Proceedings

André Santanchè and Pedro R. Andrade (Eds.)

Dados Internacionais de Catalogação na Publicação

SI57a Simpósio Brasileiro de Geoinformática (11. : 2013: Campos do Jordão,SP)

Anais do 14º Simpósio Brasileiro de Geoinformática, Campos do Jordão, SP, 24 a 27 de novembro de 2013. / editado por Pedro Ribeiro de Andrade (INPE), André Santanchè (Unicamp). – São José dos Campos, SP: MCTI/INPE, 2013.
CD + On-line
ISSN 2179-4820

1. Geoinformação. 2.Bancos de dados espaciais. 3.Análise Espacial. 4. Sistemas de Informação Geográfica (SIG). 5.Dados espaço-temporais. I. Andrade, P.R. II.Santanchè, A. III. Título.

CDU: 681.3.06

Preface

This volume of proceedings contains papers presented at the XIV Brazilian Symposium on Geoinformatics, GeoInfo 2012, held in Campos do Jordão, Brazil, November 24-27, 2013. The GeoInfo conference series, inaugurated in 1999, reached its fourteenth edition in 2013. GeoInfo continues to consolidate itself as the most important reference of quality research on geoinformatics and related fields in Brazil.

GeoInfo 2013 brought together researchers and participants from several Brazilian states, and from abroad. The number of submissions reached 43, with very high quality contributions. The Program Committee selected 17 papers submitted by 69 authors from 21 distinct Brazilian academic institutions and research centers, representing 24 different departments, from 15 different cities, 10 different states and 4 different countries: Brazil, Italy, United States and Germany. Most contributions have been presented as full papers, but both full and short papers are assigned the same time for oral presentation at the event. Short papers, which usually reflect ongoing work, receive a larger time share for questions and discussions. The conference included special keynote presentations by Helen Couclelis and Randolph W. Franklin, who followed GeoInfo's tradition of attracting some of the most prominent researchers in the world to productively interact with our community, thus generating all sorts of interesting exchanges and discussions.

This year the event added two novelties in the program: the Lightning Talks and Interactive Open Source GeoInformatics Forum. The Lightning Talks are five-minute presentations focusing on one key point. It can be a work in progress, a provocative idea, a collaboration invitation, a demonstration, or whatever that has potential to stimulate interesting discussions. The Interactive Open Source GeoInformatics Forum is an initiative of approximating the GeoInformatics Open Source and GeoInfo academic communities fostering a debate around the open source software.

We would like to thank all Program Committee members, listed below, and additional reviewers, whose work was essential to ensure the quality of every accepted paper. At least three specialists contributed with their review for each paper submitted to GeoInfo. Special thanks are also in order to the many people that were involved in the organization and execution of the symposium, particularly Simone Almeida Chaves Santanchè and INPE's invaluable support team: Daniela Seki, Janete da Cunha, Luciana Moreira, Denise Nascimento, and Gláucia Pereira da Silva.

Finally, we would like to thank GeoInfo's supporters, the São Paulo Research Foundation (FAPESP), the Brazilian Council for Scientific and Technological Development (CNPq), the Brazilian Computer Society (SBC) and the Society of Latin American Remote Sensing Specialists (SELPER-Brasil), identified at the conference's web site. The Brazilian National Institute of Space Research (Instituto Nacional de Pesquisas Espaciais, INPE) has provided much of the energy that has been required to bring together this research community now as in the past, and continues to perform this role not only through their numerous research initiatives, but by continually supporting the GeoInfo events and related activities.

Campinas and São José dos Campos, Brazil.

André Santanchè
Program Committee Chair

Pedro R. Andrade
General Chair

Conference Commitee

General Chair

Pedro R. Andrade
National Institute for Space Research, INPE

Program Chair

André Santanchè
State University of Campinas, UNICAMP

Local Organization

Daniela Seki
INPE

Luciana Moreira
INPE

Gláucia Pereira da Silva
INPE

Janete da Cunha
INPE

Denise Nascimento
INPE

Support

FAPESP - São Paulo Research Foundation
CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico
SELPER-Brasil - Associação de Especialistas Latinoamericanos em Sensoriamento Remoto
SBC - Sociedade Brasileira de Computação

Program committee

Pedro Andrade, INPE
André Santanchè, UNICAMP
Leila M. G. Fonseca, INPE
Flavia Feitosa, INPE
Joachim Gudmundsson, NICTA, Sydney, Australia
Claudio Baptista, Universidade Federal de Campina Grande
Ana Paula Afonso, Universidade de Lisboa
Sergio Faria, UFMG
Andrea Tavares, UFOP
Clodoveu Davis, UFMG
Marco Casanova, PUC-Rio
Armanda Rodrigues, Universidade Nova de Lisboa
Werner Kuhn, University of Münster
Jugurta Lisboa Filho, Universidade Federal de Viçosa
Renato Fileto, UFSC
Ricardo Ciferri, UFSCAR
Frederico Fonseca, The Pennsylvania State University
Silvana Amaral, INPE
Joao Cordeiro, INPE
Stephan Winter, University of Melbourne
Raul Feitosa, PUC-Rio
Valeria Soares, UFPB
Gilberto Queiroz, INPE
Vania Bogorny, UFSC
Lubia Vinhas, INPE
Marcus Andrade, Universidade Federal de Viçosa
Sergio Rosim, INPE
Maria Isabel Escada, INPE
Jorge Campos, UNIFACS
Camilo Renno, INPE
Ricardo Torres, UNICAMP
Valeria Times, UFPE
Karine Ferreira, INPE
Luis Alvares, UFRGS
Gilberto Camara, INPE
Marcelo Tilio, Tecgraf / PUC-Rio
Leonardo Azevedo, IBM Research Brazil
Marcelino Silva, UERN
Matt Duckham, University of Melbourne, Australia
Jin Soung Yoo, Indiana University-Purdue University,USA
Helen Couclelis, University of California, Santa Barbara
Randolph Franklin, Rensselaer Polytechnic Institute, USA
Dieter Pfoser, George Mason University, USA
Antônio Monteiro, INPE - Brazil
Laércio Namikawa, INPE - Brazil
Tiago Carneiro, UFOP - Brazil
Carla Macario, Embrapa
Jose Laurindo Campos dos Santos, INPA

External Reviewers

Júlio César Esquerdo, Embrapa
Aylton Pagamisse, FCT - Unesp

Contents

DM4VGI: A template with dynamic metadata for documenting and validating the quality of Volunteered Geographic Information

Wagner D. Souza, Jugurta Lisboa-Filho, Jarbas N. Vidal Filho, Jean H. S. Câmara

Departamento de Informática – Universidade Federal de Viçosa (UFV)
36570-000 – Viçosa – MG – Brazil

{wagnerdiasdesouza, jeanhsc2010}@gmail.com, jugurta@ufv.br, jarbasfito@hotmail.com

Abstract. Volunteered Geographic Information (VGI) is a Web phenomenon known as "user-generated content", which involves resources from Web 2.0 and geographic data. Geobrowsers are websites that collect and provide VGI. These systems, however, do not follow norms or standards for data collection or documentation, which makes it difficult to recover such data and limits the interoperability among VGI systems. In addition, the quality of the data collected by VGI systems has often been questioned. This paper proposes the DM4VGI, a template for creating dynamic metadata from volunteered geographic information provided by users through Geobrowsers or Virtual Globes. The DM4VGI is used to document and validate the quality of the VGI, as well as to facilitate data interoperability.

1. Introduction

A Web phenomenon known as "user-generated content" is increasing and diversifying the creation of data in collaborative environments. Some examples of such phenomenon are the free encyclopedia - Wikipedia[1], the world maps voluntarily produced such as OpenStreetMap[2] and Wikimapia[3], the information on public security available at Wikicrimes[4], and several websites where clients are able to make comments about the quality of a product or service provided by a company.

The term Volunteered Geographic Information (VGI), defined by Goodchild (2007), characterizes a specific type of "user-generated content", which combines three fundamental elements: Web 2.0 [O'Reilly, 2010]; collective intelligence [Lévy, 1999]; and neogeography [Turner, 2010].

A collaborative Web system is an environment where data are voluntarily contributed by the users, who can also discuss and assess the information provided by other users [Souza et al., 2012]. A Geobrowser is a type of collaborative system that enables the search, access, visualization and integration of geospatial data [Schrader-Patton et al., 2010]. According to Cooper et al. (2010) a Geobrowser that presents data in the form of a globe is called Virtual Globe (e.g.: Google Earth[5]). Websites such as

[1] http://en.wikipedia.org/wiki/Wikipedia
[2] http://www.openstreetmap.org
[3] http://wikimapia.org
[4] http://www.wikicrimes.org/main.html
[5] http://www.google.com/earth/index.html

OpenStreetMap and Wikimapia are examples of collaborative Geobrowsers.

A common problem of collaborative Geobrowsers is that such systems do not usually follow norms or standards. Thus, the geographic information they collect and provide on the Internet are usually scattered and disconnected, hindering the interoperability of different systems. Even with the aid of the modern search engines available on the Web, locating and recovering information are difficult tasks since the VGI are usually created without any documentation/metadata [Cooper et al., 2010]. In order to achieve interoperability and reuse, the produced geographic data must have an associated metadata to facilitate the search and provide the necessary details for the proper use of such information.

The VGI can be dynamic, i.e., continually modified, but the user of a collaborative environment does not normally spend time or effort to fill out metadata forms. Thus, in order to document a collaborative web-based VGI System, it is necessary to automate metadata capture during the VGI data collection process and update the values of the metadata elements to reflect the dynamic changes in the VGI. According to Cooper et al. (2010), using dynamic metadata is required in order to document a VGI.

This paper proposes a template for the creation of dynamic metadata from information voluntarily provided by Geobrowsers, namely the Dynamic Metadata for VGI (DM4VGI). In this template, the metadata are dynamic because the values of their elements are collected and modified in real-time, in order to consider the possible changes in the VGI. The DM4VGI is used to document and allow other users to verify and certify the quality of the VGI.

The DM4VGI can be used by any collaborative Geobrowser and allows the metadata of the systems to be released and found by users or other software. The DM4VGI also enables the interoperability of data collected by several websites with different and dynamic content.

The structure of the paper is as follows: Section 2 presents a review of VGI and methods developed to improve its quality, in addition to a discussion on metadata and their importance. Section 3 describes in details the elements of the template proposed to document and verify the quality of the VGI from Geobrowsers, and justifies the choice of these elements. Section 4 describes the processes of creation and management of dynamic metadata. Section 5 presents the overall conclusions of the paper and recommendations for further research.

2. Volunteered Geographic Information (VGI)

According to Goodchild and Li (2012), the quality of the VGI can be highly variable and there is no method or procedure to completely assure its reliability. Because the VGI fails to follow the scientific principles of sample design and its coverage is incomplete with respect to the production of geographic data, its quality can be questioned.

However, this does not mean the VGI does not have its value. The VGI can be a generator of numerous scientific hypotheses, which can be validated using stricter methods [Goodchild and Li, 2012]. Important information can be more easily

discovered using VGI [Georgiadou et al., 2011 and Cooper et al., 2010]. For instance, a person who has lived in a given street for a long time is likely to have knowledge that can be shared with other people using the VGI. Thus, according to Goodchild and Li (2012), the development of methods to improve the quality of VGI may provide benefits to users VGI.

Many studies have proposed and evaluated methods that attempt to assure or improve the quality of the VGI [De Longueville et al., 2010; Cooper et al., 2011; Goodchild and Li, 2012]. The main methods proposed to verify if the VGI collection process is effective are: (1) to compare the VGI with other data sources; (2) to allow the users to grade and comment a VGI, thus making them responsible for its validation; (3) to make VGI available to data production specialists or official agencies, to be analyzed and validated by people with technical knowledge in a collaborative manner.

2.1. Methods to assess and improve the quality of the VGI

Comparing VGI with similar information from other official data sources is one method to assure its quality. For instance, the website Tracks4Africa[6] receives volunteered information, which are often repeated, overlapping or similar, thus a comparison between data collected by the system with data from external sources is performed.

Cooper et al. (2011) suggests the importance of using tools that automatically correct the VGI, such as for logical consistency of a text. This type of tool may work as follows: it evaluates the information inserted by the user and, in case an invalid value is found, it requests the user to modify the contribution, pointing out the error as well as its location. These automatic correction mechanisms would be more helpful if used by several VGI Web providers, thus it is important to provide those tools in a web service format [Cooper et al., 2011].

According to De Longueville et al. (2010), another method to automatically assess the quality of the VGI is to use a zoom management system: when the user performs a voluntary contribution, the system registers the current zoom level of the map and associates it to the VGI. It is likely that the higher the zoom level at the moment of contribution, the higher the spatial accuracy of the VGI. As the zoom level increases, the system can automatically attribute a higher value to the metadata element related to accuracy. However, there are no guarantees that this method will always be successful: even though a user gives maximum zoom to provide a contribution, the process is still subject to errors, made either on purpose or by mistake [De Longueville et al., 2010].

According to Maué (2007), a reputation system similar to those seen in electronic commerce websites is important to help improve the quality of the VGI. In this method, the users of a VGI system can grade and comment on contributions provided by other volunteers, and thus co-validate or not the VGI. For instance, as the VGI receives positive feedbacks, its reliability is more likely to increase.

For Cooper et al. (2011), every volunteered information may have its quality assured by means of data review using Wiki services. Wikipedia is a successful example of good quality volunteered information, which the content is voluntarily provided and

[6] http://tracks4africa.co.za

revised [Kittur, 2008 and Cooper et al., 2011]. Thus, data review performed by other users can be an important factor to encourage the contributor to follow certain norms and standards, as well as to provide metadata [Cooper et al., 2011].

Goodchild and Li (2012) proposed three different approaches as an attempt to assure the quality of the VGI. The first approach suggests a review of the VGI by other users under the philosophy that "the more the users visualize and revise the VGI, the more acceptable the information may become", similarly to what happens with Wikipedia. The second approach suggests a reputation system that relies on a hierarchy of trusted users. For example, a user who more frequently makes positive contributions and reviews could be assigned a higher score, and be given a role as moderator or administrator of the system. The third approach suggests the use of methods for the automated triage system, to assess the credibility of the information using geographic theoretical concepts. The final score of the contribution can be calculated as the weighted average of the scores attributed by all users, considering that the scores from users in the higher levels of the hierarchy were given greater weights. The roles of the users may define permissions to remove illegal data, prevent or limit the editing of content which has already been corrected, or even ban malicious users, thus helping to control the system.

De Longueville et al. (2010), suggest the use of a list of predefined values from 0 to 5, in the "DropDown" or "ComboBox" format. Each of these values refers to a level of accuracy of the data, which is defined according to the creator of the content. Thus the higher the value selected by the user who produced the VGI, the higher the data accuracy. This value should be registered in the VGI metadata element related to its accuracy.

2.2. VGI metadata and documentation

Metadata are defined as "data about data". Documentation of official data produced by agencies or organizations must follow international documentation standards. VGI should be documented as well, either manually or automatically.

According to Cooper et al. (2011), the VGI metadata must have elements that reflect its quality and the users experience in making use of such information. However, this is no guarantee against frauds or untruths in the metadata fields related to quality, since they are filled out by the contributor himself or automatically by tools based on concepts, which are not always true. This is a critical issue even for metadata generated by professional organizations or official agencies [Cooper et al., 2011].

An example of VGI metadata is seen in the GPS Traces of the OpenStreetMap, whose elements are: file name; date when information was sent; number of points collected; initial coordinate; owner; description; keywords and visibility. Another example is the The Southern African Bird Atlas Project 2[7], whose elements are: observers, cards, records, incidentals and pentads. These metadata are related to all VGI collected in a given period of time, and not to a single VGI. This situation is similar for the Wikicrimes system metadata, which are number of registered crimes, number of occurrences and percentage of each type of occurrence per area.

[7] http://sabap2.adu.org.za/

Fields for the control and analysis of changes in the VGI are also important. For example, the fields of the History Changeset[8] of the OpenStreetMap systems identify the user who has modified a VGI, what item was altered and when the modification occurred. The OpenStreetMap is one of the few VGI systems that provide part of its data along with metadata. However, these metadata are standardized for only one specific type of contribution, such as the GPS Traces metadata. Ultimately, it is noticeable a total lack of standards for documenting the VGI, such as that existing for official geographic information. The following section presents an initial proposal of a standardized procedure for VGI documentation.

3. A template for VGI documentation

When analyzing or proposing a procedure for VGI documentation, there are two distinct paths to follow. One is the documentation of VGI from Geobrowsers and the other is the documentation of more complex and complete data, generated using software or Geographic Information System (GIS) tools, which can normally be available in a Spatial Data Infrastructure (SDI) [Cooper et al., 2011]. In this section we present the template Dynamic Metadata for VGI - DM4VGI, proposed to document VGI obtained from Geobrowsers.

According to Cooper et al. (2011), voluntary users do not usually consider spending additional time to document data, which they are already providing in a collaborative manner. Thus, the DM4VGI focus on the most relevant information about the data, in addition to being able to capture metadata automatically. Another interesting fact is that in several VGI systems, the information is not usually required to be exact. For example, the locations of a pothole in the street or occurrence of a robbery do not require high precision, i.e., the identification of a corner of such street or the block where the event was observed may be sufficient for decision makers who will respond to the VGI.

The DM4VGI was developed to achieve four objectives: to standardize and facilitate the documentation of the VGI; to provide statistical data with respect to the use of the VGI; to register data regarding VGI quality, captured through methods of VGI quality improvement; and to provide an efficient mechanism for VGI data search and access.

The template comprises 39 elements organized into five groups: Identification; Time Record; Geopositioning; VGI quality; and Auditing and Distribution. Figure 1 shows the structure of the DM4VGI and its communication with the collaborative environment through the VGI Documentation Module for data management communication.

Figure 2 shows the elements related to the data identification. Elements marked with ´*´ are mandatory. The elements Title (1.1) and Summary (1.2) are used to quickly identify the data, and they are present in several standards for data documentation. Two important elements to classify the contribution are Category (1.3) and Type (1.4), which are filled out through controlled vocabulary using “DropDown” or “ComboBox”. These elements facilitate the search for similar information or of a specific theme, such as

[8] http://www.openstreetmap.org/history

infrastructure, entertainment or public safety. They also enable a more detailed search such as for data regarding potholes in a street ("potholes in the street" could be a type of "infrastructure") or streets with the greatest number of robberies ("robberies" could be a type of the category "public safety"). Thus the DM4VGI has the elements "category" and "type", where type is always associated with only one category.

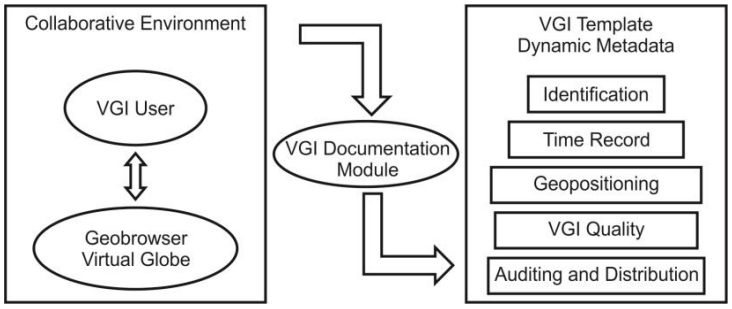


Figure 1. DM4VGI structure and communication with the collaborative environment through the VGI Documentation Module.

The DM4VGI aims to support the interoperability of data from several websites, thus two other additional elements were proposed: Software (1.5) - used in data collection (e.g.: Google Maps API[9]); and Website (1.6) - used in data collection (e.g.: Wikicrimes). This highlights the needs for auditing the VGI collection, documenting the tools used in this process as well as the organizations or websites responsible for it.

	Element	Item	Type	Auto captured
Identification	Title*	1.1	Text	X
	Abstract	1.2	Text	
	Category*	1.3	String (controlled vocabulary)	X
	Type*	1.4	String (controlled vocabulary)	X
	Software*	1.5	Text	X
	Website*	1.6	URI	X

Figure 2. DM4VGI Template – Identification Group

Figure 3 shows the fields related to the time record of the data. The relevant elements for a complete temporal search are: Date and Time of the Contribution (2.1); Date and Time of Occurrence (2.2); and Date and Time of Updates (2.3). Date and time of the contribution means the moment when the contribution was performed. Date and time of occurrence means the moment when the fact or event occurred, and not the moment it was recorded. For example, the user inserts a robbery record into the system three days after its actual occurrence date, or the user registers an event that will take place one week after the contribution date. Date and time of updates contain a list of dates and times when changes in the VGI were performed. If the system allows Wiki reviews, the VGI can be modified several times, and every change must be recorded and maintained for future analysis.

Figure 4 shows the mandatory elements related to data positioning. The VGI should be documented in a way so that it enables the spatial search of the metadata.

[9] https://developers.google.com/maps/documentation/javascript/

Thus it is necessary to maintain the data's bounding rectangle (elements 3.1 to 3.4). The element Geometry (3.5) registers the type of the VGI geometry: point; line; or polygon.

	Element	Item	Type	Auto captured
Time Record	Date and Time of the Contribution*	2.1	Timestamp	X
	Date and Time of Occurrence	2.2	Timestamp	X
	Date and Time of Updates*	2.3	Array Timestamp	X

Figure 3. DM4VGI Template – Time Record

	Element		Item	Type	Auto captured
Geopositioning	Bounding Rectangle	North*	3.1	Float	X
		South*	3.2	Float	X
		East*	3.3	Float	X
		West*	3.4	Float	X
	Geometry*		3.5	String	X

Figure 4. DM4VGI Template – Geopositioning

Figure 5 shows the elements related to the Statistics and Quality of the VGI. The element User's History (4.1) registers the personal comments made by a user who has downloaded, used, or has relevant information about the VGI. This element was included in the DM4VGI due to the advantage of maintaining post-use data, for instance, the post-use information of an Ebay[10] product, as well as the collaborative forum of the systems Wikicrimes, VGIPantanal [Souza et al., 2012] and MossoroCrimes [Vidal Filho et al., 2013].

The collaborative forum of the Wikicrimes and MossoróCrimes systems can be defined as the fields where the user can give opinions and comment on the contributions, with the intention of helping in the process of validation or showing its errors or untruths. Thus the VGI users provide a wide range of information, characteristics and consequences of the post-use of the VGI, which can also be considered as tools for the VGI quality assessment. The users can use such collaborative information as basis to identify if the VGI is true or false.

The element Number of Visualizations (4.2) is used to register the number of times the contribution has been visualized, allowing the identification of the most accessed contributions.

The first approach described by Goodchild (2012) suggests a Wiki review of volunteered data. Therefore, it is important to register in the metadata the number of Wiki reviews performed, and obtain an estimate of how much such contribution has been improved or extended. Such record makes it possible to assess the impact of Wiki reviews on the VGI in a quantitative manner. The element Number of Wiki Reviews (4.3.1) registers the number of Wiki reviews who revised a VGI. The element Number of Wiki Users (4.3.2) registers the number of different users who revised a VGI.

The element Contribution History (4.4) registers a list with a unique identifier for old contributions in order to keep a history of volunteered information and, if necessary, to enable going back to a previous state of the VGI. For instance, if a user modifies the VGI and someone or some automated data correction method identifies

[10] http://www.ebay.com/

that the status of the VGI was altered from "correct" to "incorrect", it is necessary to return it to its previous state.

	Element		Item	Type	Auto captured
VGI Quality	User's History		4.1	Array Text	
	Number of Visualizations*		4.2	Int	X
	Number of Wiki	Reviews*	4.3.1	Int	X
		Users*	4.3.2	Int	X
	Contribution History		4.4	Array Int	X
	Number of Evaluations*		4.5	Int	X
	Score	Final Score*	4.6	Float	X
		Minimum Value*	4.6.1	Float	X
		Maximum Value*	4.6.2	Float	X
	Status*		4.7	String	X
	Assessment Method		4.8	Text	X
	Zoom Contribution*		4.9	Text	X
	Zoom Wiki		4.10	Array Text	X

Figure 5. DM4VGI Template – VGI Quality

According to De Longueville et al. (2010), the contributor can provide a personal score about the accuracy or quality of his own data at the moment of the contribution. Thus the DM4VGI has fields to identify the reputation of the VGI considering, for instance, the scores provided by the users. It is important to mention that all users should be allowed to grade all contributions, in order to avoid taking into account only the opinion of the one who produced the data. The final score of a VGI could be calculated as the weighted average of all scores it received. In order to do so, the element Number of Evaluations (4.5) is necessary, since this field registers how many distinct users graded the contribution.

The users' ranking should be used, as discussed by the second approach of Goodchild (2012), which shows the importance of the hierarchy in collaborative systems. In order to calculate the final score of the contribution, the system calculates the weighted average considering the users' ranking, which is then registered in the element Final Score (4.6) of the DM4VGI. The elements Minimum Value (4.6.1) and Maximum Value (4.6.2) register the possible limits for the field Final Score.

Keeping the users' ranking is a method to help the VGI quality validation. Users who participate in a positive and constant manner should be in higher levels in the hierarchy compared with those who constantly provide erroneous contributions or untruths. A user who provides malicious, unethical or criminal contributions should be in the lowest levels and could be banned from the system. The higher the hierarchy level or position of the user grading the VGI, the greater the weight of this score in the calculus of the VGI's final score. This final score should be a weighted average considering the user's level.

The element Status (4.7) consists of keeping a record of the status of validation or judgment on the quality of the VGI, for instance, if it is still being evaluated or has already been validated by website administrators, organizations, or the users. Thus the status of the VGI can be: "Under evaluation"; "Approved"; and "Reproved".

The element Assessment Method (4.8) registers a description of the method used to assess the VGI (e.g., comparison of the VGI with other data sources, user's scores

and assessments, or the combination of one or more techniques) and who are the responsible for its final validation (e.g., website administrators, agencies or organizations, or the VGI users).

The DM4VGI is based on the suggestion by De Longueville et al. (2010) about zoom management, with an additional feature. The element Zoom Contribution (4.9) registers the zoom level of the map during the contribution. In addition, the system also captures the map zoom level at each Wiki review with the element Zoom Wiki (4.10). The zoom level can be captured automatically by some API method or function, such as the Google Maps API.

Documentation standards for geographic metadata have fields to identify those responsible for the production, update and distribution of the data along with their metadata (e.g., ISO 19115 and CSDGM). Therefore, the auditing and distribution of information should also be documented in the VGI metadata. Figure 6 shows the elements related to the auditing and distribution of the VGI. The VGI can contain several authors and editors, some even anonymous. Thus the identification element of an author should be a users' list. An "anonymous user" could be one value of the element "Name". The element Ranking (5.4) registers the name of the ranking system used, the Ranking Position (5.5) keeps its hierarchical level and the Ranking Scale (5.6) registers the hierarchical levels of the VGI ranking system.

	Element		Item	Type	Auto captured
Auditing and Distribution	Array VGI Author	Name*	5.1	String	X
		Age Group	5.2	String	X
		Email	5.3	String	X
		Ranking	5.4	Text	X
		Ranking Position	5.5	Text	X
		Ranking Scale	5.6	Text	X
		Internet Protocol*	5.7	String	X
	Array VGI Distributor	Name*	5.8	String	X
		Email*	5.9	Text	X
		Website	5.10	URI	X
		Internet Protocol*	5.11	String	X
	Link to Download or Access*		5.12	URI	X

Figure 6. DM4VGI Template – Auditing and Distribution

4. Use of dynamic metadata in the VGI Template

The processes of production and update of the metadata elements should start with data collection and continue until the last modification and/or access to the VGI. This requires a Geobrowser, a complete collaborative environment, and a VGI Documentation Module to control and monitor the users actions within the VGI system.

Metadata do not necessarily have to be provided by the user who produced the VGI. Most of the DM4VGI elements are automatically created during the processes of VGI collection and edition. Afterwards, in case of a missing metadata element, the contributor or any other user can complete the metadata in a collaborative and cooperative manner.

Figure 7 shows a suggestion of architecture for the VGI documentation in Geobrowser. The user can dynamically interact with the Geobrowser, being able to

create, edit or evaluate the data. If the user provides a new VGI, the VGI Documentation Module is responsible for the capture and generation of all necessary information registered by the user during the contribution, automatically creating most of the metadata fields. Afterwards, the contributor is required to inform the additional fields that were not filled out by the VGI Documentation Module. It is important to highlight that such step can be performed any time and by users other than the contributor.

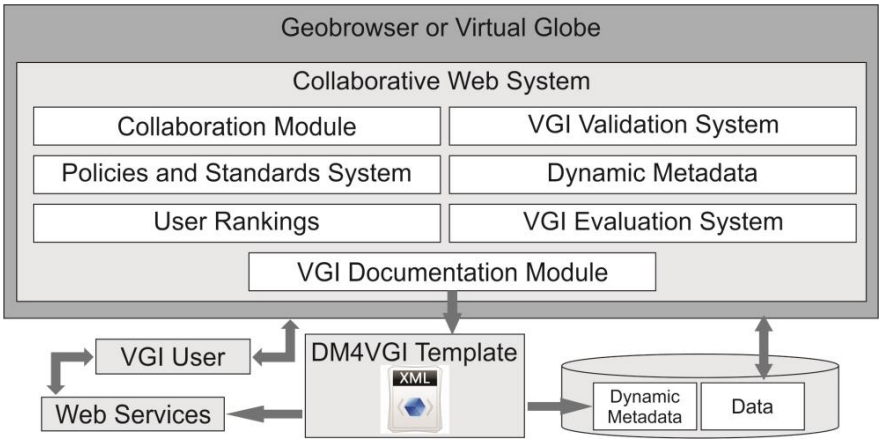


Figure 7. Architecture for VGI documentation in Geobrowser

If the user modifies an existing VGI, the VGI Documentation Module must recover the metadata related to the modified VGI, and then update or complete the necessary fields. The modifications must be done so that the metadata is compatible with the new status of the data.

The VGI Documentation Module is also responsible for saving and updating the dynamic metadata in the database, but the data are responsibility of the Geobrowser or another layer of the collaborative Web system. The web service layer is responsible for the publication of these metadata so that search mechanisms and metadata catalogs can find such information. Thus a collaborative Web system using the DM4VGI together with the proposed VGI architecture can provide greater visibility to its data and system, since other systems can have access to these metadata and the VGI and/or website that collected it.

Figure 8 shows the following quantitative comparisons: total number of elements of the DM4VGI; total number of elements automatically captured; number of mandatory elements in the DM4VGI; number of mandatory elements automatically captured; number of optional elements in the DM4VGI; number of optional elements automatically captured. It can be observed in Figure 8 that almost all elements are automatically captured, provided that this information is available in the VGI. Thus, it is possible to document the VGI with little or no demand for time and effort of the contributor. Afterwards, the elements which were not automatically filled out can be completed by other users or system administrators. A good human-computer interface can induce the user to provide part of the metadata naturally and complementary to the VGI.

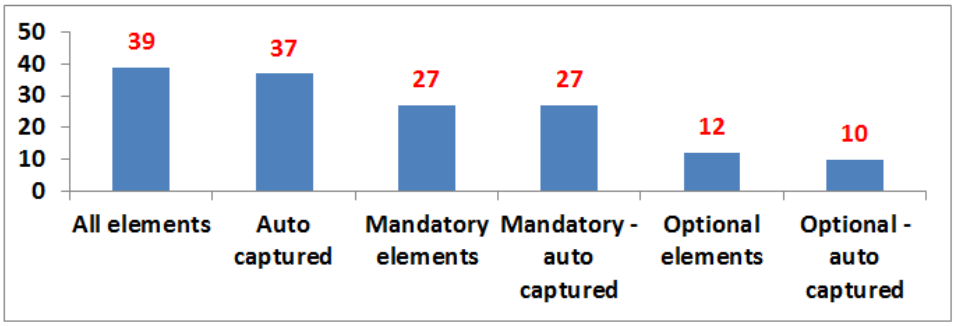


Figure 8. Number of elements of the DM4VGI

5. Conclusions

This paper proposed the DM4VGI, a template for the documentation of Volunteered Geographic Information from Geobrowsers or Virtual Globes systems. In addition to presenting auditing, descriptive and temporal information, the template also provides statistical data with respect to the use and quality of the VGI. The VGI users themselves play the role of reviewers of data and metadata, which makes documentation a completely dynamic process. In an environment where the data are dynamic, i.e., the VGI can be modified anytime by the users, the metadata should be updated as well, to guarantee coherence of the VGI documentation. Therefore, it is necessary to collect metadata in a dynamic manner to document the VGI.

This paper proposed the automatic retrieval of metadata to document a VGI using collaborative environments such as Geobrowser or Virtual Globes. Also, the paper suggested a VGI Documentation Module to control and update metadata in such environments. Since most of the metadata elements are automatically captured or calculated, the additional effort and time spent in VGI documentation are minimum.

Considering the VGI metadata database, a series of analyses about the volunteered data captured by the DM4VGI can be performed. For instance, we can obtain responses for the following questions: Which VGI had the greatest number of Wiki reviews or was more visualized? Which are the VGI of a certain type or category? What is the quality of the VGI according to its users? Has any user ever used the VGI? Has this user approved the VGI? What are the users involved and responsible for the production, update and distribution of the VGI?

The VGI metadata can be published in a system similar to the metadata catalog of an SDI. Thus the VGI can be more quickly and easily found and analyzed by humans and machines. The search and data analysis of several Geobrowsers or Virtual Globes can be integrated with many types of geographic data. Supporting the interoperability of the VGI allows the comparison of data from different systems, for instance, the crime rate of two cities that have their own collaborative system.

The DM4VGI has useful elements to aid in the process of validating the quality of the VGI. The good quality of the data provides more credibility to a collaborative system, whereas low quality data can lead the system to a complete discredit. The DM4VGI is being used in the collaborative environment CidadãoViçosa, available at <http://www.ide.ufv.br/cidadaovicosa>.

Acknowledgments
Project partially developed with funds from development agencies FAPEMIG, MCT/CNPq and CAPES. The authors also acknowledge the support of company Gapso and Funarbe.

6. Referências

Cooper, A. K., Coetzee, S., & Kourie, D. (2010) "Perceptions of virtual globes, volunteered geographical information and spatial data infrastructures", *Geomatica*, 64(1), 73-88.

Cooper, A. K., et al. (2011) "Challenges for quality in volunteered geographical information", In: Proceedings of Africa GEO 2011, Cape Town, South Africa.

De Longueville, B., Ostländer, N., & Keskitalo, C. (2010) "Addressing vagueness in Volunteered Geographic Information (VGI) - A case study". *Int. Journal of SDI Research*, v.5, 1725-0463.

Georgiadou, Y., *et al.* (2011) "Sensors, empowerment, and accountability: a Digital Earth view from East Africa", *Int. Journal of Digital Earth*, 4(4), 285-304.

Goodchild, M. F. (2007) "Citizens as voluntary sensors: spatial data infrastructure in the World of Web 2.0", *Int. Journal of SDI Research*, v.2, p. 24-32.

Goodchild, M. F., & Li, L. (2012) "Assuring the quality of volunteered geographic information", *Spatial statistics*, v.1, 110-120.

Kittur, A., & Kraut, R. E. (2008) "Harnessing the wisdom of crowds in wikipedia: quality through coordination", In: Proceedings of the ACM conference on Computer supported cooperative work, p. 37-46.

Lévy, P., & Bonomo, R. (1999) "Collective intelligence: mankind's emerging World in cyberspace", Perseus Publishing.

Maué, P. (2007) "Reputation as tool to ensure validity of VGI", In: Workshop on volunteered geographic information. VGI Specialist Meeting, Santa Barbara.

O'reilly, T. (2007) "What is Web 2.0: Design patterns and business models for the next generation of software", *Communications & strategies*, 1(17).

Schrader-Patton, C., Ager, A., & Bunzel, K. (2010) "GeoBrowser deployment in the USDA forest service: a case study". In: Proceedings of the 1st Int. Conference and Exhibition on Computing for Geospatial Research & Application (p. 28). ACM.

Souza, W. D. , et al. (2012) "Informação Geográfica Voluntária no Pantanal: um sistema Web colaborativo utilizando a API Google Maps", In: Proceedings of Simpósio de Geotecnologias no Pantanal (GeoPantanal), Bonito, MS. p. 763-772.

Turner, A. (2006) "Introduction to neogeography". O'Reilly Media, Inc.<http://brainoff.com/iac2009/IntroductionToNeogeography.pdf>. may 2013.

Vidal Filho, J. N., et al.. (2013) "Qualitative Analysis of Volunteered Geographic Information in a Spatially Enabled Society Project". In: Proceedings of Int. Conf. Computational Science and its Applications, Ho Chi Minh, Vietnam. Springer-Verlag LNCS 7973 Part III, pp.378-393.

Linked Geospatial Data: desafios e oportunidades de pesquisa

Tiago Henrique V. M. de Moura, Clodoveu A. Davis Jr.

Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG - Brasil

{thvmm,clodoveu}@dcc.ufmg.br

Abstract. *Linked Geospatial Data is a proposal for the dissemination of geographic information in an open format, based on standards adopted by the Web of Data. The semantic enrichment that comes from knowledge about relationships among spatial data is a potential source of solutions for traditional problems in geographic information retrieval, such as ambiguity resolution and the recognition of the geographic context of documents. This work discusses the applicability of linked data concepts to these problems and presents a set of challenges and opportunities for research on geographic information retrieval and related topics.*

Resumo. *Linked Geospatial Data é uma proposta para a disseminação de informações geográficas em um formato aberto, baseado em padrões da Web of Data. O enriquecimento semântico decorrente do conhecimento sobre relacionamentos entre dados geoespaciais é uma fonte potencial de soluções para problemas tradicionais em recuperação de informação geográfica, tais como resolução de ambiguidade e reconhecimento do contexto espacial de documentos. Este trabalho discute a aplicabilidade dos conceitos de* linked data *a esses problemas e apresenta uma lista de desafios e oportunidades para a pesquisa sobre recuperação de informação geográfica e tópicos correlatos.*

1. Introdução

Documentos contendo texto não estruturado são bastante comuns na Web. Esses documentos são interligados por *hyperlinks*, que podem ser interpretados como relacionamentos entre eles (Bizer *et al.* 2009). Dados podem ser organizados usando esse mesmo mecanismo de relacionamento adotado entre documentos na Web. Isso pode ajudar a enriquecê-los, tornando mais simples o processamento por máquinas, e também facilitando a integração entre diversas fontes. Essa ideia é a base da proposta denominada *Linked Data* (Berners-Lee 2006).

Dentre os dados que podem ser organizados como *linked data*, destacam-se os dados geoespaciais, por sua importância e potencial de integração. Seu uso enriquece as aplicações que envolvem dados georreferenciáveis, mas existem problemas, como a falta de fontes confiáveis e a ambiguidade de nomes de lugares. Alguns desses problemas são abordados na literatura, e soluções propostas envolvem recursos como dicionários toponímicos (*gazetteers*) e dados de contribuições voluntárias, criados e mantidos em projetos isolados. A integração dessas fontes de dados por meio dos conceitos de *linked data* no contexto geoespacial tem grande impacto potencial sobre as aplicações.

Este trabalho apresenta conceitos sobre *linked data* geoespaciais, e aborda desafios e questões para investigação sobre sua utilização em recuperação de informação geográfica. Estamos especificamente interessados no potencial suporte de *linked data*

geoespacial a problemas referentes ao reconhecimento do contexto geográfico em documentos da Web, em que frequentemente são utilizados dados de referência, como *gazetteers*. O restante do artigo está organizado da seguinte forma. A Seção 2 aborda os conceitos básicos de *linked data*. Na Seção 3 esses conceitos são expandidos para o contexto geoespacial. A Seção 4 descreve os desafios que surgem nesta nova ótica. Finalmente, as conclusões deste estudo e trabalhos futuros são apresentados na Seção 5.

2. *Linked Data*

O paradigma *linked data* pode ser definido de maneira bastante ampla como a utilização de protocolos da Web para criação de relacionamentos tipados entre diferentes fontes de dados. O objetivo é criar uma fonte de dados integrada e aberta, denominada *Web of Data*. Para que esta ideia seja aplicável, algumas premissas para publicação de dados foram estabelecidas (Berners-Lee 2006): (1) os dados precisam obedecer ao padrão RDF (*Resource Description Framework*); (2) os objetos precisam ser identificados por URIs (*Universal Resource Identifier*); (3) os dados devem estar acessíveis via protocolo HTTP; e (4) os objetos devem ser referenciados através de suas URIs.

Os dados precisam estar formatados em um padrão legível por máquinas, daí a escolha do RDF. Nesse formato, os dados são descritos formando triplas `<objetoA, predicado, objetoB>`. Assim, os objetos A e B, identificados por suas URIs, se relacionam de maneira explícita pelo predicado que compõe a tripla. Isso implica no uso de HTTP (premissa 3) para efetivar o relacionamento. A utilização de URIs como identificadores de objetos tenta simplificar a maneira de acessá-los, já que basta navegar até a URI para acessar os dados de determinado objeto. A premissa 4 é importante para interconectar as fontes de dados, criando contexto entre elas e facilitando a navegação entre os elementos que compõem a *Web of Data*.

Um exemplo de tripla RDF é indicado na Figura 1. A primeira entidade (primeira linha) representa Paris, e a segunda a França (terceira linha). O relacionamento (segunda linha) indica que Paris é uma cidade francesa.

```
<http://sws.geonames.org/2988507/,
        gn:parentCountry,
 http://sws.geonames.org/3017382/>
```

Figura 1. Exemplo de tripla RDF

Um grupo formado em janeiro de 2007 é o responsável por manter o principal projeto de criação de *linked data*, o *Linking Open Data* (LoD). O ponto de partida foi a base DBpedia, que conta com informações sobre alguns tópicos da Wikipedia (Bizer *et al.* 2009) e inicialmente integrava 12 conjuntos de dados. O projeto, que inicialmente era conduzido basicamente por pesquisadores e desenvolvedores de pequenas empresas, hoje atrai mais atenção e contava com quase 300 conjuntos de dados em 2011.

Em setembro de 2011 a *Web of Data* tinha mais de 31,5 bilhões de triplas (Tabela 1). No entanto, o número de relacionamentos entre bases de dados (*outlinks*) é relativamente baixo, pouco mais que 500 milhões de links. Isso indica que faltam referências a objetos externos, refletindo uma dificuldade na aplicação da premissa 4 da *Web of Data*. Também indica que os dados de cada fonte não estão sendo suficientemente enriquecidos pela integração a outras fontes, ou que os conjuntos de dados conteriam descrições

muito detalhadas de seus objetos, fazendo o número de triplas crescer sem o correspondente aumento nos *outlinks*.

Tabela 1. Números da Web of Data em setembro de 2011[1]

Domínio	Número de *datasets*	Triplas	%	Outlinks	%
Mídia	25	1.841.852.061	5,82 %	50.440.705	10,01 %
Geográfico	31	6.145.532.484	19,43 %	35.812.328	7,11 %
Governamental	49	13.315.009.400	42,09 %	19.343.519	3,84 %
Publicações	87	2.950.720.693	9,33 %	139.925.218	27,76 %
Múltiplos domínios	41	4.184.635.715	13,23 %	63.183.065	12,54 %
Ciências da vida	41	3.036.336.004	9,60 %	191.844.090	38,06 %
Conteúdo gerado por usuários	20	134.127.413	0,42 %	3.449.143	0,68 %
TOTAL	295	31.634.213.770		503.998.829	

3. *Linked Data* Geoespaciais: integração de fontes de dados de referência

Dados geoespaciais são comumente utilizados em sistemas de informação geográficos (SIG), mas nem sempre é fácil ou possível extraí-los para uso em outras aplicações, ou mesmo em outros SIG. *Linked data* possibilita a reutilização desses dados, facilitando o surgimento de novas aplicações e inovações.

Além das novas possibilidades criadas por *linked data*, existem desafios de recuperação de informação geográfica nos quais esse conceito pode ser muito útil. Uma classe importante de problemas envolve o reconhecimento do contexto geográfico de documentos. Nesses problemas, é necessário reconhecer nomes de lugares e outras referências espaciais no texto, mas frequentemente ocorre ambiguidade, ou seja, termos usados como nome de um lugar podem se referir a outras entidades ou a lugares homônimos. *Gazetteers* são usados como fontes de nomes válidos de lugares, permitindo o reconhecimento. Para a desambiguação, no entanto, em geral mais informação é necessária. Em um trabalho anterior (Machado et al. 2010), um gazetteer enriquecido com informação semântica foi proposto, e foi demonstrada uma técnica de desambiguação que explora o relacionamento espacial e semântico entre lugares (Machado et al. 2011).

A utilização de *linked data* nesse contexto traz diversas vantagens, relacionadas à expansão do conhecimento que se tem sobre um objeto e seus relacionamentos. Como na *Web of Data* todos os relacionamentos entre objetos são semanticamente bem definidos, estes poderiam ser utilizados no algoritmo proposto por Machado *et al.* 2011.

O *gazetteer* GeoNames, uma das primeiras fontes de dados do projeto LoD, contém atualmente mais de 8,3 milhões de referências. Tem cobertura global, porém seu nível de detalhamento é heterogêneo, não inclui muitos dados intraurbanos, e os relacionamentos entre lugares estão restritos a hierarquias de subdivisão espacial e a algumas relações de vizinhança. O *gazetteer* proposto por Machado et al. (2010), por outro lado, traz detalhamento urbano em algumas cidades brasileiras e relacionamentos semanticamente mais ricos, porém seu escopo é limitado ao Brasil. A interligação dessas fontes de

[1] http://lod-cloud.net/state/

dados, usando as técnicas de *linked data*, resultaria em uma base de conhecimento mais ampla, com melhores possibilidades de uso em problemas de desambiguação. Porém, essa interligação só pode ser realizada com segurança caso seja possível garantir a correspondência entre entidades das bases envolvidas – o que é também um problema de desambiguação e casamento de registros (*records matching*). Essa dificuldade faz parte dos desafios enfrentados atualmente para a expansão das ligações entre bases de *linked data*, no caminho de realizar todo o potencial dessa ideia. A próxima seção discute os principais desafios que identificamos para *linked data* geoespaciais

4. Desafios

A disponibilidade de grandes volumes de *linked data* geoespaciais pode ser importante para a pesquisa em recuperação de informação geográfica e para o reconhecimento de lugares em texto, a partir da integração de fontes de referência. Os tópicos a seguir descrevem alguns desafios de pesquisa nessa direção.

