, H., Kulik, L., Lee, P.U.: Wayfinding choremes – a language for modeling conceptual route knowledge. Journal of Visual Languages and Computing 16(4), 311–329 (2005)
7. Kopf, J., Agrawala, M., Bargeron, D., Salesin, D., Cohen, M.: Automatic generation of destination maps. ACM Transactions on Graphics 29(6), 158:1–158:12 (2010)
8. Langville, A.N., Meyer, C.D.: Google's PageRank and Beyond: The Science of Search Engine Rankings, ch. 4. Princeton University Press (2006)
9. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford University (1999)
10. Schmid, F.: Knowledge-based wayfinding maps for small display cartography. Journal of Location Based Services 2(1), 51–83 (2008)
11. Schmid, F., Kuntzsch, C., Winter, S., Kazerani, A., Preisig, B.: Situated local and global orientation in mobile you-are-here maps. In: Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services, MobileHCI 2010, pp. 83–92. ACM (2010)
12. Thomson, R.C., Brooks, R.: Exploiting Perceptual Grouping for Map Analysis, Understanding and Generalization: The Case of Road and River Networks. In: Blostein, D., Kwon, Y.-B. (eds.) GREC 2001. LNCS, vol. 2390, pp. 148–157. Springer, Heidelberg (2002)
13. Töpfer, F., Pillewizer, W.: The principles of selection. The Cartographic Journal 3(1), 10–16 (1966)
14. Turner, A.: The Role of Angularity in Route Choice. In: Hornsby, K.S., Claramunt, C., Denis, M., Ligozat, G. (eds.) COSIT 2009. LNCS, vol. 5756, pp. 489–504. Springer, Heidelberg (2009)
15. Winter, S.: Modeling costs of turns in route planning. GeoInformatica 6, 345–361 (2002)
16. Yamamoto, D., Ozeki, S., Takahashi, N.: Focus+glue+context: an improved fisheye approach for web map services. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2009, pp. 101–110. ACM (2009)
17. Zipf, A., Richter, K.-F.: Using focus maps to ease map reading: Developing smart applications for mobile devices. Künstliche Intelligenz (KI) 2(4), 35–37 (2002)

Author Index