Manutenção e atualização de dados de *gazetteers*: contribuição voluntária. A manutenção e atualização dos dados contidos em *gazetteers* são tarefas muito custosas, principalmente se alguma intervenção manual é necessária. Uma solução para este problema é a utilização de contribuições voluntárias (conhecidas como *Volunteered Geographic Information*, VGI) (Goodchild, 2007a; McDougall, 2009). Isso é feito em algumas aplicações na Web, como o Wikimapia[2], mas é necessário integrar dados coletados por essas aplicações à *Web of Data*. Dados informados voluntariamente podem também ser usados para aplicações que funcionam em tempo real, como o monitoramento do trânsito urbano (Davis Jr *et al.* 2011). VGI associada às diretrizes do *linked data* fornece um forte alicerce para construção de sistemas capazes de resolver consultas complexas com bastante qualidade (Keßler *et al.* 2009), já que associaria o conhecimento dos usuários aos dados já existentes de uma maneira semanticamente mais rica. Porém, observe-se que dados gerados por usuários representam ainda menos de 1% das triplas existentes na *Web of Data.*

Resolução de entidades e desambiguação. Em todo tipo de aplicação ou problema de pesquisa em que são usados nomes de lugares, existem problemas ligados à desambiguação. Existe a ambiguidade *geo/geo*, ou seja, quando mais de um lugar tem o mesmo nome, e *geo/não-geo*, quando lugares usam nomes também usados para outras entidades (Amitay *et al.* 2004). Pode-se usar *gazetteers* no reconhecimento de possíveis referências a lugares, mas para resolver a ambiguidade é necessário analisar outros fatores, como a coocorrência de nomes de lugares relacionados ou a presença de termos fortemente associados a lugares citados no texto. Os *gazetteers* tradicionais, como o GeoNames, não são uma boa fonte para esses relacionamentos, porém a integração deles a fontes como a DBPedia e a *gazetteers* com conteúdo intraurbano pode prover elementos para resolver o problema. Este é um ponto em que *linked data* podem oferecer boas soluções para ampliar o conteúdo das bases de referência, dando clareza para caracterizar os lugares citados, e para prover recursos que permitam caracterizar semanticamente relacionamentos expressos nas triplas RDF. Por outro lado, a resolução de entidades e a desambiguação de nomes são importantes também para promover maior integração entre fontes na *Web of Data*.

[2] http://wikimapia.org/

***Data Fusion* e redundância.** A redundância na *Web of Data* é um problema a partir do momento em que informações sobre a mesma entidade do mundo real ocorrem em diferentes conjuntos de dados, sendo que em cada um foi criada uma URI diferente. Existe um tipo de relacionamento na *Web of Data* que caracteriza dois objetos como iguais. Entretanto, é difícil estabelecer quando os objetos de fato coincidem, pois mesmo que correspondam à mesma entidade do mundo real, podem não compartilhar a mesma representação (Jain *et al.* 2010), por exemplo variando a escala. Nesse caso, como integrar as representações? Esse exemplo ilustra a dificuldade em se conseguir aumentar, de forma segura, a quantidade de conexões entre fontes de dados. A criação de mecanismos que reduzam a redundância na *Web of Data* e melhorem a qualidade das representações é importante para o futuro das pesquisas relacionadas a *linked data*.

Dados Temporais. A utilização de triplas RDF é capaz de ajudar a estabelecer a semântica dos relacionamentos, mas existem limitações importantes, principalmente na representação de dados temporais (Erwig *et al.* 1999). No GeoNames, por exemplo, cidades possuem um atributo referente a população, mas não existe uma data de referência para esse valor, logo o dado se torna incompleto e, dependendo do estudo que está sendo feito, até mesmo inútil. Pode ser necessário acrescentar um quarto atributo às triplas para estabelecer uma referência temporal para os dados. Um exemplo representativo da necessidade de se acrescentar atributos temporais a conjuntos de dados geoespaciais é a associação de dados censitários aos polígonos dos municípios brasileiros. Como as fronteiras municipais mudam ao longo do tempo, por exemplo no evento do desmembramento de um município, existe dificuldade em modelar e implementar uma série histórica geoespacial de dados demográficos pensando em *linked data*.

Qualidade, relevância e confiança. Os dados na *Web of Data* podem ser vistos sob duas perspectivas: (1) os *datasets* isolados e (2) os objetos que compõem cada um dos *datasets*. Em ambas, a existência de métricas para avaliação de qualidade, relevância e confiança é muito importante. Cada conjunto de dados tende a abordar um nicho específico, e o próprio LoD categoriza seus *datasets*. Dessa forma, é importante criar mecanismos confiáveis de categorização para que uma nova aplicação possa escolher os conjuntos de dados que mais se aproximem de suas necessidades. Aplicações que operam sobre toda a *Web of Data*, como por exemplo mecanismos de busca, precisariam estimar a relevância dos objetos relacionados a uma consulta, estabelecendo um ranqueamento. Uma métrica que poderia ser adaptada é o PageRank (Brin and Page 1998), mas no lugar de documentos teria que ser estimada a relevância das bases, ou até mesmo dos objetos que a compõem, diante de um perfil de usuário ou aplicação.

5. Conclusões e trabalhos futuros

Linked data é um novo paradigma para a integração de dados, simplificando os esquemas de relacionamento e enriquecendo a semântica. Embora sua utilização pareça vantajosa, a tecnologia é nova e existem diversos desafios a serem explorados e áreas de atuação a serem descobertas. Este artigo representa um trabalho de pesquisa em andamento, cujo objetivo final é verificar a aplicabilidade de *linked data* geoespaciais a problemas de recuperação de informação geográfica como alternativa à construção de bases de referência. Nos interessa avaliar o compromisso entre os ganhos decorrentes da integração de fontes diversas de dados de referência e os ganhos para o processo de recuperação de informação. Nesse sentido, os desafios apresentados constituem ao mesmo tempo barreiras para a construção imediata de aplicações e oportunidades para explora-

ção de novos conceitos. Pretende-se, inicialmente, promover a integração entre o Onto-Gazetteer (Machado *et al.* 2011) e o GeoNames, e avaliar o impacto dessa integração para as aplicações em recuperação de informação geográfica que usam bases de referência. Outras atividades incluem a avaliação das linguagens de consulta propostas para *linked data* e a continuação de esforços anteriores de pesquisa no sentido da detecção de tópicos na Wikipedia em integração com os *gazetteers* (Alencar and Davis Jr 2011), de modo a obter listas de termos *não-geo* associados a lugares.

Agradecimentos

O presente trabalho foi parcialmente financiado com recursos dos projetos CEX-PPM-00518/13 (FAPEMIG), 560027/2010-9 e 308678/2012-5 (CNPq).

6. Referências

Alencar, R.O. & Davis Jr, C.A., 2011. Geotagging aided by topic detection with Wikipedia. *14th AGILE Conference on Geographic Information Science*, Utrecht, The Netherlands, 461-478.

Amitay, E., Har'El, N., Sivan, R. & Soffer, A., 2004. Web-a-Where: Geotagging Web Content. *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Sheffield, UK, 273-280.

Berners-Lee, T., 2006. *Linked Data - design issues* [online]. http://www.w3.org/DesignIssues/LinkedData.html [Accessed Aug 5, 2013].

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

Brin, S. & Page, L., 1998. The anatomy of a large hypertextual Web search engine. *Proceedings of the 7th International Conference on the World Wide Web*, Brisbane, Australia, 107-117.

Davis Jr, C.A., Pappa, G.L., de Oliveira, D.R.R. & de L Arcanjo, F., 2011. Inferring the Location of Twitter Messages Based on User Relationships. *Transactions in GIS*, 15 (6), 735-751.

Erwig, M., Güting, R. H., Schneider, M. & Michalis Vazirgiannis. 1999. Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases.*Geoinformatica* 3, 3 (September 1999), 269-296.

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

Jain, P., Hitzler, P., Yeh, P., Verma, K. & Sheth, A., 2010. Linked data is merely nore data. *AAAI Spring Symposium: linked data meets artificial intelligence.*

Keßler, C., Janowicz, K. & Bishr, M., 2009. An agenda for the next generation gazetteer: geographic information contribution and retrieval. *17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.* Seattle, Washington, 91-100.

Machado, I.M.R., Alencar, R.O., Campos Junior, R.O. & Davis Jr, C.A., 2010. An Ontological Gazetteer for Geographic Information Retrieval. *XI Brazilian Symposium on Geoinformatics*, Campos do Jordão (SP), Brazil, 21-32.

Machado, I.M.R., Alencar, R.O., Campos Junior, R.O. & Davis Jr, C.A., 2011. An ontological gazetteer and its application for place name disambiguation in text. *Journal of the Brazilian Computer Society*, 17 (4), 267-279.

McDougall, K. 2009. The potential of citizen volunteered spatial information for building SDI. GSDI 11, *Proceedings...*, Rotterdam, Netherlands.

On Building Semantically Enhanced Location-Based Social Networks

Cláudio de Souza Baptista, Luciana Cavalcante de Menezes, Maxwell Guimarães de Oliveira, Ana Gabrielle Ramos Falcão and Leandro Balby Marinho

Information Systems Laboratory – Federal University of Campina Grande (UFCG)
Av. Aprígio Veloso 882, Bloco CN, Universitário – 58.429-140
Campina Grande – PB – Brazil

{baptista, lbmarinho}@dsc.ufcg.edu.br, maxwell@ufcg.edu.br,
{lucianacm, anagabrielle}@copin.ufcg.edu.br

Abstract. *Currently, most social content sites enable users to enrich their tags with semantic metadata, such as geographic metadata in the case of Location-Based Social Networks (LBSN). However geographic metadata alone only unveils a very specific facet of a tag, leading to the need for general purpose semantic metadata. This paper introduces DYSCS – Do it Yourself Social Content Sites – a platform that combines Web 2.0 and Semantic Web technologies for assisting users in creating their own LBSN enriched with semantics. DYSCS improves information organization and retrieval and provides distinct functionalities such as multimodal interface. Because of its ontology driven architecture, DYSCS is highly reusable and interoperable.*

1. Introduction

Social content sites are Web 2.0 applications that empower ordinary users with the ability to create, edit, publish, and annotate online Web content [John 2013]. Currently, there are many social content sites, each one driven to a specific interest or domain. For example, users can contribute and share videos in YouTube, photos in Flickr, news in Digg, and musical preferences in Last.fm. With the growing popularity of these sites, the amount of shared information has increased significantly, which, in spite of leading the Web into a more open and democratic environment, put forward new challenges for organizing and retrieving information.

A popular way of organizing shared information in social content sites is to leverage the "wisdom of the crowd" approach. In this approach, typically known as collaborative tagging, users can assign a set of freely chosen keywords, called *tags*, to any given resource (or content). These tags act as indexes for resources and are used for information retrieval.

Many social content sites enable users to take into consideration the geographic context of resources through geographic annotations (or *geotagging*) [Naaman 2011]. These social content sites enhanced with geographic information raise a new concept called Location-Based Social Networks (LBSN). LBSN add a new dimension to social networks by introducing Geo-tagged user-generated media, such as texts, photos, and videos. Hence, users annotate their resources with *geotags*, which are tags associated with geographic metadata, e.g., the latitude and longitude coordinates of the resource

being annotated. LBSN are becoming increasingly common nowadays, mainly due to cameras and smartphones that come equipped with embedded GPS devices, generating high volume of georeferenced multimedia data. Moreover, many LBSN already provide tools to facilitate enhancing tags with geographic metadata, for example, through Web interfaces enriched with interactive maps. Geographic annotations enable users to better define the geographic scope of their resources, thereby improving information retrieval regarding specific locations.

A well-known drawback of using tags for organizing and retrieving information is the lack of well-defined semantics, which can lead to problems such as: polysemy, synonymy or misspellings [Chen et al. 2012]. These problems might reduce the user's ability to find information. For example, a user who is searching for the *Java* programming language may get resources about the *Java Island* instead, since both are annotated with the tag *Java* (polysemy). In order to mitigate these problems, it is necessary to ensure an annotation that enables to enrich tags with well-defined semantics. To this end, the social semantic annotations arise, which are user-generated tags enriched with machine processable semantic metadata. Enriching social annotations with semantic metadata allows humans and machines to process the meaning of tags efficiently, hence improving, eventually, their ability to find relevant information.

Most of the existing social content sites available employ social tagging of resources through either regular social annotations (i.e., *non-semantic*) or social geographic annotations. However, social semantic annotation is still rarely featured in these sites. Moreover, we observed that geographic metadata is underexploited in the LBSN, such as Flickr, for example, in which the geographic search is restricted to names of places and visualization of shared resources on a map. To better exploit the geographic metadata, spatial topological functions, such as *buffer*, *contains* and *not contains*, could be used in order to define, precisely, the geographic search scope, allowing, thus, the retrieval of more precise results. For example, a query about bookstores within a radius of 20 km from a given point, could be fulfilled through the function *buffer*, while a query about gas stations in a given region marked on the map, could be fulfilled through the spatial function *contains*.

In order to address the aforementioned problems, we introduce the DYSCS - Do it Yourself Social Content Sites – platform, that combines technologies from Web 2.0, Semantic Web and advanced exploitation of geographic metadata, for assisting users in creating LBSN. The platform uses an ontology-based approach to model LBSN and user interactions. In order to organize and retrieve information on the LBSN built with the platform, a strategy that combines social semantic and geographic annotations is used. Thus, tags can be associated with both general purpose semantics and geographic metadata.

The DYSCS platform was developed with the aim of assisting users in creating LBSN, focusing on improving the organization and retrieval of information through the use of semantic and geographic metadata. The platform is based on an underlying ontology that models the interactions between users, multimedia resources, semantic and geographic tags, and other features of the site, such as description and geographic scope. Ontologies have appealing features that can improve the overall quality of the

application such as interoperability, reusability and they are highly processable by machines [Simperl 2009].

Social content sites may be used as a communication channel between the society and government agencies. An example with this purpose may be cited, where an individual can use the DYSCS platform to create a site so that people can report street problems they find on the streets of the city of Rio de Janeiro, Brazil. The citizens, then, are able to mark on the map the places where they witnessed problems such as, graffiti, bad lighting or trash on the streets and also share images regarding the problem. An advantage of a social content site created with DYSCS is the ability to use geographical and semantic tags to organize the publications in the site and improve the information search.

The main contributions of DYSCS are: ontology built for LBSN, definition of geographic scopes of a LBSN, addition of semantic tags, addition of GeoTags, and a web-based multimodal interface.

The remainder of this paper is structured as follows. Section 2 presents the OntoDYSCS ontology. Section 3 focuses on DYSCS architecture. Section 4 addresses the use of the DYSCS platform. Section 5 discusses related work. Finally, section 6 concludes the paper and highlights further work to be undertaken.

2. ONTODYSCS: Ontology for the DYSCS Platform

This section presents OntoDYSCS, an ontology developed to serve as a model for the DYSCS platform. The main goal of this ontology is combining social semantic and geographic annotations.

A base ontology is used for creating other ontologies, as it contains concepts and instances that can serve multiple domains. According to Simperl (2009), reusing ontologies can reduce the development costs since it avoids re-implementing components already available, which could be incorporated into another ontology, after eventual minor adjustments. Thus, OntoDYSCS extended and reused several base ontologies for the representation of people, online communities, geographical elements, date, time and digital media.

To store the information of the DYSCS platform (such as the social content sites and the semantic and geographical tags) and to perform the inference on such data, the Jena Framework is used. When using Jena, a Semantic Web language needs to be chosen and a Model object created, responsible for storing a reference to the Semantic Web data. RDF, RDFS and OWL are examples of possible Semantic Web languages, and the latter was chosen for our platform.

For representing the semantic aspects of DYSCS LBSN, the MOAT (“Meaning of a Tag”) ontology, proposed by Passant and Laublet (2008), was used. MOAT is an extension of the “Tag Ontology”, which in turn, is based on the SIOC (“Semantically Interlinked Online Communities”) [Breslin et al. 2005] and FOAF (“Friend of a Friend”) ontologies. The MOAT ontology enables meaning to be added to the regular user generated tags through the *moat:tagMeaning* property (from the *Meaning class*), hence creating the semantic tag.

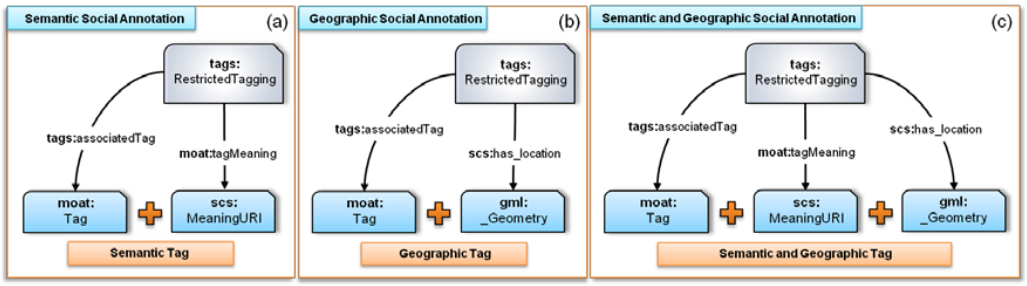


Figure 1. Representation of the semantic tag.

For modeling the geographic aspects of a LBSN, we used GeoOWL, a W3C ontology that uses the GeoRSS vocabulary for modeling classes and geographic properties. This ontology supports the concept of points (i.e., a pair of coordinates), lines (i.e., one or more pairs of coordinates), boxes (i.e., exactly two pairs of coordinates), and polygons (i.e., at least four pairs of coordinates).

The class *tags:RestrictedTagging* is used to represent a social annotation performed by a specific user to a specific resource, assigning to such resource a tag, a semantic metadata and/or a geographic metadata.

The class *tags:RestrictedTagging* is originated from the "Tag Ontology" and its main properties are:

- *tags:associatedTag:* associates a tag to the social annotation. The tag is modeled by the class moat:Tag;
- *tags:taggedResource:* identifies a resource that is being annotated. The class used to model the resource must be a subclass of owl:Thing;
- *moat:tagMeaning:* assigns a meaning to the tag used in the annotation. This meaning is represented by the class scs:MeaningURI;
- *tags:taggedBy:* identifies the person creating the annotation. The class that represents the person must be a subclass of foaf:Person; and
- *scs:has_location:* associates in the annotation a geographical location, for such, gml:_Geometry objects are created.

In OntoDYSCS, the semantic tag is derived from the social annotation. Hence, a tag is called a semantic tag if the properties *tags:associatedTags* and *tags:tagMeaning*, from the social annotation it belongs to, are filled. In this case, the ontology provides a semantic social annotation (Figure 1 (a)). The same way, the geographic tag must have in its social annotation the properties *tags:associatedTag* and *scs:has_location* - if this happens, the ontology provides a geographic social annotation (Figure 1 (b)). In the case the tag has all three properties, it is considered a semantic and geographic tag and its social annotation is called a semantic and geographic social annotation (Figure 1 (c)).

In order to allow the modeling of semantic and geographic annotation, we extend the Passant and Laublet model (*MOAT ontology*), which is based on quadruples of the form (*user, resource, tag, meaning of tags*), with geographic metadata (*GeoOWL ontology*).

Besides these ontologies, we used FOAF to represent users and their social networks, and SIOC to integrate the published information into the LBSN site.

3. DYSCS: Do it Yourself Social Content Sites Platform

This section presents DYSCS, a platform proposal for assisting users in creating LBSN that uses OntoDYSCS ontology and combines technologies from Web 2.0, Semantic Web and advanced exploitation of geographic metadata.

In DYSCS, when a user annotates a resource with a regular tag, he can further extend this annotation with semantic and geographic metadata. He can also annotate a resource using only semantic or geographic tags.

In order to create a semantic tag, the user must choose a "meaning" for the tag. For choosing tag meanings, DYSCS provides an approach based on the principles of Linked Data, in which URIs of existing resources are used to define the tag meanings. These URIs are retrieved from the Freebase database.

When the user chooses a topic from Freebase data repository, the system associates the URI related to that topic to the tag chosen by the user. Next, this semantic annotation is stored in the knowledge base of the DYSCS platform, which contains the relationships between the user, resource, the tag, and its meanings.

The assignment of geographic metadata (latitude and longitude coordinates), in order to generate the geotags, is facilitated by the mashup of Google Maps and also by the latitude and longitude properties of Freebase topics of the type *location*.

Figure 2 depicts the DYSCS interface that allows the assignment of semantic and geographic metadata. The text fields 'Location' and 'Semantic' use an auto-complete feature that displays the topics extracted from the Freebase database referring to the word typed in.

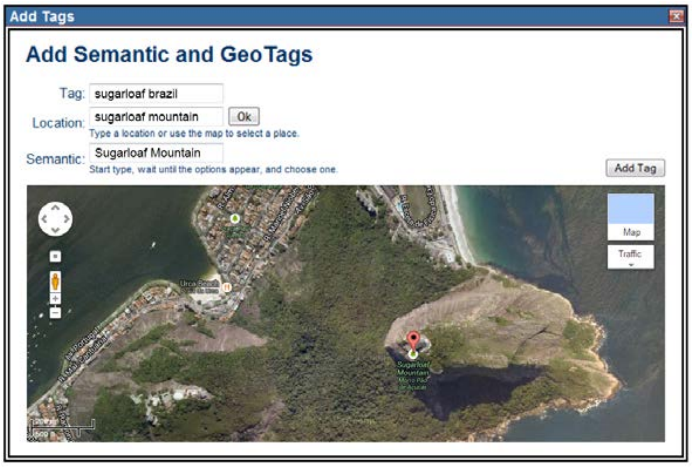


Figure 2. Interface for creating geotags, semantic tags, and the combination of both.

When the user types a word in the field 'Location', Freebase topics related to the type location are displayed. After selecting the topic, the semantic and geographic tag manager is triggered in order to retrieve the latitude and longitude coordinates of the

topic. If they are found, the point is marked on the map so the user can visualize the results.

When the coordinates of the place are not found in Freebase, the system uses the Google Geocoding API for trying to retrieve this information. However, before using the API, the names of the city and the country containing the place of interest is added to the name of place of interest. This basic information about the place of interest is retrieved from Freebase and is used aiming to increase the chance for the algorithm from Google to retrieve the coordinates of the place correctly.

For example, suppose that Freebase does not have the coordinates for 'Sugarloaf Mountain', then the names of the city and the country containing the 'Sugarloaf Mountain' is added, resulting in 'Sugarloaf, Rio de Janeiro, Brazil'. Next, using the Geocoding API, we try to retrieve the latitude and longitude for this place. Finally, if none of the aforementioned strategies work, the user can mark the coordinates of the place of interest in the map.

The algorithm developed for creating and maintaining social semantic annotations in the DYSCS platform is shown in Figure 3.

```
createDYSCsSocialAnnotation(tag, semantic, location, resource, user) {

   RestrictedTagging socialAnnotation = OntoDYSCSManipulator.createSocialAnnotation;
   socialAnnotation.addTag = tag;
   socialAnnotation.addThing = resource;
   socialAnnotation.addPerson = user;

   if (semantic != null)  then
        MeaningURI semantic = OntoDYSCSManipulator.createSemantic;

   socialAnnotation.addSemantic = semantic;

   if (location != null) then
        _Geometry location = OntoDYSCSManipulator.createLocation;

   socialAnnotation.addLocation = location;
   PlatformManipulator.addDYSCsSocialAnnotation = socialAnnotation;
}

ontoDYSCsSearch(what, wherer, who, when, tag) {
   Model ontoDYSCS = OntoDYSCSManipulator.instantiateModel;
   Query query = OntoDYSCSManipulator.createQuery(what, wherer, who, when, tag);
   Response response = OntoDYSCSManipulator.executeQuery(query, model);
   while (response.hasNext)
       // process the result to the client
}
```

Figure 3. The DYSCS Algorithm.

3.1. Architecture

The Model-View-Controller (MVC) design pattern was adopted in the development of the DYSCS architecture. Such architecture is composed of three layers: visualization, control and persistence. Figure 4 depicts the three-layer based architecture of DYSCS.

The visualization layer deals with the elements of the application that are visible to the end users, i.e., the interface between the system and the users. In DYSCS, users can access the visualization layer for either building LBSN sites or contributing with information on existing ones.

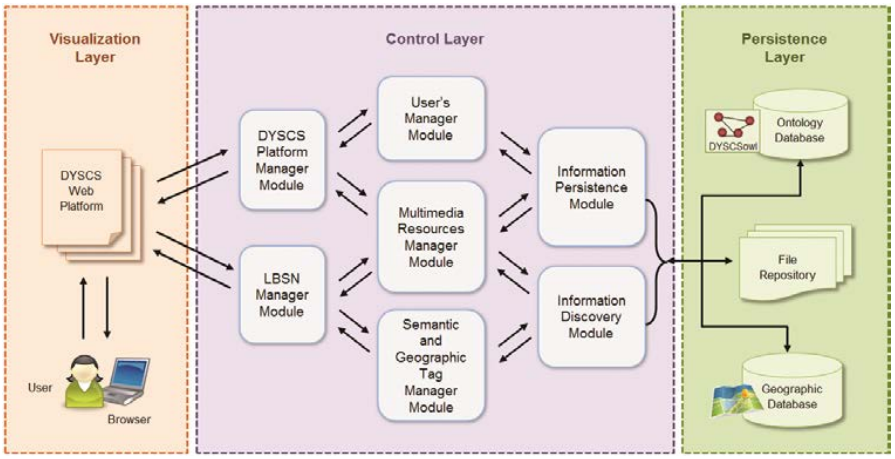


Figure 4. The DYSCS architecture.

The control layer is responsible for processing the operations required by users, and contains the logic of the system. As Figure 4 shows, the DYSCS control layer is composed of seven main modules: the platform manager, the LBSN manager, the user's manager, the multimedia resources manager, the semantic and geographic tag manager, the information persistence module, and the information discovery module.

The *platform manager module* represents the kernel of the DYSCS platform and is responsible for managing the social content sites built with the platform, and also users who register with the aim of owning a social content site. This module receives the necessary information from the visualization layer in order to create, remove, edit, or search for a site or site owner, verify the consistency of the information and pass it to the module responsible for realizing the desired action.

The *LBSN manager module* is responsible for managing the content of all the LBSN sites created through DYSCS. Moreover, it manages the searches for information contributed by users.

The *user's manager module* manages the site owners, authenticated and anonymous, and their social content sites. This module interacts with the multimedia resources manager module, in order to store the photos that identify the users; with the information persistence module, in order to persist the information about users; and with the information discovery module, used for creating, editing, and authenticating users.

The *multimedia resources manager module* manages the multimedia resources of the social content sites. This module interacts with the information persistence module in order to save the existing multimedia files in a multimedia repository. The multimedia resources manager also communicates with the information discovery module, in order to conduct searches for multimedia resources.

The *semantic and geographic tag manager module* deals with the semantic and geographic tags in the DYSCS platform. The information persistence module, in turn, is responsible for providing data persistence services. This module has communication interfaces with the DYSCS persistence layer. Finally, the searches realized in the LBSN are processed by the information discovery module, which also communicates with the persistence layer through several interfaces.

Finally, the persistence layer contains the data that will be processed and/or generated by the control layer and visualized by the user in the visualization layer. We use three data repositories in the DYSCS platform. The first one is used to store the information related to the LBSN sites, according to the OntoDYSCS ontology. The control layer communicates to this repository through the Jena Framework. The second is a file repository, used to store multimedia files shared by the users, and pages containing the templates of each existing LBSN. The third and last repository is a database created to support the spatial functions and is used to store the geographic metadata of geotags and the geographic scope of the LBSN sites. In this repository the spatial functions required by the users through the visualization layer are processed.

4. A DYSCS LBSN Instance

The creation of a LBSN site through DYSCS involves user authentication, choosing a name for the site, configuring general information about the site, defining the geographic scope, and creating map markers. The entire creation process is executed through Web interfaces and the utilization of the underlying ontology is transparent to the user. Finishing these steps, the site is ready for use.

Figure 5 presents the final interface of the LBSN site built with the purpose of sharing information about the urban problems of the city of Rio de Janeiro, Brazil. Note that there is a polygon in the map (Figure 5 (a)), which is used to delimit the geographic scope chosen for the site. Therefore, any information that contains geographic metadata must be within the geographic scope, otherwise it cannot be published into the site. The geographic scope can only be defined by the owner of the site, who can edit it anytime, however, care must be taken since changing the geographic scope can lead to loss of shared information that might be out of the limits of the new geographic scope.

For conducting tag-based search, the user must use the text field shown in Figure 5 (b). For advanced search, the user must access the respective Web page by clicking in the link ‘Advanced Search’, positioned at the right hand side of the tag-based search text field (Figure 5 (b)). The advanced search allows the user to employ the spatial operators *contains*, *not contains*, and *buffer*, in order to better filter the results, and also allows to filter resources according to the people who published it. Furthermore, it is possible to combine semantic, geographic, and social elements in the search. The results of performed searches can be visualized in the interactive map or in a new window with textual information.

For sharing information, both authenticated and anonymous users are allowed. Figure 5 (c) shows a list of map markers that can be used for sharing information directly on the map. These markers are defined by the site owner. In the specific case of the social content site depicted in Figure 5, the map markers, among others, are: *trash, public lighting, hole, graffiti, and sewage*. Figure 5 (d) depicts the sharing area and the searching for information published in the site that does not use map markers. This information can be annotated with semantic and/or geographic tags during publishing.

The members of the LBSN community can be visualized in the area shown in Figure 5 (e). Moreover, registered users can search for other users and establish friendship relations with them.

Figure 5. LBSN site created with the DYSCS platform.

For conducting advanced searches in sites created with DYSCS, users should access the advanced search interface. In this interface, users can configure several parameters in order to obtain the desired information. An user of the site 'Street problems in Rio de Janeiro' can search, for example, for the posts annotated with the semantic metadata 'Asphalt concrete' and with geographic metadata within a buffer of 2.5 km radius from the location named 'Copacabana Beach'. The results of this search are displayed in textual format and the ones among those associated with geographic metadata can be visualized in the map.

Several types of queries can be submitted. In general, users can personalize their queries by:

- *Regular Tags*: The user informs the tags that will be used by the search engine. The system returns as results all the resources that were annotated with these tags;
- *Semantic Tags*: In this option, the user must inform the tags and their associated meanings to the search interface. The system then returns all the resources annotated with these tags that are associated with the given meanings. The user can also conduct searches passing to the system only the meanings of the resources, without explicitly specifying the tags associated to these meanings. In this particular case, all the resources annotated with the tags associated with the informed meanings are returned;
- *Geographic Tags:* The user informs the tags and their geographic coordinates or the name of the place to the search interface. He can use spatial operators, such as buffer, contains, and not contains to refine the search. The result is shown in a

mashup of Google Maps. As well as with semantic tags, it is possible to conduct searches using only geographic metadata, without having to explicitly inform an associated tag;

- *Semantic and Geographic Tags*: The user can search for resources informing both semantic and geographic tags to the search interface. To this end, the user is required to fill up the fields corresponding to the tag and its semantic and geographic metadata;
- *Members and Date*: The user can search for resources based on the users (members or anonymous) who publish it, the date when the resource was shared, and also on the type of information published.

5. Related Work

Currently, there are several applications built with Web 2.0 technologies such as blogs, wikis, social network sites, social content sites and also software platforms that assist users in creating their own LBSN. Common to all these applications is the fact that the information available in the system is contributed by the users. The utilization of social annotations to promote information sharing among users is a common practice in the context of the Web 2.0 paradigm.

Marchetti et al. (2007) introduce SemKey, a semi-automatic approach to enrich tags with semantic metadata. In their work, users are responsible for assigning semantics to a tag. DYSCS uses the same principle, where users can enrich tags with semantic metadata. However, additionally, DYSCS's users can also assign geographic metadata to their tags.

Regarding the development of social networks, Web 2.0 platforms such as Ning and Elgg provide customized services for the users to create their own networks, with functionalities like multimedia sharing and discussion forums. Nevertheless, they are limited in the sense that neither of them support semantic or geographical tags. Additionally, DYSCS distinctly provides several other useful functionalities such as the definition of a geographical scope for the site, the ability to perform semantic or spatial searches on the available resources and multimodal interface.

Mahmud et al. (2010) focus on using social network and user’s context (e.g. user location), in a ‘Help-me!’ scenario in a vehicular network. They obtain timely and relevant information from close peers, considering their areas of expertise and spatial closeness.

Still regarding user's context, another relevant project is the WeGov [Wandhöfer et al. 2012], which addresses the citizens connection with governmental policy makers through popular social networks, such as Twitter and Facebook, showing the retrieved information on a map. The geographical information, though, is also not fully explored.

Bilandzic et al. (2008) present a system enabling visitors and new residents in a city to obtain the knowledge and experiences of local residents. That system aims to facilitate social navigation in urban places. Lee and Sumiya (2010) developed a geo-social local event detection system by monitoring crowd behaviors from geo-tagged microblogging sites, such as Twitter, using posts about such events. Doytsher et al.

(2010) present an integrated socio-spatial graph to monitor life patterns through the integration of social network and spatial network graphs.

Baykurt (2011) discusses the FixMyStreet LBSN, where UK citizens can report problems about their streets - such as bad pavement or lack of lighting - and this interaction is done through a map.

Santos and Furtado (2012) proposed a service-oriented architecture called SeMaps that makes it possible to generate semantic crowd maps "that have the power to perform inferences and/or access external sources that constitute useful and appropriate information to the map context". This approach also explores the semantic value of the information, other than the geographic metadata, and was integrated with Wikimapps in order to see it in action.

6. Conclusion and Future Work

Currently, there is a strong trend towards LBSN, in which users can contribute and share georeferenced content. Most of these sites still have lack of semantics of user tags, which can lead to poor annotations and retrieval of information. Furthermore, it is important to provide more sophisticated geographic search tools, which can reduce precision.

In this paper we introduced the DYSCS platform, which aims at addressing the aforementioned limitations of existing LBSN sites. DYSCS aims to help users in the creation of geo-social content sites enhanced with general purpose semantic metadata, improving the organization of the sites information and the complexity of searches.

In order to facilitate the creation of semantic metadata, we developed an interface that access the Freebase repository for retrieving URIs that define the existing resources. The creation of geographic tags was also facilitated by Freebase, through the retrieval of the latitude and longitude coordinates from topics of the type location, and the mashup of GoogleMaps, that enables users to interact with a map to define the metadata of the geotag.

As future work, we intend to incorporate trust features, concerning the shared information, into OntoDYSCS; to improve the search for textual documents through traditional keyword searching using inverted indexes of terms; and to add recommender system services in order to suggest users, resources, and tags that match user preferences. Moreover, we intend to conduct usability tests to assess the user satisfaction regarding the Web interface of the LBSN sites built with DYSCS.

7. Acknowledgments

The authors are grateful to the Brazilian Research Council - CNPQ - for funding this research.

References

Baykurt, B. (2011) “Redefining Citizenship and Civic Engagement: political values embodied in FixMyStreet.com”, In: Proceedings of the 12th Annual Conference of the Association of Internet Researchers, Seattle, USA.

Bilandzic, M., Foth, M. and Luca, A.D. (2008) “CityFlocks: Designing Social Navigation for Urban Mobile Information Systems”, In: Proceedings Conference on Designing Interactive Systems, Cape Town, South Africa, p. 174–183.

Breslin, J., Harth, A., Bojars, U. and Decker, S. (2005) “Towards Semantic Interlinked Online Communities”, In: Proceedings of the 2nd European Semantic Web Conference, LNCS, p. 500–514.

Chen, Y. A., Hwang, R. and Wang, C. (2012) “Development and Evaluation of a Web 2.0 Annotation System as a Learning Tool in an E-learning Environment”, Computers & Education, Elsevier Science Ltd., vol. 58, no. 4, p. 1094–1105.

Doytsher, Y., Galon, B. and Kanza, Y. (2010) “Querying Geo-social Data by Bridging Spatial Networks and Social Networks”, GIS-LBSN, p. 39–46.

John, N. A. (2013) “Sharing and Web 2.0: The Emergence of a Keyword”, New Media and Society, vol. 15, no. 2, p. 167–182.

Lee, R. and Sumiya, K. (2010) “Measuring Geographical Regularities of Crowd Behaviors for Twitter-based Geo-social Event Detection”, In: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, p. 1–10.

Mahmud, N., Aksenov, P., Yasar, A., Preuveneers, D., Luyten, K., Coninx, K. and Berbers, Y. (2010) “Geo-social Interaction: Context-aware Help in Large Scale Public Spaces”, In: de Ruyter, B. (Ed.), LNCS (6439), First International Joint Conference on Ambient Intelligence (AmI'10), Malaga, Spain, p. 107–116.

Marchetti, A., Tesconi, M. and Ronzano, F. (2007) “SemKey: A Semantic Collaborative Tagging System”, In: Proceedings of the WWW Workshop Tagging and Metadata for Social Information Organization, Banff, Canada.

Naaman, M. (2011) “Geographic Information from Georeferenced Social Media Data”, ACM SIGSPATIAL Special, vol. 3, no. 2, p. 54–61.

Passant, A. and Laublet, P. (2008) “Meaning of a Tag: A Collaborative Approach to Bridge the Gap Between Tagging and Linked Data”, In: Proceedings of the Linked Data on the Web (LDOW) workshop at WWW2008, Beijing, China.

Santos, H. and Furtado, V. (2012) “A Service-oriented architecture for assisting the authoring of semantic crowd maps”, In: SBIA'12 Proceedings of the 21st Brazilian Conference on Advances in Artificial Intelligence, Curitiba, Brazil, p. 32–41.

Simperl, E. (2009) “Reusing Ontologies on the Semantic Web: A Feasibility Study”, Data and Knowledge Engineering, Elsevier Science Publishers, vol. 68, no. 10, p. 905–925.

Wandhöfer, T., Van Eeckhaut, C., Taylor, S. and Fernandez, M. (2012) “WeGov Analysis Tools to Connect Policy Makers with Citizens Online”, In: tGovernment Workshop 2012, London, Great Britain.

Drainage Paths derived from TIN-based Digital Elevation Models

Henrique R. A. Freitas, Sergio Rosim, João R. F. Oliveira, Corina C. Freitas

National Institute for Space Research – Image Processing Division
PO BOX 515 – 12.227-010 – São José dos Campos – SP – Brazil

{henrique,sergio,joao,corina}@dpi.inpe.br

Abstract. *Triangular Irregular Networks efficiently define Digital Elevation Models that represent terrain surfaces and drainage paths can be calculated from these terrain models. This paper describes a method for calculating drainage paths from a triangulated irregular terrain model that was obtained from contour lines and points. Contour lines crossed by triangles edges and flat areas, which prevent path continuity, are removed by edge rotations and by inserting interpolated points into the triangulation, respectively. Drainage paths are connected by processing the triangles with an associated priority. Results achieved are consistent with an available drainage network and with real-world terrain information from a RapidEye image.*

1. Introduction

Digital Elevation Models (DEM) can be defined by Triangular Irregular Networks (TIN) in order to represent terrain models. A TIN is a very efficient terrain model as the density of information can vary from region to region in a way that more points are included where there is more elevation variation while fewer points are necessary in regions of less elevation variation avoiding data redundancy.

The triangulation is calculated from a set of points where each point is defined by its x, y coordinates on the plane and an elevation z. These points contain the main features and characteristics of the terrain and the most common triangulation used is the Delaunay triangulation that maximizes the minimum angle among all triangles thus creating less skinny triangles [De Berg et al. 2008].

In this work, all the points used as input for calculating the triangulation define contour lines and elevation points, and as the original Delaunay triangulation could produce edges crossing these contour lines, it turns out to be necessary to apply a further procedure that removes these intersections in order to modify connections between points that could result in wrong terrain features. This procedure defines a Constrained Delaunay Triangulation [Zhu and Yan 2010] where every contour line is considered as a restriction line.

Besides intersections of triangles edges and contour lines, another problem that may arise when using a TIN as a terrain model is the existence of flat triangles. These triangles define flat areas where it is not possible to determine a flow direction because all three points or vertices of each triangle have the same elevation. This problem is solved by the insertion of new points into the triangulation with interpolated elevation values in order to guarantee that every new triangle created, after a re-triangulation with these new points, has a defined flow direction and drainage paths have no discontinuities.

DEMs are very important in many research areas and they have useful applications especially in Hydrology where drainage patterns calculated from a DEM are essential in the understanding of many hydrologic processes of nature. This paper focuses on a method for calculating drainage paths in a TIN where the flow direction in a triangle is determined by a gradient vector. Results are consistent with real-world terrain information and with an available drainage network indicating that a TIN is an appropriate alternative structure for terrain modeling and hydrologic applications.

The paper is organized as follows. Next section mentions works already developed and the motivation for the theme of drainage paths derived from TIN. Section 3 contains the methodology including a description of the Constrained Delaunay Triangulation, the procedure for removing flat areas and the gradient method for tracing drainage paths. In section 4, drainage paths are compared to an available drainage network of the analyzed region where similarities between them indicate that these drainage paths represent good approximations and are also consistent with water flow patterns. Computational times took by the procedures are also given. Section 5 presents the conclusions as well as suggestions for future work. References are placed at the end.

2. Related Work and Motivation

Some authors investigated and developed techniques to calculate drainage paths directly from TIN terrain models. Many important concepts were described by [Jones et al. 1990] considering the flow direction in each triangle defined by the gradient of the plane that contains it. Another approach was developed by [Silfer et al. 1987] determining how water should be routed across the surface of a TIN distinguishing from two different conditions between TIN facets. More recently, a trickle path procedure by [Tsirogiannis 2011] traces a sequence of edges and vertices determined from intersections between points and terrain features.

The above-mentioned techniques can be added as hydrology-specific functionalities in Geographic Information Systems (GIS) as these systems are able of storing and processing a wide range of georeferenced data. Many GIS applications that process terrain models have limited capabilities when it comes to flow modeling in TIN because they require the design of more robust data structures and algorithms in order to solve problems of computational geometry so that this type of functionality is less developed than for the most common and simple DEM defined by regular grids.

TIN datasets used for terrain modeling and analysis raise many challenges in the development of efficient algorithms that can process and extract useful results from them because their use usually involves complex tasks. Computing flow-related structures on TIN such as drainage paths can present a worst-case complexity of $\Theta(n^3)$ when considering the whole river network with n triangles where this complexity is measured by the number of segments of all paths [Agarwal et al 1996].

This work addresses the problem of automatically calculating drainage paths in a TIN obtained from a dense set of points after removing inconsistencies such as contour lines crossing triangles edges and flat areas that can occur when using triangulated structures as terrain models. The aforementioned works do not define specific procedures for removing flat areas comprised of several flat adjacent triangles branching in different directions and do not specify how drainage paths can be

connected in order to make it possible to calculate accumulated flows and drainage networks.

3. Methodology

The set of triangles that defines a TIN is a good approximation to the irregularities inherent to a terrain structure. This structure can be characterized by surface-specific points and lines representing terrain features that are considered as the backbone of the surface [Fowler and Little 1979]. In the present work, a TIN is calculated by a Constrained Delaunay Triangulation algorithm from a set of points that defines contour lines and elevation points.

3.1. Constrained Delaunay Triangulation

There are several different triangulations that can be calculated from the same set of points and a good approximation used for terrain modeling is given by the Delaunay triangulation [De Berg et al. 2008]. The main property of the Delaunay triangulation is that every triangle defines a circle through its three vertices that does not contain any other point of the set inside it. This property is also considered as criteria for calculating the triangulation [Tsai 1993] which indicates that a Delaunay triangulation consists of more equiangular triangles and therefore the minimum angle among all triangles is maximized. Figure 1 shows a Delaunay triangulation calculated from a set of points and its criteria for a circle defined by the three vertices of a triangle.

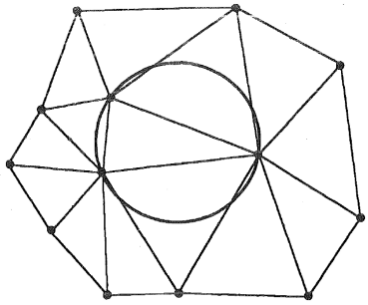


Figure 1. Delaunay triangulation criteria (modified from [Jones et al. 1990])

Many algorithms that calculate the Delaunay triangulation can be found in the literature. Some of them are: Bowyer-Watson [Bowyer 1981, Watson 1981], Incremental [Guibas et al. 1992, De Berg et al. 2008], Divide-and-Conquer [Cignoni 1998], Fortune [Fortune 1987] and Brute Force [O'Rourke 1998]. For the present work, the Incremental algorithm was used because its time complexity is $O(n \log n)$ [De Berg et al. 2008] where n is the number of points. A C++ implementation was developed because the structures and procedures from the source code could be easily modified in the future so that they work with the Terralib library [Câmara et al. 2000].

The algorithm works by initially determining a triangle that contains the set of points all inside it, then inserting each point one at a time, and when a point is inserted into the triangulation, a new triangulation is calculated with possible local changes in the current one. The algorithm also defines a tree structure for storage and search of the triangles and this structure is modified after every insertion of a point. When a triangle contains the inserted point it is divided into new triangles and this triangle division is reflected on the tree structure in a way that old and new triangles are connected by a

hierarchy link in the tree. At the end, all Delaunay triangles are leaf nodes of the tree and the initial triangle together with all its incident edges are discarded. The tree height is proportional to log(n) where n is the number of points, which determines that the search for a triangle that contains some point can be computed in logarithmic time.

It is noteworthy that if the points used as input for calculating the triangulation do not have any kind of specific connection between them, the Delaunay triangulation suffices to define a TIN as a terrain model. However, if the set of points define contour lines, as it is the case in this work, every segment of a contour line must be considered as a restriction line that cannot be crossed by a triangle edge otherwise that intersection would create inconsistencies with the terrain surface. In order to solve this problem, an initial Delaunay triangulation is calculated and a further procedure removes the intersections between contour lines segments and triangles edges. Figure 2 shows triangles edges (dashed lines) that intersect contour lines segments (solid lines) and the resulting triangulation after removing these intersections.

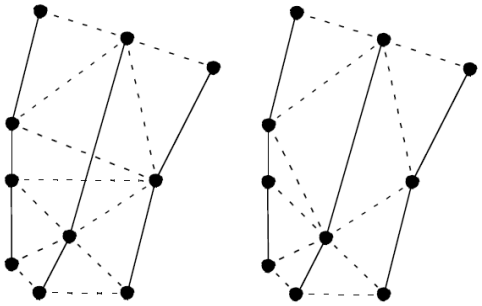


Figure 2. Triangulations before and after removing intersections (taken from [Eastman 2001])

The procedure for removing intersections initially detects for every contour line segment a triangle connected to one segment endpoint such that its edge opposite to the endpoint intersects the segment. This triangle is then processed by a search procedure that verifies each of its adjacent triangles and checks if there is one triangle not processed yet that also intersects the segment. If another triangle is found, this adjacent triangle is tested similarly and the search continues until no more intersections are found, that is, when the other segment endpoint is reached.

All triangles found in the search process are inserted into a queue structure and every pair of triangles in the queue is processed in order to remove all intersections. If an edge of a triangle in the queue intersects the segment (not in a vertex), then its adjacent triangle in the queue also intersects the segment, so their common edge is rotated and the new modified triangles are inserted back into the queue for a further verification. This procedure continues as long as there are intersections between triangles edges and contour lines segments.

3.2 Flat Areas

Flat areas are not common over terrain surfaces, except in plateau areas that consist of relatively flat terrains, but terrain models are prone to such inconsistencies as they are near-to-real approximations. TIN used as terrain models can present flat areas whenever all three vertices of a triangle have the same elevation. This situation must be avoided as

the flow direction over a flat area is undefined which turns out to be a major problem in hydrologic computations.

Every flat triangle is removed by first detecting its critical edges that are identified by two cases: a) an edge that connects two non-consecutive points in the same contour line; b) an edge connecting two points in different contour lines but of equal elevation. These two cases are illustrated in figure 3 where solid lines are contour lines and dashed lines are triangles edges with critical edges in red.

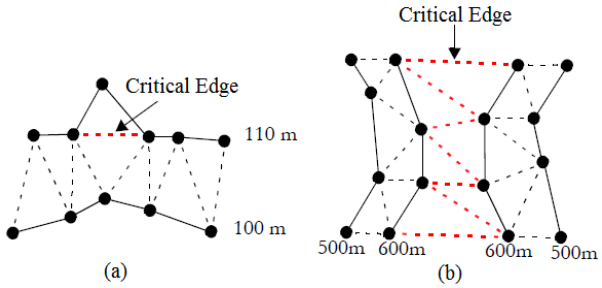


Figure 3. Flat triangles and critical edges (modified from [Eastman 2001])

The procedure for removing flat triangles inserts critical points into the triangulation that are placed exactly in the middle of critical edges assigning to each of these critical points a linearly interpolated elevation value that is calculated after a path of flat triangles is determined by a search process.

Initially, every search process defines a path by starting at corner triangles which are triangles that contain one critical edge and two edges that are non-critical, that is, either contour lines segments or edges connecting points of different elevations. The point that connects the two non-critical edges is defined as the initial point for interpolation. This search process continues always going from the current triangle to one of its adjacent triangles that share a common critical edge, following through the critical edge that contains the closest critical point in relation to the last point considered for interpolation. The search terminates when there are no more adjacent triangles to be visited (and the last point is a critical point) or the current triangle is another corner triangle (in this case, the last point has a defined elevation). All critical points found in the search process have their elevation values linearly interpolated between the elevation values of the initial and final points.

Branches found in the search process (triangles with three critical edges) are processed after the interpolation procedure has assigned an elevation value to every critical point included in the path. The search process repeats once again beginning at every branching triangle found and the same procedure is executed until all critical points have been assigned an interpolated elevation value. Finally, these critical points are inserted into the triangulation and the areas around them are then re-triangulated.

This procedure is illustrated in figure 4 where contour lines segments are dark lines with their endpoints in light green, triangles edges are in red and the critical points in magenta. Flat triangles are in light blue and corner triangles in yellow. In this example, the initial point used for interpolation is circled in red at the top corner triangle and the final point is also circled in red at the bottom. The path from the initial point to the final point following through critical edges is in dark green with branches in cyan.

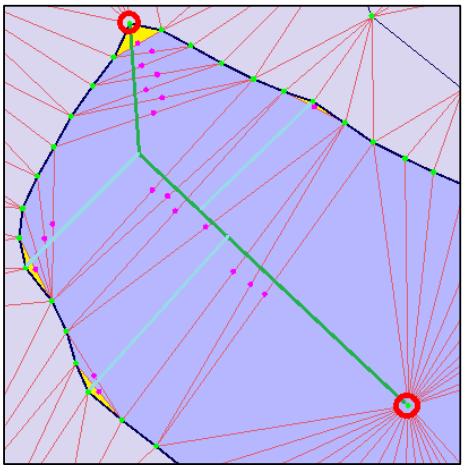


Figure 4. Paths for interpolation of critical points

As mentioned before, a linear interpolation of the critical points is performed considering both initial and final points found in the path. If the final point elevation is not defined (in the case of a critical point) then the elevation variation from the contour line that encloses the flat area in relation to its neighboring contour lines indicates whether the interpolated elevation values to be assigned to every critical point should be increasing (neighboring contour lines values are lower) or decreasing (neighboring contour lines values are higher).

3.3 Drainage Paths

Terrain models represented by TIN consist of several adjacent triangles of different sizes and shapes. Each triangle defines a plane surface that passes through its three vertices and drainage paths can be calculated from any starting point in a triangle following the path of steepest descent given by the plane gradient [Jones et al. 1990]. A plane equation and its coefficients are determined by the equations below, where each (x_i, y_i, z_i) is a triangle vertex with index i = 1,2,3:

$$Ax + By + Cz + D = 0 \tag{1}$$

$$A = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2) \tag{2a}$$

$$B = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2) \tag{2b}$$

$$C = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2) \tag{2c}$$

$$D = -Ax_1 - By_1 - Cz_1 \tag{2d}$$

Writing the plane equation (1) with z as a function of x and y, then calculating the negative gradient of this function by partial derivatives, the direction of steepest descent projected onto the xy plane is defined by equation (4) which determines the flow direction from a point in a triangle.

$$z = f(x, y) = -\left(\frac{A}{C}x + \frac{B}{C}y + \frac{D}{C}\right) \tag{3}$$

$$-\nabla f = -\left(\frac{\partial f}{\partial x}\boldsymbol{i} + \frac{\partial f}{\partial y}\boldsymbol{j}\right) = \frac{A}{C}\boldsymbol{i} + \frac{B}{C}\boldsymbol{j} \tag{4}$$

Every drainage path begins at a starting point in a triangle always following the direction given by the gradient vector of each triangle. When tracing a drainage path, different situations can occur regarding the intersections between gradient vectors and triangles edges. If the gradient vector of a triangle intersects one of its edges and the gradient vector of the adjacent triangle opposite to that edge points back to the first triangle, thus forming a channel edge, then the drainage path should continue along the edge towards the vertex of lowest elevation otherwise the path continues across the adjacent triangle.

When the intersection is exactly in a triangle vertex, then all the edges and triangles incident to that vertex are checked in order to find the lowest elevation point reached from the vertex. Each edge is first verified if it is a channel edge (both gradient vectors of the adjoining triangles by the edge point to each other) and then if the other vertex of the edge has a lower elevation. Triangles are tested by checking if there is an intersection between its gradient vector based at the current vertex and its edge that is opposite to the vertex (the gradient vector lies between the other two edges) and if this intersection has also a lower elevation. After the lowest elevation point has been found, the drainage path continues through an edge to another vertex (in the case of a channel edge) or across a triangle and the process is repeated.

Part of a drainage path can be visualized in figure 5 which contains interpolated elevation values on each plane and the path that every gradient vector follows across triangles and edges beginning at the starting point *a*.

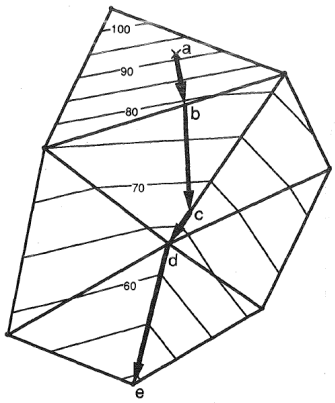


Figure 5. Path of steepest descent in a TIN (taken from [Jones et al. 1990])

The procedure described for constructing drainage paths can be applied by selecting any point as the starting point. In this work, the points selected as starting points are the triangles centroids which approximately represent the elevation of the triangles. Every starting point has its elevation considered as a priority value associated to the point defining the order in which all the points will be processed in the calculation of drainage paths. This approach indicates that it is possible to delineate potential drainage patterns by calculating drainage paths beginning at these starting points ordered from highest to lowest elevations. Another important aspect of this procedure is that when a drainage path is being traced and it reaches a triangle where another path has already been defined, then the current path is connected to the existing path. This procedure terminates after every starting point has been processed and all drainage paths have been connected.

4. Results

All results were obtained from contour lines and elevation points of an area in the city of São José dos Campos – Brazil in a geographic region with bounds 396000.0m-427400.0m West and 7421000.0m-7445000.0m South given in UTM coordinates and SAD69 projection. These UTM coordinates correspond to the geographic coordinates ranging from -46.017 to -45.709 in longitude and from -23.317 to -23.102 in latitude. The entire dataset used as input is from a database named "Cidade Viva" that is updated every 6 months and made publicly available since 2003 by the city's Geoprocessing Service of the Urban Planning Department in a format that is easily imported by a GIS.

As the main focus of this work is to calculate drainage paths from a triangulated terrain model, a TIN was defined by the Constrained Delaunay Triangulation detailed in section 3.1 and the terrain model was calculated from ~20 m *xy* resolution contour lines and elevation points with neighboring contour lines having a 5 m elevation difference represented by approximately 200000 points. Flat areas and drainage paths were processed by the procedures described in sections 3.2 and 3.3.

Figure 6 shows in blue the drainage network available from the previously mentioned database over a RapidEye image of 5 m spatial resolution for part of the total region. This drainage network is considered as the reference drainage for comparison with the drainage paths. The dashed rectangle indicates an area that is shown next in figure 7.

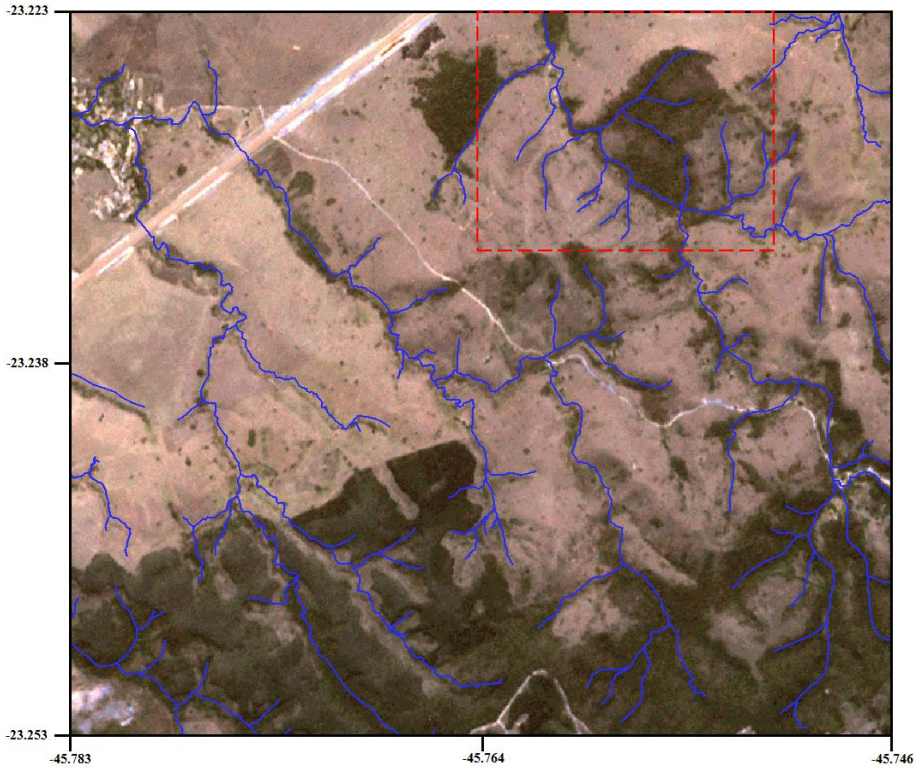


Figure 6. Drainage network from the "Cidade Viva" database over a RapidEye image

Drainage paths in cyan can be visualized in figure 7 together with the reference drainage network for the small region took from the upper-right part of figure 6 that is

highlighted by the dashed rectangle. It can be noticed that these drainage paths approximately converge to the drainage network, thus forming drainage patterns very close to the real hydrologic processes governed by the terrain surface.

Discrepancies between the two drainage patterns may be due to the precision of the input data, i.e., the contour lines and elevation points, as it can change the direction of flow from triangle to triangle. Discontinuities in the drainage paths occur by the presence of pits that are located at vertices where flow does not follow through an edge or a triangle because the gradient conditions are not satisfied. Once again, a dashed rectangle indicates another area which is detailed in figure 8 that follows in sequence.

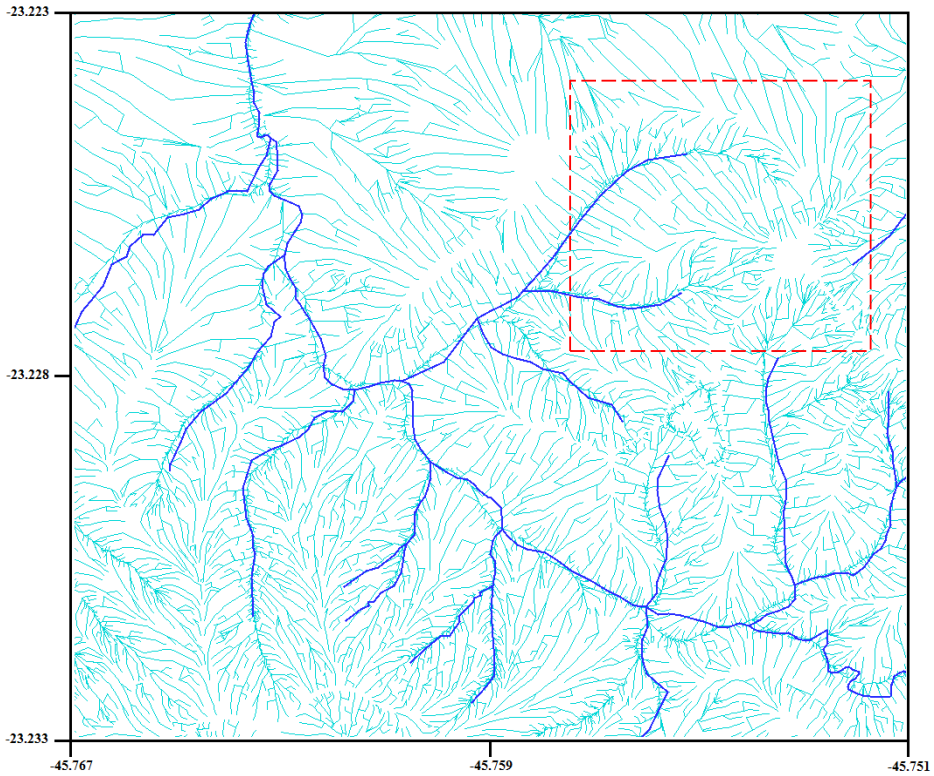


Figure 7. Drainage paths converge to the reference drainage network

For a more precise view of how the drainage paths are distributed across the triangles of the TIN used as terrain model, a closer look at both the drainage paths in cyan and the triangulation in red is given in figure 8 that contains the area bounded by the dashed rectangle of figure 7.

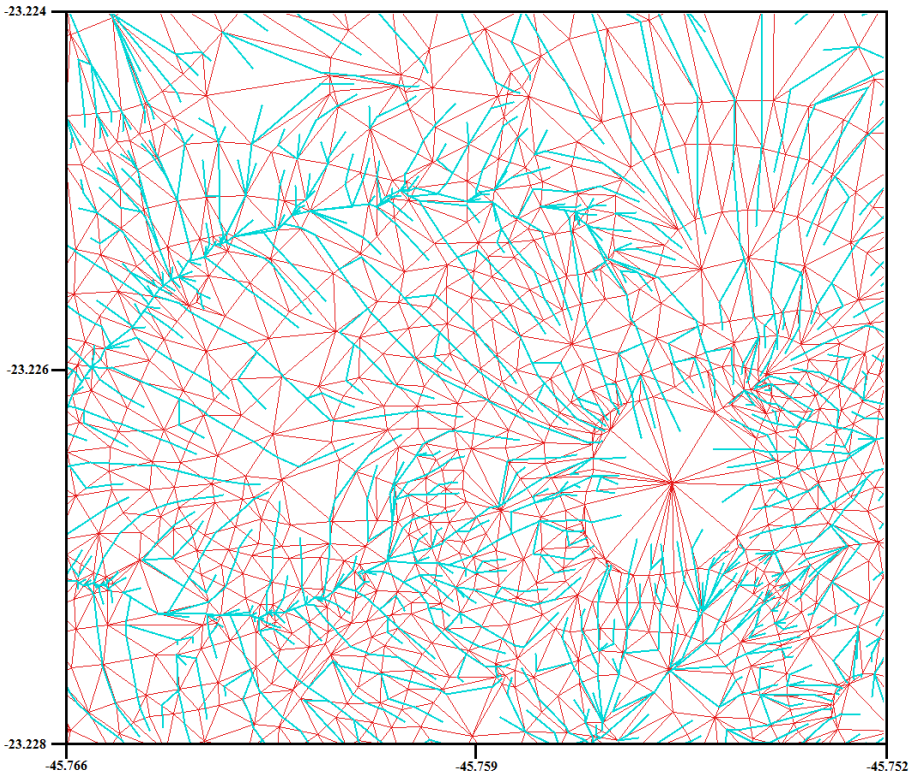


Figure 8. Drainage paths over a TIN

The primary and most significant concern to be considered when analyzing the effectiveness of the methods is the quality in the results obtained after applying all the procedures to the TIN terrain model, i.e., the drainage paths converging to streams of a drainage network, although computational times are also an important aspect related to the complexity of drainage-related structures.

The number of triangles in the final TIN and computational times took by the algorithms described in this work are given in table 1 for different numbers of input points. The total times shown below include the execution times of the procedures for removing the intersections between triangles edges and contour lines, interpolating new elevation values to the critical points in order to remove flat areas, re-triangulating the entire set of points after the addition of these new critical points into the set, calculating the plane gradient and all the drainage paths defined from each triangle. The algorithms were compiled for 64-bit and executed on a PC with Intel Core i7 2.93 GHz CPU and 8 GB of RAM memory.

Table 1. Details on TIN and execution times

Number of points	Number of triangles	Total execution time (s)
50000	148857	1.95
100000	265069	3.33
150000	396958	4.92
200000	512437	6.26

5. Conclusions and Future Work

Triangulated irregular terrain models are structures that can efficiently represent terrain surfaces. These models are calculated from terrain-specific points scattered over a region obtained from a land survey.

The algorithms and procedures developed for processing a TIN have low computational complexities which make this model an attractive alternative to other terrain models. Drainage paths following the streams of the drainage network illustrated in the previous section indicate that these patterns represent good approximations that are consistent to potential surface water flows and can be used in decision-making systems supporting studies of their impacts in hydrologic processes.

In this work, flat areas were removed by a procedure that defines a path of flat triangles and interpolates elevation values of critical points. Branches found in the path are also processed in order to complete paths previously found. The delineation of drainage paths traced by starting at each triangle centroid, ordered by their elevation values and also connected to each other result in very good water flow patterns that are consistent to real-world terrain surfaces.

Next steps to be taken in future works are careful investigations of precise definitions about the concepts of flow accumulation and contributing areas for the delineation of watersheds given by a drainage network. Pit removal must also be considered as the flow directions need to be continuous between all the triangles. Computational times could be improved by a detailed analysis and further optimizations in the algorithms.

The assignment of flow directions obtained from drainage paths to triangles and vertices in flow computation processes can produce important quantifications of water flow distribution that are essential to Hydrology.

References

Agarwal, P., De Berg, M., Bose, P., Dobrint, K., Van Kreveld, M., Overmars, M., De Groot, M., Roos, T., Snoeyink, J., Yu, S. (1996). The complexity of rivers in triangulated terrains. In *8th Canadian Conference on Computational Geometry*, pages 325–330.

Barbalić, D., Omerbegović, V. (1999). "Correction of horizontal areas in TIN terrain modeling–algorithm", http://proceedings.esri.com/library/userconf/proc99/proceed/papers/pap924/p924.htm

Bowyer, A. (1981). Computing Dirichlet tessellations. In *The Computer Journal*, pages 162–166.

Câmara, G., Souza, R. C. M., Pedrosa, B. M., Vinhas, L., Monteiro, A. M. V., Paiva, J. A., Carvalho, M. T., Gattass, M. (2000). TerraLib: technology in support of GIS innovation. In *II Brazilian Symposium on Geoinformatics, GeoInfo2000*, pages 1–8.

Cignoni, P., Montani, C., Scopigno, R. (1998). DeWall: A fast divide & conquer Delaunay triangulation algorithm in E^d. In *Computer-Aided Design*, pages 333–341.

De Berg, M., Cheong, O., Van Kreveld, M. and Overmars, M. (2008). Computational Geometry – Algorithms and Applications, Springer, 3rd edition.

Eastman, J. R. (2001). Idrisi32 Release 2 – Guide to GIS and Image Processing, Volume 2, Clark Labs.

Fortune, S. (1987). A sweepline algorithm for Voronoi diagrams. In *Algorithmica*, pages 153–174.

Fowler, R. J., Little, J. J. (1979). Automatic extraction of irregular network digital terrain models. In *ACM SIGGRAPH Computer Graphics*, pages 199–207.

Felgueiras, C. A., Goodchild, M. F. (1995). An incremental constrained Delaunay triangulation. In *NCGIA Technical Report 95-2*, pages 31–46.

Guibas, L. J., Knuth, D. E., Sharir, M. (1992). Randomized incremental construction of Delaunay and Voronoi diagrams. In *Algorithmica*, pages 381–413.

Jones, N. L., Wright, S. G., Maidment, D. R. (1990). Watershed delineation with triangle-based terrain models. In *Journal of Hydraulic Engineering*, pages 1232–1251.

O'Rourke, J. (1998). Computational Geometry in C, Cambridge University Press, 2nd edition.

Prefeitura Municipal de São José dos Campos. (2003). Base de Dados "Cidade Viva". Departamento de Planejamento Urbano, Serviço de Geoprocessamento, http://www.sjc.sp.gov.br/secretarias/planejamento_urbano/geoprocessamento.aspx (in Portuguese).

Silfer, A. T., Kinn, G. J., Hassett, J. M. (1987). A geographic information system utilizing the triangulated irregular network as a basis for hydrologic modeling. In *8th International Symposium on Computer-Assisted Cartography*, pages 129–136.

Tsai, V. J. D. (1993). Delaunay triangulations in TIN creation: an overview and a linear-time algorithm. In *International Journal of Geographical Information Systems*, pages 501–524.

Tsirogiannis, C. P. (2011). Analysis of flow and visibility on triangulated terrains. PhD Thesis. Eindhoven University of Technology.

Watson, D. F. (1981). Computing the n-dimensional Delaunay tessellation with application to Voronoi polytypes. In *The Computer Journal*, pages 167–172.

Zhu, Y. and Yan, L. (2010). An improved algorithm of constrained Delaunay triangulation based on the diagonal exchange. In *2nd International Conference on Future Computer and Communication*, pages 827–830.

Addition of the Directionality Concept in Spatial Queries on SDMSs Using the Union of the Cone-Based and Projection-Based Models

Jefferson A. Da Silva[1], Karla D. Fook[2]

[1]Maranhão State University (UEMA)
P.O. Box 15.064 – 91.501-970 – São Luis – MA, Brazil

[2]Department of Informatics, Maranhão Federal Institute for Education, Science and Technology (IFMA). Av. Getúlio Vargas, 04, Monte Castelo. CEP 65030-005. São Luis – MA, Brazil

jefferson.amarals@gmail.com, karladf@ifma.edu.br

Abstract. *This paper describes a proposed model for defining directional relationships between geometries into Spatial Data Management Systems (SDMSs) uniting the characteristics of the Cone-Based Model and the Projection-Based Model. The proposal also includes the implementation of the created model built as SDMS's extension, unlike other existing implementations which are produced in the form of external SDMS tools.*

1. Introduction

Directional relationships are strongly linked to spatial queries and spatial reasoning in general [Tang et al. 2008], and are naturally perceived by the human being. Directional relations concern the order in which geographic entities are willing.

One of the problems in dealing with directional relationships is that, unlike the case of topological relationships where there seems to be a widely accepted set of relationships [Egenhofer et al. 1990], there is no unified definition of direction relations [Theodoridis et al. 1996]. As a consequence of this lack of unification, there are several models that define the directional relationships, each with their own characteristics.

This work aims to use the Cone-Based and Projection-Based Models in order to create a hybrid conception that both utilises their respective advantages and reduces their respective limitations. Subsequently, the intention is to carry out the implementation of a framework incorporating the reasoning established, which integrates practical content into the work. The created framework is reusable, and for that, this study will implement this directly into the Spatial Data Management System (SDMS), so it can not only be used by any third party software, but also so its functionality can be merged with the existing system in the SDMS. This way, it is possible to use the functionalities related to directional relationships coupled with the functionalities geared towards topological and metric relationships, already well accepted and implemented by SDMSs, creating the possibility of hybrid spatial queries.

2. Theoretical Foundation and Related Work

2.1. Models for Definition of Directional Relationships

According to Xia et al. (2007), the basic models for defining directional relationships fall into two major categories: Cone-Based Models and Projection-Based Models. The Cone-Based Models partition the space by using lines with an origin angle α, as shown in Figure 1. Typical models include the 4-direction Model, Figure 1(a), the 8-direction model, Figure 1(b), and the triangle model, Figure 1(c) [Tang et al. 2008]. The Cone-Based Models can give an accurate identification of directional relationships in the case of point geometries, whereas misleading directional relations may be produced when reference objects are lines or polygons. [Tang et al. 2008].

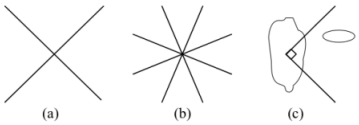


Figure 1. Cone-Based Model

The Projection-Based Models partition the space by using lines parallel to the axes [Spiros et al. 2007]. The space around an object reference *A* is partitioned into nine areas: north (N_A), northeast (NE_A), east (E_A), southeast (SE_A), south (S_A), southwest (SW_A), west (W_A) e northwest (NW_A), that refer to the cardinal and ordinal directions, and one additional region corresponding to the Minimum Bounding Rectangle (MBR) of the reference geometry (0_A), as shown in Figure 2. In this category, the MBR Model is prominent [Tang et al. 2008].

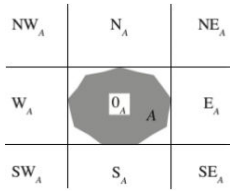


Figure 2. Projection-Based Model

The MBR Model expresses directional relationships by the relationship between the MBR of the reference object *A* and the primary object *B*. Egenhofer et al. (2000) introduces a MBR Model, which uses the 9-intersection matrix [Egenhofer and Herring 1991], projecting a grid over the concerned geometries. This model has the flexibility for the attribution of weights if an object *B* occupies the space of more than one direction using the following formula:

$$Dir_{RR}(A,B) = \begin{bmatrix} \frac{area(NW_A \cap B)}{area(B)} & \frac{area(N_A \cap B)}{area(B)} & \frac{area(NE_A \cap B)}{area(B)} \\ \frac{area(W_A \cap B)}{area(B)} & \frac{area(0_A \cap B)}{area(B)} & \frac{area(E_A \cap B)}{area(B)} \\ \frac{area(SW_A \cap B)}{area(B)} & \frac{area(S_A \cap B)}{area(B)} & \frac{area(SE_A \cap B)}{area(B)} \end{bmatrix} \quad (1)$$

Egenhofer's work formalised a method concerning the treatment of geometries that occupy more than one direction according to the projection. However, the work was theoretical, not presenting implementation, either in third party software or directly on a

SDMS. Furthermore, in the projection, as one moves away from the reference geometry, the areas of the ordinal directions become too large in comparison with the cardinal directions areas. Moreover, the MBR Model is not suitable for treating points, since they do not actually have MBR.

Zhu et al. (2012) presents a model for defining directional relationships between geometries based on Geo-Ontologies (gazetteers). In this model, the directional relationships are determined from secondary queries made on Geo-Ontologies. The Zhu's model is interesting, primarily due to the fact that adding semantics to the research enables the tapping of knowledge pertaining to the directionality in the objects represented in the ontology. However, the addition of this semantic implies the existence of data arranged in the form of ontological basis on the studied area – this could potentially result in existing spatial databases becoming incompatible with this model, if there are no ontologies regarding their spatial context in question. Furthermore, the ontological database is external to the SDMS, resulting in the need for two databases, one spatial and one ontological, separated to perform the search. Thus, integration with the existing resources of the SDMS becomes problematic and complex. The model is theoretical, being treated the implementation as a future work.

2.2. Implementation in SDMSs

The implementation of additional functionalities for Spatial Queries can be made in several ways, however the possibility of creating such features as SDMS extensions from their own source code is interesting. This approach brings benefits, among which are as follows:

- Reuse of existing SDMS resources in the extensions creation;
- Possibility of performing SQL/SF-SQL queries using a combination of the pre-existing functionalities and the functionalities added by the extensions directly in SDMS without the need for a third party software;
- Transparent utilisation of the extensions by third party software, since extensions would be incorporated into the SDMS;
- Distribution facility of the extensions since these (after compiled) would be compatible with any valid installation of the SDMS used as the source for the development.

This implementation model allows of created extensions can be positioned in the GIS communication scheme as shown in Figure 3.

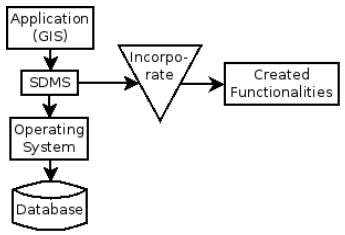


Figure 3. Illustration of the layer in which are inserted the extensions

3. Proposed Work

The proposed work contains four steps, from the conception to the implementation of the model. These steps are explained, in general terms, as follows:

Formal definition of the model: In this step, the paper formalises the proposed model that combines the features of the 8-direction Cone-Based Model and Projection-Based Model introduced by Egenhofer et al. (2000).

Elaboration of a pseudo-implementation of the functionalities: The pseudo-implementation should be generic to the point of allowing it to be translated into a variety of programming languages and on different layers – in this case, the third party application or database.

Development of functionalities in a SDMS: The implementation will be made from the source code of PostgreSQL and PostGIS. Uchoa et al. (2005) puts this set forth as a robust option for enterprise GIS implementations; a suggestion that was considered as an option for the work's implementation. PostgreSQL has also been adopted due to its flexibility in developing new modules [UCHOA et al. 2005], counting on architecture specifically designed for this purpose called PGXS – a construct that facilitates the integration and distribution of the created extensions. To use this architecture, you must create the following features: Control File, SQL Descriptor, Makefile and the Created Extension file (source or compiled), as shown in Figure 4. If the source code of the created extension is used, the Makefile should contain an entry specifying the compilation process and the compiler.

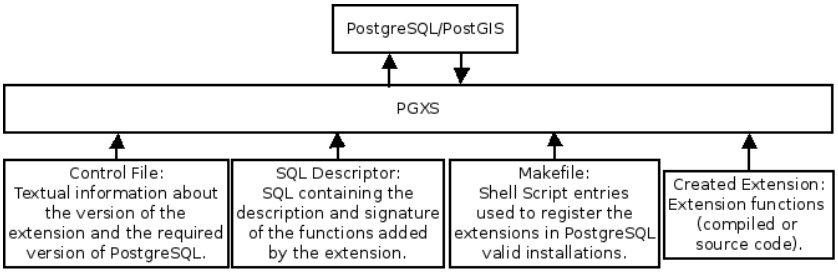


Figure 4. Requirements of the PGXS architecture

Development of a GIS module that uses the created extension: Aims to validate the functionalities added by the SDMS extension.

3.1. Current Status

In order to ascertain where the projection grid and the conic grid diverged, it was necessary to overlay the former with the latter; this procedure was used as a way to reduce the discrepancy between the existing areas in the Projection Model. In the created model, the disagreement areas were named in correspondence with Table 1, where the ordinal directions, according to the projection, are given by $P(NE_A)$, $P(NW_A)$, $P(SE_A)$ and $P(SW_A)$, and the cardinal directions according to the Cone by $C(N_A)$, $C(S_A)$, $C(E_A)$ and $C(W_A)$. The remaining areas of divergence were not considered because, if they were, it would remove an area of cardinal directions, thereby reducing the decrease of discrepancy that was achieved.

Table 1. Nomenclature of divergence areas according to the created model

Region	Name	Notation
$(P(NE_A) \bigcap C(N_A) \neq \emptyset) \vee (P(NW_A) \bigcap C(N_A) \neq \emptyset)$	*slightly_north(A)*	N_A^S
$(P(SE_A) \bigcap C(S_A) \neq \emptyset) \vee (P(SW_A) \bigcap C(S_A) \neq \emptyset)$	*slightly_south(A)*	S_A^S
$(P(NE_A) \bigcap C(E_A) \neq \emptyset) \vee (P(SE_A) \bigcap C(E_A) \neq \emptyset)$	*slightly_east(A)*	E_A^S
$(P(NW_A) \bigcap C(W_A) \neq \emptyset) \vee (P(SW_A) \bigcap C(W_A) \neq \emptyset)$	*slightly_west(A)*	W_A^S

This formalisation adds regions for the standard definition of Projection-Based Models and can be used for relations between lines and polygons. It is, however, not possible to use said model if there are relationships involving points as reference geometry.

The formula used by Egenhofer in his work also applies to the created model. This is due to the fact that, though arranged in a matrix form, the value corresponding to each cell follows the pattern "geometry area within the region in question" divided by "geometry area", thus easily fitting into the created model.

The implementation of the created model, still in the initial stage, is made through spatial functions that indicate to what extent a geometry is in a certain direction. These functions should be used in SQL queries such as "SELECT E1.name FROM state E1, state E2 WHERE stx_slightly_north(E2.geometry, E1.geometry) >= 0.2 AND NOT st_touches(E1.geometry, E2.geometry) AND E2.name ilike('GOIAS') AND E1.country_name ilike('BRAZIL');". Figure 5 illustrates the result of this query, highlighting the reference geometry, Figure 5(a), and the return of the query, Figure 5(b), along with the cone and projection grids.

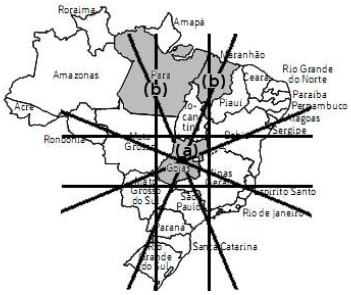


Figure 5. Query used as example shown graphically

In the previous example, it is possible to get a listing of the names of the states of Brazil that have at least one fifth of its area slightly north of the 'GOIAS' state and which do not meet it. The given example uses functions implemented by the extension (stx_slightly_north) and topological functions existing in the SDMS (ST_Touches) in the same query. The implemented functions follow the algorithm in Figure 6 in order to measure to what extent a geometric area is in a direction relating to a specific reference:

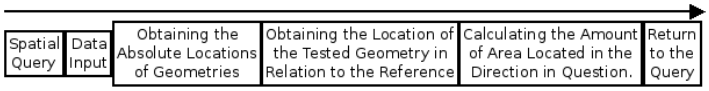


Figure 6. Generic algorithm used in implemented spatial functions

4. Final Considerations

The presented models for defining directional relationships can determine the direction accurately, but with certain limitations. Cone-Models are suitable for point geometries and Projection-Models for lines and polygons, thus opening the possibility to study their characteristics and combine them in order to provide a more global model.

In general, there is a dearth of implementations of the models. Even when performed, is generally concentrated in SDMS external tools unlike this work which focuses on an implementation incorporated within the SDMS. So far has developed a prototype that is only compatible with points as target geometries. In order to improve this work, further studies must be, and are indeed being, conducted at the moment.

References

EGENHOFER, M.; HERRING, J. Categorizing Binary Topological Relationships Between Regions, Lines, and Points in Geographic Databases. Orono, ME: Department of Surveying Engineering, University of Maine, 1991.

EGENHOFER, M.; FRANZOSA, R. On the Equivalence of Topological Relations. International Journal of Geographical Information Systems, v. 9, n.2, p. 133-152, 1995.

EGENHOFER, M., Qualitative Spatial-Relation Reasoning for Design. National Center for Geographic Information and Analysis. Department of Spatial Information Science and Engineering Department of Computer Science. University of Maine Orono, ME 044690-5711, USA 2000.

SPIROS, S., NIKOS, S., TIMOS, S., MANOLIS, K., 2007. A Family of Directional Relation Models for Extended Objects, IEEE Transactions on Knowledge and Data Engineering, 19(8), pp.1116-1129.

TANG, X., MENG, L., QIN, K., Study On The Uncertain Directional Relations Model. Based On Cloud Model. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B2. Beijing 2008.

THEODORIDIS, Y., PAPADIAS, D., STEFANAKIS, E., Supporting Direction Relations in Spatial Database Systems, Spatial data handling International symposium; 7th, Spatial data handling, ISBN: 0748405917, 1997.

UCHOA, H.N; COUTINHO, R. C.; FERREIRA, P. R.; FILHO, L.C. T.; BRITO, J.L.; Análise do módulo PostGis (OpenGIS) para armazenamento e tratamento de dados geográficos com alta performance e baixo custo, OpenGEO, XXV Congresso da Sociedade Brasileira de Computação, Canela – RS, Brasil 2005.

XIA, Y., ZHU, X., LI, D., QIN, K., Research on spatial directional relation description model, Science of Surveying and Mapping, 32(5), pp.94-97, 2007.

ZHU, X., CHEN, D., ZHOU, C., LI, M., XIAO, W., Cardinal Direction Relations Query Modeling Based on Geo-Ontology, State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXIX-B2, 2012 XXII ISPRS Congress, 25 August – 01 September 2012, Melbourne, Australia 2012.

Discovering Trajectory Outliers between Regions of Interest

Vitor Cunha Fontes[1], Lucas Andre de Alencar[1], Chiara Renso[2], Vania Bogorny[1]

[1]Dep. de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88.040-900 – Florianópolis – SC – Brazil

[2]KDD-LAB – University of Pisa
Pisa, It.

{vitor.fontes,lucas.alencar}@posgrad.ufsc.br, vania.bogorny@ufsc.br

chiara.renso@isti.cnr.it

Abstract. *Different algorithms have been proposed in the last few years for discovering different types of behaviors in trajectory data. Existing approaches, in general, deal only with the outliers, and do not consider the standards routes and regions of interest. In this paper we propose a new algorithm for trajectory outlier detection between regions of interest. We show with two experiments on real data that the method correctly finds outlier patterns.*

1. Introduction and Motivation

Current advances in mobile technology have increased the interest in mobility data analysis in several application domains. Very simple actions as carrying a mobile phone may register the trace of an object. Some devices specially developed for tracking like GPS or sensor networks may capture the movement of people, animals, cars, boats, buses and natural phenomena. These tracks are called *trajectories of moving objects*. Several data mining methods have been proposed for extracting different types of patterns from trajectories. Some examples of trajectory patterns are chasing [de Lucca Siqueira and Bogorny 2011], objects moving together in flocks [Laube et al. 2005], sequences of visited places [Giannotti et al. 2007], periodic movements [Cao et al. 2007], outliers [Lee et al. 2008], and avoidance [Alvares et al. 2011]. In this paper we focus on trajectory outlier detection.

Trajectory outliers can be very useful in traffic analysis. This type of movement analysis between regions of interest is useful to help to understand the flow of people that move between the regions, how this flow is distributed and what are the characteristics of the movements. In high traffic areas outliers can show alternative paths that can reduce the volume of cars, or reveal the best or worst path that connects two regions. Moreover, the outliers can be interesting to discover suspicious behaviors, like company cars that scape from their normal route.

Figure 1 shows some examples of trajectory outliers moving between two regions. There are six trajectories that move from region R_1 to region R_2. Trajectories T_2, T_3 and T_4 move close to each other, using a similar path (probably the same) to move from R_1 to R_2, and form the standard path. If we consider that R_1 is a Shopping center area in a city and R_2 is the downtown region, there is a high probability that T_2, T_3 and T_4 followed a common route to move between the regions, while T_1 used an alternative way in its movement. Trajectory T_1 is far from the group (T_2, T_3 and T_4), so it may characterize

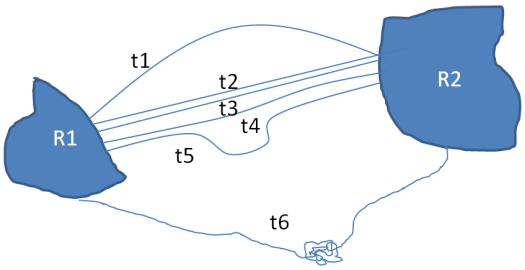


Figure 1. Examples of trajectory outliers.

an outlier in relation of the group. Trajectory T_5 took an alternative route in part of its movement (made a detour) in the middle of the way from R_1 to R_2. Trajectory T_6 made a very long detour, on the way from R_1 to R_2. By observing the movement of T_1, T_5 and T_6 we notice that these trajectories made a movement different from the rest of the trajectories (the standard path), what characterizes an outlier.

In this paper we present an algorithm to find spatial and spatio-temporal outliers between trajectories, and in summary, we make the following contributions in relation to existing approaches: (i) Define a different type of outlier pattern in trajectory data analysis, (ii) find both the standard path and the outlier patterns between regions of interest and (iii) define a new algorithm for discovering spatial and spatio-temporal outliers. It is important to mention that in this paper (at this first step) we are not interested in discovering why an object avoided a group, but to discover the main route and alternative ways to move between regions of interest.

The rest of the paper is organized as follows: section 2 presents the related works. Section 3 presents the main definitions and the algorithm. Section 4 presents experiments on real trajectory data. Finally, section 5 concludes the paper and suggests directions of future research.

2. Related Works

Several types of patterns can be extracted from trajectories. Laube in 2005 [Laube et al. 2005] proposed five types of trajectory patterns based on movement, direction, and location, which are very well known: convergence, encounter, flock, leadership, and recurrence.

Lee [Lee et al. 2008] proposed an algorithm to find outliers, which are the trajectories that move differently from the rest of the trajectories in the dataset. No regions of interest, standard path or time is considered. In [Li et al. 2007] an approach is proposed to find hot routes. These routes are discovered based on the density of the roads, and not among trajectories that move together in space between regions of interest. A similar work for discovering popular routes is proposed by [Chen et al. 2011]. This approach considers as regions of interest the origin and destination of the trajectories and hot routes are discovered based on trajectory turns. The routes where several trajectories make turns are considered popular.

A closer approach to our method could be the T-pattern [Giannotti et al. 2007]. It is a sequential trajectory pattern mining algorithm that first generates regions of interest

considering dense areas in space, and than computes sequences of visited regions, taking into account transition time from one region to another and minimum support. Although it finds the trajectories that move between regions, it does not look at the path followed by the objects, if they move together, or if there is a standard route. The basic idea of our approach is to detect if there is a standard path to move between places and to find the trajectories that avoid this path. In the following section we present the basic concepts for outlier patterns and the proposed algorithm.

3. Mining outlier patterns from Trajectories

Before defining the outlier we present some definitions like point and trajectory.

Definition 1 *Point. A point p is a tuple (x, y, t), where x and y are spatial coordinates and t is the time instant in which the coordinates were collected.*

Definition 2 *Trajectory. A trajectory T is a list of points $\langle p_1, p_2, p_3, ..., p_n \rangle$, where $p_i = (x_i, y_i, t_i)$ and $t_1 < t_2 < t_3 < ... < t_n$.*

Usually the patterns do not hold for the whole trajectory or during the complete trajectory life. Trajectory patterns occur in part of the trajectories, and this is specially true for outlier. Therefore, we make use of subtrajectories, that is a concept commonly used in trajectory research.

Definition 3 *Subtrajectories. Let $T = \langle p_1, p_2, ..., p_n \rangle$ be a trajectory. A subtrajectory S of T is a list of consecutive points $\langle p_k, p_{k+1}, ..., p_m \rangle$, where $p \in T, k \geq 1$, and $m \leq n$.*

Most existing works for trajectory pattern mining look for patterns in the whole dataset, without having a specific interest. For instance, for chasing patterns, flocks or outliers, the whole dataset is searched. When looking for *outlier patterns* in trajectory data we first look for trajectories that move around the same places. It would not make much sense to compare a trajectory that moves in Paris around Eiffel Tower with a trajectory moving around Hotel des Invalides. Trajectories should be in close areas to deviate from others. Therefore, we look for outlier patterns between *regions of interest*.

Regions of interest can have different size and format, depending on the application. Regions of interest can be districts, dense areas, hot spots, important places, etc. A region can be a pre-defined important place or computed by an algorithm that finds dense areas. How to find these regions is not the focus of this work, but we consider a region as a polygon, as in [Giannotti et al. 2007], that is a well known concept in GIS comunity.

The use of regions allows filtering from the whole dataset only the subtrajectories that move between the same regions, and outliers will be searched among these sets, what significantly reduces the search space for outlier. It is important to mention that at this point, among the trajectories that cross specific areas, we are only interested in the part of the trajectories (subtrajectories) that move between the regions, and not in the trajectory inside the area. We call these subtrajectories that move between regions as *candidates*.

We define candidate as the smallest subtrajectory that moves between two regions, i.e., we take the last point of the subtrajectory that intersects the first region and the first point that intersects the final region, as shown in Figure 2(left). In this example the candidate has the points from p_i to p_m.

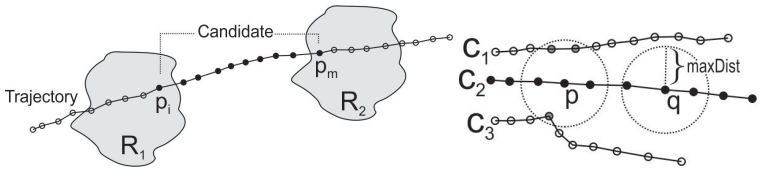


Figure 2. (left)Example of candidate (right) Example of neighborhood.

Definition 4 *Candidate. Let R_1 and R_2 be two regions such that $R_1 \cap R_2 \neq 0$ and T a trajectory. A candidate from R_1 to R_2 is the subtrajectory $S = \langle p_i, p_{i+1}, ..., p_m \rangle$ of T, where $(S \cap R_1) = \{p_i\}$ and $(S \cap R_2) = \{p_m\}$.*

After defining the set of candidates we start looking for outliers. A candidate will be an outlier when it follows a different path in relation to the majority of the candidates from its group. We can say that a path that is different from the route used by most candidates is of low density, and it has less trajectories around, while a crowded path has many trajectories in its neighborhood. In order to discover these two types of paths we introduce the concept of neighborhood, that is computed for each point of the candidate. A candidate is a neighbor of a point if it is close to the point. If a point has a few candidates in its neighborhood, then at that time the moving object was following a path different from the majority of candidates. The maximal distance for a candidate to be a neighbor of a point is called *maxDist*.

Definition 5 *Neighborhood. Let p be a point. The neighborhood of p $N(p, maxDist) = \{c_i | c_i$ is a candidate and $\exists q \in c_i, dist(p, q) \leq maxDist\}$.*

Figure 2(right) shows an example of neighborhood. The neighborhood of point p are the candidates C_1 and C_3, since these two candidates have at least one point inside the radious of size *maxDist* around p. Notice that point q has no candidates inside its radios of size *maxDist*, so its neighborhood is empty. We can conclude that at point p, C_2 was moving with C_1 and C_3 (same path), but at point q, C_2 was moving far from C_1 and C_3 (different path).

In general, there exist one or more frequent paths (main routes) to move from one region to another, and which are more frequently used than alternative ways. To find these standard paths we use the minimum support concept (*minSup*), which is the minimal amount of candidates that a point should have in its neighborhood to be part of a crowded or dense path. In the example in Figure 2(right), considering *minSup* $= 2$, the point p in candidate C_2 is in a dense path, while the point q in C_2 moves alone. A candidate that has all its points in a crowded path is considered a *standard*.

Definition 6 *Standard. Let $c = \langle p_1, p_2, p_3, ..., p_n \rangle$ be a candidate, c is a standard candidate if and only if $\forall p_i \in c, \mid N(p_i, maxDist) \mid \geq minSup$.*

The candidates that have at least one point where the cardinality of its neighborhood is less than *minSup* are called *potential outlier*. Therefore, the candidates are split in standards and potential outliers, such that a candidate will always be either a standard or a potential outlier. When all candidates between two regions are potential outlier, there is no standard. As a consequence, there is no standard path that an object could avoid or

deviate. On the other hand, if there is at least one standard path, then the potential outlier did really perform a detour, and becomes a *spatial outlier*.

An important remark here is that no outlier will exist if there is no standard path. This is one of the main difference of our approach in relation to existing works on trajectory pattern mining. So the first assumption to define an outlier is that it should move between two regions of interest. The second is that there must be a standard path that connects the regions such that the outlier should deviate from it. Therefore, any subtrajectory that uses a path different from the standard is an outlier.

Definition 7 *Outlier. Let C be the set of candidates between two regions. A potential outlier is an outlier O if $\exists c \in C | c$ is a standard.*

When two candidates leave the start region at the same time interval we can say that they are *synchronized*. For instance, when two students leave the university together to go to the cinema, we can say their trajectories are synchronized. Two candidates leave the same region at the same time interval if the difference between the timestamps of the first point of the candidates is less than a given *time tolerance*. When the trajectories in the standard path are synchronized with the outlier, then the outlier becomes *spatio-temporal*.

Definition 8 *Spatio-temporal outlier. Let C be the set of candidates between two regions. An outlier O is a spatio-temporal outlier if $\exists c \in C | c$ is standard and c is synchronized with O.*

In this work we analyze the time that the objects leave the starting region, since the objective is to know if they have a synchronized departure, and it is not relevant here if they keep the synchronization during the entire movement until reaching the destination. After defining the main concepts related to outlier patterns, we show in listing 1 the pseudo-code of the algorithm. The input of the algorithm is a set of trajectories T, a set of regions of interest R, the maximal distance (*maxDist*), the minimum support (*minSup*) and the *TimeTolerance*.

Listing 1. Algorithm

```
1  INPUT:
2  T;  // Set of trajectories
3  R;  // Set of regions
4  maxDist; // maximum distance
5  minSup;  // minimal number of neighbour
6  TimeTolerance;
7
8  OUTPUT:
9  Set of semantic spatial and spatio-temporal outliers.
10
11 METHOD:
12 FOR EACH PAR OF REGIONS (startRegion, endRegion) in R{
13     C = findCandidates(T, startRegion, endRegion); // find candidates.
14     StandardSet = findStandard(C, maxDist, minSup); // find standards.
15     IF ( StandardSet != EmpytSet ) {
16         SpatialOutSet = C - StandardSet; // Set of spatial outliers
17         FOR EACH outlier out in SpatialOutSet {
18             out.Time_granularity_refinement;
19             out.Comput_synchronized_standards(TimeTolerance);
20             IF (out.duration > avg_duration_standards)
21                 Out.is("slower outlier");
22             IF (out.duration < avg_duration_standards)
23                 out.is("faster outlier ");
24
25
26 } } }
27 return out
```

For each pair of regions (line 12), the algorithm starts by computing the candidates that move from *startRegion* to *endRegion* (line 13), with the function *findCandidates*. This function checks for every trajectory if it intersects the pair of regions. Once the candidates

are computed, the algorithm searches for the standards with the function *findStandard* (line 14), considering the parameters *maxDist* and *minSup*. The function *findStandard* checks for all points of a candidate in the set if the number of neighbors is greater than *minSup*. If this is the case, then the candidate is considered a standard. If the set of standards is not empty (line 15), then it goes for finding the spatial outliers, since there is a standard path that connects both regions.

Spatial outlier are all candidates which are not standards (line 16). Once we have the standard path and the spatial outlier, the algorithm starts the time analysis. For each spatial outlier (line 17) the algorithm discretizes the time dimension (line 18). Instead of simply showing the timestamp of the spatial and spatio-temporal outlier, as has been done in most data mining algorithms, we automatically discretize the time for the user to rapidly identify the periods of the outlier. Such discretization simplifies postprocessing steps. For this purpose, the algorithm extracts from the timestamp several information, including: the day of the week that the outlier occurred, the period of the day , and the month of the year. Such granularity refinement is useful to interpret the patterns.

It is important to notice that we first discover the patterns, and afterwards interpret them (discretize the time). If a time interval was defined a priori and the data filtered by this time interval in preprocessing steps, the method would be very limited and several patterns of previously unknown periods would never emerge. So the idea is to discover the standard path and outlier trajectories for then checking when these patterns occur.

The next step of the algorithm is to check if the outlier is synchronized with any standard, i.e., if there are standards that leave the start region at a similar time as the spatial outlier (line 19). In case there is a synchronized standard, then the spatial outlier becomes a spatio-temporal outlier. In the last step the algorithm verifies if the duration of the outlier is greater than the average duration of the standards (line 20). When the outlier is spatio-temporal, the average duration is compared only with the synchronized standards. If the duration of the outlier is greater, it means that the outlier took more time to move between the regions, and is classified as slower outlier. If its path was faster, the outlier is classified as faster outlier.

In this section we presented the main concepts related to trajectory outlier detection and presented an algorithm to find both the standard and outlier subtrajectories. The following section presents two experiments with real trajectories.

4. Experimental Results

In this section we evaluate the proposed method with two datasets with different characteristics. The first are trajectories of cars of people that leave and work in the city of Porto Alegre. It is a dense dataset, where trajectory points are collected every second. The second dataset are trajectories of taxi drivers in the city of San Francisco, and the trajectory points are collected in an average of one minute. More detailed experiments with other datasets and a better comparison with the method TRAOD can be found in [Fontes and Bogorny 2013].

4.1. Porto Alegre Dataset

This experiment considers a dataset with 241 trajectories, with a set of 197959 points. As mentioned before, the sampling rate is one second. Figure 3 shows this dataset over

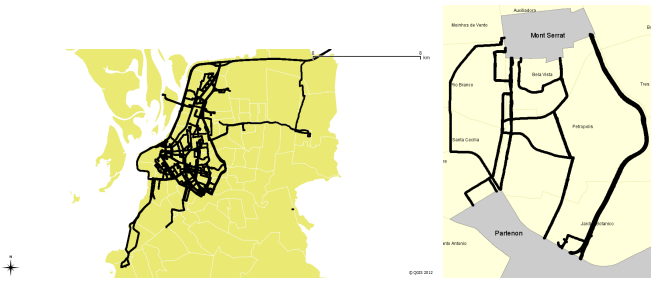


Figure 3. (left) Car Trajectories in Porto Alegre (right) Candidate trajectories between districts MontSerrat and Partenon.

a map of districts of the city. In this experiment we considered as interesting regions two districts which are crossed by the highest number of trajectories: Montserrat and Partenon. Among the 241 trajectories, 59 cross these districts, so there are 59 candidates, shown in Figure 3 (right).

This experiment was performed considering 50 and 80 meters as the *maxDist*, but we show the results for 50 meters only since the results were quite similar. Minimum support *minSup* was set to 10, indicating that at least 10 candidates should move in a distance of around 50 meters for generating a standard path. The *TimeTolerance* was set to 10 minutes, but no spatio-temporal pattern was found, because the dataset has not many synchronized trajectories.

Among the 59 candidates, 29 subtrajectories move from Partenon in direction to Montserrat, and from the 29 subtrajectories, 26 move in the standard path, which is shown in Figure 4(left), and only 3 are outlier. This shows that the standard path is used by the majority of the trajectories that move between these districts. The standard path corresponds to the Carlos Gomes Avenue, which is a popular street that crosses Porto Alegre. The average duration of the trajectories in the standard path is 6 minutes. Only two standards took more time than the average (11 and 13 minutes), and both happened at the end of the day. The majority of the standards (17) happened at morning.

We show two outlier patterns moving between these regions. Figure 4(center) shows an outlier moving from Partenon to Montserrat which was faster than the standard path, taking 4 minutes to make his trip (35% faster than the standards). As can be seen in the Figure, this path is shorter than the standard, and can be a good alternative for avoiding travel on the standard path. This pattern follows the Lucas de Oliveira Avenue. Another outlier pattern took 10 minutes in its movement, making a longer trip, as can be seen in Figure 4(right).

We compare the output found in this experiment with the TRAOD algorithm [Lee et al. 2008]. This comparison is performed to show that both methods discover different patterns, which is mainly obvious since the proposals are different. The algorithm TRAOD does not consider regions, the standard path and it does not perform any further analysis over outliers, but in order to compare the results of both algorithms we considered the same trajectory candidates as input for both methods. Different input would generate different output. TRAOD has as input the maximal distance between trajectory partitions (D), the maximal percentage of trajectories (p) for not being outliers and the fraction (F)

Figure 4. (left) standard path from Partenon to Montserrat; (center) faster outlier moving from Partenon to Montserrat; (right) slower outlier moving from Partenon to Montserrat.

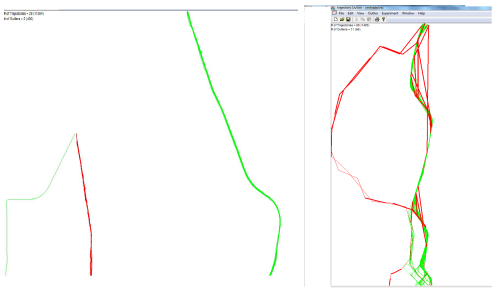


Figure 5. (left) Results for TRAOD (D=50, p=0.7, F0.2) for the POA dataset and (right) Results with parameters D=100, p=0.9, F=0.2 for the taxi dataset.

of partitions that a trajectory should have to be an outlier. We ran the TRAOD, with the parameters D = 50, p = 0.7 and F = 0.2 (these parameters are close to the parameters used in the experiment with our algorithm). We keep the original algorithm output, therefore outliers are shown in red while trajectories are shown in green, so it is not possible to overlap the output with the geographic map and show the regions. TRAOD transforms subtrajectories in lines, what makes the result a bit different. The algorithm found only 2 outliers, as shown in Figure 5(left). It is important to notice that the output of TRAOD is the total number of outliers and the outliers presented over the set of trajectories.

The size of the regions may influence the standard path and outlier patterns. However, in this paper we are only interested in the part of the trajectories that move between the regions, and not in the trajectory inside the area. The analysis of the part of the trajectory inside the region can be interesting to understand why an object avoided the standard path, but this is out of the scope of this paper. To avoid much influence of the size of the region in the patterns, the size of the regions should not be so large. In the next experiment we considered very small regions, such that the size of the region should not much influence the selected route.

4.2. Taxi trajectories in San Francisco

This experiment was performed with trajectory data collected in the city of San Francisco, California. This dataset contains trajectories of taxi drivers. We considered trajectories of one month, with 1.8 million points. One trajectory corresponds to the movement of one taxi driver during the whole day, and the time collection interval is in average 1 minute.

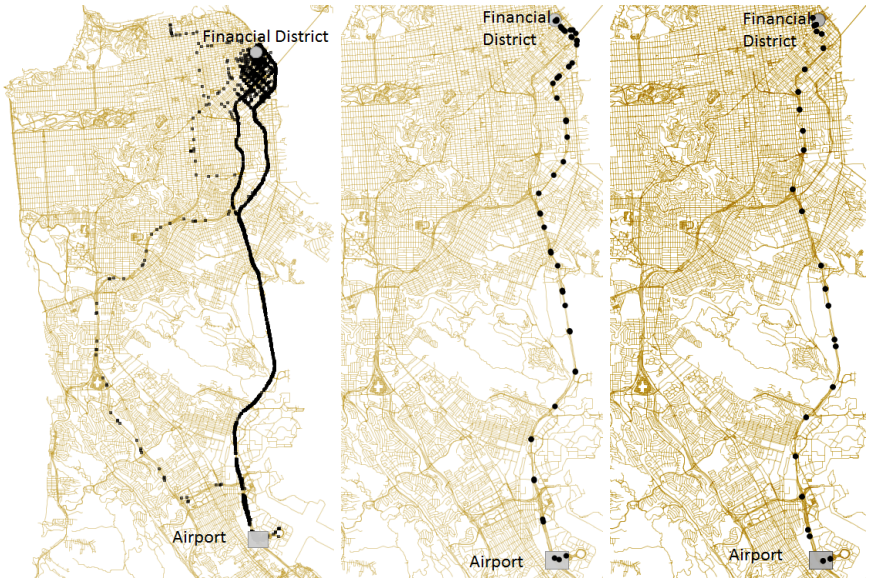


Figure 6. (left) candidates that move from airport (gray rectangle) to the financial district (gray circle), (center) standard path from Airport to Financial District, (right) standard path from Financial District to Airport.

Even with such a large time interval between every two points (which in general is every second) the method obtained very good results for the standard path and outlier patterns, what shows that the algorithm can deal with different types of trajectories.

Each taxi trajectory has an attribute *occupation*, which states if the taxi has passengers or is empty. Here we are interested in discovering the standard path and outlier patterns only when the taxi has passengers. This is interesting to discover those drivers that make detours from the main route. Therefore, we removed the trajectories with no passengers and split each trajectory of the same driver in a different one when the passenger changes. After splitting the trajectories the dataset resulted in 76.885 trajectories, having a total of 842.455 points.

In this experiment we want to analyze the movement between some specific places. We considered the trajectories of taxis moving from the airport to the Financial district in San Francisco. The parameters were distance of 100 meters for finding the neighbors, because of the large distance between the trajectory points, minimun support was set to 30 and time tolerance to 20 minutes. A total of 154 candidates was generated, what means that 154 objects traveled from the airport to the financial district. Between airport and financial district two standard paths were found, one from airport to financial area and another in the opposite direction. Figure 6 shows the candidates (left), the standard path from airport to the financial district (center) and the standard path in the opposite way (right). The average travel time on the standard path in this case is 18 minutes and has a length of 26 km. The standard path leaving the airport starts at Bayshore Freeway (US 101), changes to John F. Foran Freeway and follows to the King Street, later to Folsom Street and finally turns to Fremont Street.

Figure 7 shows different examples of outliers, where triangles are the outliers and

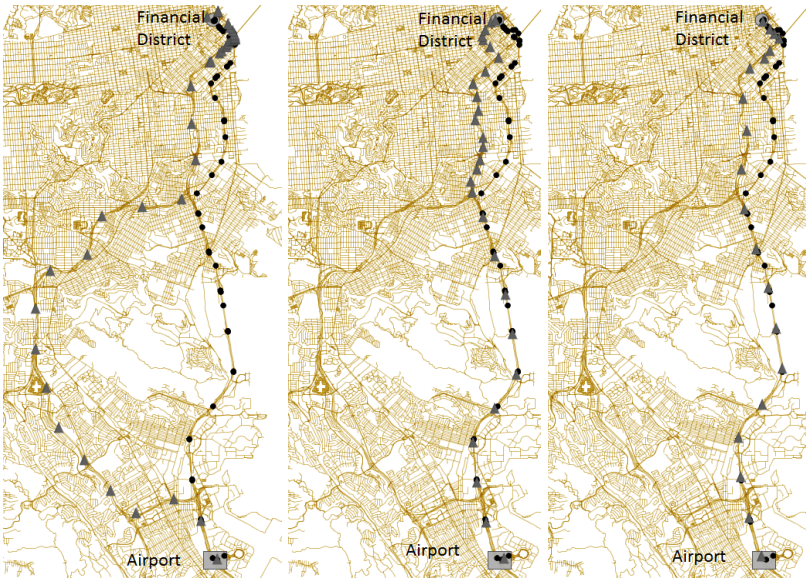


Figure 7. (left) very slow Outlier - 47 minutes (center) slow Outlier - 25 minutes (right) fastest Outlier - 17 minutes

circles the standard path. One outlier made a long detour taking 47 minutes and traveling 36 km (Figure 7 (left)). Another one also made a longer detour, taking 25 minutes and traveling 27 km (Figure 7(center)). Both trajectories that traveled a longer distance and took more time than the standard path were generated on Sunday afternoon and Saturday evening, respectively, characterizing a strange behavior for a weekend, where traffic flow should be normal. The last outlier shown in Figure 7 (right) was a little bit faster than the standard path (taking 17 minutes). This is a spatio-temporal outlier, i.e., this subtrajectory left the airport at the same time as the trajectories in the standard path.

A last analysis is on the standard path from the Financial Area to the Airport. The standard path which connects these regions is different from the previous one, as can be seen in Figure 6 (right), and is faster, taking in average 15 minutes, while the previous one takes 18. The traveled distance is the same, 26 km. Among the three examples of outlier shown in Figure 8, all examples are slower, taking respectively 30, 21, and 17 minutes, showing that for this direction the standard path is the best option. One driver made a very big detour. Two outlier trajectories partially followed the standard path, but when leaving the financial area each one took a different route, i.e., a slower one. The outlier on the right side in the figure is spatio-temporal. We compare the output found in this experiment with the TRAOD algorithm [Lee et al. 2008] that found different outliers even within the standard path (Figure 5(right)).

In general, most outlier patterns take more time to travel between the regions, and the standard path should be a better option if the user is more familiar with it.

4.3. Parameter Analysis

As in any data mining algorithm, the parameter definition is a concern, and it directly influences the results of the algorithm. The algorithm makes use of three parameters only: *maxDist*; *minSup* and *TimeTolerance*. *maxDist* is used to check if trajectories use

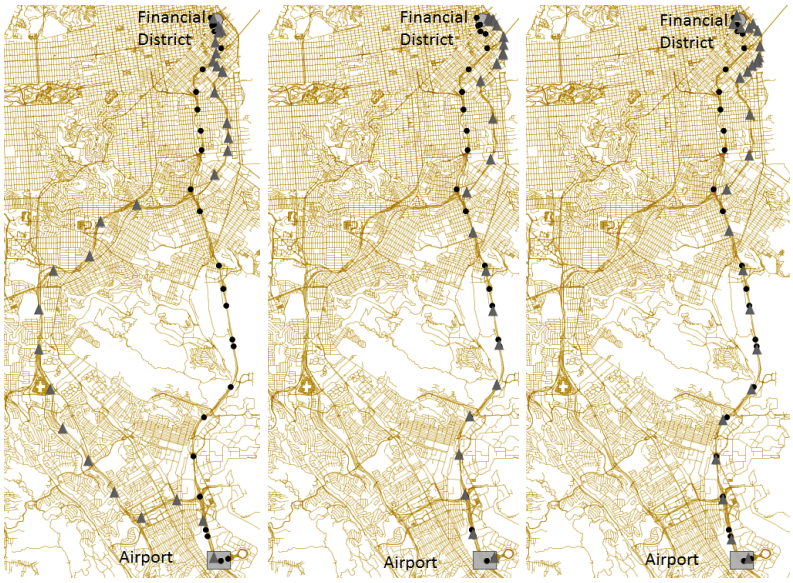


Figure 8. (left) very slow Outlier - 30 minutes (center) slow Outlier - 21 minutes (right) fastest Outlier - 17 minutes

the same path to move between regions. The best value for this parameter is the width of the streets, since outlier patterns are interesting for trajectories in cities. For instance, in cities where the average width of a street is 50 meters, $maxDist$ can be set as 80 meters, considering so 15 meters on each side of the street for GPS impreciseness. In cities with larger streets like 80 or 100 meters, $maxDist$ can be defined as 100 or even as 120 meters. It will depend on size of the streets where trajectories are collected. In narrow streets, $maxDist$ should be lower, while in larger streets it should be higher.

A small *maxDist* may split objects that move in the same path, making it more difficult to find the standards. A very high *maxDist* may join objects that move in different paths (distant paths) in the same group. Therefore, this parameter depends on the application. In our experiments in San Francisco the best parameter was 100 meters, but good results were also discovered with 80 meters.

Minimun support will depend on the density of the dataset. The higher the number of trajectories to be in the standard path, the more difficult it will be to find the standard route. A low minimal support may find several standard paths and less outliers, while a high $minSup$ will generate large amounts of outliers. The minimal support is also application dependent, so it can be high for a dataset where dense regions have several trajectories passing by. The *TimeTolerance* influences the amount of spatio-temporal outlier. The higher the *TimeTolerance* the higher is the chance for several trajectories being traveling within the time window. However, a very high *TimeTolerance* may be meaning less in the sense that trajectories should be moving together.

5. Conclusion and Future Works

In this paper we presented a method for discovering the standard path which connects regions that are interesting for an application domain and the alternative routes to move between these regions, that are called outliers. We presented the definition and an algo-

rithm to discover the outlier trajectories and the standard path. Both dimensions of space and time are considered, therefore allowing the interpretation of the outlier, like: when did it happen; which path is faster; and their duration.

The method presented in this paper is a first step towards trajectory outlier detection and interpretation, and several future works are ongoing. The first one is to distinguish the standard paths between the same regions moving in the same direction. So far we consider as standard path all standard candidates. A second one includes a deep analysis on the standard path and the use of context information around it aiming to discover the intent of the outlier. For instance, if there is a traffic jam in the standard path or an event like a police patrol, such information can help to interpret the outlier. In this method if a subtrajectory has a small portion of it which avoids the standard path it is considered an outlier. In next steps we will evaluate the use of minimal size of an outlier.

6. Acknowledgment

This work has been partially supported by EU project FP7-PEOPLE SEEK (N.295179 http://www.seek-project.eu) and the Brazilian agencies CAPES and CNPQ. Authors would like to thank also for the partial support from CNR-CNPQ Joint project 2012, DataSIM FP7-ICT-270833, and UFSC.

References

Alvares, L. O., Loy, A. M., Renso, C., and Bogorny, V. (2011). An algorithm to identify avoidance behavior in moving object trajectories. *J. Braz. Comp. Soc.*, 17(3):193–203.

Cao, H., Mamoulis, N., and Cheung, D. W. (2007). Discovery of periodic patterns in spatiotemporal sequences. *IEEE Trans. Knowl. Data Eng.*, 19(4):453–467.

Chen, Z., Shen, H. T., and Zhou, X. (2011). Discovering popular routes from trajectories. In Abiteboul, S., Böhm, K., Koch, C., and Tan, K.-L., editors, *ICDE*, pages 900–911. IEEE Computer Society.

de Lucca Siqueira, F. and Bogorny, V. (2011). Discovering chasing behavior in moving object trajectories. *T. GIS*, 15(5):667–688.

Fontes, V. C. and Bogorny, V. (2013). Discovering semantic spatial and spatio-temporal outliers from moving object trajectories. *CoRR*, abs/1303.5132.

Giannotti, F., Nanni, M., Pinelli, F., and Pedreschi, D. (2007). Trajectory pattern mining. In Berkhin, P., Caruana, R., and Wu, X., editors, *KDD*, pages 330–339. ACM Press.

Laube, P., Imfeld, S., and Weibel, R. (2005). Discovering relative motion patterns in groups of moving point objects. *International Journal of Geographical Information Science*, 19(6):639–668.

Lee, J.-G., Han, J., and Li, X. (2008). Trajectory outlier detection: A partition-and-detect framework. In *ICDE*, pages 140–149. IEEE.

Li, X., Han, J., Lee, J.-G., and Gonzalez, H. (2007). Traffic density-based discovery of hot routes in road networks. In Papadias, D., Zhang, D., and Kollios, G., editors, *SSTD*, volume 4605 of *Lecture Notes in Computer Science*, pages 441–459. Springer.

Towards efficient prospective detection of multiple spatio-temporal clusters

Bráulio Veloso[1], **Andréa Iabrudi**[1], **Thais Correa**[2]

[1] Computer Science Department

[2]Statistics Department
Universidade Federal de Ouro Preto (UFOP) – Ouro Preto, MG – Brazil

{andrea.iabrudi,thaiscorrea}@iceb.ufop.br, brauliocic091@gmail.com

Abstract. *In this paper we propose a novel technique to efficiently detect multiple emergent clusters in a space-time point process. Emergent cluster detection in large datasets is a ubiquitous task in any application area where fast response is crucial, like epidemic surveillance, criminology or social networks behavior changing. Although different authors investigate aspects of efficient spatio-temporal cluster detection, they handle either multiple or prospective detection of spatio-temporal clusters. Our work concomitantly presents a solution for both aspects: prospective and multiple cluster efficient detection in space and time. Our results with synthetic data are very encouraging, since with a wide range of parameters, we are able to detect multiple clusters in about 90% of the scenarios with very low type I and II errors (less than 2%), without increasing delay time.*

1. Introduction

This work presents a new method for accurate and computationally efficient prospective detection of multiple clusters in space-time event databases, suitable for intensive generating processes. A spatio-temporal cluster is an aggregate of points that are grouped together in space and time with an abnormally high incidence, which has a low probability to have occurred by chance alone. A process that detects such a cluster at earlier stage – an emergent or live cluster – is called surveillance system [Höhle 2007]. Surveillance system development is a ubiquitous task in any application area where fast response is crucial. These application areas include public health and safety surveillance, real life event detection from social network data, traffic control, among others.

Spatio-temporal data is increasingly available as geo-tagged procedures are more popular [Richardson 2013]. Geographic information system community are actively proposing methods to tackle with many different issues [Oliveira and Baptista 2012]: storage, information recovery, ontologies and visualization methods specific for spatio-temporal data. In particular, specialized clustering is a promising and important area for the GIScience and KDD communities [Bogorny and Shekhar 2010, Goodchild 2010]. [**?**] classify spatio-temporal types in: ST Events, Geo-Referenced Variables, Moving Objects and Trajectories. Moving Objects and Trajectories are recent in Geographic Information [Gudmundsson et al. 2012], while Geo-Referenced Variables are usual in Data Mining applications [Birant and Kut 2007] and ST Event, in Computational Statistics [Kulldorff 2001]. Our method classifies as an early distance-based ST event clustering detection procedure.

Epidemiology is traditionally a proficuous area for surveillance systems, as disease outbreak detection is a crucial task, requiring very fast reaction from public agents. Through the years, many different methods have been proposed [Marshall et al. 2007, Tango et al. 2011], investigating alternative point process hypothesis, various cluster shapes, and interaction of Spatio, Spatio-Temporal (ST) and Non-ST data. Usually, the method's quality is measured by the Type I (false clusters identification) and Type II (no detection of true clusters) errors and, for prospective detection, the delay time (elapsed time between the cluster start and its actual identification). There is no single winner method that applies for all situations and applications, and generally the underlying hypothesis are very different and sometimes restrictive. Besides that, recent increase in data availability strengthenes the need of computationally efficient approaches.

In our work, we identify whether there are one or more anomalous concentrations of point events in the very early stage. The underlying assumption is that the points are generated by a Poisson process, with rates that may vary both in space and time. The specific distribution parameters are directly estimated from the data, and may be heterogeneous and non stationary, guaranteeing broad applicability. Furthermore, there is explicit control of Type I error. Computational cost is controlled by considering only cylindrical clusters and using simple likelihood statistics to identify unexpected concentrations, as in [Li et al. 2011].

In a previous work [Veloso et al. 2012], we showed that our method is suitable to handle large volumes of spatio-temporal data to discover actual events from social network message. This work is a step towards bringing together Statistical Computing and GIScience applications, by allowing existence of multiple clusters, which is likely when the generating process is intense. Detection of multiple clusters has been investigated by spatial and spatial-temporal retrospective cluster detection community. In [Zhang et al. 2010], a sequential version of the spatial scan statistic procedure is adjusted for the presence of other clusters in the study region, by sequential deletion of the previously detected clusters, much like as our approach. In [Li et al. 2011], the authors consider the existence of multiple clusters directly by the alternative hypothesis and show better power in terms of both rejecting the null hypothesis and accurately detecting the coexisting clusters. Both methods are not suitable to emergent detection and have significant computational cost. Adopting a graph-based strategy, [Demattei and Cucala 2010] identify clusters by linking the events closest than a given distance and thus defining a graph associated to the point process. The set of possible clusters is then restricted to windows including the connected components of the graph. This allow detection of multiple, arbitrary shaped clusters with relatively efficiency. There is no extension for prospective detection, as far as we know.

Our main contribution in this paper is to extend the method in [Assunção and Correa 2009] to identify multiple clusters, using a sequential strategy of deleting some events of already identified clusters. The proposed method is evaluated for a wide range of synthetic scenarios, simulating rare to intensive processes, where two spatially separated clusters are inserted. In this controlled setup, our results are very promising, with more than 93% of correct detections. We first present the original method, followed by our extension to couple with multiple clusters. The metrics used to evaluate the method are detailed and results are then presented and discussed.

Final remarks and possible future directions close the article.

2. Method

We extended the method proposed by [Assunção and Correa 2009] for detection of multiple space-time clusters. In section 2.1, the original method, designed for an unique space-time cluster detection is briefly presented. We state our extension in section 2.2, proposing the algorithms for it.

2.1. Review: on line detection of an unique space-time cluster

Consider a point process observed in a three dimensional area $A \times (0, T]$ where A represents the space and $(0, T]$ is the time. The original method looks for a live space-time cluster with cylindric shape. The radius ρ of the circular base must be specified by the user. The method consist on monitoring a simple statistic that doesn't depend on the marginals space and time intensities of events. When this statistic exceeds a threshold the method rings an alarm and the detected cluster is identified.

Events are sequentially observed at times $t_1, t_2, \ldots$ and they are processed as soon as they happen. The spatial coordinates for an event observed at time t_i are (x_i, y_i). Let $C_{k,n} \in A \times (0, T]$ be a cylinder with circular base spatially centered at (x_k, y_k). The height of $C_{k,n}$ is given by $t_n - t_k$, where t_n is the time of the last observed event. Note that, considering t_n as the current time, $C_{k,n}$ is a live cylinder, since it reaches t_n. Let $N(C_{k,n})$ be the number of events inside the cylinder $C_{k,n}$, we assume that it follows a Poisson distribution, $N(C_{k,n}) \sim \text{Poisson}(\mu(C_{k,n}))$. The Poisson distribution is broadly used in methods of cluster detection due to its suitability for modeling count data. If there is no cluster, no space-time dependency exists, implying that space-time event intensity $\lambda(x, y, t)$ is separable and may be written as the product of the space and time marginal intensities:

$$\lambda(x, y, t) = \mu\, \lambda_s(x, y)\, \lambda_t(t), \qquad \mu = \int_A \int_{(0,T]} \lambda(x, y, t)\, dt\, dx\, dy.$$

However, if a cluster $C_{k,n}$ emerges at time t_k, dependency change in distribution is captured by a constant $\varepsilon > 0$ such that

$$\lambda(x, y, t) = \mu\, \lambda_s(x, y)\, \lambda_t(t)\, (1 + \varepsilon I_{C_{k,n}}(x, y, t))\,,$$

where $I_{C_{k,n}}$ is an indicator function for $(x, y, t) \in C_{k,n}$ and ε represents the intensity increase inside the cluster. The excess ε is a method parameter specified by the user.

The test statistic compares the null hypothesis of no cluster existence against a localized cylindrical cluster alternative. Let L_∞ be the likelihood of the spacetime Poisson process when there is no cluster and let L_k be the likelihood of this same process when there is a cluster $C_{k,n}$, both for n observed events. The test statistic is the sum of the likelihood ratio over all possibilities for the cylinder $C_{k,n}$:

$$R_n = \sum_{k=1}^{n} \frac{L_k}{L_\infty} = \sum_{k=1}^{n} \Lambda_{k,n} = \sum_{k=1}^{n} (1 + \varepsilon)^{N(C_{k,n})}\, \exp(-\varepsilon\, \mu(C_{k,n}))\,.$$

The method uses a non parametric estimate for the mean $\mu(C_{k,n})$, as in practice it is unknown. Assuming that $\lambda(x, y, t)$ is separable, our method estimates this quantity by:

$$\hat{\mu}(C_{k,n}) = \frac{N(\ B(k,\rho) \times (0, t_n]\ )\ N(\ \mathcal{A} \times (t_k, t_n]\ )}{n}\ ,$$

where $N(\ B_{k,\rho} \times (0, t_n]\ )$ is the number of events inside the circular base of cylinder $C_{k,n}$ irrespective of time, $N(\mathcal{A} \times (t_k, t_n])$ is the number of event between time t_k and t_n irrespective of space, and n is the total number of events.

When the test statistic exceeds a threshold A, the method rings an alarm indicating there is evidence of a live cluster. As the test statistic is a sum over all possible live clusters, the estimate of the detected live cluster is the one with largest contribution to the test statistic. That is, if $R_n > A$, then $\Lambda_{k*,n} = \max\{\Lambda_{k,n}, 1 \leq k \leq n\}$ and the estimated cluster is $C_{k*,n}$.

Assume that there is a cluster starting at time t_k. If the test statistic exceeds the threshold A at some time $t < t_k$, then it is a false alarm. Otherwise, if threshold A is exceeded at time $t > t_k$, it is a motivated alarm. Surveillance systems must address the trade-off between fast detection and low false alarm rate. In our method, this is accomplished by setting the threshold A to be equal to the desired value of the Average Run Length (ARL), the expected number of events before a false alarm. It ensures that, on average, the user will wait at least A events before a false alarm, expressing user tolerance to wrong alerts.

2.2. Our extension for simultaneous space-time clusters

Now suppose we are using the method above and the alarm rings. It means there is evidence of a live cluster. In this case, is there evidence of a second live cluster? What about a third live cluster? We answer these questions with our extension for the situation where more than one cluster start at the same time t_k. The radius and the increase in the intensity for all clusters are the same: ρ and ε, respectively. For simplicity, we describe the extension for two clusters, but it can be generalized for any number of simultaneous clusters.

In the original method with a threshold A, consider there is an alarm at time t_n. The estimated cluster is $C_{k*,n}$. Our sequential approach consists in deleting the excess of events inside $C_{k*,n}$ in a random way and reapplying the original method to this new reduced database. The excess of events is the difference between the observed and the expected number of events: $\Delta(C_{k*,n}) = N(C_{k*,n}) - \hat{\mu}_{k*,n}$.

After deletion of the exceeding events, we have a reduced database. We then evaluate the test statistic at the current time t_n for this new reduced database. We will refer to the statistic for the reduced database as R'_n to distinguish from R_n. Since we artificially erased the cluster $C_{k*,n}$, if R'_n is more than a new threshold it is due to a second live cluster, it is hoping one different from $C_{k*,n}$. The estimate for this second cluster is the one with largest contribution to the test statistic R'_n, just as in the original method. The new threshold A' is equal to the old one, except for the events we delete: $A' = A - \Delta(C_{k*,n})$.

Algorithms 1, 2, 3 and 4 show our extension.

Algorithm 1 Simultaneous Spatio-Temporal Clusters Detection

Require: E order by time.

function SIM-STCD(n-array of spatio-temporal events E, radius ρ, increase in the intensity ε, threshold A, number of events n)
  Let C be a cluster set initially empty.
  $\Lambda,\ N,\ \hat{\mu},\ M \Leftarrow STCD(\ E,\ \rho,\ \varepsilon,\ n\ )$
  $R_n \Leftarrow \sum_{j=1}^{n}(\ \Lambda_{j,n}\ )$
  alarm $\Leftarrow (\ R_n > A\ )$
  while alarm **do**
    $(C_{k,n},\ E\star) \Leftarrow findCluster(\ E,\ \Lambda,\ M)$
    $C \Leftarrow C \cup \{(C_{k,n},\ E\star)\}$
    $\Delta = N(C_{k,n}) - \hat{\mu}(C_{k,n})$
    $E \Leftarrow removeExcessEvents(\ E,\ E\star,\ \Delta\ )$
    $A \Leftarrow A - \Delta$
    $n \Leftarrow n - \Delta$
    $\Lambda,\ N,\ \hat{\mu},\ M \Leftarrow STCD(\ E,\ \rho,\ \varepsilon,\ n\ )$
    $R_n \Leftarrow \sum_{j=1}^{n}(\ \Lambda_{k,n}\ )$
    alarm $\Leftarrow (\ R_n > A\ )$
  end while
  return C
end function

Algorithm 2 Spatio-Temporal Cluster Detection

Require: E order by time.

function STCD(n-array of spatio-temporal events E, radius ρ, increase in the intensity ε, id of the last event n)
  Let N be an n-array of number of events inside a cylinder
  Let $\hat{\mu}$ be an n-array of expected number of events inside a cylinder
  Let M be an all-zero $n \times n$-matrix of events neighborhood
  Let Λ be an n-array of parcels
  for i $\leftarrow 1$ **to** n **do**
    for j $\leftarrow 1$ **to** n **do**
      $d \Leftarrow spatialDistance(\ E_i = (x_i, y_i),\ E_j = (x_j, y_j)\ )$
      $M_{i,j} \Leftarrow (\ (d \leqslant \rho) \wedge (i \neq j)\ )$
    end for
  end for
  for k $\leftarrow 1$ **to** n **do**
    $N(C_{k,n}) \Leftarrow \sum_{i=k}^{n}(\ M_{n,i}\ )$
    $\hat{\mu}(C_{k,n}) \Leftarrow \frac{(n-k+1)}{n} \sum_{i=1}^{n}(\ M_{k,i}\ )$
    $\Lambda_{k,n} \Leftarrow (1+\varepsilon)^{N(C_{k,n})}\ \exp(-\varepsilon\, \hat{\mu}(C_{k,n}))$
  end for
  return $\Lambda,\ N,\ \hat{\mu},\ M$
end function

Algorithm 3 function findCluster

function FINDCLUSTER(n-array of spatio-temporal events E, parcel's array Λ, events neighborhood matrix M)
  $\Lambda_{k*,n} \Leftarrow MAX\{\Lambda_{k,n}, 1 \leqslant k \leqslant n\}$
  Then, let $C_{k*,n}$ be the cylinder that define the cluster found beginning in event $k*$ and finishing in time of event n
  Let $E^\star$ be a dataset empty
  $E^\star \Leftarrow E^\star \cup \{E_{k*}\}$
  for i $\leftarrow (k*+1)$ **to** n **do** ▷ note: As data are ordered by time, only the events greater than $k*$ can be inside the cylinder $C_{k*,n}$.
    if event i is a $k*$ neighbor, ie., $M_{i,k*} = 1$
      then $E^\star \Leftarrow E^\star \cup \{E_i\}$
    end if
  end for
  return ($C_{k*,n}$, $E^\star$)
end function

Algorithm 4 procedure removeExcessEvents

procedure REMOVEEXCESSEVENTS(spatio-temporal events dataset E, spatio-temporal events sub-dataset $E^\star$, number of events in excess Δ)
  Let E' be an empty dataset
  for i $\leftarrow 1$ **to** Δ **do**
    Sort an event E_k of $E^\star$, always a different one.
    $E' \Leftarrow E' \cup \{E_k\}$
  end for
  $E \Leftarrow E - E'$
end procedure

3. Evaluation metrics

We defined some metrics to evaluate the method for simulated databases. We first describe these metrics when an unique cluster is present. In this case, we analyzed the percentages of *No Alarm*, *Incorrect Alarm*, and *Correct Alarm*. A *No Alarm* occurs when $R_i < A$ for all $i = 1, \ldots, n$, where n is the total number of events in the database. An *Incorrect Alarm* occurs when $R_i > A$ for some $i = 1, \ldots, n$ and the events set of the estimated cluster has no intersection with the events set of the real one. A *Correct Alarm* happens when $R_i > A$ for some $i = 1, \ldots, n$ and the events set of estimated cluster has any intersection with the events set of real one. For each database we used, we let the time moves forward until a *Correct Alarm* and record the total number of different alarms. If $R_i > A$, $R_{i+1} < A$, $R_{i+2} > A$, then the alarms at times i and $i + 2$ was considered as different alarms. When $R_i > A$, $R_{i+1} > A$ and the estimated clusters at times i and $i + 1$ are the same, we considered the alarms at times i and $i + 1$ as the same alarm. If $R_i > A$, $R_{i+1} > A$ and the estimated clusters at time i and $i + 1$ are not the same, we considered the alarms at times i and $i + 1$ as different alarms. We consider the estimated clusters as the same if the distance between their centers is greater than 2ρ.

The percentages of *No Alarm*, *Incorrect Alarm*, and *Correct Alarm* was calculated in relation to the total number of different alarms in all databases. To enable the calculation of these percentages, the total number of different alarms was consider as one for a *No Alarm* database. Figure 1 illustrates these concepts. Situation (a) is the case where the first alarm is a correct one. In (b) there is a period of incorrect alarms before the correct one. We count the number of different incorrect alarms that appear in this period. It is also possible to alternate periods between no alarm and incorrect alarms before reach a correct one. If there isn't any correct alarm, as in situation (c), we count only the number of different incorrect alarms. Finally, situation (d) shows a *No Alarm* case: there is no alarm, neither correct or incorrect.

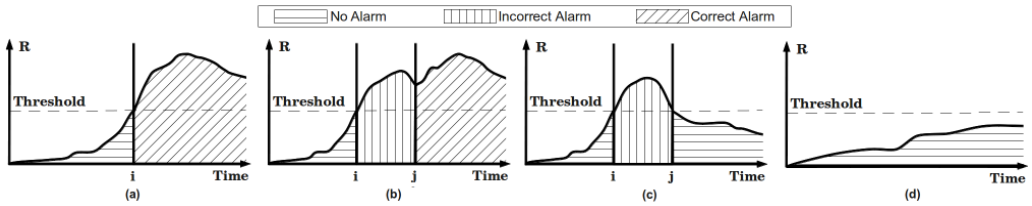


Figure 1. Alarms for some situations with one cluster.

For databases with one cluster, we also record the delay for a *Correct Alarm*, in time unit. This delay is the difference between the time of the the *Correct Alarm* and the real time of the beginning of the cluster.

We now describe the measures used for the situation with two clusters. In this case, an alarm can be *Single* or *Double*. It is a *Single Alarm* when $R_i > A$ and $R'_i < A'$; it is a *Double Alarm* when $R_i > A$ and $R'_i > A'$. Figure 2 illustrates the possibilities for the two clusters situation. Initially it has no alarm (0). If only the first threshold is exceeded (1), then it's a *Single Alarm*. If the first and second thresholds are exceeded (2), then it's a *Double Alarm*. From (1) there is three possibilities. The first one: the estimated cluster changes (3), it's a new *Single Alarm*. The second one: the first threshold is not exceeded anymore (4), then it returns to the case of no alarm. The last one: the second threshold is also exceeded (5), then it's a *Double Alarm*. From (2) there is also three possibilities. If at least one of the estimated clusters change (6), it's a new *Double Alarm*. If the first and second thresholds are not exceeded anymore (7), it returns to the case of no alarm. If only the second threshold is not exceeded anymore (8), then it's a *Single Alarm*.

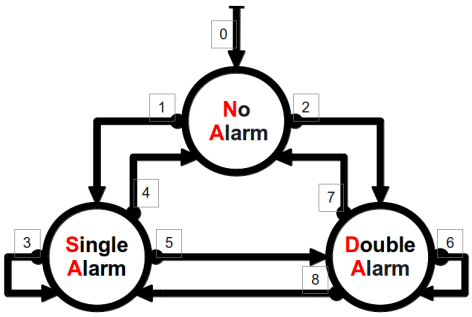


Figure 2. Alarms for the situation with two clusters.

A *Single Alarm* can be *Correct* or *Incorrect*. A *Single Alarm* is *Single Correct Alarm* if the estimated cluster has any intersection with one of the two real clusters, named here $C1$ and $C2$. If the estimated cluster has no intersection with $C1$ and no intersection with $C2$ it is a *Single Incorrect Alarm*. A *Double Alarm* can be *Correct*, *Incorrect* or *Half-Correct*. A *Double Correct Alarm* occurs when one of the estimated clusters has any intersection with $C1$ and the other estimated cluster has any intersection with $C2$. A *Double Incorrect Alarm* occurs the estimated clusters has no intersection with both $C1$ and $C2$. Finally, it is a *Double Half-Correct Alarm* when only one of the estimated clusters has any intersection with $C1$ or $C2$.

For each database with two clusters, we let the time moves forward until a *Double Correct Alarm* and record the total number of different alarms. We analyzed the percentages of *No Alarm*, *Incorrect Alarm*, *Incomplete Alarm*, and *Complete Alarm* in relation to the total number of different alarms in all databases. Again, to enable the calculation of these percentages, the total number of different alarms was consider as one for a *No Alarm* database. Here, *Single Incorrect Alarm* and *Double Incorrect Alarm* were both considered as *Incorrect Alarm*. A *Complete Alarm* is a *Double Correct Alarm*. A *Single Correct Alarm* is an *Incomplete Alarm*. One *Double Half-Correct Alarm* represents $1/2$ *Incorrect Alarm* and $1/2$ *Incomplete Alarm*.

4. Results

In this section we present the results we obtained applying both the original method and our extension to artificial (simulated) data. The original method was applied to databases with an unique cluster and the extension was applied to databases with two simultaneous clusters. We first describe the artificial databases we used in subsection 4.1 and then show the results in the following subsections.

4.1. Simulated databases

In all databases we considered a 10×10-square as the space area and the the time ranging from 0 to 10. We used four different values for the spatial radius $\rho : 0.5, 1.0, 1.5, 2.0$. For the increase in the intensity inside the cluster ε we tried three different values: $1, 3, 10$. All clusters finish at time 10. We also varied the time the cluster emerges. We used $5, 7, 8$ for this initial time. In case of two clusters, time, radius ρ and the excess ε are varied equality for both clusters. The clusters' centers are distant by at least 4ρ, guaranteeing that no cluster candidate intersects both of them simultaneously. In subsections 4.2 and 4.3 the true values for ρ and ε was used as input for these parameters. The threshold for the alarm was always set as the total number of event in the database.

For of each combination of ρ, ε and initial time, we generated 100 databases. Figure 3 presents examples for database's cases with one and two clusters. We show the simulations results in the following subsections. The initial time of the cluster proved not to be significant, and then we show here the average of each measure for the three different values we tried for this time. We also disregard the results for the combinations $\rho = 0.5$, $\varepsilon = 1$ and $\rho = 2.0$, $\varepsilon = 10$, since the method proved to be inadequate in these extremes.

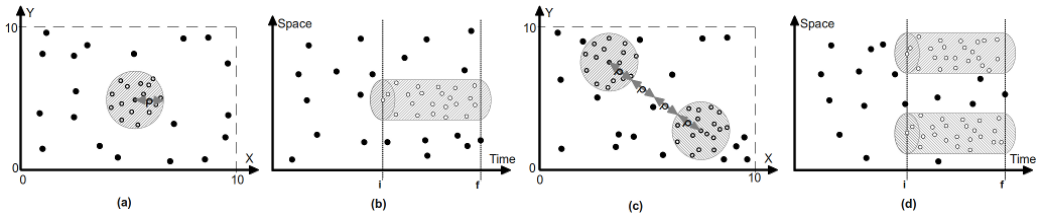


Figure 3. Examples of simulated database. (a) and (b) unique cluster database. (c) and (d) simultaneous clusters database. (a) and (c) space distribution. (b) and (d) space-time distribution.

4.2. Simulation results for an unique cluster

We applied the original method for simulated databases with one cluster. For each database we let the time move forward until a *Correct Alarm* or the final time ($t = 10$). We analyze the percentage of *No Alarm*, *Incorrect Alarm*, and *Correct Alarm*, shown in Figure 4. In this figure the bars represent the percentage of *No Alarm*, *Incorrect Alarm* and *Correct Alarm*, in this order. The segment in each bar is the 95% confidence interval. In all cases the percentage of *No Alarm* and *Incorrect Alarm* are very small.

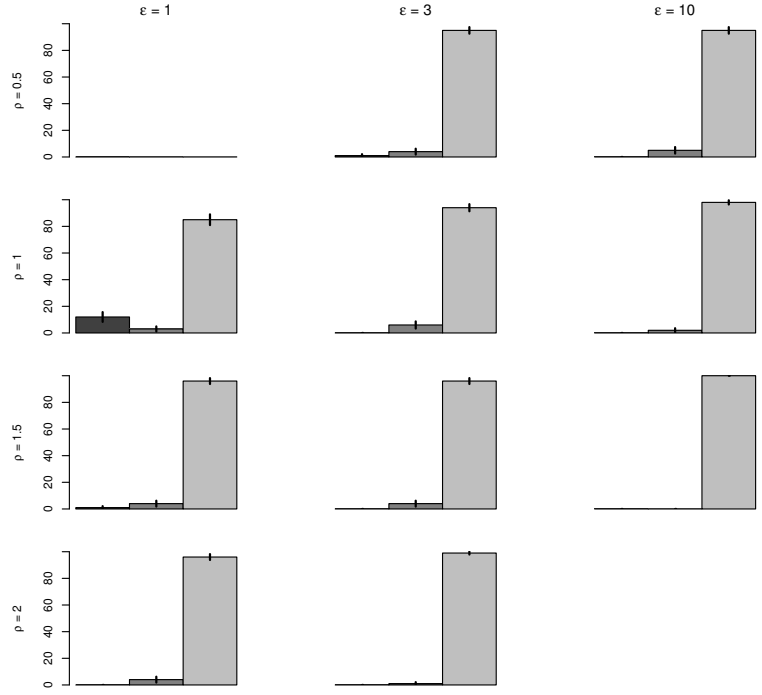


Figure 4. Number of Alarms for an unique cluster. The bars represent the percentage of *No Alarm*, *Incorrect Alarm* and *Correct Alarm*, in this order. The segment in each bar is the 95% confidence interval.

We also measured the delay for a correct alarm, in unit time. On average, it is 0.277, and the 95% confidence interval is $(0.067,\ 0.775)$.

4.3. Simulation results for simultaneous clusters

We applied our extension for simulated databases with two simultaneous clusters, $C1$ and $C2$. For each database we let the time move forward until a *Double Correct Alarm* or the final time ($t = 10$) and count the number of distinct alarms per case into database. The results are shown in Figure 5. In this figure the bars represent the percentage of *No Alarm*, *Incorrect Alarm*, *Incomplete Alarm* and *Complete Alarm*, in this order. The segment in each bar is the 95% confidence interval. The percentage of *No Alarm* and *Incorrect Alarm* are always very small, as in the case of an unique cluster.

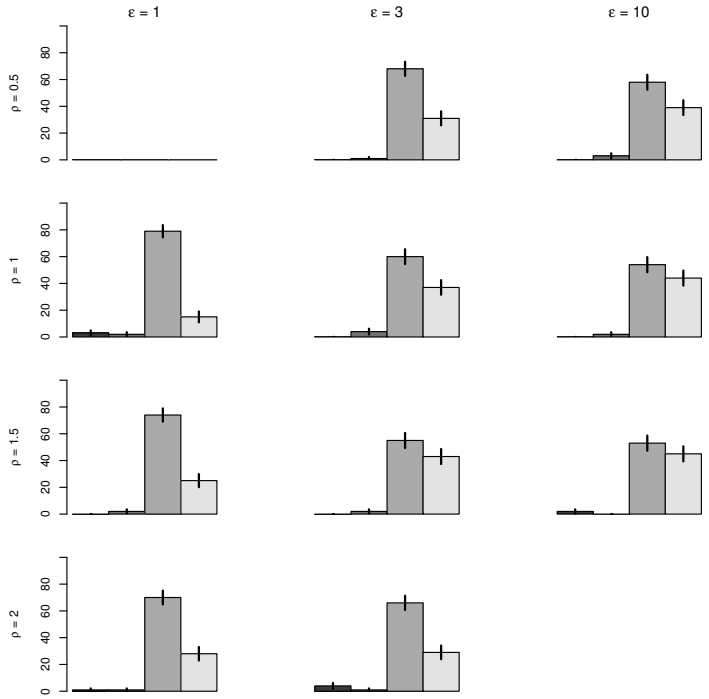


Figure 5. Number of Alarms for two clusters. The bars represent the percentage of *No Alarm*, *Incorrect Alarm*, *Incomplete Alarm* and *Complete Alarm*, in this order. The segment in each bar is the 95% confidence interval.

Figure 6 shows the delay for both the cases with one cluster and two simultaneous clusters. The first bar represents the delay for a *Correct Alarm* when there is an unique cluster (*Delay 1*). The second, third and fourth bars show the delay when there is two simultaneous clusters. These bars represents, in this order, the delay until: the first *Correct Alarm* (*Delay Min*), the detection of cluster $C1$ (*Delay C1*), the detection of cluster $C2$ (*Delay C2*), and a *Double Correct Alarm* (*Delay Double*). The segment in each bar is the 95% confidence band: percentiles 2.5% and 97.5%. We did not found significant difference for the average delay between *Delay 1* and *Delay Min*. It means that, the extension for multiple clusters detects one cluster as fast as the original method. As expected, there is also no significant difference for the average delay between *Delay C1* and *Delay C2*. For all others combinations we found significant difference for the average delay. Note that, on average, *Delay Double* is usually more than *Delay C1* and *Delay C2* and the percentage of *Incomplete Alarm* is always more than the percentage of *Complete*

Alarm. It's means: even before a *Double Correct Alarm*, our extension detects both clusters in different alarms (*Single Correct Alarms*).

Independent of the number of alarms, on average, our extension reached a *Complete Alarm* in 88.2% of cases into database, while 10.6% of cases it only identifies one cluster of two expected (*Incomplete Alarms*) all the time. In 0.2% of cases have only *Incorrect Alarms* and 1% of cases have *No Alarm*.

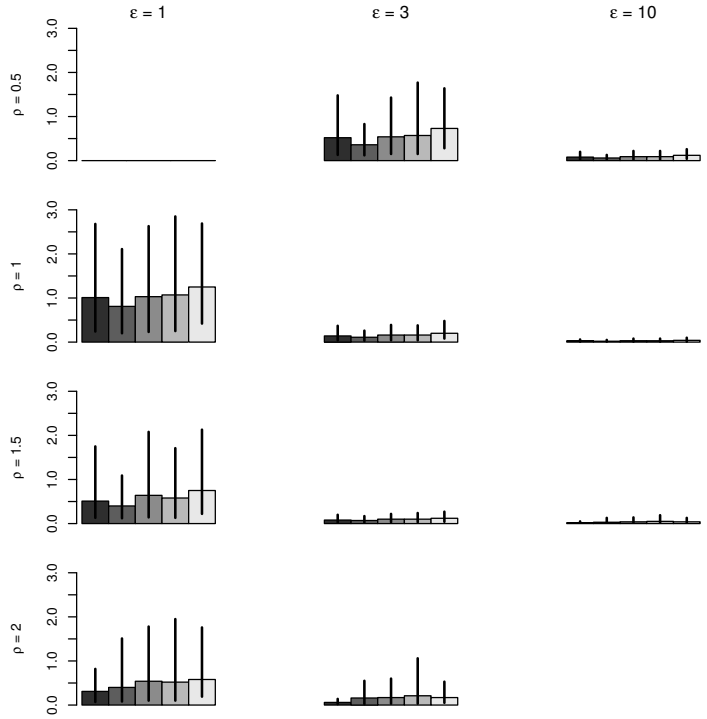


Figure 6. Delays in time unit. The segment in each bar is the 95% confidence band for: *Correct Alarm for one cluster case*, *first Correct Alarm for two cluster case*, *detection of cluster C1*, *detection of cluster C2*, and *Double Correct Alarm*.

5. Final considerations

Simulation results in the previous section show that our extension for multiple cluster is quite satisfactory for spatially separate clusters. It detects one of the multiple clusters as fast as the original method. The percentage of detection for both clusters is around 88%, and the delay is reasonably small. These are initial results, evaluating our method capability of detecting simultaneous clusters with respect to the original one.

Here we took ρ and ε parameters to be exactly their true values on datasets. The impact of changing these parameters was evaluated for the original method in [Assunção and Correa 2009]. The same should be done for the extension for multiple clusters in a future work, as well as the application of the extension to real data. Other future direction is to compare our approach to others, establishing its relative efficiency and effectiveness. Important issues to be considered hereafter are the automatic calibration for these parameters and removing the restriction on the cylindrical shape of the clusters, allowing for arbitrary shaped ones.

References

Assunção, R. and Correa, T. (2009). Surveillance to detect emerging space-time clusters. *Comput. Stat. Data Anal.*, 53(8):2817–2830.

Birant, D. and Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221.

Bogorny, V. and Shekhar, S. (2010). Spatial and Spatio-temporal Data Mining. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 1217–1217.

Demattei, C. and Cucala, L. (2010). Multiple spatio-temporal cluster detection for case event data: an ordering-based approach. *Communications in Statistics-Theory and Methods*, 40(2):358–372.

Goodchild, M. F. (2010). Twenty years of progress: GIScience in 2010. *Journal of Spatial Information Science*, (1):3–20.

Gudmundsson, J., Laube, P., and Wolle, T. (2012). Computational Movement Analysis. In Kresse, W. and Danko, D. M., editors, *Springer Handbook of Geographic Information*, pages 423–438. Springer Berlin Heidelberg.

Höhle, M. (2007). surveillance: An R package for the monitoring of infectious diseases. *Computational Statistics*, 22:571–582.

Kulldorff, M. (2001). Prospective time periodic geographical disease surveillance using a scan statistic. *Journal of the Royal Statistical Society, Series A*, 164:61–72.

Li, X.-Z., Wang, J.-F., Yang, W.-Z., Li, Z.-J., and Lai, S.-J. (2011). A spatial scan statistic for multiple clusters. *Mathematical biosciences*, 233(2):135–142.

Marshall, J. B., Spitzner, B. D., and Woodall, W. H. (2007). Use of the local Knox statistic for the prospective monitoring of disease occurrences in space and time. *Statistics in Medicine*, 26:1579–1593.

Oliveira, M. G. d. and Baptista, C. d. S. (2012). GeoSTAT: A system for visualization, analysis and clustering of distributed spatiotemporal data. In *Proceedings XIII GEOINFO,*, pages 108–119.

Richardson, D. B. (2013). Real-Time Space–Time Integration in GIScience and Geography. *Annals of the Association of American Geographers*, 103(5):1062–1071.

Tango, T., Takahashi, K., and Kohriyamma, K. (2011). A space-time scan statistic for detecting emerging outbreaks. *Biometrics*, 67:106–115.

Veloso, B. M., Iabrudi, A. I., and Correa, T. R. (2012). Localização em tempo real de acontecimentos através de vigilância espaço-temporal de microblogs. page 12. IX Encontro Nacional de Inteligência Artificial.

Zhang, Z., Assunção, R., and Kulldorff, M. (2010). Spatial scan statistics adjusted for multiple clusters. *Journal of Probability and Statistics*, 2010.

A method to automatically identify road centerlines from georeferenced smartphone data

George Henrique Rangel Costa, Fabiano Baldo

Graduate Program in Applied Computing
Computer Science Department
Santa Catarina State University - UDESC/Joinville

{dcc6ghrc,baldo}@joinville.udesc.br

Abstract. *Digital road maps have gained fundamental role in population's daily life, so they need to be accurate and up-to-date. A viable solution is to generate maps by processing GPS data. However, one of the most challenging tasks regarding this approach is how to extract road centerlines from the cloud of georeferenced GPS points. The literature presents various methods that do it, but none have been found that is prepared for the continuous update of the map and refinement of its roads' accuracy. In this context, the objective of this work is to propose a method to identify road centerlines using an evolutive algorithm in order to generate and update road maps. This work uses as source of data GPS traces collected by smartphones. Although suitable to obtain a large amount of georeferenced data, these devices bring the additional problem of identifying the transport vehicle used along each trace. Preliminary results indicate that the method's performance is satisfactory, with an average variation of 2.95 meters in relation to satellite images.*

1. Introduction

Digital road maps are supporting tools that have gained fundamental role in population's daily life. For this reason it is essential that the maps reflect reality as well as possible, that is, they must be generated from accurate data. Periodic updates are also necessary to adapt them to modifications on the roads. These factors must be taken into consideration when describing a solution to generate maps.

Usually, maps are generated and updated by photogrammetric methods applied on pictures taken by airplanes and, more recently, on satellite images [Jang et. al. 2010]. The results are good, but need manual adjustments in areas that are hard to map, e.g. where treetops block visibility of the road. Another disadvantage is that these maps tend to not be updated regularly, so after some time they might contain inconsistencies in relation to the actual roads. Due to these limitations new solutions are being promoted, and the GPS is one of the options.

By storing the sequence of georeferenced points collected by a GPS receiver, a trace that represents the trajectory of the moving object is created [Braz and Bogorny 2012]. Combining traces from one or more moving objects, it is possible to build a map containing the paths they traveled. Assuming that the moving object is a motor vehicle, the resulting map can be considered a road map.

Collaborative projects such as OpenStreetMap [OSM 2013] follow this idea. They allow users to upload traces and to use them to create or update maps. However, all map

editing is done manually. An automatic solution would be more effective, since it could allow maps to be updated more quickly. Several studies demonstrate the feasibility of this approach, like Brüntrup et. al. (2005) and Cao and Krumm (2009).

One of the main challenges in this approach is how to analyze the collected traces in order to identify road centerlines. Since civilian GPS receivers are not completely accurate the traces form a cloud of points along the roads, and the only way to find their centerlines is by approximation. In this context, a suitable technique may be an evolutive algorithm that can approximate a good solution after many iterations. This technique is part of the area of Evolutive Computing and can be classified into the subarea of search and optimization.

As more traces are collected, the existing road centerlines can be improved and new ones can be found. Thus, a new – updated – map should be generated. The literature contains many map generating methods, but none have been found that is prepared for the continuous update of the map and refinement of its roads' accuracy. This depends on how new are the traces used to generate maps. Therefore, a method to find road centerlines must take into consideration the date when the traces have been recorded.

Another question that raises when thinking about generating maps is how to obtain the georeferenced data, and one possible solution is to use smartphones. The adoption of this type of device has been increasing fast over the years and this stimulates its usage in this work. They have several sensors, including a GPS receiver, all of which can be used to obtain a range of information about how, when and where the users are moving. As the device is generally carried close to its user during the whole day, the volume of data collected tends to be high. On the other hand, this brings the additional problem of identifying the transport vehicle used along each trace.

Based on all the challenges and assumptions presented, the objective of this work is to propose a method to identify road centerlines using an evolutive algorithm in order to generate and update road maps.

The methodological procedure utilized in this work begins with a literature review of related work. In parallel, a system to record and collect data for testing purposes is developed. Based on the knowledge acquired an initial solution is defined. Then, this solution evolves cyclically, with the gradual implementation of improvements and analysis of results. The evaluation of the proposed method is performed comparing the road centerlines with satellite images, where the test scenario is the city of Joinville-Brazil.

The paper is structured as follows. Section 2 presents related work and compares it with the proposed solution. Section 3 details the proposed method. Section 4 presents the evaluation of the results obtained. Finally, conclusion and possible future work are drawn.

2. Other proposals to identify road centerlines

The literature related to generation of maps contains many different proposals on how to identify road centerlines. Brüntrup et. al. (2005) do it from traces from any GPS receiver, but assume that they contain only motor vehicles trajectories. After the client device collects the traces, the server filters them to remove noise, based on limits of speed,

acceleration, and relation between distance and time for two sequential points. After that, it divides each trace into segments and, for each of them, uses a clustering technique based on Artificial Intelligence to identify which points should be added to the map. This method does not depend on an initial map, so it can create maps from scratch. Also, it can add roads to previously existing maps as new traces are collected, but no details have been given about how it updates already existing road centerlines in case there is any change.

Cao and Krumm's (2009) work uses GPS data recorded with in-vehicle GPS loggers. The traces are filtered by the server based on limits of distance, time and direction of movement between two sequential points. After that they find the road centerlines, which are divided by direction of movement. To do that, they developed a technique based on principles of physical attraction and repulsion and applied it to each of the points collected, relating them to all the other points nearby. As in the previous work, this method also does not depend on an initial map. The authors do not discuss about updating the map.

Jang et. al. (2010) collect traces using mobile devices equipped with GPS receivers. Like Brüntrup et. al. (2005), they also assume that all traces contain only motor vehicle trajectories. No steps for filtering the traces have been described. To generate the map, they first divide the area where traces have been recorded in squares of one meter. After that, the points on each square are clustered based on the distance to nearby squares. The clusters are then connected to make the road network, and the analysis of the shape and angle of the streets is used to correct any problems. This step is also not detailed by the authors. As in the previous works, they do not use an initial map. Finally, the authors comment that the method to update the map has not been automatized and, therefore, is inefficient.

Niu et. al. (2011) use smartphones (Blackberry and iPhone) as client devices to collect data, but also assume that all traces contain only motor vehicle trajectories. Their method to identify road centerlines begins by filtering traces based on limits of accuracy and direction of movement. Points with speed equal to zero or repeated points in the same coordinates have also been discarded. Finally, they use a combination of Robust Loess and subtractive clustering. Differently from the previous works, this one depends on an initial map to serve as reference. The authors do not discuss about how to update the map.

Although each work proposes a different method to find road centerlines, there are a few similarities between these solutions. Some are interesting and the present work follows the same ideas. One such example is the independence of initial maps. Only Niu et. al. (2011) use a map as reference, but that is because their objective is to improve existing maps.

Another interesting similarity is related to the filtering of traces to remove noise. Except for Jang et. al. (2010), all other works defined heuristics based on the attributes provided by the GPS, such as speed, acceleration, distance and so on. The present work uses traces collected with smartphones, therefore, the most relevant heuristics are those defined by Niu et. al. (2011) – i.e. filtering based on accuracy, direction of movement, speed and proximity of other points – since they too used this type of device.

On the other hand, the present work approaches some steps of the solution differently. For example, all these proposals assume that the traces contain only motor vehicle

trajectories. This is not true when collecting data with smartphones, so it is necessary to identify the transport vehicle used along each trace.

Another concern is that these works rarely discuss about updating the map. This is an important question because new roads can be created and existing ones might be altered in some way, even if temporarily. It is possible to think that generating a new map as more traces are collected would be enough. However, this is not true, because when these four related works are identifying road centerlines they do not take into consideration the date when the traces have been recorded. Because of that, the result would be a mixture of old trajectories and new ones. Therefore, the present work analyzes the traces collected over time to identify changes and correctly reflect them on the map.

3. Proposed solution

The traces with georeferenced data must be (i) collected and (ii) preprocessed to filter out noise. Then, (iii) the road centerlines are identified by applying an evolutive algorithm. The following subsections detail each of these steps, with emphasis given to the identification of road centerlines.

3.1. Data collection

To test the solution it is necessary to collect traces with georeferenced data. To this end, a service oriented system has been developed. This system is composed of a smartphone application that collects the data and a web application that stores it. The smartphone application runs in devices with Android operating system and is available at <http://bdes.dcc.joinville.udesc.br:100/coletor>. The user decides when to start and stop recording data and no personal information is logged. The collected data are the GPS coordinates, recorded once per second, and the accelerometer oscillation, twenty times per second. That means that each point of the trace is composed of one GPS coordinate and twenty accelerometer oscillations. When the smartphone connects to the Internet the traces are sent to the server, where they are kept stored until the next step begins.

3.2. Preprocessing

Many points of the traces contain noise. These points must be discarded, in order to find more accurate road centerlines. There are several types of noise and each of them can be filtered using different methods. Following, the types of noise identified are presented.

A single trace might contain data from different means of transport, i.e. recorded while the user is walking, riding, running or driving. However, in order to generate road maps it is necessary to use only motor vehicle trajectories. That is, points that have not been recorded while using motor vehicles must be discarded. This can be achieved using many different methods, such as generating a Decision Tree combining GPS and accelerometer data, as proposed by Reddy et. al. (2010).

Many points have bad accuracy or incorrect information that can decrease the accuracy of the whole map, therefore they should not be used to identify road centerlines. One strategy to filter them out is to simply define a limit value for each of those characteristics and eliminate any points that go above these limits. Points too close to each other or with speed equal to zero can be considered unnecessary and should be discarded as well.

Lastly, there are algorithms that can compress traces by removing less significant points, such as the Douglas-Peucker and Opening Window algorithms [Hershberger and Snoeyink 1992; Meratnia and de By 2004]. These algorithms can substantially reduce the number of points of a trace without changing its shape.

On average, more than 75% of the points of each trace are discarded after applying these filters. Although this percentage is high it is important to consider that, usually, there will be many traces passing through the same road, and all their remaining points must be combined to find the road centerline.

3.3. Identifying road centerlines with an evolutive algorithm

After the preprocessing step, the roads are covered with points from all traces that pass through them. Since GPS has variable accuracy, the points are scattered all around the road, in the shape of a cloud. For this reason, this work assumes that none of the existing points is correctly positioned and therefore new points must be created to represent the road centerlines.

For each portion of the cloud that is analyzed, one road centerline is defined. These points cannot be defined deterministically, so it is necessary to use an approximation method. A suitable approach is to use evolutive algorithms. They are based on the concept of evolution by selection of the most adequate individuals after each generation. That is, given a set of candidates, the best ones are selected and used to create new sets.

In this work, each candidate (individual) contains only a geographic coordinate. They are selected according to the results of a fitness function, that weights them considering three values extracted from the points in the same portion of the cloud: (i) recording date, (ii) accuracy and (iii) distance between the candidate and the points nearby. The recording date is essential to correctly update the map, since it helps identify changes on the roads. Combining it with the other values, the candidate selected to be in the road centerline will be the one closest to the highest concentration of recent and accurate points.

The fitness function is composed of equations that define the influence of each of those characteristics (distance, accuracy and date) in finding the road centerlines. Each characteristic can have a different weight, so it is possible to control which has more influence. After calculating the fitness of all candidates, the selection procedure identifies which will be kept for the next generation, which will be discarded, and which will be used to create new candidates through random modifications to its latitude and longitude. That way, it is possible to avoid local maximums [Haupt and Haupt 2004].

This cycle of executing the fitness function and creating new generations is repeated many times. At the end, the best candidate from the last generation is chosen to represent the road centerline for that segment of the road. Algorithm 1 gives an overview of the entire method. It is worth mentioning that all of the method's parameters have been chosen based on tests with a set of 1500 points.

With this contextualization in mind it is possible to detail the method's execution. Each of the q traces collected is composed of n points. The traces and their respective points can be represented as $P_{j,k}(j = 1, 2, .., q; k = 1, 2, .., n)$. After sorting the traces from most recent to oldest and from most accurate to least, the process starts from $P_{1,1}$

(first point of the most recent and accurate trace) and is repeated to every point of every trace.

```
01 Initialize road_centerlines as an empty list;
02 Query database to get all traces ordered by date and accuracy;
03 foreach trace do
04     foreach point P of the trace do
05         if P has not yet been marked as used then
06             Create the S set by querying the database to identify points near P
               and with similar direction of movement;
07             Add P into the S set;
08             Define the domain based on the S set;
09             Create first generation based on the S set;
10             repeat
11                 foreach candidate in that generation do
12                     Calculate the candidate's fitness;
13                 Order candidates according to their fitness;
14                 Create new generation based on the best candidates and the
                   domain;
15             until 60th generation has been created;
16             Add the best candidate into road_centerlines;
17             foreach point P' in the S set do
18                 Mark P' as used;
```

Algorithm 1: Pseudocode of the road centerlines identification method

First, it is necessary to define what is the area considered close to $P_{j,k}$. Experimentally, it has been defined that this area is a circle with 15-meter radius, centered on $P_{j,k}$. The same idea is also used to represent the points in the cloud: their circle's radius is their accuracy and they are centered on their own coordinates. To simplify the explanation, from now on the circle of a point in the cloud will be called its "circle of accuracy".

To find the point that represents the road centerline in the area close to $P_{j,k}$, all n' points with a circle of accuracy that intersects this 15-meter radius circle are identified, no matter from which trace they are. The points with a direction of movement different than $P_{j,k}$ are eliminated, and the remaining ones (n'') compose the set of selected points $S_i(i = 1, 2, .., n'')$ (line 06 of Algorithm 1).

The domain specifies the range of valid coordinates that can be assigned to the candidates created in each generation, and it is defined (line 08 of Algorithm 1) as a rectangle bounded by the coordinates of the four most distant points in S. After defining it, the first generation is created and each candidate is represented as $C_x(x = 1, 2, .., z)$ (line 09 of Algorithm 1). In the first generation z is equal to n''; in every other generation z is fixed to 20. This exception occurs because the first generation is formed by a copy of the full S set, that is, for each S_i there is a candidate C_x located exactly on the same coordinates. This is done to speed up the creation of the first generation. On the other hand, if S has less than 20 points, those will be used as seed to create the points remaining to complete the C set.

After that, the fitness of each candidate is calculated (lines 11-12 of Algorithm

1). For each C_x, its relationship to each S_i is analyzed individually. The final fitness of C_x is the sum of the contributions of each point from S_i, as represented by equation (1). That equation is composed of equations (2), (3) and (5), that calculate the influence of recording time (IT), accuracy (IA) and distance (ID), respectively. Their results are in the range between 0 and 1, inclusive ($[0..1]$). The influence of those characteristics is weighted through the variables named MT, MA and MD, where $MT+MA+MD = 1$. In tests, $MT = 0.4$, $MA = 0.3$ and $MD = 0.3$.

$$FITNESS(C_x) = \sum_{i=1}^{n''} IT(S_i) \cdot MT + IA(S_i) \cdot MA + ID(C_x, S_i) \cdot MD \tag{1}$$

Equation (2) has been created to define the influence of time. The function $T(S_i, S_r)$ calculates the elapsed time (in days) between the recording of S_i and the recording of the most recent point of the S set (S_r). The variable t_{max} represents the maximum elapsed time allowed. In tests it has been used $t_{max} = 90$ days, and points older than t_{max} are removed from S to keep the map updated. This value has been chosen because this work aims to quickly identify modifications made on roads and reflect them on the map. Thus, recently collected traces must have a greater fitness value than older ones.

$$IT(S_i) = \frac{t_{max} - |T(S_i, S_r)|}{t_{max}} \tag{2}$$

Equation (3) defines the influence of accuracy, where $A(S_i)$ is the accuracy of the selected point S_i. A quadratic equation has been chosen because it allows points below a certain threshold (considered good accuracy) to be overestimated while points above it are underestimated. This equation has the following conditions: (i) if $A(S_i) = 0$, then $IA(S_i) = 1$; (ii) if $A(S_i) = a_{lim}$, then $IA(S_i) = a_{limV}$; (iii) if $A(S_i) = a_{max}$, then $IA(S_i) = 0$. In tests, $a_{max} = 20$ meters and $a_{lim} = 10$ meters, while $a_{limV} = 0.75$. With these values, points with really good accuracy ($A(S_i) \leq 5$) contribute a lot to the candidate's fitness value, points with good accuracy ($5 < A(S_i) \leq 10$) contribute not too much, and points with average accuracy ($10 < A(S_i) \leq 20$) contribute little. Applying these values to equation (3), equation (4) is reached.

$$\begin{aligned} IA(S_i) &= \alpha A(S_i)^2 + \left(\frac{-1 - \alpha \cdot {a_{max}}^2}{a_{max}} \right) A(S_i) + 1 \\ where \quad \alpha &= \frac{-a_{lim} - a_{max}\left(a_{limV} - 1\right)}{a_{max} a_{lim}\left(a_{max} - a_{lim}\right)} \end{aligned} \tag{3}$$

$$IA(S_i) = 1 - 0.0025 A(S_i)^2 \tag{4}$$

Lastly, equation (5) defines the influence of distance, where $D(C_x, S_i)$ is the distance between a candidate C_x and a point S_i. A quadratic equation has also been chosen because it increases the fitness of candidate points located near the selected points (within a threshold) while decreasing the fitness of candidate points far from them. This equation has the following conditions: (i) if $D(C_x, S_i) = 0$, then $ID(C_x, S_i) = 1$; (ii) if $D(C_x, S_i) = d_{lim}$, then $ID(C_x, S_i) = d_{limV}$; (iii) if $D(C_x, S_i) = d_{max}$, then

$ID(C_x, S_i) = 0$. In tests, $d_{max} = 50$ meters and $d_{lim} = 10$ meters, while $d_{limV} = 0.9$. With these values, points with a distance smaller than 10 meters contribute a lot to the candidate's fitness, and points farther than this gradually decrease their contribution. This equation is complemented by the previous one (influence of accuracy) because it is not enough for a candidate to be near many points, they must have good accuracy too. Applying these values to equation (5), equation (6) is reached.

$$\begin{gathered} ID(C_x, S_i) = \beta D(C_x, S_i)^2 + \left(\frac{-1 - \beta \cdot {d_{max}}^2}{d_{max}} \right) D(C_x, S_i) + 1 \\ where \quad \beta = \frac{-d_{lim} - d_{max}\left(d_{limV} - 1\right)}{d_{max} d_{lim}\left(d_{max} - d_{lim}\right)} \end{gathered} \tag{5}$$

$$ID(C_x, S_i) = 1 - 0.0075 D(C_x, S_i) - 0.00025 D(C_x, S_i)^2 \tag{6}$$

After calculating the fitness of each candidate, the two best ones (highest values) are kept for the next generation and 18 new candidates are generated through small random modifications of the eight best results. A new candidate will be 2.5 meters away from its seed's location, at most (lines 13-14 of Algorithm 1). The same process (equation (1)) is applied to this new generation and so forth until 60 generations have been calculated. At this moment, the candidate with better fitness is considered the road centerline in the area close to $P_{j,k}$ (line 16 of Algorithm 1).

The next area where the road centerline would be calculated is the area close to $P_{j,k+1}$. As the GPS records one point per second, even after the preprocessing the chance of $P_{j,k+1}$ being near $P_{j,k}$ is very high. If this indeed happen, it means that $P_{j,k+1}$ has already been used to calculate the road centerline in the area close to $P_{j,k}$, because its circle of accuracy intersected the 15-meter radius circle centered on $P_{j,k}$ (as explained during the creation of the S set). Thus, it is not necessary to calculate the road centerline near $P_{j,k+1}$. The same thought is valid to every point included in the S set of $P_{j,k}$, so they are "marked" to not be used again (lines 17-18 of Algorithm 1). Therefore, before calculating the road centerline near any given point $P_{j,k}(j \in q; k \in n)$ first it is necessary to verify if it has been marked yet (line 05 of Algorithm 1). If so, then $P_{j,k}$ is ignored and the same verification is done to the following points. After processing every point of all traces collected, all road centerlines are found.

4. Results Assessment

As the aim of this work is to represent the road centerline based on a cloud of points collected by smartphone's GPS, its evaluation consists in comparing the resulting maps to satellite images. The test scenarios are regions of the city of Joinville-Brazil that contain complex road structures. These places can indicate the efficiency of the proposed method. If it works well for these scenarios, by induction, the same is valid for regions with simpler road structures.

The first test scenario presented in Figure 1a contains a region where two highways cross. Based on the collected data showed in Figure 1b, the difficulties to find road centerlines are related to three factors: (i) two-ways on each highway; (ii) one highway

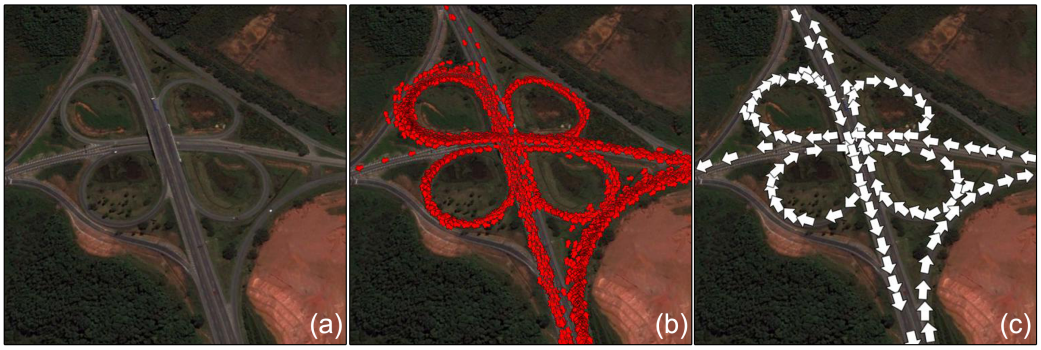


Figure 1. First test: (a) satellite image, (b) points after preprocessing, (c) road centerlines

with two lanes in each direction; (iii) cloverleaf interchange causing multiple join and disjoin intersections.

As can be seen in Figure 1c, in order to have a more realistic result, it is important to distinguish the direction of movement of each roadway, otherwise the road centerline will tend to the side of the roadway with more traffic. In the method developed, the roadways are distinguished before executing the evolutive algorithm. The method does not try to identify the number of lanes in a roadway and only returns one centerline. Therefore, the centerline will be closer to the lane with more traffic. It is assumed that traffic is almost equally distributed across all lanes, so the final result should not be affected. Besides that, the roadway differentiation also identifies when a road has a join or a disjoin point, as well as when two roads overlap. Figure 1c shows that the proposed method distinguishes the centerline of the two highways and the cloverleaf interchange, without mixing them.

The second test showed in Figure 2 has been performed on a large roundabound that has access points to two universities and to various places of the city. From all test scenarios, this one is considered by the authors as the most complex because of: (i) roads with multiple lanes; (ii) proximity of the roads, with similar or different directions of movement.

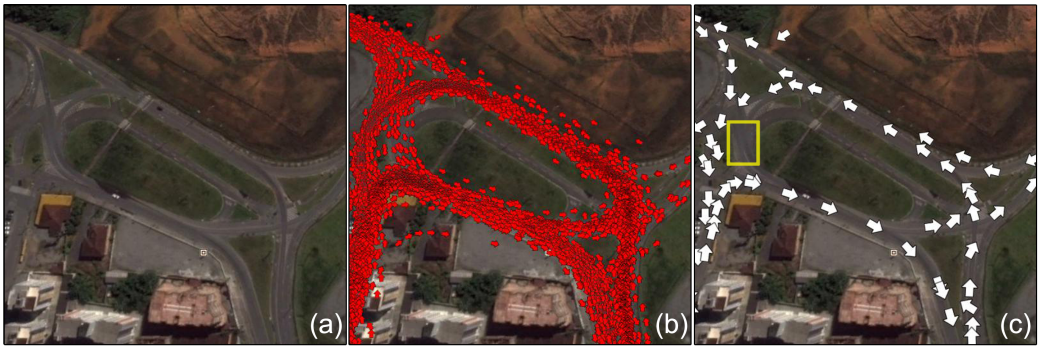


Figure 2. Second test: (a) satellite image; (b) points after preprocessing; (c) road centerlines with incorrectly mapped area highlighted

Once again, the differentiation by direction of movement is essential to achieve that result. Unfortunately, in some cases the parameters have not been enough to differ-

entiate nearby roads. As example, in Figure 2c there is a road – highlighted by the yellow rectangle – without centerline, but in Figure 2b it is possible to see that points have been collected in that area. This happened because the direction of movement of the road to the left is similar and they are close to each other, so the method treated them as two lanes from the same roadway. In other tests, the same problem is observed in highways with parallel roads nearby. It is believed that this problem can be fixed by tweaking the parameters of the preprocessing step and evolutive algorithm.

As can be roughly perceived in the two test scenarios presented in Figure 1 and Figure 2, the difference between the road centerlines and the satellite images in both tests is small. However, it must be taken into account that that the baseline images provided by Google Earth might be a little shifted from their correct position. This means that it is not possible to confirm whether the road centerlines or the satellite images is more correct, but at least it is possible to observe that there are no considerable differences.

In order to collect some statistical evidences the average perpendicular distance between the road centerlines and the road at the satellite images has been calculated using 100 randomly selected points. The result is an average distance of 2.95 meters.

The method has been implemented using the Python programming language. The DBMS chosen is PostgreSQL with PostGIS extension to support geographical data. The database scheme contains two tables. One stores the filtered traces and that is accessed by the method to retrieve information about points being processed. The other stores the circle of accuracy of each point and that is used to find the points that fill the S set (as explained in Section 3.3). The method's output is a Keyhole Markup Language (KML) file that is opened with the Google Earth program, used to evaluate the results and calculate the average distance.

To process all data collected between January 27 and June 15 2013 (6475 Kilometers collected in 301 hours; 4237 files with traces and a total of 966698 points), the Python implementation took four hours and 42 minutes. The computer used on the tests had an Intel Core i5 2400 3.1GHz, 10GB RAM and Windows 7 operating system. This execution time can be shortened, because the method has been implemented without any optimizations and without following the parallel programming paradigm.

5. Conclusion and future work

Digital maps are becoming increasingly more important. However, producing and keeping them updated is a complex problem. Road maps can be created using georeferenced data provided by GPS receptors, and smartphones can facilitate its collection. Due to their popularization, these devices make possible to collect a large amount of traces, which can be used to generate a complete map of a city.

Related work propose methods to generate maps, but few of them use data from smartphones. This can be partially explained by the complexity of using data from a device that is not dedicated to such task. This makes necessary to use filters to identify and discard traces that are not motor vehicle trajectories. Besides that, no work has been found that take into consideration the influence of the traces' date of recording to the process of finding road centerlines. This is essential to automatically update the map, otherwise the result would be a mixture of the old path with the new one.

This work tackles one of the main challenges of automatic generating road maps, which is identifying road centerlines. The proposed solution assumes that an approximation method can be used to create points that better represent the centerline of a road. This approach has been implemented applying an evolutive algorithm that analyzes recording date, accuracy and distance between the points candidate to center of the road and the points collected with smartphones.

The method's parameters have been refined through many test cycles, and the results obtained showed little difference to the satellite images, with an average difference of 2.95 meters. However, it is still possible to optimize the parameters to achieve better results. It would be interesting to compare this method and the proposals from related work using the same dataset, but that was not possible because their implementations could not be found publicly.

Regarding future work, it is suggested to study ways to improve collected traces' reliability. For example, by applying a method to infer that the accuracy of a certain point is better than what the GPS informed. If this would be possible, fewer points would be eliminated during the preprocessing step and the evolutive algorithm could find more accurate road centerline results. In this context, the Kalman Filter [Grewal and Andrews 2001] seems to be a worthy approach to be investigated. Another future work would be to define an update policy. Traces are not collected everywhere with the same frequency. Therefore, different parameters should be used for each region. For example, in downtown only points with good accuracy could be used, since more traces are collected there. On the other hand, in rural regions points that are not so good could be used, since traces are not collected so frequently there. One last future work would be to analyze the data collected to extract more georeferenced information, such as the location of semaphores or potholes, so as to make the map more complete.

6. Acknowledgements

This project is supported by a research scholarship granted by the Brazilian Coordination for Enhancement of Higher Education Personnel (CAPES).

References

Braz, F. J. and Bogorny, V. (2012) Introdução a trajetórias de objetos móveis. Editora Univille.

Brüntrup, R., Edelkamp, S., Jabbar, S. and Scholz, B. (2005) "Incremental map generation with GPS traces". In: Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems, p. 413-418. IEEE.

Cao, L. and Krumm, J. (2009) "From GPS traces to a routable road map". In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09, p. 3-12. New York, USA: ACM Press.

Grewal, M. S. and Andrews, A. P. (2001) Kalman Filtering: Theory and Practice Using MATLAB. John Wiley & Sons, Inc. 2. ed. 410p.

Haupt, R. L. and Haupt, S. E. (2004) Practical genetic algorithms. Wiley-Interscience. 2. ed. 253p.

Hershberger, J. and Snoeyink, J. (1992) ”Speeding Up the Douglas-Peucker Line-Simplification Algorithm”. In: Proceedings of the 5th International Symposium on Spatial Data Handling, p. 134-143.

Jang, S., Kim, T. and Lee, E.(2010) ”Map Generation System with Lightweight GPS Trace Data”. In: International Conference on Advanced Communication Technology - ICACT, p. 1489-1493.

Meratnia, N. and de By, R. A. (2004) ”Spatiotemporal Compression Techniques”. In: Lecture Notes in Computer Science, 2992, p. 765-782. Springer-Verlag.

Niu, Z., Li, S. and Pousaeid, N. (2011) ”Road extraction using smart phones GPS”. In: Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications - COM.Geo ’11. New York, USA: ACM Press.

OSM. OpenStreetMap (2013) ”The Free Wiki World Map”. Disponível em: <http://www.openstreetmap.org>. Acesso em: 05 ago. 2013.

Reddy, S. et.al. (2010) ”Using mobile phones to determine transportation modes”. In: ACM Transactions on Sensor Networks, 6(2), p. 1-27.

A Parallel Sweep Line Algorithm for Visibility Computation

Chaulio R. Ferreira[1], **Marcus V. A. Andrade**[1],
Salles V. G. Magalhes[1], **W. R. Franklin**[2], **Guilherme C. Pena**[1]

[1]Departamento de Informática – Universidade Federal de Viçosa (UFV)
Campus da UFV – 36.570-000 – Viçosa – MG – Brazil

[2]Rensselaer Polytechnic Institute – Troy – NY – USA

{chaulio.ferreira,marcus,salles,guilherme.pena}@ufv.br,
wrf@ecse.rpi.edu

Abstract. *This paper describes a new parallel raster terrain visibility (or viewshed) algorithm, based on the sweep-line model of [Van Kreveld 1996]. Computing the terrain visible from a given observer is required for many GIS applications, with applications ranging from radio tower siting to aesthetics. Processing the newly available higher resolution terrain data requires faster architectures and algorithms. Since the main improvements on modern processors come from multi-core architectures, parallel programming provides a promising means for developing faster algorithms. Our algorithm uses the economical and widely available shared memory model with OpenMP. Experimentally, our parallel speedup is almost linear. On 16 parallel processors, our algorithm is up to 12 times faster than the serial implementation.*

1. Introduction

An important group of Geographical Information Science (GIS) applications on terrains concerns visibility, i.e., determining the set of points on the terrain that are visible from some particular observer, which is usually located at some height above the terrain. This set of points is known as *viewshed* [Franklin and Ray 1994] and its applications range from visual nuisance abatement to radio transmitter siting and surveillance, such as minimizing the number of cellular phone towers required to cover a region [Ben-Moshe et al. 2007], optimizing the number and position of guards to cover a region [Magalhães et al. 2011], analysing the influences on property prices in an urban environment [Lake et al. 1998] and optimizing path planning [Lee and Stucky 1998]. Other applications are presented in [Champion and Lavery 2002].

Since visibility computation is quite compute-intensive, the recent increase in the volume of high resolution terrestrial data brings a need for faster platforms and algorithms. Considering that some factors (such as processor sizes, transmission speeds

and economic limitations) create practical limits and difficulties for building faster serial computers, the parallel computing paradigm has become a promising alternative for such computing-intensive applications [Barney et al. 2010]. Also, parallel architectures have recently become widely available at low costs. Thus, they have been applied in many domains of engineering and scientific computing, allowing researchers to solve bigger problems in feasible amounts of time.

In this paper, we present a new parallel algorithm for computing the viewshed of a given observer on a terrain. Our parallel algorithm is based on the (serial) sweep line algorithm firstly proposed by [Van Kreveld 1996], which is described in Section 2.3.3. Comparing to the original algorithm, our new algorithm achieved speedup of up to 12 times using 16 parallel cores, and up to 3.9 times using four parallel cores.

2. Related Work

2.1. Terrain representation

In what follows, our region of interest is small compared to the radius of the earth, thus, for this discussion the earth can be considered to be flat.

A *terrain* τ is a $2\frac{1}{2}$ dimensional surface where any vertical line intersects τ in at most one point. The terrain is usually represented approximately either by a *Triangulated Irregular Network (TIN)* or a *Raster Digital Elevation Model (DEM)* [Li et al. 2005]. A TIN is a partition of the surface into planar triangles, i.e., a piecewise linear triangular spline, where the elevation of a point p is a bilinear interpolation of the elevations of the vertices of the triangle containing the projection of p. On the other hand, a DEM is simply a matrix storing the elevations of regularly spaced positions or posts, where the spacing may be either a constant number of meters or a constant angle in latitude and longitude. In this paper, we will use the DEM representation because of its simpler data structure, ease of analysis, and ability to represent discontinuities (cliffs) more naturally. Finally, there is a huge amount of data available as DEMs.

2.2. The viewshed problem

An *observer* is a point in space from where other points (the *targets*) will be visualized. Both the observer and the targets can be at given heights above τ, respectively indicated by h_o and h_t. We often assume that the observer can see only targets that are closer than the *radius of interest*, ρ. We say that all cells whose distance from O is at most ρ form the *region of interest* of O. A target T is visible from O if and only if the distance of T from O is, at most, ρ and the straight line, the *line of sight*, from O to T is always strictly above τ; see Figure 1.

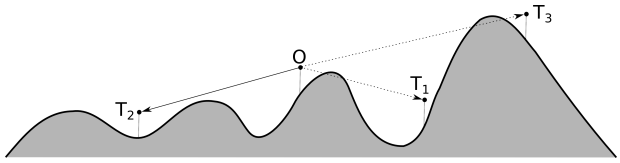


Figure 1. Targets' visibility: T_1 and T_3 are not visible but T_2 is.

The *viewshed* of O is the set of all terrain points vertically below targets that can be seen by O; formally,

$$viewshed(O) = \{p \in \tau \mid \text{the target above } p \text{ is visible from } O\}$$

with ρ implicit. The viewshed representation is a square $(2\rho + 1) \times (2\rho + 1)$ bitmap with the observer at the center.

Theoretically, determining whether a target T is visible from O requires verifying all points in the line of sight connecting O to T. But since τ is represented with a finite resolution, only points close to the rasterized line segment connecting the projections of O and T onto the horizontal plane will be verified. Which points those might be, is one difference between competing algorithms, as the ones we will describe in Section 2.3. The visibility depends on the line segment rasterization method used, see Figure 2, and how the elevation is interpolated on those cells where the segment does not intersect the cell center.

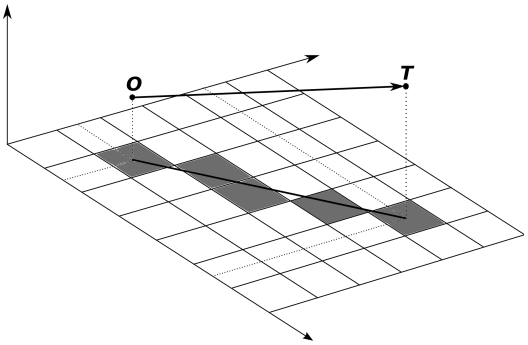


Figure 2. The rasterization of the line of sight projection.

The visibility of a target above a cell c_t can be determined by checking the slope of the line connecting O and T and the cells' elevation on the rasterized segment. More precisely, suppose the segment is composed of cells $c_0, c_1, \cdots, c_t$ where c_0 and c_t correspond to the projections of O and T respectively. Let α_i be the slope of the line connecting O to c_i, that is,

$$\alpha_i = \frac{\zeta(c_i) - (\zeta(c_0) + h_o)}{dist(c_0, c_i)} \tag{1}$$

where $\zeta(c_0)$ and $\zeta(c_i)$ are, respectively, the elevation of cells c_0 and c_i and $dist(c_0, c_i)$ is the 'distance' (in number of cells) between these two cells. The target on c_t is visible if and only if the slope $\frac{\zeta(c_t)+h_t-(\zeta(c_0)+h_o)}{dist(c_0,c_t)}$ is greater than α_i for all $0 < i < t$. If yes, the corresponding cell in the viewshed matrix is set to 1; otherwise, to 0.

2.3. Viewshed algorithms

Different terrain representations call for different algorithms. A TIN can be processed by the algorithms proposed by [Cole and Sharir 1989] and [De Floriani and Magillo 2003]. For a DEM, we can point out [Van Kreveld 1996] and RFVS [Franklin and Ray 1994], two very efficient algorithms. Another option for processing DEMs is the well-known R3 algorithm [Shapira 1990]. Although this one is not as efficient as the other two, it has higher accuracy and may be suitable for small datasets.

These three algorithms differ from each other not only on their efficiency, but also on the visibility models adopted. For instance, R3 and Van Kreveld's algorithms use a center-of-cell to center-of-cell visibility, that is, a cell c is visible if and only if the ray connecting the observer (in the center of its cell) to the center of c does not intersect a cell blocking c. On the other hand, RFVS uses a less restrictive approach where a cell c may be considered visible if its center is not visible but another part of c is.

Therefore, the viewsheds obtained by these methods may be different. Without knowing the application and having a model for the terrain's elevation between the known points, it is impossible to say which one is better. Some applications may prefer a viewshed biased in one direction or the other, while others may want to minimize error computed under some formal terrain model. For instance, since Van Kreveld's algorithm presents a great tradeoff between efficiency and accuracy [Fishman et al. 2009], it may be indicated for applications that require a high degree of accuracy. On the other hand, if efficiency is more important than accuracy, the RFVS algorithm could be preferred.

Considering that each one of these algorithms might be suitable for different applications, we will describe them briefly in the next sections.

2.3.1. R3 algorithm

The R3 algorithm provides a straightforward method of determining the viewshed of a given observer O with a radius of interest ρ. Although it is considered to have great accuracy [Franklin et al. 1994], this algorithm runs in $\Theta(n^{\frac{3}{2}})$, where $n = \Theta(\rho^2)$. It works as follows: for each cell c inside the observer's region of interest, it uses the digital differential analyzer (DDA) [Maćiorov 1964] to determine which cells the line of sight (from O to the center of c) intersects. Then, the visibility of c is determined by calculating the

slope of all cells intersected by this line of sight, as described in Section 2.2. In this process, many rules to interpolate the elevation between adjacent posts may be used, such as average, linear, or nearest neighbour interpolations.

2.3.2. RFVS algorithm

RFVS [Franklin and Ray 1994] is a fast approximation algorithm that runs in $\Theta(n)$. It computes the terrain cells' visibility along rays (line segments) connecting the observer (in the center of a cell) to the center of all cells in the boundary of a square of side $2\rho + 1$ centered at the observer (see Figure 3(a)). In each column, it tests the line of sight against the closest cell. Although a square was chosen for implementation simplicity, other shapes such as a circle would also work.

RFVS creates a ray connecting the observer to a cell on the boundary of this square, and then rotates it counter-clockwise around the observer to follow along the boundary cells (see Figure 3(a)). The visibility of each ray's cells is determined by walking along the segment, which is rasterized following [Bresenham 1965]. Suppose the segment is composed of cells $c_0, c_1, \cdots, c_k$ where c_0 is the observer's cell and c_k is a cell in the square boundary. Let α_i be the slope of the line connecting the observer to c_i determined according to Equation (1) in Section 2.2. Let μ be the highest slope seen so far when processing c_i, i.e., $\mu = \max\{\alpha_1, \alpha_2, \cdots, \alpha_{i-1}\}$. The target above c_i is visible if and only if the slope $\left(\zeta(c_i) + h_t - (\zeta(c_0) + h_o)\right) / dist(c_0, c_i)$ is greater than μ. If yes, the corresponding cell in the viewshed matrix is set to 1; otherwise, to 0. Also, if $\alpha_i > \mu$ then μ is updated to α_i. We say that a cell c_i blocks the visibility of the target above c_j if c_i belongs to the segment $\overline{c_0 c_j}$ and α_i is greater or equal to the slope of the line connecting the observer to the target above c_j.

2.3.3. Van Kreveld's algorithm

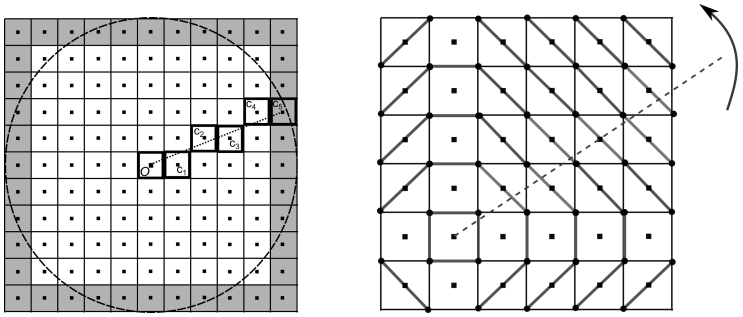
Van Kreveld's algorithm [Van Kreveld 1996] is another fast viewshed algorithm. According to [Zhao et al. 2013], it has accuracy equivalent to the R3 algorithm, while running in $\Theta(n\ log\ n)$. Its basic idea is to rotate a sweep line around the observer and compute the visibility of each cell when the sweep line passes over its center (see Figure 3(b)). For that, it maintains a balanced binary tree (the *agenda*) that stores the slope of all cells currently being intersected by the sweep line, keyed by their distance from the observer. When this sweep line passes over the center of a cell c, the *agenda* is searched to check c's visibility. More specifically, this algorithm works as follows:

For each cell, it defines three types of events: *enter*, *center*, and *exit* events to indi-

cate, respectively, when the sweep line starts to intersect a cell, passes over the cell center and stops to intersect a cell. The algorithm creates a list *E* containing these three types of events for all cells inside the region of interest. The events are then sorted according to their azimuth angle.

To compute the viewshed, the algorithm sweeps the list *E* and for each event it decides what to do depending on the type of the event:

- If it is an *enter* event, the cell is inserted into the *agenda*.
- If it is an *center* event of cell c, the *agenda* is searched to check if it contains any cell that lies closer to the observer than c and has slope greater or equal to the slope of the line of sight to c; if yes, then c is not visible, otherwise it is.
- If it is an *exit* event, the cell is removed from the *agenda*.



(a) (b)

Figure 3. Viewshed algorithms: (a) RFVS; (b) Van Kreveld - adapted from [Fishman et al. 2009].

2.3.4. Parallel viewshed algorithms

Parallel computing has become a mainstream of scientific computing and recently some parallel algorithms for viewshed computation have been proposed. [Zhao et al. 2013] proposed a parallel implementation of the R3 algorithm using Graphics Processing Units (GPUs). The RFVS algorithm was also adapted for parallel processing on GPUs by [Osterman 2012]. [Chao et al. 2011] proposed a different approach for parallel viewshed computation using a GPU, where the algorithm runs entirely within the GPUs visualization pipeline used to render 3D terrains. [Zhao et al. 2013] also discuss other parallel approaches.

However, we have not found any previous work proposing a parallel implementation of Van Kreveld's algorithm. In fact, [Zhao et al. 2013] stated that "a high degree of sequential dependencies in Van Kreveld's algorithm makes it less suitable to exploit

parallelism". In Section 3 we show how we have overcome this difficulty and describe our parallel implementation of Van Kreveld's sweep line algorithm.

2.4. Parallel Programming models

There are several parallel programming models, such as distributed memory/message passing, shared memory, hybrid models, among others [Barney et al. 2010]. In this work, we used the shared memory model, where the main program creates a certain number of tasks (*threads*) that can be scheduled and carried out by the operating system concurrently. Each thread has local data, but the main program and all threads share a common address space, which can be read from and written to asynchronously. In order to control the concurrent access to shared resources, some mechanisms such as locks and semaphores may be used. An advantage of this model is that there is no need to specify explicitly the communication between threads, simplifying the development of parallel applications.

For the algorithm implementation, we used OpenMP (*Open Multi-Processing*) [Dagum and Menon 1998], a portable parallel programming API designed for shared memory architectures. It is available for C++ and Fortran programming languages and consists of a set of compiler directives that can be added to serial programs to influence their run-time behaviour, making them parallel.

3. Our parallel sweep line algorithm

As described in Section 2.3.3, Van Kreveld's algorithm needs information about the cells intersected by the sweep line. It maintains these information by processing the *enter* and *exit* events to keep the *agenda* up to date as the sweep line rotates. Therefore, processing a *center* event is dependent upon all earlier *enter* and *exit* events.

In order to design a parallel implementation of this algorithm, this dependency had to be eliminated. We did that by subdividing the observer's region of interest in S sectors around the observer, O (see Figure 4(a), where S = 8). Our idea is to process each one of these sectors independently using Van Kreveld's sweep line algorithm, such that it can be done in parallel.

More specifically, consider sector s defined by the interval $[\alpha, \beta)$, where α and β are azimuth angles. Let a and b be the line segments connecting O to the perimeter of its region of interest, with azimuth angles α and β, respectively (see Figure 4(a)). To process s, the algorithm creates rays connecting O to all cells on the perimeter of the region of interest that are between (or intersected by) a and b (see Figure 4(b)). These rays are rasterized using the DDA method [Maćiorov 1964] and the events related to the intersected cells are inserted into s's own list of events, E_s. Since the grid cells are convex,

this process inserts into E_s the events for all cells inside s or intersected by a or b. The inserted cells are shown in Figure 4(b).

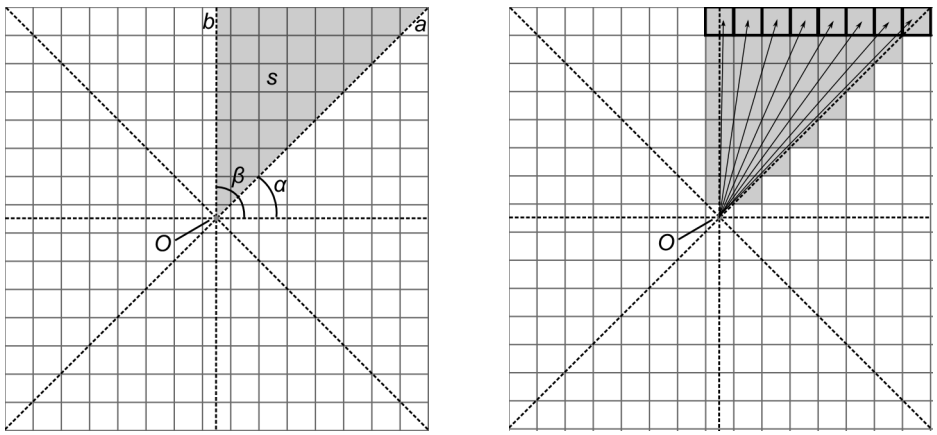


Figure 4. Sector definition: (a) The subdivision of the region of interest and the sector s, defined by the interval $[\alpha, \beta)$; (b) The cells in the perimeter of the region of interest, the rays used to determine which cells are intersected by s and the cells inserted into E_s (shaded cells).

Then, the algorithm sorts E_s by the events' azimuth angles and sweeps it in the same manner as Van Kreveld's algorithm. Note that, because we have distributed the events into different lists and each list contains all events that are relevant to its sector, each sector may be processed independently, each one with its own *agenda*. This allows a straightforward parallelization of such processing. Also, note that the events of a cell may be included in more than one sector's event list and therefore some cells may be processed twice. But that is not a problem, since this will happen only to a few cells, and it will not affect the resulting viewshed.

It is also important to note that our algorithm might be faster than the original one even with non-parallel architectures. For instance, we achieved up to 20% speedup using only one processor (see Section 4). This happens because both implementations have to sort their lists of events and, while the original (serial) algorithm sorts a list of size n, our algorithm sorts S lists of size about $\frac{n}{S}$. Since sorting can be done in $\Theta(n\ logn)$, the latter one is faster. In practice, we empirically concluded that, for a computer with N cores, using $S > N$ achieved better results than using $S = N$. This will be further discussed in Section 4, as long with our experimental results.

4. Experimental results

We implemented our algorithm in C++ using OpenMP. We also implemented the original (serial) Van Kreveld's algorithm in C++. Both algorithms were compiled with g++ 4.6.4

and optimization level -O3. Our experimental platform was a Dual Intel Xeon E5-2687 3.1GHz 8 core. The operational system was Ubuntu 12.04 LTS, Linux 3.5 Kernel.

The tests were done using six different terrains from SRTM datasets and, in all experiments, the observer was sited in the center of the terrain, with $h_O = 100$ meters and $h_T = 0$. The radius of interest, ρ, was set to be large enough to cover the whole terrain.

Another important parameter for our program is the number of sectors S into which the region of interest will be subdivided. Changing the number of sectors may significantly modigy the algorithm performance. Empirically, we determined that good results are achieved when the region is subdivided such that each sector contained about 40 cells from the perimeter of the region of interest, so we adopted that strategy. Other strategies for choosing the number of sectors should be further investigated and it could be an interesting topic for future work.

To evaluate our algorithm performance, we compared it to the original (serial) algorithm. We ran several experiments limiting the number of parallel threads to the following values: 16, 8, 4, 2 and 1. The results are given in Table 1 and plotted in Figure 5(a), where the times are given in seconds and refer just to the time needed to compute the viewshed. That is, we excluded the time taken to load the terrain data and to write the computed viewshed into disk, since it was insignificant (less than 1% of the total time in all cases). Also, the time represents the average time for five different runs of the same experiment.

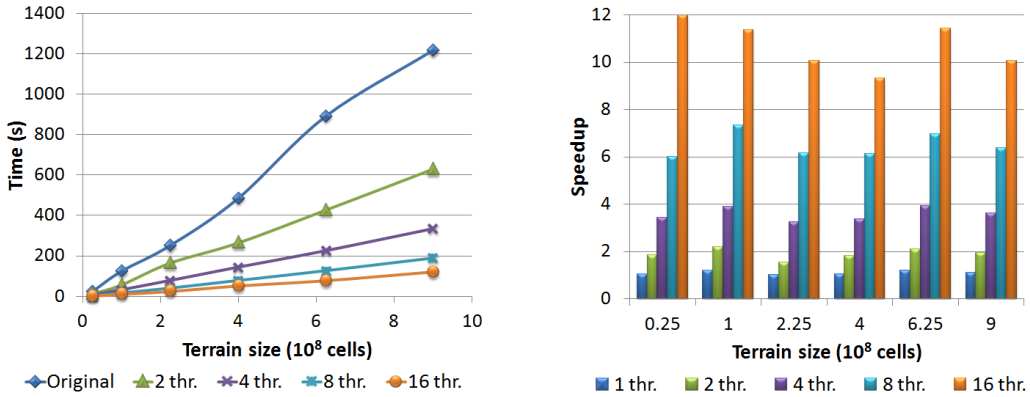
Table 1. Running times for the serial algorithm and the parallel algorithm with different number of threads.

Terrain size		Serial Alg.	Parallel Alg. Number of threads				
# cells	GiB		16	8	4	2	1
$5\,000^2$	0.09	24	2	4	7	13	23
$10\,000^2$	0.37	125	11	17	32	57	105
$15\,000^2$	0.83	252	25	41	78	165	246
$20\,000^2$	1.49	485	52	79	144	265	464
$25\,000^2$	2.33	891	78	128	226	427	740
$30\,000^2$	3.35	1 216	121	191	335	629	1 100

We calculated our algorithm speedup compared to the original algorithm. They are presented in Table 2 and plotted in Figure 5(b). Our algorithm has shown very good performance, achieving up to 12 times speedup, when running 16 concurrent threads. It is also important to notice that with only four threads we achieved a speedup of 3.9 times for two terrains and more than 3 times for all other terrains. Considering that processors

Table 2. Speedups achieved by our parallel algorithm, with different number of threads.

Terrain size		Parallel Alg. Number of threads				
# cells	GiB	16	8	4	2	1
$5\,000^2$	0.09	12.00	6.00	3.43	1.85	1.04
$10\,000^2$	0.37	11.36	7.35	3.91	2.19	1.19
$15\,000^2$	0.83	10.08	6.15	3.23	1.53	1.02
$20\,000^2$	1.49	9.33	6.14	3.37	1.83	1.05
$25\,000^2$	2.33	11.42	6.96	3.94	2.09	1.20
$30\,000^2$	3.35	10.05	6.37	3.63	1.93	1.11



(a) (b)

Figure 5. (a) Running times for the serial algorithm and the parallel algorithm with different number of threads; (b) Speedups achieved by our parallel algorithm, with different number of threads.

with four cores have become usual and relatively cheap nowadays, these improvements may be useful for real users with regular computers. Finally, as discussed in Section 3, the experiments with only one thread show that our strategy can be faster than the original program even with serial architectures.

5. Conclusions and future work

We proposed a new parallel sweep line algorithm for viewshed computation, based on an adaptation of Van Kreveld's algorithm. Compared to the original (serial) algorithm, we achieved speedup of up to 12 times with 16 concurrent threads, and up to 3.9 times using four threads. Even with a single thread, our algorithm was better than the original one, running up to 20% faster.

Compared to other parallel viewshed algorithms, ours seems to be the only to use

Van Kreveld's model, which presents a great tradeoff between efficiency and accuracy [Fishman et al. 2009]. Also, most of them use other parallel models, such as distributed memory/message passing and general purpose GPU programming. On the other hand, ours uses the shared memory model, which is simpler, requires cheaper architectures and is supported by most current computers.

As future work, we can point out the development of other strategies for defining S, the number of sectors into which the region of interest is subdivided. We also intent to develop another adaptation of Van Kreveld's model using GPU programming. Since GPU architectures are much more complex, this will not be a straightforward adaptation.

Acknowledgements

This research was partially supported by FAPEMIG, CAPES, CNPq and NSF.

References

Barney, B. et al. (2010). Introduction to parallel computing. *Lawrence Livermore National Laboratory*, 6(13):10.

Ben-Moshe, B., Ben-Shimol, Y., and Y. Ben-Yehezkel, A. Dvir, M. S. (2007). Automated antenna positioning algorithms for wireless fixed-access networks. *Journal of Heuristics*, 13(3):243–263.

Bresenham, J. (1965). An incremental algorithm for digital plotting. *IBM Systems Journal*, 4(1):25–30.

Champion, D. C. and Lavery, J. E. (2002). Line of sight in natural terrain determined by L_1-spline and conventional methods. In *23rd Army Science Conference*, Orlando, Florida.

Chao, F., Chongjun, Y., Zhuo, C., Xiaojing, Y., and Hantao, G. (2011). Parallel algorithm for viewshed analysis on a modern gpu. *International Journal of Digital Earth*, 4(6):471–486.

Cole, R. and Sharir, M. (1989). Visibility problems for polyhedral terrains. *J. Symb. Comput.*, 7(1):11–30.

Dagum, L. and Menon, R. (1998). Openmp: an industry standard api for shared-memory programming. *Computational Science & Engineering, IEEE*, 5(1):46–55.

De Floriani, L. and Magillo, P. (2003). Algorithms for visibility computation on terrains: a survey. *Environment and Planning B: Planning and Design*, 30(5):709–728.

Fishman, J., Haverkort, H. J., and Toma, L. (2009). Improved visibility computation on massive grid terrains. In Wolfson, O., Agrawal, D., and Lu, C.-T., editors, *GIS*, pages 121–130. ACM.

Franklin, W. R. and Ray, C. (1994). Higher isn't necessarily better: Visibility algorithms and experiments. In Waugh, T. C. and Healey, R. G., editors, *Advances in GIS Research: Sixth International Symposium on Spatial Data Handling*, pages 751–770, Edinburgh. Taylor & Francis.

Franklin, W. R., Ray, C. K., Randolph, P. W., Clark, L., Ray, K., and Mehta, P. S. (1994). Geometric algorithms for siting of air defense missile batteries.

Lake, I. R., Lovett, A. A., Bateman, I. J., and Langford, I. H. (1998). Modelling environmental influences on property prices in an urban environment. *Computers, Environment and Urban Systems*, 22(2):121–136.

Lee, J. and Stucky, D. (1998). On applying viewshed analysis for determining least-cost paths on digital elevation models. *International Journal of Geographical Information Science*, 12(8):891–905.

Li, Z., Zhu, Q., and Gold, C. (2005). *Digital Terrain Modeling — principles and methodology*. CRC Press.

Maćiorov, F. (1964). *Electronic digital integrating computers: digital differential analyzers*. Iliffe Books (London and New York).

Magalhães, S. V. G., Andrade, M. V. A., and Franklin, W. R. (2011). Multiple observer siting in huge terrains stored in external memory. *International Journal of Computer Information Systems and Industrial Management (IJCISIM)*, 3:143–149.

Osterman, A. (2012). Implementation of the r. cuda. los module in the open source grass gis by using parallel computation on the nvidia cuda graphic cards. *ELEKTROTEHNISKI VESTNIK*, 79(1-2):19–24.

Shapira, A. (1990). Visibility and terrain labeling. *Master's thesis, Rensselaer Polytechnic Institute*.

Van Kreveld, M. (1996). Variations on sweep algorithms: efficient computation of extended viewsheds and class intervals. In *In Proceedings of the Symposium on Spatial Data Handling*, pages 15–27.

Zhao, Y., Padmanabhan, A., and Wang, S. (2013). A parallel computing approach to viewshed analysis of large terrain data using graphics processing units. *International Journal of Geographical Information Science*, 27(2):363–384.

Algoritmo paralelo usando GPU para o posicionamento de observadores em terrenos

Guilherme C. Pena[1], **Salles V. G. Magalhães**[1], **Marcus V. A. Andrade**[1], **Chaulio R. Ferreira**[1]

[1]Departamento de Informática - Universidade Federal de Viçosa (UFV)
Campus da UFV – 36.570-000 – Viçosa – MG – Brasil

{guilherme.pena,salles,marcus,chaulio.ferreira}@ufv.br

Abstract. *This paper presents an efficient method for sitting a set of observers to maximize the coverage of a given terrain. It is based on an efficient implementation of a local search heuristic using dynamic programming and GPU parallel programming. The tests showed that the proposed method can be more than 200 times faster than the conventional implementation (with no use of dynamic programming and GPU parallel programmming).*

Resumo. *Este artigo apresenta um método eficiente para o posicionamento de um conjunto de observadores de modo a maximizar a cobertura de um dado terreno. A estratégia proposta se baseia na implementação eficiente de uma heurística de busca local utilizando programação dinâmica e programação paralela (em GPU). Os testes mostraram que esse método chega a ser acima de 200 vezes mais rápido que a implementação convencional (sem o uso de programação dinâmica e paralela).*

1. Introdução

Um importante grupo de aplicações na área de Sistemas de Informação Geográficas envolve o conceito de visibilidade em terrenos, que consiste em determinar os pontos do terreno que são visíveis a partir de um dado ponto (*observador*). Muitas aplicações práticas, tais como telecomunicações, planejamento ambiental, e monitoramento militar, envolvem esse conceito [Ben-Moshe et al. 2007, Bespamyatnikh et al. 2001]. Um problema importante relacionado à visibilidade é o posicionamento de observadores para maximizar a "cobertura do terreno", sendo que esses "observadores" podem ser câmeras, torres de vigilância ou de telefonia móvel, etc. [Ben-Moshe et al. 2007, Franklin and Vogt 2006, Eidenbenz 2002]. Conforme demonstrado em [Nagy 1994] esse problema é NP-Completo e, portanto, não se conhece um algoritmo eficiente para resolvê-lo de maneira exata.

Mas até mesmo a obtenção de soluções aproximadas para esse problema de otimização demanda um alto tempo de processamento, principalmente ao processar grandes volumes de dados. Uma forma de reduzir o tempo de processamento consiste em desenvolver algoritmos paralelos baseados nas unidades de processamento gráfico de propósito geral (*General Purpose Graphics Processing Units (GPGPUs)*), que estão presentes na maioria das placas gráficas atuais.

Este trabalho apresenta uma implementação paralela utilizando *Graphics Processing Units (GPUs)* de uma heurística para solucionar o problema do posicionamento de

observadores em um terreno representado por um Modelo Digital de Elevação (MDE) *raster*. Mais precisamente, a heurística proposta visa maximizar a área do terreno coberta por um dado número de observadores.

2. Referencial Teórico

2.1. Visibilidade - Definições

Um *terreno* é uma representação da elevação da superfície terrestre em uma determinada região que, neste trabalho, será representado por uma matriz de elevação (ou *MDE*) que armazena a elevação de amostras do terreno regularmente espaçadas [Felgueiras 2001].

Um *observador* é um ponto do espaço que "deseja" visualizar ou se comunicar com outros pontos do espaço chamados de *alvos*. Tanto o observador como o alvo podem estar a uma certa altura acima do terreno. Existem vários algoritmos para o cálculo da visibilidade e, em geral, eles adotam que um alvo T é *visível* a partir de um observador O se, e somente se, T estiver dentro do raio de interesse de O e não houver nenhum ponto do terreno bloqueando o segmento de reta que conecta O a T.

O *problema de posicionamento de observadores* consiste em posicionar um conjunto de observadores de modo a maximizar o número de pontos visíveis no mapa de visibilidade (*viewshed*) acumulado desses observadores [Young-Hoon Kim et al. 2004, Magalhães et al. 2010].

2.2. CUDA e programação paralela de propósito geral

Atualmente, uma técnica de programação paralela que tem sido muito utilizada é a *GPGPU* [CUDA Programming Guide 2007], que consiste em se utilizar os vários núcleos de processamento presentes em *GPUs* para processar dados. O modelo de programação *Compute Unified Device Architecture (CUDA)* facilita o desenvolvimento de aplicativos capazes de aproveitar o potencial de processamento das *GPUs* da NVIDIA, que são compostas por vários processadores do tipo *Single Instruction, Multiple Data (SIMD)* e possibilitam a execução de milhares de *threads* de forma paralela.

A arquitetura *CUDA* permite a execução de código tanto no processador principal (*CPU*) quanto na *GPU*. Assim, o *CPU* pode executar trechos de código que envolvam menos paralelismo e maior quantidade de estruturas de controle de fluxo de execução (que normalmente não são processadas de forma eficiente em arquiteturas *SIMD*) e designar para a execução na *GPU* funções que possam ser aplicadas de forma paralela sobre diferentes elementos de dados. Dessa forma, a *GPU* é utilizada como um co-processador que é capaz de executar certos tipos de tarefas de forma mais eficiente do que a *CPU*.

3. A heurística proposta

A solução proposta neste artigo consiste em implementar uma heurística de busca local (*Swap*) de forma eficiente utilizando *CUDA* e programação dinâmica. Mais precisamente, a heurística será utilizada para aumentar a área visível de soluções previamente obtidas utilizando o método *Site* [Franklin and Vogt 2006], que é baseado em uma heurística gulosa. Assim, dada uma solução obtida pelo método *Site* (ou por qualquer outro método), o método proposto tenta melhorar esta solução realizando operações de busca local baseadas em trocas de observadores (*viewsheds*) para aumentar o número de pontos visíveis no *viewshed* acumulado, mantendo fixo o número de observadores utilizados na solução.

É importante ressaltar que em [Magalhães et al. 2011] é apresentada uma solução, usando programação paralela em GPU, para uma variação do problema de posicionamento de observadores onde o objetivo é "minimizar" o número de observadores selecionados para alcançar uma taxa de cobertura do terreno. Embora os problemas sejam semelhantes, as soluções (e implementações) adotadas são bem diferentes.

3.1. Heurística de busca local *Swap*

Dado um conjunto de n observadores candidatos, seja $V = \{V_1, \cdots V_n\}$ o conjunto com os respectivos *viewsheds* e seja S um subconjunto de V contendo k *viewsheds* representando uma solução para o problema de posicionamento de observadores. Assim, o objetivo da heurística *Swap* é realizar uma busca local para maximizar o número de pontos visíveis no *viewshed* acumulado de k observadores selecionados dentre os n candidatos.

Mais especificamente, a idéia da heurística *Swap* consiste em verificar iterativamente todas as soluções vizinhas à solução S e, a cada passo, substituir S pela solução vizinha com maior área visível. Dada uma solução S, as vizinhas de S são todas as soluções S' onde exatamente um dos *viewsheds* em S' é diferente de um *viewshed* de S. Por exemplo, se $V = \{V_1, V_2, V_3, V_4\}$ e $S = \{V_1, V_2, V_3\}$, as soluções vizinhas a S são: $\{V_1, V_2, V_4\}$, $\{V_1, V_4, V_3\}$, $\{V_4, V_2, V_3\}$. O processo de trocar a solução atual pela melhor vizinha é repetido até ser obtida uma solução que não possua melhores vizinhas.

Assim, considerando cada *viewshed* representado por uma matriz linearizada de inteiros (onde os números 0 e 1 representam, respectivamente, pontos não visíveis e visíveis), a heurística *Swap* pode ser implementada (sequencialmente) da seguinte forma: dados os *viewsheds* candidatos $V_1, \cdots V_n$, seja S uma solução composta por k *viewsheds*, isto é, $S = \{V_{i_1}, \cdots, V_{i_k}\}$ cujo *viewshed* acumulado é dado por $V_{i_1} \oplus \cdots \oplus V_{i_k}$ onde $\oplus$ representa a união de dois *viewsheds*[1]. Além disso, seja Υ_r o *viewshed* acumulado envolvendo todos os *viewsheds* em S exceto o *viewshed* V_{i_r}. Daí, em cada iteração da heurística são geradas todas as soluções vizinhas a S e a melhor delas é selecionada para ser a solução corrente utilizada na próxima iteração. As soluções vizinhas a S são geradas fazendo-se $\Upsilon_r \oplus V_j$ para $r = 1 \cdots k$ e $j = 1 \cdots n$.

A etapa que demanda maior tempo de processamento na heurística *Swap* é o cálculo da área visível para cada solução vizinha. O Algoritmo 1 apresenta o pseudocódigo dessa etapa, onde $Area[r][j]$ representa o número de pontos visíveis em $\Upsilon_r \oplus V_j$ e, portanto, essa matriz pode ser utilizada para se obter qual é a melhor vizinha de S.

Algoritmo 1 Calcula a matriz $Area$; *VSize* é o número de pontos em cada *viewshed*.

```
1: Area[1..k][1..n] ← {{0}}
2: for r = 1 → k
3:    for j = 1 → n
4:       for w = 1 → VSize
5:          Area[r][j] ← Area[r][j] + (Υ[r][w] or V[j][w])
```

Por motivos de eficiência, neste trabalho os *viewsheds* foram codificados em palavras de 64 bits (onde cada palavra representa a visibilidade de 64 pontos no terreno).

[1]A união de dois *viewsheds* pode ser realizada aplicando-se a operação **or** binária entre cada ponto de um *viewshed* e o ponto correspondente do outro.

Com isso, a união de *viewsheds* e o cálculo da área visível podem ser realizadas utilizando, respectivamente, o operador **or** de bits e funções de contagem de população, que são operações disponibilizadas no *hardware* da maioria dos computadores atuais.

3.2. Implementação eficiente da heurística *Swap*

Note que, para se gerar os valores de Υ_r com $r = 1 \cdots k$, o algoritmo descrito na seção 3.1 realiza $k \times n = \Theta(kn)$ operações $\oplus$ de união de *viewsheds* (sendo que cada união envolve Θ*(VSize)* posições na matriz dos *viewsheds*).

Conforme descrito a seguir, a eficiência dessa etapa do algoritmo pode ser melhorada consideravelmente utilizando-se programação dinâmica. Para simplificar a notação, sejam $\nu_1, \cdots \nu_k$ os k *viewsheds* selecionados para formar a solução S, ou seja, $\nu_1, \cdots \nu_k = V_{i_1} \cdots V_{i_k}$. Assim, $\Upsilon_r = (\nu_1 \oplus \nu_2 \oplus ... \oplus \nu_{r-1}) \oplus (\nu_{r+1} \oplus \nu_{r+2} \oplus \cdots \oplus \nu_k)$. Sejam $\lambda_r = (\nu_1 \oplus \nu_2 \oplus ... \oplus \nu_{r-1})$ e $\overline{\lambda}_r = (\nu_{r+1} \oplus \nu_{r+2} \oplus \cdots \oplus \nu_k)$. Note que λ_r pode ser obtido com base na seguinte equação de recorrência: $\lambda_1 = \emptyset$ e $\lambda_r = (\nu_1 \oplus \nu_2 \oplus ... \oplus \nu_{r-1}) = (\lambda_{r-1} \oplus \nu_{r-1})$. De forma análoga, temos que $\overline{\lambda}_k = \emptyset$ e $\overline{\lambda}_r = (\nu_{r+1} \oplus \nu_{i+2} \oplus \cdots \oplus \nu_k) = (\nu_{r+1} \oplus \overline{\lambda}_{r+1})$. Dai, $\Upsilon_r = \lambda_r \oplus \overline{\lambda}_r$. Essas relações de recorrência foram utilizadas para implementar um algoritmo baseado em programação dinâmica para o cálculo da matriz que representa os valores de Υ, sendo que esse algoritmo realiza $\Theta(k)$ uniões de *viewsheds*.

Após o cálculo de Υ, o próximo passo consiste em utilizar esses valores para realizar uma iteração da heurística *Swap*. Uma forma de se implementar essa etapa na GPU consiste em manter todos os *viewsheds* na memória global da GPU e, então, utilizar cada *thread* para calcular cada um dos elementos da matriz $Area$. Essa implementação convencional não aproveitaria de forma eficiente todo o potencial da GPU, pois os acessos seriam realizados na memória global, que é mais lenta do que outras memórias, como a memória compartilhada. Como a capacidade da memória compartilhada é muito pequena, ela não é capaz de armazenar todos os dados e, portanto, é necessário adotar uma estratégia de dividir o processamento para que os cálculos possam ser realizados por partes.

Note que, no Algoritmo 1, as matrizes $Area$, Υ e V são acessadas de forma similar ao padrão de acesso utilizado em um algoritmo de multiplicação de matrizes, onde as matrizes Υ e V^T são multiplicadas para gerar a matriz $Area$. A única diferença entre o Algoritmo 1 e um algoritmo de multiplicação de matrizes é que, no algoritmo de multiplicação, a operação $Area[r][j] \leftarrow Area[r][j] + (\Upsilon[r][w]$ **or** $V[j][w])$ (linha 5) é substituída pela operação $Area[r][j] \leftarrow Area[r][j] + \Upsilon[r][w] \times V^T[w][j]$.

Assim, podemos utilizar as técnicas de otimização adotadas na implementação em GPU de algoritmos de multiplicação de matrizes. Neste trabalho, adaptou-se o algoritmo de multiplicação de matrizes descrito em [CUDA Programming Guide 2007], substituindo a operação de multiplicação por uma operação de **or** binário seguida por uma operação de contagem de população. Esse algoritmo divide as matrizes em blocos, que são carregados iterativamente na memória compartilhada à medida em que o processo de multiplicação é realizado. Com isso, a maior parte dos acessos realizados pelo algoritmo é feita na memória compartilhada, que é mais rápida do que a memória global.

4. Resultados experimentais

Para avaliar o desempenho do método proposto, foram implementadas duas versões da heurística *Swap*: uma versão utilizando o método convencional para cálculo de Υ e

o algoritmo sequencial para o cálculo da matriz $Area$ e uma outra versão utilizando programação dinâmica para o cálculo de Υ e programação paralela (em GPU) para a obtenção da matriz $Area$. Essas duas versões foram implementadas em C++ e compiladas, respectivamente, com os compiladores *g++* 4.6.4 e *nvcc* 4.0 (ambos com nível máximo de otimização). Foram realizados testes em um computador com processador Intel Xeon E5-2687 3.1GHz, 128GiB de memória RAM e GPU NVidia Tesla Kepler K20x, que possui 2688 núcleos de processamento CUDA e 6GiB de memória global.

As implementações foram testadas em dois terrenos obtidos a partir do projeto *SRTM* da NASA, com dimensões 1201×1201 e 3601×3601 células. Os testes para cada terreno foram executados variando-se o número total de pontos candidatos e o número de observadores selecionados. A Tabela 1 apresenta os resultados obtidos.

Tabela 1. Tempos de execução da heurística convencional (em CPU) e da heurística proposta; neste caso, são apresentados os tempos para obtenção do *viewshed* acumulado (Υ), da área visível e tempo total incluindo operações de entrada e saida.

Ter.	#Can.	#Obs.	Cob.	Tempo de processamento (em seg.)						
				$\Upsilon(s)$		Area (s)		Total (s)		
				Conv.	P.D.	CPU	GPU	Conv.	Proposta	
1201 × 1201	500	16	25%	0.1	0.1	17.4	0.1	17.6	0.9	(20x)
		32	39%	1.3	0.1	98.7	0.5	100	1.6	(63x)
		64	47%	9.2	0.3	351	1.5	360	3.4	(106x)
	1000	32	55%	1.1	0.1	175	0.7	177	1.9	(93x)
		64	83%	10.7	0.4	829	3.1	839	5.2	(161x)
		128	95%	94	1.6	3363	12	3457	18	(192x)
		256	98%	640	5.5	11129	39.1	11769	56.8	(207x)
3601 × 3601	500	16	1%	0.4	0.1	52.3	0.4	53	2.6	(20x)
		32	1%	2.6	0.1	183	0.9	186	3.6	(52x)
		64	2%	18	0.6	635	2.8	654	6.9	(95x)
	1000	32	2%	2.2	0.2	314	1.3	317	5.1	(62x)
		64	3%	23.9	0.8	1689	6.2	1713	12.8	(134x)
		128	5%	175	3	6083	21.7	6259	35.2	(178x)
		256	7%	1375	12	24114	83.7	25489	126	(202x)
	2000	32	2%	2.2	0.2	636	2.3	639	8.7	(73x)
		64	4%	13.4	0.5	1880	6.7	1895	14.2	(133x)
		128	6%	192	3.3	13381	46.9	13575	64	(212x)
		256	10%	1320	11.5	46008	159	47329	203	(234x)

Os resultados apresentados na Tabela 1 representam todas as iterações da heurística até atingir um máximo local, ou seja, até ser obtida uma solução que não possui melhores vizinhas. Além disso, é apresentado o *speedup* da implementação proposta em relação à implementação convencional das heurísticas e como pode-se notar, os ganhos são melhores à medida que as configurações de número de observadores, número de *viewsheds* candidatos e quantidade de células do terreno aumentam.

5. Conclusão

Foi proposta uma heurística de busca local *Swap* eficiente para a solução do problema de posicionamento de observadores. O método proposto combina técnicas de programação dinâmica com programação paralela em CUDA e chega a ser acima de 200 vezes mais rápido do que uma implementação convencional.

A eficiência obtida é importante não só para o processamento de grandes quantidades de dados, mas também para a implementação de outras heurísticas (como *GRASP* e *ILS*) [Magalhães et al. 2010], que normalmente executam várias vezes heurísticas de busca local (como a *Swap*).

Como trabalhos futuros, pretende-se analisar o impacto da heurística proposta em outros métodos que realizam buscas locais. Além disso, pretende-se também desenvolver outras métodos de busca local para o problema de posicionamento de observadores.

Agradecimentos

Este trabalho foi parcialmente financiado pela FAPEMIG, CNPq e CAPES.

Referências

Ben-Moshe, B., Ben-Shimol, Y., and Y. Ben-Yehezkel, A. Dvir, M. S. (2007). Automated antenna positioning algorithms for wireless fixed-access networks. *Journal of Heuristics*, 13(3):243–263.

Bespamyatnikh, S., Chen, Z., Wang, K., and Zhu, B. (2001). On the planar two-watchtower problem. In *In 7th International Computing and Combinatorics Conference*, pages 121–130, London, UK. Springer-Verlag.

Eidenbenz, S. (2002). Approximation algorithms for terrain guarding. *Inf. Process. Lett.*, 82(2):99–105.

Felgueiras, C. A. (2001). Modelagem numérica de terreno. In G. Câmara, C. Davis, A. M. V. M., editor, *Introdução à Ciência da Geoinformação*, volume 1. INPE.

Franklin, W. R. and Vogt, C. (2006). Tradeoffs when multiple observer siting on large terrain cells. In Springer-Verlag, editor, *12th International Symposium on Spatial Data Handling*, pages 845–861.

Magalhães, S. V. G., Andrade, M. V. A., and Ferreira, C. (2010). Heuristics to site observers in a terrain represented by a digital elevation matrix. In *GeoInfo*, pages 110–121.

Magalhães, S. V. G., Andrade, M. V. A., and Ferreira, R. S. (2011). Using gpu to accelerate heuristics to site observers in dem terrains. In *IADIS Applied Computing (AC 2011)*, pages 127–133. Rio de Janeiro.

CUDA Programming Guide (2007). `http://www.nvidia.com/object/cuda_develop.html` (accessed on Aug 2013).

Nagy, G. (1994). Terrain visibility. *Computers & graphics*, 18(6):763–773.

Young-Hoon Kim, Rana, S., and Wise, S. (2004). Exploring multiple viewshed analysis using terrain features and optimisation techniques. *Computers & Geosciences*, 30:1019–1032.

A geo-ontology to support the semantic integration of geoinformation from the National Spatial Data Infrastructure

Paulo J. A. Gimenez, Astério K. Tanaka, Fernanda Baião

Programa de Pós-Graduação em Informática – Departamento de Informática Aplicada
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
{paulo.gimenez, tanaka, fernanda.baiao }@uniriotec.br

Abstract. *The profusion of geoinformation and the diversity of providers increasingly demand availability of integrated geoinformation. The Brazilian National Spatial Data Infrastructure (INDE) and the Web 2.0 technologies provide the existence of such solutions. In this context, geoinformation needs to be discovered and processed, seamlessly and semantically interoperable through the services offered by INDE, independently of the existing technological arrangement. It is essential to have an information model that represents the involved concepts, their characteristics and their relationships. The development of a geo-ontology, in order to consolidate aspects of spatial metadata, spatial features and geographic names, is the challenge addressed by this work.*

1. Introduction

Over the last decades of evolution and exponential growth of geospatial content, there have been increasing demands for interoperability, integration and sharing of geoinformation, and increased ability to provide geospatial knowledge, especially by public organizations to attend multidisciplinary demands.

Recent studies address geospatial semantics through the use of consistent and well formed ontologies, as applied to the semantic objects when querying a heterogeneous geospatial database [Bishr 2008].

In order to provide an effective and efficient access, as well as to make spatial data available for all sociopolitical entities, many governments and public organizations have developed their Spatial Data Infrastructures (SDI), which aggregate the main functions of communication and spatial reasoning [Bishr 2008] of governmental actions and policies, emphasizing transparency and governance [Gimenez et al. 2013] [CONCAR 2010].

An SDI provides the representation of syntactic Web [Janowicz et al. 2012] based on the technology stack from the Open Geospatial Consortium (OGC) standards of geo-services with the architecture defined by the Global Spatial Data Infrastructure Association (GSDI). The services available in an SDI support syntactic and systemic integration describing interfaces for using geospatial data, in contrast to Semantic Web technologies.

The National Spatial Data Infrastructure (INDE) is a Brazilian initiative to set up an SDI, with national scope, bringing together public and private organizations, with focus on dissemination of geoinformation in Brazilian context. Its three key

characteristics are the use of standardized metadata, semantic and syntactic interoperability [CONCAR 2010].

The INDE follows the definitions of the e-PING[1] architecture, which in turn references the OGC standards, adding other standards and specifications of the geographical domain in the Brazilian context: (i) MGB (Brazilian Geographic Metadata), a profile of the geospatial metadata standard ISO 19115:2003, which provides facilities for publishing, searching and exploring geospatial data; (ii) ET-EDGV (Technical Specification for Geographic Vector Data Structure), which describes the classes of geographic objects (geo-objects) and their interrelationships, providing a conceptual data model with attributes detailing [Gimenez et al. 2013].

The need for formalization and representation of the geoinformation at the semantic level, to face the challenges of SDI interoperability, was addressed by the OGC [Lieberman 2007]. In that work, the OGC issues regarding geospatiality (features and geometries of features, geographic and non-geographic relationships, systems and coordinates, scales conflicts) and geosemantics (discernment of a feature, spatial reasoning and representation dissonance) were the biggest challenges for semantic integration based on the characteristics of the domain. This field is still open, in that there is an increased availability of geoinformation that amplifies the need for geospatial semantic processing, in order to dynamically provide more elaborate geoinformation without redundancy [Gimenez et al. 2013][Diaz et al. 2012].

The objective of this research is to present an approach to the creation of a geo-ontology for representing geographic objects within the context of the INDE under the ET-EDGV and other associated standards, using the MGB profile and existing geographical names, as well as enabling the discovery and integration of geoinformation.

The remainder of this paper is organized as follows. Section 2 discusses the concepts and technologies applied to the field of geoinformation, the INDE and the development of geo-ontology. Section 3 presents the methodological approach for ontology engineering. Section 4 presents the sets of ontologies defined. Section 5 presents an application scenario of the geo-ontology. Section 6 presents some related works and proposals. Finally, Section 7 concludes and outlines future work.

2. Geoinformation and geo-ontologies: the Brazilian context

A geospatial data is a particular case of spatial data in which the spatial component refers to its position on the Earth and its space in a specific moment or period of time [Soares, Tanaka and Baião 2010]. The term comes from the geospatial association to the geoid concept, which is the physical model for the shape of the Earth or the equipotential surface (surface of constant gravitational potential) obtained by considering the mean value of the average level of the sea. The geoid surface is in fact more irregular than the ellipsoid of revolution usually used to approximate the shape of the planet and represent entities in relation to it [Bédard, Rivest and Proulx 2005].

In this way, expressions and mathematical models are used to derive the shape of the geoid. While there is an international ellipsoid (globally representing the Earth from a

[1] http://www.governoeletronico.gov.br/acoes-e-projetos/e-ping-padroes-de-interoperabilidade

"zero-point", or point of reference), countries and international organizations are allowed to prepare their own local ellipsoids in order to view their region from an internal central point, thus generating a new geographic reference system. In Brazil, SIRGAS2000[2] is defined as the standard geographic reference system.

Geographic entities can be conceptualized in two different perspectives: geo-field (spatial data as a set of continuous distribution) and geo-object (spatial data that is discrete and identifiable throughout the world) [Fonseca 2008]. A geospatial object (geo-object perspective) is defined by its attributes and their spatial distribution [Zhou 2008].

2.1 The Brazilian context

The National Digital Cartographic Library (MND) [Lunard and Augusto 2006] represents the Brazilian geographic space in three parts: (i) matrix data, (ii) vector data and (iii) metadata.

Matrix data represents the geo-field perspective, and follows the ET-PCDG (Technical Specification for products of Geospatial Datasets) standard [CONCAR 2010].

Vector data represents the geo-object perspective, and follows the ET-EDGV standard [CEMND-CONCAR, 2008]. Its current version (2.1) addresses reference geospatial data for Topographic Systematic Terrestrial Mapping [CONCAR 2010], and a future version will include Cadastral Systematic Terrestrial Mapping.

The metadata structure is defined by the MGB Profile [CEMG-CONCAR 2009], which aims to promote documentation, integration and deployment of geospatial data as well as to enable search and exploration, while avoiding duplications. The MGB profile is structured in sections according to their objectives: (i) to identify the producer and technical production responsibility, (ii) to standardize terminology, (iii) to ensure data sharing and transfer, (iv) to facilitate information integration, (v) to enable quality control, and (vi) to ensure minimum availability requirements. The MGB profile is recommended for the description of geospatial reference data and has two versions: a full version and a summarized version, which is based on "Core Metadata for Geographic Datasets" ISO 19115:2003 with the addition of the Status attribute. Nowadays only includes metadata specification for data sets, does not cover services (ISO 19119), and also the implementation (ISO 19139).

Additionally, the Action Plan for the Implementation of INDE [CONCAR 2010] intends to consider the ET-BNGB standard, that is still under development, but is already being implemented on the Brazilian Geographical Names database [IBGE 2010], which deals with the geographical names used in the Brazilian systematic mapping.

2.2 Geo-ontology

A geo-ontology (or geospatial ontology) is an ontology that aims at describing spatial factors, spatial relationships, physical facts, subjects, collections of data and geospatial computing models [Di and Zhao 2008].

[2] ftp://geoftp.ibge.gov.br/documentos/geodesia/projeto_mudanca_referencial_geodesico/legislacao file rpr_01_25fev2005.pdf

Wang et al. [Wang, Li and Song 2008] proposed the following formulation:

Geo-ontology = {C, R, A, X, I}, where: C is the set of concepts of a geographical object, R is a set of relationships and the description of this set on the concepts, A is the set of attributes of the geographic object, X represents axioms and constraint rules on concepts, relations and attributes, and I is a set of instances.

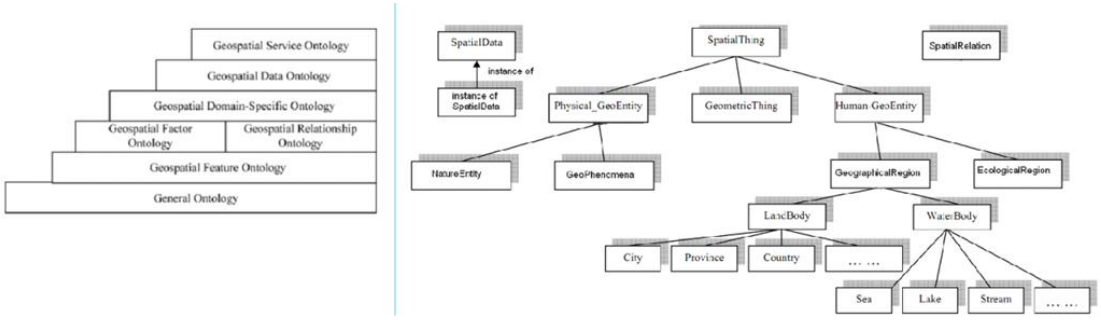


Figure 1: (a) Hierarchy of Geospatial ontologism [Di and Zhao 2008]. (b) Structure of geographic space [Adapted from Wang ET AL. 2007]

The geographic entities defined in the ET-EDGV standard may be arranged according to Wang's proposal [Wang ET AL. 2007] (illustrated in Figure 1(b)). With this arrangement, some concepts, relationships, conditions and attributes can represent geospatial ontologies of different levels as proposed by Di and Zhao [2008] (illustrated in Figure 1(a)).

3. Methodological approach for the construction of the proposed geo-ontology

This section presents the methodological approach we follow for constructing a geo-ontology for INDE. The proposed ontology was built by combining existing methodologies from the literature: the Simple Knowledge-Engineering Methodology defined in [Noy and McGuinness 2001] and the works from Wang, Li and Song [2008] and [Bishr 2008]. The Simple Knowledge-Engineering Methodology was adopted due to its simplicity and efficiency, while the methodologies of Wang, Li and Song [2008] and [Bishr, 2008] were considered to specifically address the geographical domain.

According to the Simple Knowledge-Engineering Methodology, the ontology designer defines a set of competency questions (CQ) to determine the scope of the ontology. After that, he/she follows a set of steps to: (1) determine the domain and scope; (2) reuse existing ontologies; (3) list important terms in the ontology; (4) define classes and class hierarchies; (5) define class properties; (6) define facets of these properties; and (7) create instances.

The methodology from Wang et al. [2008] is composed by the following steps: (i) confirm the scope of geo-ontology, (ii) list ontological properties (properties that describe the object in essence) for the geographical concept, (iii) ensure the relationship between geographical concepts; (iv) collect concepts meaning, their attributes, images and instances, and (v) build the prototype model / geospatial ontology system. Wang's approach uses the "Concept lattice", which is defined as sets of objects and attributes

from geographic concepts that represent the main aspects of the geospatial domain: is-a, kind-of, part-whole, dependency, instantiation and member relationships, as well as the relationship between attributes and concepts.

Bishr [Bishr 2008] states that some elements must be observed during the construction of the geo-ontology concepts: (i) the context: establish the set of assertions and conditions considering a restricted vocabulary and spatial-temporal perspectives; (ii) identity criteria: establish sufficient conditions to determine the identity of a concept, organize the taxonomy of concepts and persist in time; (iii) spatial reference system: to represent the location concept for absolute or relative position; (iv) Mereotopology: incorporate relationships between assemblies, parts, parts of parts and boundaries between the parts in space; (v) limits: distinguish boundaries between "bona-fide" (intrinsic things) and "fiat" (marked as human cognitive effect) for the generalization of the concepts and treatment of co-localization of spatial objects; and (vi) the shape and size: characterize qualitatively ("has hole", is hollow, is a piece of something larger, is complete) and quantitatively (may assist in identification) for a feature size.

4. The proposed INDE geo-ontology

This section presents our proposed geo-ontology, following the steps of the methodology presented in the previous section. The proposed geo-ontology will serve as a basis for the semantic integration of spatial data within INDE.

The following competency questions were established, as required by the methodology from [Noy and McGuinness 2001]:

CQ1: Which conditions or characteristics are required by a Geographic name so that it addresses (identifies) a Geographic Feature?
CQ2: Which conditions or characteristics are required by a Geographic Metadata so that it can be associated to a Geographic Name when identifying Geographic Feature?
CQ3: How are the needs for cartographic generalization of geographic features be identified?
CQ4: How can we identify the same object being represented as distinct cartographic features using different scales?

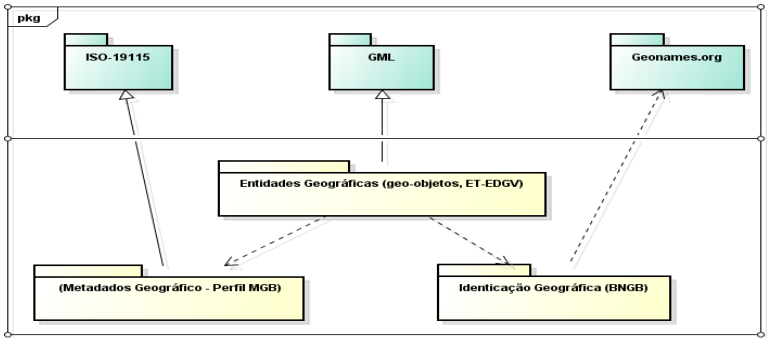


Figure 2: Overview of the INDE geo-ontology

To address these issues and serve as a basis for the semantic integration of different knowledge bases containing geographical data, we propose a geo-ontology composed by three sub-ontologies, as illustrated in Figure 2: (i) geographical names

ontology ("*Identificação Geográfica*"); (ii) MGB profile ontology ("Metadados *Geográficos*"); (iii) geographic entities based on geo-objects.

4.1. Geographical Names ontology

This sub-ontology describes the basic features for the representation of Geographic Names, as defined in [IBGE 2010] and in [Lima 2011]. The Geographic name or toponym standard allows the identification of a Geographic Feature or Accident. A toponymic phrase consists of two parts: the element on the geographical entity that receives the name (generic term) and the element that distinguishes the identity of the geographic element (specific term) [IBGE 2010, Lima 2011 apud Dick 1990].

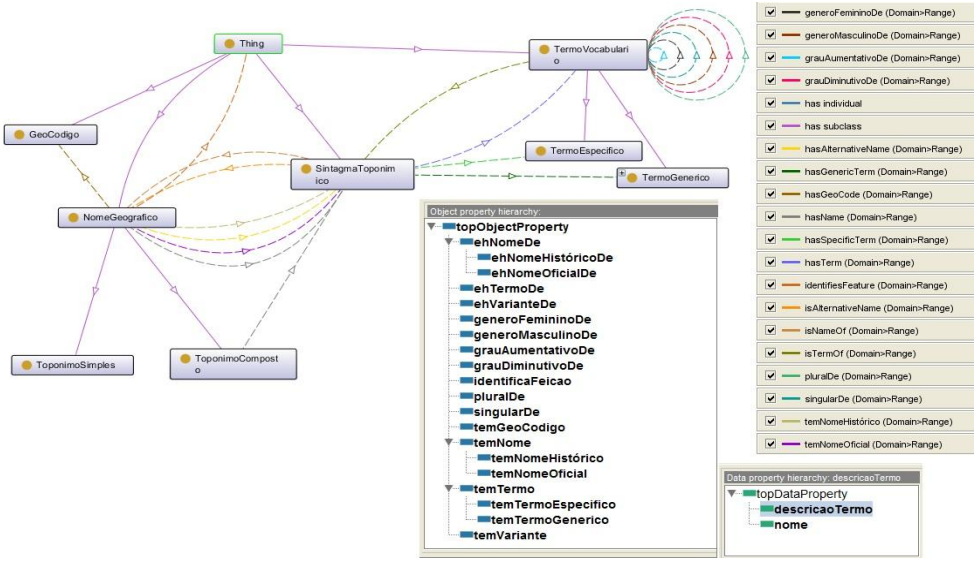


Figure 3: The Geographical Names ontology – basic view

Figure 3 shows the correlation between term and toponyms. It also encompasses lists of generic terms (as proposed in [IBGE 2010]), names denoting variation in gender (male x female) and number (plural x singular), alternative toponymic phrases for a particular geographical area, as well as the possible existence of geographical names composed of multiple toponymic phrases [IBGE 2010] [Lima 2011].

4.2. Brazilian Geographic Metadata Profile ontology

Ontologies for geographic metadata (e.g., ISO 19115 ontology) add semantic meaning and relationship to describe the underlying data [Di and Zhao 2008]. This extends the capability of geospatial data being discovered, used and semantically interoperable from the geo-ontologies of geospatial data or services.

This sub-ontology describes the basic features for the representation of concepts defined in the MGB Profile. The nature of the terms, their interrelations and dependencies, composition and classification between the two versions of the MGB Profile (Summarized and Full) were considered. The sections and entities referred in the profile specification were generally represented as classes in the ontology, while information and elements were typically represented as properties and enumerated lists.

The MGB Profile [CEMG-CONCAR 2009] has several information elements shared among several sections, many of them being referred as distinct terms in different sections. In those cases, we have preserved the distinct terminology and added synonym relationships to the ontology.

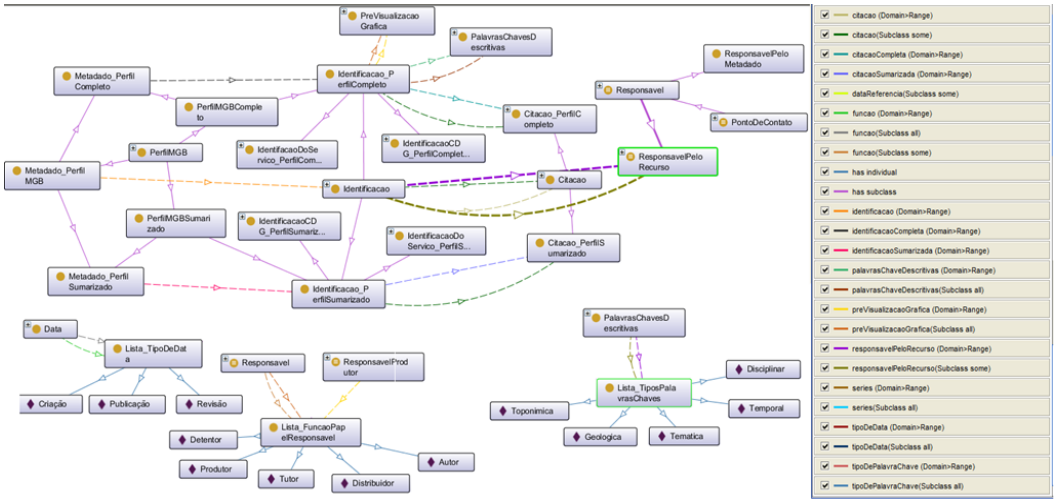


Figure 4: Brazilian Geographic Metadata (MGB Profile) ontology – partial view

As seen in Figure 4, the proposed MGB Profile ontology concepts is based on the ISO-19115 ontology (Metadata Application), which in turn makes use of several other ISO ontologies, including ISO-19103 (Conceptual Schema Language), ISO-19107 (Spatial Schema), GML (Geographic Markup Language), ISO 19111 (Spatial Referencing by Coordinates), ISO 19112 (Spatial Referencing by Geographic Identifier), ISO 19109 (Rules for Application Schema) and ISO-19108 (Temporal Schema). The concepts were all based on the MGB profile, but their characterizations as properties and fields followed the ISO-19115 where not defined by the profile [CEMG-CONCAR 2009] or in cases of distinct definitions [Diniz 2013] [Pascoal, Carvalho and Xavier 2013]. Cardinality restrictions were mapped according to the specifications in Table 1.

Table 1 – Mapping Cardinality for Ontology

MGB Profile		Ontology
Mandatory	**Cardinality**	**Mapping**
Yes	**1**	**[ExactCardinality 1]**
Yes	**N (multiple)**	**[someValueFrom]**
No	**1**	**[MinCardinality 0 MaxCardinality 1]**
No	**N (multiple)**	**[MinCardinality 0]**
Conditional	**1**	**[subClasses] hierarchy and/or [ExactCardinality 1] or [MinCardinality 0 MaxCardinality 1]**
Conditional	**N (multiple)**	**[subClasses] hierarchy and/or [someValueFrom] or [MinCardinality 0]**

4.3. Brazilian Geographic Domain ontology

The ontology of Brazilian Geographic Domain was based on the characteristics of geographic objects, spatial relationships and spatial primitives described in ET-EDGV [CEMND-CONCAR 2008] and the guidelines for the construction of each element and concept defined in ET-ADGV [DSG-EB 2011]. The Relationship, Classes and Objects (RCO) attached to ET-EDGV were also evaluated in order to broaden the semantic level of the concepts involved through class definitions and subtypes described. Additionally,

the hierarchical classification of concepts was adjusted to represent the categories provided by their own reference specifications and classifications of geo-concepts proposed by Wang [Wang, Li and Song 2008]. The ontology is partially illustrated in Figure 5.

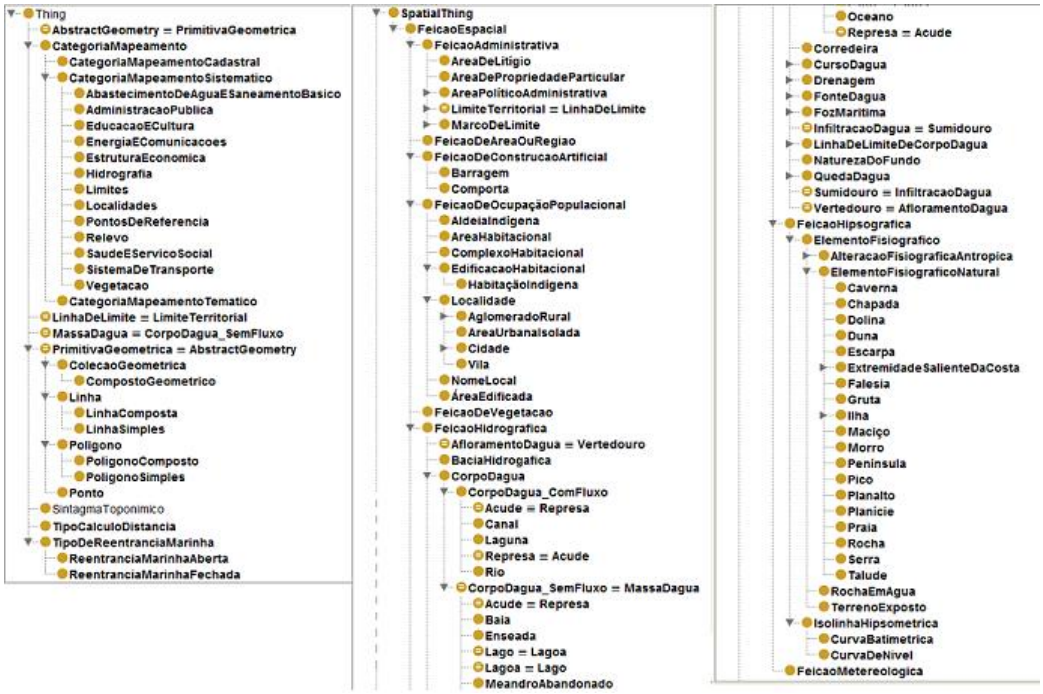


Figure 5: Brazilian Geographic Domain ontology – partial view

As seen in Figure 5 and Figure 2, the proposed ontology represents geo-objects of the domain covered in the ET-EDGV and ET-ADGV specifications, and makes use of the concepts of the ontology specification for GML (Geospatial Markup Language), that defines one geographic feature and the forms of spatial representation for geometric primitives among other characteristics.

The spatial relations and cardinalities presented by reference specifications are mapped to the ontology proposed as shown in Table 2 below:

Table 2 – Mapping Cardinality and Conditionality relations for Ontology

Cardinality and conditions from ET-EDGV	Ontology Mapping
0..1	**[MinCardinality 0] and [MaxCardinality 1]**
0..N (0..*)	**[MinCardinality 0]**
1..N (1..*)	**[someValueFrom]**
1..2	**[MinCardinality 1] and [MaxCardinality 2]**
1..1	**[ExactCardinality 1]**
Condittion {if 'tipoMassaDagua' = “Oceano” or “Baía” or “Enseada”}	**[OnlyValueFrom “Oceano”] or [OnlyValueFrom “Baia”] or [OnlyValueFrom “Enseada”]**

The subtypes are reported in RCO as subclasses of the class hierarchies for original, with consideration of the description of each element and the class itself, the categories were represented as a hierarchy. Figure 6 shows these correspondences demonstrating the case of class "*CorpoDagua*" (water body) and hierarchy of categories ("*CategoriaMapeamentoSistematico*" and others). The conceptual connection with the Brazilian Geographic Metadata Profile ontology is also presented.

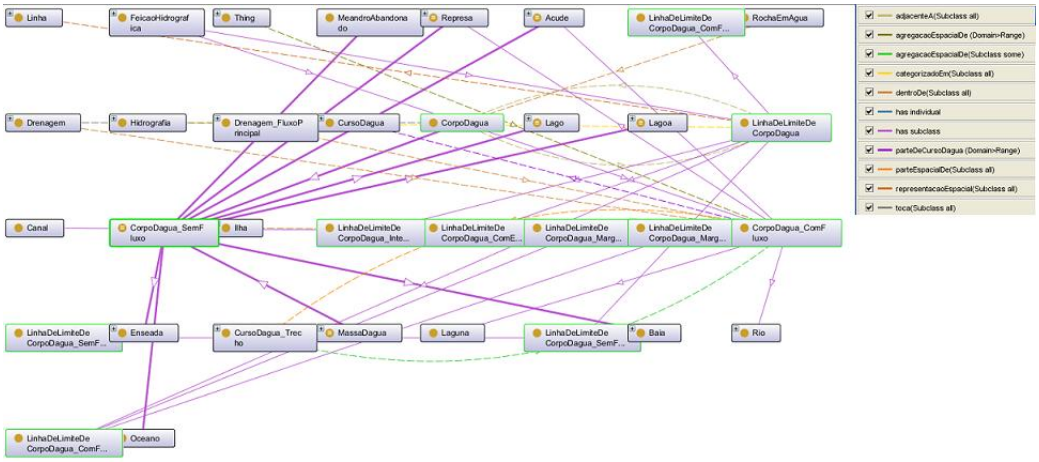


Figure 6: Class hierarchy and relationships to represent subtypes of RCO - case "CorpoDagua"

Whereas @OntoNG, @OntoMG and @ OntoFG respectively as sub-ontologies of Geographical Names, Geographical Metadata and Geographic Domain, the Competency questions listed before in this section could be answered through set of assertions in Table 3 below:

Table 3 – Set of assertions that address the Competency Questions

Set of assertions	CQ addressed

• There is a name x queried.
• There is some GeographicName g:
  • @OntoNG:GeographicName.hasToponym"->Toponym.compositeName(x) OR
  • @OntoNG:GeographicName.hasAlternativeToponym"->Toponym..compositeName(x)
• And there is some GeographicFeature f:
  • @OntoFG:GeographicFeature.identifiedByName(g)
	CQ nº 1

• There is a name **x** queried.
• There is some GeographicName **g**:
  • @OntoNG:GeographicName.hasToponym->Toponym.compositeName(**x**) OR
  • @OntoNG:GeographicName.hasAlternativeToponym->Toponym..compositeName(**x**)
• And there is some GeographicMetadata **m**:
  • @OntoMG:GeographicMetadata.hasIdentifcationOfCDG.hasKeywords(x) OR
  • @OntoMG:GeographicMetadata.hasIdentifcation.hasAbstract(x)
• And there is some GeographicFeature **f**:
  • @OntoFG:GeographicName.identifiesFeature(**f**) AND @OntoFG:GeographicMetadata.describesFeature(**f**)
	CQ nº 2

▪ Considering GeographicFeature **f1** and **f2**. ▪ There is some Resolution **res1** as @ontoMG:GeographicMetadata.describesFeature(**f1**).hasIndentificationOfCDG->IdenticationOfCDF..hasResolution(). ▪ There is some Resolution **res2** as @ontoMG:GeographicMetadata.describesFeature(**f2**).hasIndentificationOfCDG->IdenticationOfCDF..hasResolution(). ▪ And **res2** <> **res1**.	**CQ nº 3**
• Considering GeographicFeature **f1** and **f2**. • There is some Resolution **res1** as @ontoMG:GeographicMetadata.describesFeature(**f1**).hasIndentificationOfCDG->IdenticationOfCDF..hasResolution(). • There is some Resolution **res2** as @ontoMG:GeographicMetadata.describesFeature(**f2**).hasIndentificationOfCDG->IdenticationOfCDF..hasResolution(). • With res2 <> res1 then generalizes f1 as gf1 and f2 as gf2, both using max value between res1 and res2. • And **gf1** is spatially equals **gf2**.	**CQ nº 4**

5. Application of the proposed geo-ontology

The example scenario is presented in Table 4 [Gimenez let al. 2013], where integrated geoinformation is obtained from the basic geoinformation available in INDE. Moreover, integration of the associated metadata and the correlation with the geographical names are also carried out. The final step of the integration comprises the alignment of the ontology that describes the INDE geo-services with the geo-ontology proposed by this work, and then followed by the geo-processing of the retrieved data to compose the integrated geo-information with its resulting metadata.

Table 4 – Application query sample [Gimenez et al. 2013]

Consulta	Um especialista ambiental deseja saber quais são as áreas urbanas de maior risco de deslizamento de encostas por região de interesse.
Recursos requeridos	• Dados de localidades e áreas urbanas. Provedor: IBGE. Temas: Localidades e Malhas Municipal e Estadual. Conceitos principais: Cidades, Municípios, Estados, Áreas Edificadas. • Dados de precipitação pluviométrica histórica por região. Provedor: INPE/CPTEC. Tema: Clima. • Dados de geologia e geomorfologia por região. Provedor CPRM. Temas: Geologia e Geomorfologia. Conceitos: declividade, formação do solo. • Dados de incidência histórica de catástrofes naturais por região. Possível provedor: Ministério das Cidades. Tema: Mapeamento de Áreas de risco. • Dados de relevo. Provedor: IBGE e Estados. Tema: Cartografia – Relevo. Conceitos principais: relevos, curva de nível. • Dados de hidrografia. Provedor IBGE, ANA e Estados. Conceitos principais: hidrografia, bacias hidrográficas, trechos de drenagem, curso d'água, massa d'água.

Thus, to provide the data needs of localities and urban areas, topography and hydrography of the query specified in Table 4, which are covered by the ET-EDGV standard, would require the combination of the corresponding concepts in the sub-ontology of geographic features. This combination is based on the evaluation of the geographical identity of each element, considering the sub-ontology of geographic names, and using the description provided by the geographic metadata sub-ontology.

6. Related works

Some studies have been made to define and specify the possible structuring of geo-ontologies sets to represent geographic space [Bishr 2008] [Di and Zhao 2008] [Kun, Wang and Shuang-Yun 2005] [Wang, Li and Song 2008].

Di and Zhao [Di and Zhao 2008] defines several levels of abstraction for geo-ontologies, as seen in Figure 1(a): (i) General Ontology – the core upper level vocabulary representing the common human consensus reality that all other ontologies must reference; (ii) Geospatial Feature Ontology – defines the geospatial entities and physical phenomena that form ontological foundation for geospatial information; (iii) Geospatial Factor Ontology – describes geospatial location, unit conversions factors and numerical extensions; (iv) Geospatial Relationship Ontology – represents geospatial and logical relationships between geospatial features to enable geospatial topological, proximity and contextual reasoning; (v) Geospatial Domain-Specific Ontology – represents the domain concepts by using proprietary vocabularies; (vi) Geospatial Data Ontology – provide a dataset description including representation, storage, modeling, format, resources, services and distributions; (vii) Geospatial Service Ontology – describes who provides the service, what the service does and other properties that the service has that it discoverable, as well as other characteristics of the service.

Our proposal differs from existing works by being specific to the Brazilian

context and to the INDE technological stack of its applied patterns, datasets, MGB profile, and the particular characteristics of the associated Brazilian Geographical Names. We also adopt existing ontologies, such as Geonames.org, ISO 19000 series, GML and so on, in a simplified perspective that covers all applied patterns to INDE.

7. Final Considerations

In this paper we outline the conceptual organization of the geographic entities defined to the Brazilian context based on INDE and ET-EDGV specification about geo-objects, as well as information about BNGB (Brazilian Bank of Geographic Names) and MGB Profile. Much has to be done yet, to achieve a geo-ontology that can be accepted as the basis for semantic integration of several heterogeneous sources as to the themes and producers in the Brazilian context. This work has the intention to (re)open the discussion and the application prospect to maximize the use of basic geoinformation available in INDE. The focus of our current work is to use the geo ontology proposed applied on architecture for semantic integration for INDE.

The main contribution of this proposal is the combination of concepts from the geographic names, metadata and geographic entities, providing support for analysis, applications and multifaceted uses.

Besides promoting the technical and academic contribution for the research centers and the institutions participating in the INDE, in order to mature the geo-ontology proposal, we can envision some future work: (i) extension of the proposed geo-ontology to cover the needs of Systematic Cadastral Mapping as soon as the ET-EDGV specifies them; (ii) extension of geo-ontology to represent metadata of geo-services that are not yet covered by the MGB Profile and adaptation of coded values lists to reflect the national context; (iii) extension of geo-ontology for geo-field in alignment to ET-PCDG[3] under elaboration; (iv) creation of geographic quality control ontology for validation and verification of geospatial data quality for alignment with the future ET-CQPCDG[3] specification; (v) expansion of Brazilian Geographic Domain Ontology to match ET-EDGV specification in a complete way, considering all rules and orientations in there; (vi) expansion of Brazilian Geographic Names Ontology to treat the concepts associated with historical, ethnological and linguistic characteristics of toponyms.

References

Bédard, Y., Rivest, S. and Proulx, M.-J. (2005). Spatial On-Line Analytical Processing (SOLAP): Concepts, Architectures and Solutions from a Geomatics Enginnering Perspective. In: Robert Wrembel & Christian Koncilia (Eds.), Data Warehouses and OLAP: Concepts, Architectures and Solutions, Chap. 13, IRM Press (Idea Group), London, UK, pp. 298-319. 2007.

Bishr, Y. (2008) "Geospatial Semantic Web: Applications". In: Shekhar, S.; Xiong, H; (Eds), Encyclopedia of GIS, p.391-398, Springer, New York, USA.

CEMG-CONCAR. (2009). "Perfil de Metadados Geoespaciais do Brasil – Perfil MGB. Versão Homologada". CEMG-CONCAR, MPOG, Rio de Janeiro.

CEMND-CONCAR. (2008). "Especificação Técnica para a Estruturação de Dados Geoespaciais Vetoriais (ET-EDGV)". Edição 2.1.3 – Outubro 2010. CEMD-CONCAR, MPOG, RJ.

CONCAR. (2010) "Plano de Ação para Implantação da Infraestrutura Nacional de Dados

[3]See more details for these specifications in CONCAR (2010)

Espaciais". CINDE- CONCAR, MPOG, Rio de Janeiro.

Di, L., Zhao, P. (2008) "Geospatial Semantic Web, Interoperability". In: Shekhar, S.; Xiong, H; (Eds), Encyclopedia of GIS, p.398-403, Springer, New York, USA.

Diaz, L. et al. (2012) "Future SDI – Impulses from Geoinformatics Research and IT Trends". IJSDIR.

Diniz, F. C. (2013) "Qualidade de metadados geoespaciais conforme a ET-PCDG". Anais XVI SBSR, Foz do Iguaçu, PR. Brasil.

DSG-EB (2011) "Especificação Técnica para A Aquisição de Dados Geoespaciais Vetoriais". Ministério da Defesa, Exército Brasileiro, Diretoria de Serviços Geográficos. Norma Cartográfica Brasileira, ET-ADGV, v. 2.0. 2011.

Fonseca, F. (2008) "Geospatial Semantic Web". In: Shekhar, S.; Xiong, H; (Eds), Encyclopedia of GIS, p.388-391, Springer, New York, USA.

Gimenez, P. J. A., Tanaka, A. K., Baião, F. (2013) "Uma proposta de integração semântica para a Infraestrutura Nacional de Dados Espaciais usando geo-ontologias". Proc. 5Th WCGE – SBSI 2013.

IBGE (2010) "Glossário dos Termos Genéricos dos Nomes Geográficos Utilizados no Mapeamento Sistemático do Brasil". IBGE, Diretoria de Geociências, Coordenação de Cartografia. v.1 Escala 1: 1000000 Base Cartográfica Contínua do Brasil ao Milionésimo.

Janowicz, K. ET AL. (2012) "Geospatial Semantics and Linked Spatiotemporal Data – Past, Present, and Future". Semantic Web, IOS Press.

Kun, Y., Wang, J., Shuang-Yun, P. (2005) "The Research and Practice of Geo-Ontology Construction". In: XXXVI-2/W25 International Symposium on Spatio-temporal Modeling, Spatial Reasoning, Analysis, Data Mining and Data Fusion, 27-29 Aug, Beijing, China. Proceedings … Beijing, ISPRS, 2005.

Lieberman, J. (2006) "Geospatial Semantic Web Interoperability Experiment Report. OGC Discussion Paper". OGC Site. http://www.opengeospatial.org/

Lima, R. de V. (2011) "Compostos Toponímicos – Uma Abordagem para o Processamento Automático do Português do Brasil". PERcursos Linguísticos, Vitoria, ES, Brasil, v.2 n. 1 p.140-151.

Lunardi, O. A.; Augusto, M.J. C. Infra-Estrutura dos Dados Espaciais Brasileira –Mapoteca Nacional Digital. In: Congresso Brasileiro de Cadastro Técnico Multifinalitário, Florianópolis, 15-19 de Outubro, 2006. Anais.

Noy, N. F., McGuinness, D. L. (2001) "Ontology Development 101: A Guide to Creating Your First Ontology". Stanford Knowledge Systems Laboratory Technical Report, KLS-01-05. March, 2001.

Pascoal, A. P., Carvalho, R. B., Xavier, E. M. A. (2013) "Materialização do Perfil de Metadados Geoespaciais do Brasil em esquema XML derivado da ISO 19139". Anais XVI SBSR, Foz do Iguaçu, PR. Brasil.

Soares, P. G., Tanaka, A., Baião, F. (2010) "Estudo dos Principais Conceitos sobre Integração de Dados Geoespaciais". Relat-DIA n.0018/2010. UNIRIO. Rio de Janeiro, Brasil.

Wang, H., Li, L., Song, P.-C. (2008) "Design of Geo-Ontology based on Concept Lattice". In: XXIst ISPRS Congress, Technical Commission II, v. XXXVII, part B2 p.715-720, 3-11 Jul, Beijing, China. Proceedings. Beijing, ISPRS, 2008.

Wang ET AL (2007) "Geo-ontology design ad its logic reasoning". Geoinformatics 2007: Geospatial Information Science vol. 6753.

Towards Semantic Trajectory Outlier Detection

Artur Ribeiro de Aquino[1], Luis Otavio Alvares [1], Chiara Renso [2],Vania Bogorny[1]

[1]Departamento de Informatica e Estatistica – UFSC

[2]KDD Lab - Pisa, Italy

artur.aquino@posgrad.ufsc.br, alvares@inf.ufsc.br

renso.chiara@isti.cnr.it,vania.bogorny@ufsc.br

Abstract. *Only a few works in trajectory data mining have focused on outlier detection, and to the best of our knowledge, no works so far have made a deeper analysis to either understand or to give a meaning to the outliers. In this paper we present an algorithm to add meaning to trajectory outliers considering three main possible reasons for a detour: stops outside the standard route, events, and traffic jams in the standard path. We show with experiments on real data that the method correctly finds the different types of outliers.*

1. Introduction and Motivation

The current advances in mobile technology made mobility traces more present. As a consequence, there is an increasing need for developing new methods for interpreting these traces to provide more information to the decision maker.

Mobility traces, well known as trajectories of moving objects, are collected as raw data, with the position of the object in space and time. Several works have already been developed for trajectory data analysis. Different types of patterns can be extracted from mobility data, but only a few attempts have been made to either discover the meaning of a pattern or the reason of certain behaviors of moving objects. This is specially true for trajectory outliers. Existing works for trajectory outlier detection as ([Lee et al. 2008], [Yuan et al. 2011], [Chen et al. 2011] look for trajectories that simply behave differently from the majority of the trajectories in a dataset, but no further analysis is performed to discover when the outliers occur or which are the reasons for a different behavior. Trajectory outliers can be interesting to discover suspicious behaviors in a group of people, to find alternative routes in traffic analysis or to reveal the best or worse paths that connect regions of interest. The interpretation of outliers can provide more information to the decision maker and help to answer questions like: has a driver deviate from the main group because he needed to pickup up someone nearby? Was he avoiding a police check? Is this an alternate route to reach a specific place? Is that a suspicious driver? More information about an outlier can be useful to answer these questions in different application domains. However, to discover why and object followed a different route is very complex, since normally only the raw trajectory data are available and no information is given about the moving object and his intents.

In this paper we try to go one step further in trajectory outlier detection, aiming to infer the possible reason why an object made a different movement. More specifically, we extend the work of Fontes [Fontes et al. 2013], which finds outliers between regions of interest. Figure 1 shows an example of trajectories moving from region R_1 to region R_2, where most of the trajectories follow the same path and two objects deviate from this path, c_1 and c_5, which are the *outliers*.

In this paper we consider three main aspects which could be the reason of an outlier: (i) an event on the standard path, (ii) a traffic jam in the standard path, and (iii) a stop in the outlier. In summary, we make the following contributions to the state of the art in trajectory outlier detection: try to interpret the reason of an outlier, define different types of trajectory outliers and propose an algorithm to classify the outliers.

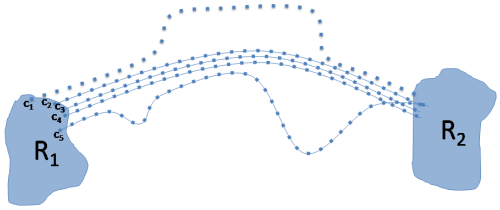


Figure 1. Example of outliers.

The rest of the paper is organized as follows: Section 2 presents the related works, Section 3 presents the main definitions used as base to the new concepts and algorithm presented in Section 4. Section 5 presents experiments on real trajectory data. Finally, Section 6 concludes the paper and suggests directions of future research.

2. Related Works

Only a few works are specifically developed for finding outliers in trajectories [Lee et al. 2008], [Yuan et al. 2011], and [Chen et al. 2011]. The first two approaches split trajectories in subtrajectories to find outliers. Outliers are the trajectories with a fraction of partitions distant from other partitions and must have a certain length. The distance function considers position and direction. In both approaches time is not taken into account and no standard path must exist for trajectories to be classified as outliers. None of these algorithms consider any semantic information or give more meaning to the outliers, their concern is more geometrical, while our work focuses on the meaning.

The last work [Chen et al. 2011] is the closest to the work of this paper. It proposes an algorithm to identify the most popular routes in a trajectory dataset. They build a graph where the nodes are the starting and ending points of the trajectories and the places where the trajectories cross each other, and the arrows are the possible paths from one node to another. For each node the probability is calculated. The standard paths and outliers can be easily derived from this probability. Its purpose is very similar to the work of Fontes [Fontes et al. 2013], but no further analysis is performed over the outliers.

The approaches of [Alvares et al. 2011] can also be used to find outliers. The algorithm proposed by [Alvares et al. 2011] finds trajectories that avoid or deviate from target objects as surveillance cameras, traffic jams, or other pre-defined static objects. It verifies for each trajectory if it avoids static objects and if there is a valid path that crosses the region of the avoided object. This work looks for outliers between trajectories and static objects, while we deal with outliers among trajectories. Indeed, no further interpretation is made on the outliers.

Gupta in [Gupta et al. 2013] presents a survey on the most recent approaches for outlier detection in temporal and spatio-temporal data.

[Fontes et al. 2013] finds outliers between regions of interest. In a first step the algorithm extracts the candidates, that are the subtrajectories moving between pairs of

regions. In a second step the standards and the outliers are discovered. In this paper we extend the work of [Fontes et al. 2013] to find the specific standard path that the outlier deviated and propose to give a meaning to the outliers looking for episodes (i.e. events and traffic jam) in the standard path.

3. Basic Concepts

In this section we introduce some basic concepts about trajectories that will help to understand our approach. Most of these concepts are based on Fontes [Fontes et al. 2013], which is the work we extended here. We start the definitions with the very basic concept of point.

Definition 1 (Point)**.** A point p is a tuple (x, y, t), where x and y are spatial coordinates and t is the time instant in which the coordinates were collected.

A list of points ordered in time is a trajectory.

Definition 2 (Trajectory)**.** A trajectory T is a list of points $\langle p_1, p_2, p_3, ..., p_n \rangle$, where $p_i = (x_i, y_i, t_i)$ and $t_1 < t_2 < t_3 < \ldots < t_n$.

In general, trajectory patterns do not exist in the whole trajectory or during the complete trajectory life. Trajectory patterns occur in parts of the trajectories, therefore, we make use of subtrajectories, that is a well known concept in trajectory research.

Definition 3 (Subtrajectory)**.** A subtrajectory S of T is a list of consecutive points $\langle p_k, p_{k+1}, ..., p_{k+l} \rangle$, where $p_i \in T, k \geq 1$, and $k + l \leq n$.

In this work we try to understand the reason why an object made a detour in relation to the path followed by the majority of the trajectories to move between regions of interest. The trajectories that intersect a pair of regions are the candidates [Fontes et al. 2013], as shown in Figure 1(five candidates).

A candidate is the smallest subtrajectory that moves between two regions. It corresponds to the last point of the subtrajectory that intersects the first region and the first point that intersects the final region, as shown in Figure 2 (left).

Definition 4 (Candidate)**.** Let R_1 and R_2 be two regions such that $R_1 \cap R_2 = \emptyset$ and T a trajectory. A candidate from R_1 to R_2 is the subtrajectory $S = \langle p_i, p_{i+1}, ..., p_m \rangle$ of T, where $(S \cap R_1) = \{p_i\}$ and $(S \cap R_2) = \{p_m\}$.

A candidate that moves from a region R1 to a region R2 is different from a candidate that moves in the opposite direction. In the example of Figure 2 (left) the movement is from R_1 to R_2, thus the candidate has the points from p_i to p_m.

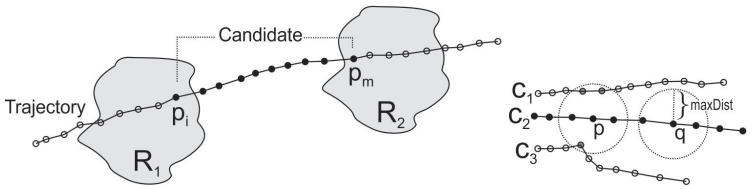


Figure 2. Example of candidate and neighborhood. [Fontes et al. 2013]

According to [Fontes et al. 2013], outliers are discovered among candidates that move in the same direction. The first characteristic that a candidate should have to be an outlier is to move apart from other candidates. Therefore, for each point of a candidate

its neighbors are computed. A candidate is a neighbor of a point if it is close to the point. If a point has a few candidates in its neighborhood, then at that time the moving object was following a path different from the majority of the candidates. The maximal distance (Euclidean distance) for a candidate to be a neighbor of a point is called $maxDist$.

Definition 5 (Neighborhood)**.** Let p be a point. The neighborhood of a point p wrt to $maxDist$ is
$N(p, maxDist) = \{c_i | c_i$ is a candidate and $\exists q \in c_i, dist(p, q) \leq maxDist\}$.

Here we consider spatial neighborhood, because we are interested in all candidates that move together in space. In order to find *traffic avoiding outliers* (as can be seen in Section 4) we analyze the duration of both synchronized and non-synchronized candidates. Figure 2 (right) shows an example of neighborhood. The neighborhood of point p are the candidates c_1 and c_3, since these two candidates have at least one point inside the radius of size $maxDist$ around p. Point q has no candidates inside its radius of size $maxDist$, so its neighborhood is empty. We can conclude that at point p, c_2 was moving with c_1 and c_3 (same path), but at point q, c_2 was moving far from c_1 and c_3 (different path).

When each point of a candidate has a number of candidates in its neighborhood that is higher than a threshold for minimal support, called ***minSup***, then this candidate is called a *standard*.

Definition 6 (Standard)**.** Let $c = \langle p_1, p_2, ..., p_n \rangle$ be a candidate, c is a standard candidate wrt $maxDist$ and $minSup$ if and only if $\forall p_i \in c$,
$|N(p_i, maxDist)| \geq minSup$, where $|X|$ means the cardinality of X.

Figure 1 shows an example of standard candidates, where c2, c3 and c4 are always moving together. So in this example there are three standard candidates considering ***minSup***=2. In the following section we present the new concepts, including a new definition for outlier and an algorithm to add meaning to the outliers.

4. Adding Meaning to Outliers

The main goal of this paper is to add meaning to the outliers. The reasons for an outlier are broad, and can be related to several things. In this paper we consider three main cases that can be the reason of an outlier: (a) stops in the outlier trajectories, where the moving object had the intent to stop somewhere else out of the standard path, (b) there is an event in the standard path that could block it or cause traffic jams in the area of the standard path, (c) a traffic jam in the standard path. We name these three types of outliers as *stop outlier*, *event outlier* and *traffic outlier*, respectively. Each case is detailed in the following subsections.

Before going into detail of the three cases of outliers, we define the concept of *standard path*, redefine the concept of outlier, and define an outlier segment. These definitions are needed because it is necessary to know which standards move in the same path that the outlier deviated. Two standards can be directly connected or reachable from another standard.

Definition 7 (Directly Connected)**.** A standard d is directly connected to a standard e wrt maxDist if $\forall p_i \in d,\ e \in N(p_i, maxDist)$.

Definition 8 (Reachable)**.** A standard d is reachable from a standard e wrt maxDist if there is a chain of standards $d_1, d_2, ..., d_n$ where $d_1 = e, d_n = d$ such that d_{i+1} is directly connected from d_i.

When two or more standards are reachable from any other standard we can define a standard path.

Definition 9 (Standard Path)**.** Let D be a set of standards moving from region R_1 to region R_2. A standard path H wrt maxDist is a non-empty subset of D satisfying the following conditions:

- $\forall d, e \in D$: if $d \in H$ and e is reachable from d wrt maxDist, then $e \in H$.
- $\forall d, e \in H$: d is reachable from e wrt maxDist.

In this work we do not consider an outlier when only a few points of a trajectory deviated the standard path. Therefore, we define the concept of outlier such that it should have a minimal deviation length, called *minLenght*.

Definition 10 (Outlier)**.** Let R_1 and R_2 be two regions and C the set of candidates from R_1 to R_2. A candidate $o \in C$ is an outlier wrt $maxDist$, $minSup$ and $minLenght$ if $\exists c \in C$. c is a standard $\wedge$ $\exists s$ .s is a subtrajectory of o, $s = \langle p_i, p_{i+1}, \ldots, p_n \rangle$. $\forall p_k \in s$, $|N(P_k, maxDist)| < minSup \wedge \sum_{j=i}^{n-1} dist(p_j, p_{j+1}) > minLenght$.

Figure 1 shows an example of outlier, considering $minSup = 60\%$. In this example there are 5 candidates that move from R_1 to R_2, where c_1 and c_5 are the outliers and c_2, c_3 and c_4 are the standard path from R_1 to R_2.

In order to interpret the meaning of an outlier, we want to analyze only the part of the outlier trajectory that made the detour. Figure 3 (left) shows an outlier example where the subtrajectories from p_1 to p_7 and from q_1 to q_14 will be analyzed. These subtrajectories are called *outlier segments*.

Definition 11 (Outlier Segment)**.** Let o be an outlier. Let $s = \langle p_i, p_{i+1}, ..., p_n \rangle$ be a subtrajectory of o. s is an outlier segment wrt $maxDist$, $minSup$, and $minLenght$ if $\forall p_k \in s$, $\mid N(p_k, maxDist) \mid < minSup$ and $\mid N(p_{i-1}, maxDist) \mid \geq minSup$ and $\mid N(p_{n+1}, maxDist) \mid \geq minSup$ and $\sum_{j=i}^{n-1} dist(p_j, p_{j+1}) > minLenght$.

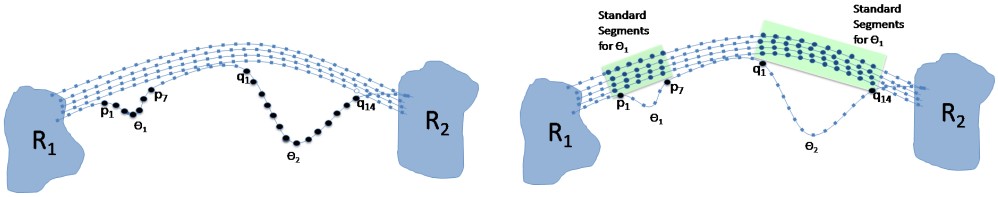


Figure 3. Example of outlier segments and standard segments.

In the example of Figure 3 (left), if *minLenght* is defined as 10%, both θ_1 and θ_2 are outlier segments. If *minLenght*, for instance, is defined as 30%, only θ_2 is an outlier segment.

Having defined the outlier segments we try to interpret them looking for stops, traffic jams and events around these segments, as detailed in the following subsections.

4.1. Stop Outliers

A stop outlier occurs when the moving object made a stop for some time during the deviation. She/he had an appointment, a meeting, or something to do somewhere else that was not in the standard path. This is an intentional detour with a reason. Very common

examples can be go shopping after work, pick up the children at school, go to a happy hour with friends, pass by a bakery, visit a friend or relative, or having a work meeting. When these objectives are out of the standard path, they can be the reasons for the outliers.

To discover if an outlier has a stop we need to look for stops not in the complete outlier trajectory, but only in the subtrajectory that corresponds to the outlier (deviation), i.e., the outlier segment. We consider as a stop a subtrajectory that had speed close to zero for a minimal amount of time (*minTime*). We consider a $maxSpeed$ threshold and not the value zero because, in reality, due to GPS imprecision, it is not common to have points with exact the same coordinates when the object has stopped.

Definition 12 (Stop)**.** Let θ be an outlier segment. A subtrajectory $s \subseteq \theta, s = \langle p_1, p_2, ..., p_n \rangle$ is a stop of θ wrt $minTime$ and $maxSpeed$ if $t_n - t_1 \geq minTime$ and $\frac{\sum dist(p_i, p_{i+1})}{t_n - t_1} \leq maxSpeed$

Definition 13 (Stop Outlier)**.** An outlier segment θ is a stop outlier iff it made a stop.

In the following section we detail the event outliers.

4.2. Event Outliers

The stop outlier is the most simple case of an outlier. The second case is more complex. An *event avoiding outlier* is a detour from the standard path because an event is going on close to the standard path. The complexity starts because there might be more than one standard path connecting two regions in the same direction, as shown in Figure 4 (left), or two standard paths may start together and split later, as shown in Figure 4 (right).

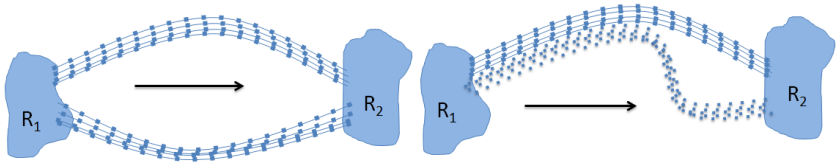


Figure 4. Examples of standard paths from R_1 to R_2.

An outlier is an event outlier when the standard path has an event nearby, i.e., in the area where the outlier avoided the standard path. However, to discover if the standard path has an event nearby is not a trivial task. We cannot simply look if there is an event at any part of the standard path. For an outlier to avoid an event in the standard path it is necessary to analyze only the part of the standard path that was deviated by the outlier. Therefore, we first need to discover which standard path an outlier segment has avoided. Then we need to find the corresponding *segments* of the standard path that the outlier deviated, since the event should be around these segments.

As there might be more than one standard path connecting the regions, we look for the standard path closest to the outlier, in order to analyze this area. Figure3 (right) shows two examples where the outlier segments θ_1 and θ_2 deviated from a different set of standard segments in the same standard path. The segments in a standard path that the outlier deviated are called *standard segments*.

Definition 14 (Standard Segment)**.** Let o be an outlier, $\theta = \langle p_1, p_2, \ldots, p_n \rangle$ an outlier segment of o, p_0 the point of o immediately before p_1, p_{n+1} the point of o immediately after p_n and d a standard in the same path as p_0 and p_{n+1}. A subtrajectory $s = \langle p_k, \ldots, p_l \rangle$

of d is a standard segment of θ if and only if $s \subset d$ and $p_k = closest(p_0, d) \wedge |s| < |d| \wedge p_l = closest(p_{n+1}, d)$, where $closest(p, c)$ is a function that returns the closest point of a candidate c from a point p. If p_0 does not exist then p_k is the first point of d. If p_{n+1} does not exist then p_l is the last point of d.

An event is represented by its region of effect and the times in which the effect starts and ends. The effect of an event is its influence on the movement around.

Definition 15 (Event)**.** An event e is a triple $\langle R, t_0, t_f \rangle$, where R is a region, and t_0 and t_f are the starting and ending time, respectively.

In an event outlier, the outlier segment should not intersect the event, that must happen at the same time of the deviation, but the event should intersect the standard segment (Figure 5 (right) shows an example). Moreover, there must be a standard segment that is synchronized with the outlier to be sure that the standard path was free during the deviation. All standard segments that are close to the outlier in the time that the outlier started, are called *synchronized standard segments*. Figure 5 (left) shows an example of three synchronized standard segments with respect to outlier θ_2 and one non synchronized.

Definition 16 (Synchronized Standard Segment)**.** Let D be the set of standard segments of outlier segment θ moving in the same standard path. A subtrajectory s is a synchronized standard segment with respect to θ when $s \in D$ and $abs(s.t - \theta.t) \leq timeTol$, where $s.t$ and $\theta.t$ represent the time of the first point of s and θ, respectively.

When there is at least one standard segment that intersects an event but the outlier segment does not, and there is an overlap between the outlier segment and the event times, then the outlier segment is called *event avoiding outlier*.

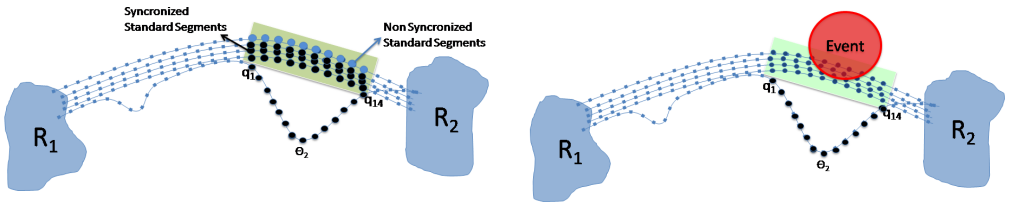


Figure 5. Example of traffic avoiding outlier and event avoiding outier.

Definition 17 (Event Avoiding Outlier)**.** Let e be an event, R_e its region, t_{e_0} its start time and t_{e_f} its end time. Let d be a standard segment of an outlier segment θ, t_{θ_0} its start time and t_{θ_f} its end time. The outlier segment θ is an event avoiding outlier if and only if it is not a stop outlier and $R_e \cap d \neq \emptyset \wedge R_e \cap \theta = \emptyset \ \wedge \ (t_{e_0} < t_{\theta_0} < t_{e_f})$.

In the following section we explain a traffic avoiding outlier.

4.3. Traffic Avoiding Outlier

The last possibility treated so far is that an outlier may be deviating from a traffic jam in the standard path. Normally, traffic jams at the main paths or roads in a city are well known by most people, mainly at rush hours. However, some objects may follow the standard path anyway, while others that know alternative routes leading to the destination may prefer them. Therefore, in this section we look for slow traffic in the standard path at the time of the outlier. Here we assume that the outlier has no stop in its subtrajectory and there is no event around the standard path.

To measure the time on the standard path at the same moment that the outlier deviated from it we need to look only at the synchronized standard segments. Figure 5 (left) shows an example of outlier (θ_2) and the respective synchronized and non-synchronized standard segments. For each outlier segment, the average speed of all synchronized standard segments in the same standard path are compared to the speed of the non-synchronized standard segments in the same path. This way it is possible to know if at the moment of the deviation the traffic was slower than normal traffic (fastest segments). We consider that there is a traffic jam when the average speed of the synchronized standard segments is less than half of the average speed of the non-synchronized standard segments. If this is the case, then we define the outlier as a traffic avoiding outlier.

Definition 18 (Traffic Avoiding Outlier)**.** Let θ be an outlier segment, M a set of synchronized standard segments with θ and N a set of non-synchronized standard segments of θ. θ is a traffic avoiding outlier if and only if it is neither a stop outlier nor an event avoiding outlier and $avgSpeed(M)/avgSpeed(N) \leq 0.5$.

A trajectory can have many outlier segments, and each one is interpreted independently. This means that a trajectory that has 3 outlier segments may have, for instance, 3 types of outliers. After defining the three types of semantic outliers we can finally present an algorithm to automatically interpret the outliers.

4.4. Algorithm

In this section we present an algorithm to interpret the outliers. Algorithm 1 shows the pseudo-code of the main algorithm. The input of the algorithm are thresholds detailed in the definitions. The output is a set of classified outliers. The first step is to find the outlier segments and the standard segments (lines 16 and 17), according to Definition 11 and Definition 14, respectively.

Algorithm 1. Main algorithm pseudocode

```
1  INPUT:
2    C // set of candidates between 2 regions
3    E // set of Events
4    maxDist // for neighborhood
5    minLenght  // for the outlier segments
6    minSup // for a standard path
7    minTime // for a stop
8    timeTol // for synchronized  standard segments
9
10 OUTPUT:
11   SO  // set of stop outliers
12   EAO // set of event avoiding outliers
13   TAO // set of traffic avoiding outliers
14
15 METHOD:
16 OutSegs = findOutlierSegments(C, maxDist, minSup);
17 StandardSegs = findStandardSegments(C,OutSegs,maxDist,minSup);
18 SO = findStopOutlier(OutSegs,mintime);
19 EAO = findEventAvoidingOutlier(OutSegs,StandardSegs, E, SO);
20 TAO = findTrafficAvoidingOutlier(OutSegs,StandardSeg, SO,EAO,timeTol);
21 RETURN SO, EAO, TAO
```

The function findStopOutlier() (line 18) computes the stop outliers checking if each outlier segment has a stop for at least $minTime$, according to Definition 12 (Stop). If the outlier segment has a stop, it is a stop outlier. The maxSpeed threshold used in Definition 12 is computed as 5% of the average speed of the candidate.

The functions findEventAvoidingOutlier() (line 19) and findTrafficAvoidingOutier() (line 20) are detailed in algorithm 2 and algorithm 3, respectively. In order to find event

avoiding outliers, the first step of algorithm 2 is to find all the outlier segments without stops (line 11). Then, for each outlier segment, the algorithm gets its standard segments (line 13) according to Definition 14 and checks the intersection with the set of events (line 14). If there is an event which intersects any of these standard segments without intersecting the outlier segment and at least part of the event happens during the detour, then it is added to the event avoiding outlier list (line 16).

Algorithm 2. findEventAvoidingOutlier pseudocode

```
1  INPUT:
2    OutSegs // set of outlier segments
3    StandardSegs // set of standard segments
4    E // set of events
5    SO // set of stop outliers
6
7  OUTPUT:
8    EAO // set of event avoiding outliers
9
10 METHOD:
11 O = OutSegs − SO;
12 FOR EACH(o in O)
13   S = StandardSegs.getStandardSegments(o);
14   e = getIntersection(S, E);
15   IF(e != NULL && !hasIntersection(o, e) && timeOverlaps(o, e))
16     EAO.add(o);
17   ENDIF
18 ENDFOR
19 RETURN EAO;
```

The function findTrafficAvoidingOutlier() is shown in algorithm 3. The first step is to remove from the set of outlier segments the ones that are stop outliers and event outliers (line 12). Then, the algorithm computes the standard segments that are synchronized (according to Definition 16) and the ones not synchronized (lines 14 and 15). For the non-synchronized the algorithm considers 5% of the fastest standard segments. This way we obtain the speed of the path when there is no traffic. Then the algorithm is able to compare the average speed of both synchronized and non synchronized segments, and infer if there was a traffic jam at that moment (line 16). When the average time of the synchronized standard segments is half of the average of the set of non-synchronized ones, we say that there was a traffic jam at that moment in the standard path, thus the outlier is classified as a traffic avoiding outlier.

Algorithm 3. findTrafficAvoidingOutlier pseudocode

```
1
2  INPUT:
3    OutSegs // set of Outlier Segments
4    StandardSegs // set of Standard Segments
5    SO // set of stop outliers
6    EAO // set of event avoiding outliers
7
8  OUTPUT:
9    TAO // set of traffic avoiding outliers
10
11 METHOD:
12 O = OutSegs − SO − EAO;
13 FOR EACH(o in O)
14   sync = StandardSegs.getSyncStandardSegments(o);
15   notSync = StandardSegs.getNotSyncStandardSegments(o);
16   IF(avgtime(sync.time) < avgtime(notSync.time)/2)
17     TAO.add(o);
18   ENDIF
19 ENDFOR
20 RETURN TAO;
```

5. Experimental Results

In this section we present the results of preliminary experiments on the taxi trajectory dataset collected in San Francisco, California, in May and June 2008 [Crawdad 2013]. This dataset is interesting for analyzing outliers because taxi drivers, in general, know several paths to reach the same place. Therefore, we want to find the alternative routes (outliers) in relation to the standard path and to discover if the alternative route was made to avoid an event in the standard path, to avoid low traffic in the standard path or if the taxi driver had a stop in the detour.

This dataset contains trajectories of taxi drivers during one month, with over 11 million points. One trajectory corresponds to the movement of one taxi driver during several days (with a trajectory identifier for each driver, and not each trajectory), having very long trajectories with sampling rate around 1 minute. Even with such a large time interval between every two points the method was able to find semantic outliers. In order to analyze the behavior of the driver between regions of interest we split the trajectories using the occupation attribute (which states if the taxi has passengers or not), so instead of having only one trajectory of a driver, we have several trajectories of the same object.

When analyzing trajectory patterns we should consider the time periods, since the movement patterns can be different during the day and night, and during weekdays and weekends. In this experiment we separated trajectories of weekdays and weekends, and because of space limitations, we analyzed only the trajectories from Monday to Friday. After this preprocessing step a set of 537.098 trajectories with 6.314.120 points was obtained.

We selected as regions of interest the Airport of San Francisco and a downtown area where most hotels are located, because there is a high taxi flow between these regions. To find the standard path and the outliers we considered 120 meters as *maxDist*, 5% as $minSup$ (number of candidates in the neighborhood for discovering a standard path between the regions) to find the standard path, and $minLenght$ as 10%, i.e., at least 10% of a candidate should be moving far from the standard path in order to be considered an outlier. Figure 6 (left) shows the standard path from the Airport to the central area.

In order to evaluate the method proposed in this paper, which is to give a meaning to the outliers, we simulated an event at Bayshore Freeway (US 101) with start time 17:30 and end time at 21:30. For discovering stop outliers we considered 15 minutes ($minTime$) as the minimal time for an outlier to be a stop outlier and 15 minutes as $timeTol$ for synchronized trajectories.

The method discovered 73 stop outliers (for $minTime$ 15 minutes), 6 traffic avoiding outliers and one event avoiding outlier. Because of space limitations we show one example of each type of outlier. Figure 6 (right) shows an event avoiding outlier, where the method correctly detected the outlier which deviated the standard path with an event. The outlier segment is represented in the figure by the triangles, the event by a circle in the standard path, and the standard segments are represented in black.

Figure 7 (left) shows a stop outlier, where the stop in the outlier segment had a duration of 44:13 minutes. The stop in the outlier segment is zoomed in the figure. Figure 7 (right) shows an example of traffic avoiding outlier, highlighting the outlier segment (triangles) and the corresponding standard segments (black). In this case, the average time of the synchronized standard segments was 7:05 minutes while the average duration of the non-synchronized segments was 3:40 minutes, characterizing a small traffic jam in

the standard path at the moment of the outlier. Most detected traffic jams were near the downtown area, and not in the main road from the airport. Therefore, for this execution we reduced the parameter $minLenght$ in order to detect short detours as the one shown in Figure 7 (right).

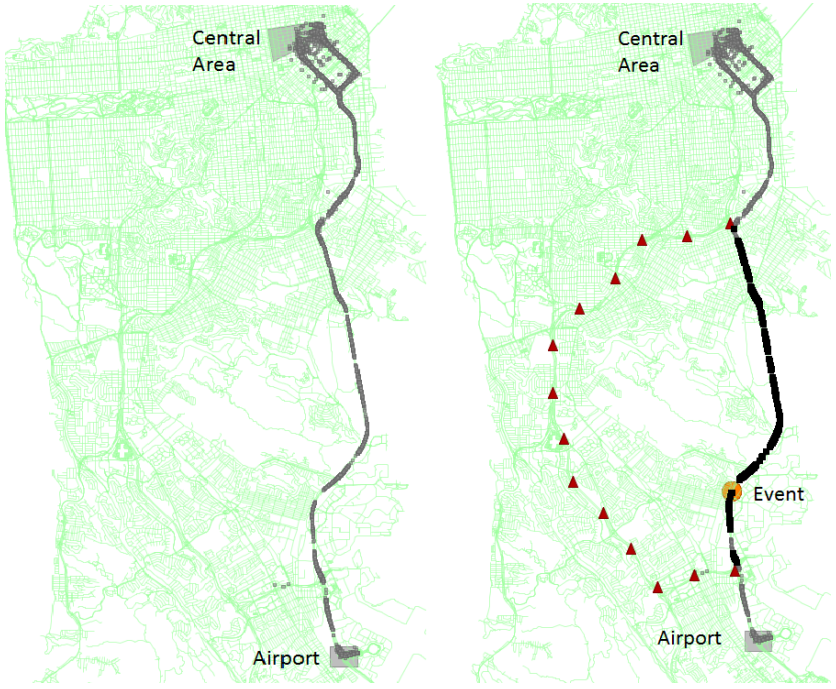


Figure 6. (left) Standard path from Airport to Central area and (right) event outlier.

Figure 7. (left) Stop outlier and (right)Traffic avoiding outlier

6. Conclusion and Future Works

Several algorithms have been proposed for trajectory data mining, but only a few consider trajectory outlier detection. Existing approaches for trajectory outliers do not make deeper analysis of the discovered patterns to give more meaning or semantics. In this paper we look for outliers among trajectories that move between the same regions and try to interpret them.

For all outliers the proposed method finds and interprets each outlier segment, which is the part of the outlier that corresponds to the detour itself. So far the interpretation is separated in three cases which represent some possibilities of deviations: *stop outliers*, *event avoiding outliers*, and *traffic avoiding outliers*.

Future work includes outlier classification according to the type of detour made for each segment, and to give weights for each outlier, depending on the types of its segments. Indeed, more experiments have to be performed to better evaluate the method and the parameters need to be better studied and reduced.

7. Acknowledgments

This work has been partially supported by EU project FP7-PEOPLE SEEK (N. 295179 http://www.seek-project.eu) and the Brazilian agencies CAPES and CNPq. Authors would like to thank also for the partial support from CNR-CNPQ Joint project 2012, DataSIM FP7-ICT-270833, and UFSC.

References

Alvares, L. O., Loy, A. M., Renso, C., and Bogorny, V. (2011). An algorithm to identify avoidance behavior in moving object trajectories. *J. Braz. Comp. Soc.*, 17(3):193–203.

Chen, Z., Shen, H. T., and Zhou, X. (2011). Discovering popular routes from trajectories. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pages 900–911. IEEE.

Crawdad (2013). Crawdad a community resource for archiving wireless data at dartmouth. `http://crawdad.cs.dartmouth.edu/meta.php?name=epfl/mobility`. Accessed: 2013-05-10.

Fontes, V. C., de Alencar, L. A., Renso, C., and Bogorny, V. (2013). Discovering trajectory outliers between regions of interest. In *GeoInfo*.

Gupta, M., Gao, J., Aggarwal, C. C., and Han, J. (2013). Outlier detection for temporal data: A survey. *TKDE*, 25.

Lee, J.-G., Han, J., and Li, X. (2008). Trajectory outlier detection: A partition-and-detect framework. In Alonso, G., Blakeley, J. A., and Chen, A. L. P., editors, *ICDE*, pages 140–149. IEEE.

Yuan, G., Xia, S., Zhang, L., Zhou, Y., and Ji, C. (2011). Trajectory outlier detection algorithm based on structural features. *Journal of Computational Information Systems*, 7(11):4137–4144.

Exploração visual interativa de dados coletados pelo Sistema Integrado de Monitoramento Ambiental - SIMA

Alisson Fernando Coelho do Carmo[1], **Milton Hirokazu Shimabukuro**[1], **Enner Herenio de Alcântara**[1]

[1]Programa de Pós-Graduação em Ciências Cartográficas (PPGCC)
Faculdade de Ciências e Tecnologias (FCT)
Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP)
Presidente Prudente – SP – Brasil

alisondocarmo@gmail.com, {miltonhs,enner}@fct.unesp.br

Abstract. *Researches on environmental phenomena behavior have been benefited from sensor technological advances. Periodic collecting produces long time series, with large data sets composed by a great number of variables. Interactive visual exploration and analysis techniques can be applied to support such data sets processing in order to facilitate the identification of erros and trends, as well as, the detection of patterns. In this context, this paper presents an approach using visual representations for data, associated with interaction techniques, to support the process of identification and understanding the data structures and possible failures in data collected and stored by Integrated System for Environmental Monitoring.*

Resumo. *Estudos sobre o comportamento de fenômenos ambientais tem se beneficiado da evolução dos sensores. Coletas periódicas geram longas séries temporais, com grande quantidade de dados e variáveis. Técnicas de exploração e análise visual interativa de dados podem ser utilizadas para auxiliar a análise do grande volume de dados e facilitar a identificação de erros, tendências e observação de padrões. Neste contexto, este trabalho apresenta uma abordagem que utiliza técnicas de representação visual e interativa de dados para auxiliar a identificação e compreensão das estruturas de dados e possíveis falhas presentes no conjunto de dados coletados e armazenados pelo Sistema Integrado de Monitoramento Ambiental.*

1. Introdução

A necessidade de registrar, acompanhar e entender os fenômenos e comportamentos do meio ambiente sempre esteve presente no cotidiano do homem e é um de seus objetos de estudo. Tal tarefa tem se beneficiado do desenvolvimento tecnológico, principalmente relacionado à evolução da tecnologia empregada em dispositivos sensores. A utilização de sensores na coleta de dados ambientais permite a aquisição automática e periódica de dados. A periodicidade oferece uma nova possibilidade de análise, possibilitando a integração do atributo tempo com os valores coletados, originando longas séries temporais.

A principal característica da análise de séries temporais é o histórico dos valores registrados. Desta forma, é possível investigar o conjunto de dados em busca de padrões

e dependências que podem ser evidenciados durante a observação do comportamento dos registros ao longo do tempo.

No entanto, a manipulação de conjuntos de dados de séries temporais requer alguns cuidados. A constante aquisição dos dados faz com que o volume de dados coletados permaneça em crescimento frequente. Para que esta grande quantidade de dados possa ser analisada e interpretada, é necessária a utilização de recursos computacionais capazes de processar e sumarizar o conjunto.

Existem diversas metodologias que podem ser aplicadas para a análise de séries temporais. Um dos recursos que podem potencializar a análise é a utilização de técnicas de Visualização de Informação. Estas técnicas buscam representar visualmente os dados para facilitar a interpretação. Esta operação pode ser integrada com outros recursos que se beneficiem da percepção do analista durante a exploração dos dados, e garantam a interação durante a exploração dos dados, cenário conhecido como Visual Analytics.

Neste contexto, este trabalho tem o objetivo de apresentar os resultados parciais obtidos com a investigação sobre a utilização de técnicas de exploração e análise visual de dados oriundos de sensores ambientais. O presente texto discute alguns fatores relacionados com as características do conjunto de dados coletados pelo Sistema Integrado de Monitoramento Ambiental (SIMA).

As demais Seções descrevem os principais aspectos sobre o desenvolvimento deste trabalho, que compõe uma pesquisa em andamento. Na Seção 2 são apresentados alguns conceitos relacionados. A exploração e manipulação dos dados é descrita na Seção 3. Os resultados dos testes realizados com a exploração do conjunto de dados, sobretudo com a representação visual dos dados são abordados na Seção 4 para então, subsidiar as considerações finais apresentadas na Seção 5.

2. Conceitos relacionados

Nesta Seção são apresentados os principais conceitos relacionados com o desenvolvimento deste trabalho.

2.1. Séries Temporais

As séries temporais constituem uma importante configuração de dados, as quais os representam de forma ordenada em relação ao tempo. O estudo de séries temporais geralmente é focado em dois principais fatores que são diretamente relacionados, referentes à compreensão da forma que os valores da série são gerados e ao estudo do comportamento da série, permitindo a estimativa de valores ausentes em instantes de tempo da série, bem como a predição de valores futuros. Existem diversas técnicas tradicionais para análise de séries temporais, principalmente baseadas em cálculos estatísticos. Para potencializar os resultados obtidos na análise de séries temporais, outros recursos computacionais podem ser utilizados, como abordado por Esling e Agon [Esling and Agon 2012], que apresentam um levantamento sobre diferentes algoritmos e ferramentas que permitem aplicar técnicas de mineração de dados para a descoberta de conhecimento em séries temporais por meio do comportamento geométrico da variação dos dados.

2.2. Sistema Integrado de Monitoramento Ambiental

O SIMA é formado por um conjunto de tecnologias aplicadas à coleta de dados e monitoramento da hidrosfera [INPE 2013]. Ele é composto de uma rede de plataformas que

possui sensores aquáticos e sensores capazes de coletar atributos relacionados ao ar. As plataformas SIMA realizam a leitura dos sinais dos sensores com a periodicidade de uma hora. Após a leitura, os dados coletados são transmitidos via satélite, estando este visível para a plataforma ou não. Por esta razão, pode ocorrer de alguns dados serem perdidos no momento da transmissão. Servidores intermediários em estações terrestres são responsáveis por receber os dados transmitidos e realizar a verificação da existência de erros na transmissão dos sinais. Após esta validação dos dados, estes são transmitidos ao servidor no centro de armazenamento, os quais passam pelo processo de decodificação, processamento e armazenamento, e ficam disponíveis em um portal da internet. Alcântara et al [Alcântara et al. 2013] apresentam uma análise utilizando algumas métricas estatísticas sobre as plataformas e discutem sobre os principais problemas que podem estar presentes, relacionados à degradação dos sensores e comunicação com o satélite.

2.3. Visual Analytics

A complexidade envolvida no processo de exploração e análise de dados pode ser incrementada em razão de alguns aspectos intrínsecos, como o volume de dados a ser considerado, a quantidade de parâmetros a serem interpretados, e a integridade e qualidade associada a estes dados. Neste sentido, técnicas de análises e recursos computacionais podem oferecer as ferramentas necessárias para viabilizar esta tarefa e torná-la mais natural. O termo Visual Analytics (VA) foi apresentado por Thomas e Cook [Thomas and Cook 2005] no cenário em que a representação visual não era suficiente para viabilizar a análise direta de grandes quantidades de dados. Então, técnicas de interação e manipulação visual foram agregadas ao processo de análise para garantir a permanência do analista no centro do processo, de modo que sua capacidade de percepção e cognição visual pudesse ser utilizada para refinar o processo de exploração e construção do raciocínio analítico. No contexto de análise de dados obtidos por sensores, o fator tempo pode oferecer outras oportunidades de análise, como apresentado por Maciejewski et al [Maciejewski et al. 2010], que discutem sobre alguns benefícios que podem ser obtidos com a integração entre as representações tradicionais de séries temporais com outras técnicas de visualizações espaço temporais.

3. Exploração dos dados

Para a realização deste trabalho, foram considerados os dados coletados e armazenados pelo projeto SIMA. Para tanto, foram utilizadas as planilhas eletrônicas de dados exportados pelo portal na internet, levando em consideração o período de funcionamento de todas as plataformas registradas até início do ano de 2013. Os dados foram inseridos em um Sistema Gerenciador de Banco de Dados (SGBD) para facilitar a manipulação e filtragem dos dados a serem processados. O SGBD utilizado foi o PostgreSQL, escolha motivada por ser um sistema *open source* que possui integração com a extensão espacial PostGIS, também *open source*. O primeiro fator observado durante a exploração dos dados registrados, foi o tempo de atividade de cada plataforma, como pode ser visto na Figura 1, que apresenta o período em que as plataformas estiveram ativas - coletando e transmitindo dados.

Explorando os dados foi possível observar que mesmo durante o período de atividade da plataforma algumas amostras não estavam registradas. A ausência de dados de

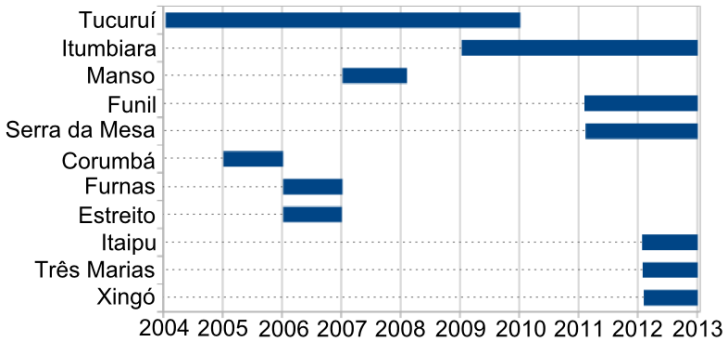


Figura 1. Períodos de atividades das plataformas SIMA

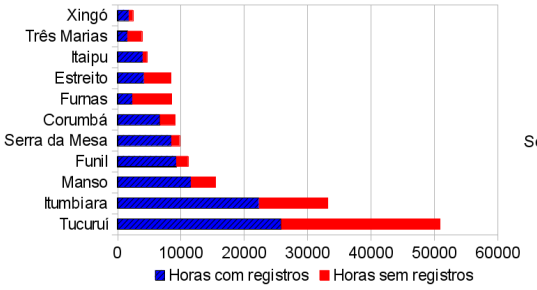


Figura 2. Horas ativas das plataformas

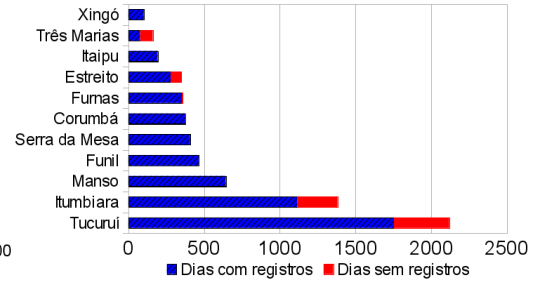


Figura 3. Dias ativos das plataformas

coletas ocorre tanto em determinadas horas do dia, mostrada na Figura 2, como também em dias completos, exibida na Figura 3.

Os gráficos apresentados nas Figuras 1, 2 e 3 permitem observar a ausência de dados de um modo geral. No entanto, esta visualização considera apenas a existência/ausência de registros, e não aborda a integridade dos dados existentes. A Seção 4 apresenta abordagens iniciais para a observação da integridade dos dados das plataformas SIMA.

4. Resultados preliminares

Inicialmente, para verificar o comportamento geral do estado das variáveis e sua integridade, foi utilizada uma visualização capaz de representar uma matriz de dados ordenados de acordo com o tempo. A Figura 4 exibe a representação de um período de dados da plataforma Três Marias, na qual cada linha representa uma variável diferente. A plataforma Três Marias foi escolhida em razão do maior índice proporcional de ausência de dados - objeto de estudo deste teste - tanto relativo a cada hora como ao dia completo.

A escala de cores utilizada na Figura 4 representa os valores de acordo com sua intensidade, variando em uma escala linear contínua. Nesta visualização foram considerados apenas os registros diários existentes no conjunto de dados, tornando a linha do tempo sequencial, sem lacunas. Vale ressaltar que a escala de tempo nesta representação não é linear, pois considera apenas os registros armazenados, organizando-os sequencialmente, independente da ausência de dados.

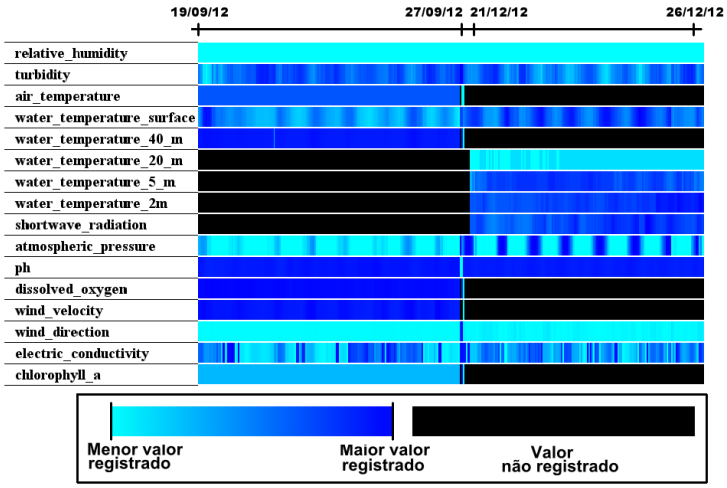


Figura 4. Visualização em matriz de um período da plataforma Três Marias

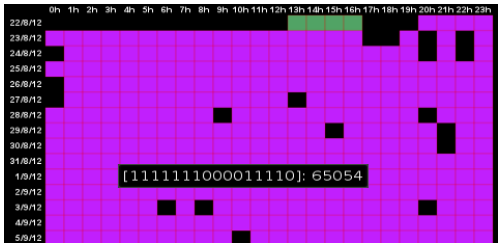


Figura 5. Visualização Calendar View (início do intervalo) dos dados da plataforma Três Marias

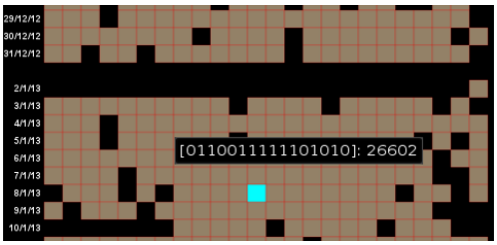


Figura 6. Visualização Calendar View (fim do intervalo) dos dados da plataforma Três Marias

Para evidenciar as falhas dos sensores, foi utilizada a técnica de representação conhecida como *Calendar View*. Nesta representação, a relação de tempo é segmentada em observações diárias em cada linha da matriz, e as colunas identificam as horas do dia. Os valores registrados de cada variável foram convertidos em uma sequência binária de 16 bits, em que cada bit representa a presença (1) ou ausência (0) do dado da respectiva variável. Uma vez definida a sequência binária para cada registro e realização do mapeamento discreto para a escala de cores do espaço RGB (*Red, Green, Blue*), é possível identificar o conjunto de sensores com falhas por meio da cor que representa o registro, facilitando a observação de padrões de falhas. As Figuras 5 e 6 apresentam um exemplo de dois períodos, inicial e final, respectivamente, da plataforma Três Marias.

Nesta representação a conversão da sequência binária ocorre de forma discreta, de modo que todas as combinações de falhas nas 16 variáveis (2^{16}) são mapeadas proporcionalmente para o espaço de cor RGB, garantindo que cada uma das combinações seja representada por uma cor diferente. Na Figura 5, que representa o início do período ativo da plataforma, predomina a cor referente à sequência binária "1111111000011110"que representa o erro nos dados de cinco sensores. Já nos últimos registros, representados na Figura 6, a cor predominante se altera, indicando a sequência "0110011111101010", ou seja, com erros em seis sensores, sendo todos diferentes do primeiro cenário.

5. Considerações finais

No contexto de análise de séries temporais geradas por sensores é comum a inconsistência de alguns dados, pois o processo de aquisição e armazenamento dos valores envolve diversos fatores. Os erros dos dados podem ser inseridos em razão de falha do sensor na leitura, erro na transmissão ou ainda na conversão do sinal elétrico do sensor para um valor discreto. Por esta razão, a manipulação e estudo do comportamento das séries temporais pode trazer benefícios associados à completitude do conjunto de dados, pois os valores perdidos podem ser estimados considerando o histórico temporal.

Existem diferentes abordagens para a análise de séries temporais. Este trabalho apresentou a utilização de técnicas de análise visual para auxiliar a exploração dos dados, contando com a capacidade humana de cognição visual. Os recursos de interatividade, por meio da manipulação da representação visual, permite que o analista refine os resultados gerados a partir de sua própria percepção. Comparando a visualização apresentada na Figura 4 com a representação dos mesmos dados utilizando a representação *Calendar View*, exibidas nas Figuras 5 e 6, é possível verificar a existência de falhas nos respectivos sensores. Desta forma, pode ser observado que diferentes visualizações podem ser utilizadas para representar o mesmo conjunto de dados, sendo que cada uma delas implica em leituras diferentes da representação, evidenciando características específicas.

Agradecimentos

Os autores agradecem o Programa de Pós-Graduação em Ciências Cartográficas (PPGCC) da Faculdade de Ciências e Tecnologia/UNESP (FCT/UNESP)- Campus de Presidente Prudente - por permitir o desenvolvimento desta investigação como tema de projeto de mestrado; a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo auxílio dedicado ao projeto; e ao Instituto Nacional de Pesquisas Espaciais (INPE) pela cessão dos dados SIMA.

Referências

[Alcântara et al. 2013] Alcântara, E., Curtarelli, M., Ogashawara, I., Stech, J., and Souza, A. (2013). A system for environmental monitoring of hydroelectric reservoirs in brazil. *Ambiente e Água - An Interdisciplinary Journal of Applied Science*, 8(1).

[Esling and Agon 2012] Esling, P. and Agon, C. (2012). Time-series data mining. *ACM Comput. Surv.*, 45(1):12:1–12:34.

[INPE 2013] INPE, H. (2013). Sima: Sistema integrado de monitoramento ambiental. `http://www.dsr.inpe.br/hidrosfera/sima/`. Acessado em Agosto de 2013.

[Maciejewski et al. 2010] Maciejewski, R., Rudolph, S., Hafen, R., Abusalah, A., Yakout, M., Ouzzani, M., Cleveland, W. S., Grannis, S. J., and Ebert, D. S. (2010). A visual analytics approach to understanding spatiotemporal hotspots. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):205–220.

[Thomas and Cook 2005] Thomas, J. J. and Cook, K. A. (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr.

Measuring Allocation Errors in Land Change Models in Amazonia

Luiz Gustavo Diniz[1], Merret Buurman[2], Pedro R. Andrade[3], Gilberto Camara[1,2], Edzer Pebesma[2]

[1] Image Processing Division (DPI) - National Institute for Space Research (INPE) Avenida dos Astronautas, 1758, Jardim da Granja, São José dos Campos, SP – Brazil, 12227-010

[2] Institute for Geoinformatics, University of Muenster, Weseler Strasse 253, 48153, Muenster, Germany

[3] Earth System Science Center (CCST) - National Institute for Space Research (INPE) Avenida dos Astronautas, 1758 Jardim da Granja, São José dos Campos, SP – Brazil, 12227-010

lgdo@dpi.inpe.br, merretbu@gmail.com, pedro.andrade@inpe.br, gilberto.camara@inpe.br, edzer.pebesma@uni-muenster.de

Abstract. *Top-down land change models use computer simulation to allocate demands for change over the spatial region under study. This paper presents a metric to estimate the goodness of fit of land change models. It focuses only on changed areas, preventing inflated goodness of fit values due to large fractions of the landscape remaining unchanged. We use the proposed metric to evaluate land change models for Amazonia. Despite large quality differences between them, all models have problems to predict new frontiers and expansion areas. The best model considered in this paper only performs slightly better than a simple model that predicts a cell's deforestation based on the deforestation in neighboring cells.*

1 Introduction

Changes in land use and land cover have increased worldwide substantially in the second half of the 20th century, mostly as part of the economic growth of emerging nations such as China, India, Brazil and Indonesia. Land cover is the biophysical state of the earth's surface; land use is the purpose for which humans use the land [1]. Forest and cropland are examples of land cover and agricultural and pasture is an example of land use. We use the term "land change" to refer to land use and land cover change. Land changes result from people acting on ecosystems, based on demographic, social, and economic factors. Planners and policy makers need models that represent how humans change the land [2]. Despite the challenges involved in building them, these models have an important role, as they serve as tools to understand human-environment interactions and to help public policy making.

Many papers on land change models make strong policy recommendations based on their results [3, 4]. Thus, measuring the quality of land change models is important to assess the likelihood of the scenarios expressed by the models. The problem is of course that these models project future changes and therefore testing them is unattainable at the moment when the model is conceived. What can be done is an *ex-post analysis*: looking at the past from the future. Some years later, we can compare the

model projections with the reality. To do this comparison properly, we need a metric that expresses differences between the model projections and the facts.

This paper evaluates the results of several models of deforestation in the Brazilian Amazon in an ex-post analysis. We propose a goodness of fit metric extending the works of [5] and [6]. The metric uses a multi-resolution approach to account for the scale-dependency of spatial patterns. Using this approach, we aim at better understanding the strong points and the flaws of the different models and their underlying conjectures. To do this, we also compared the models to a simple model based on the previous year's deforestation as the only explaining variable.

2 Review of current goodness of fit metrics for spatial models

There are two complementary views on the literature on the issue of measuring the goodness of fit of land change models. One of them is the multiple resolution approach proposed by Costanza [5] to evaluate land change models [7, 8]. In his paper, Costanza proposes a method to compute the goodness of fit for categorical data. Arguing that simple cell-by-cell comparison is misleading because allocation of change can occur at neighboring cells, he proposes comparing the two maps at several resolutions, using moving windows. He starts with a window whose size is a single cell. For each window position, he computes a goodness of fit metric. After moving the window over the whole map, he gets the average of this metric and uses it as the goodness of fit measure at the window's resolution. Then, he doubles the window size and evaluates the metric again. The result is a set of metrics, one for each window resolution.

A second view in the literature argues for taking persistence into account when computing model goodness of fit. In land change modeling, the primary interest is finding out whether the cells representing land change were correctly placed. As Pontius et al. [6] point out, in most land change models a lot of areas remain the same from one time step to another. Usually, the changes are a small proportion of the total area. Thus, to properly assess goodness of fit in land change models, we have to account for persistence and should only consider the areas where change occurred. To compare different models, Pontius et al. [9] propose the metric "figure of merit" to compute the ratio of the area that was correctly predicted as change and all the areas that were observed or modeled as change:

$$FM = \frac{Change_{correct}}{Change_{correct} + Change_{ref} + Change_{mod} + Change_{wrongcat}}$$

Where

$Change_{correct}$ = Area that is change in both the model result and the reference map

$Change_{ref}$ = Area that is change in the reference map, but not change in the model result

$Change_{mod}$ = Area that is change in the model result, but not change in the reference map

$Change_{wrongcat}$ = Area that is change in both the model result and the reference map, but was predicted a wrong category of change.

In our work, we combine Costanza's multi resolution metric [5] with Pontius et al. [9] "figure of merit". We extend Costanza's metric for cell spaces where the cells have numerical values, as described in the next section. Constanza's metric accounts for misallocation of cells with change and the Pontius extension avoids inflated large goodness of fit value due to large amounts of unchanged area.

3 A metric for goodness of fit in land change models

Top-down land change models usually have three sub-models: *demand*, *potential*, and *allocation* [7, 10, 11]. The demand depends on the underlying causes of change and represents how much change will happen. Usually, it is calculated externally by tools that consider economic, demographic and social trends. The demand is then spatially allocated based on the potential for change of each cell. For example, increase in global food consumption results in greater demand for agricultural areas.

Each place has a *potential for transition* between land cover classes. This potential depends on the relative importance of driving forces of change in that place. The potential represents the proximate causes, which are the factors directly linked to the locations. It combines data from different sources, such as distance to roads, soil quality, and protected areas to estimate the possibility of change from one given land cover to another. The result identifies the areas more likely to change. Finally, the *allocation* combines the demand and the potential to simulate where land change will take place. Given the demand is usually external to the model, modelers need to estimate the transition potential well so their simulations get closer to reality.

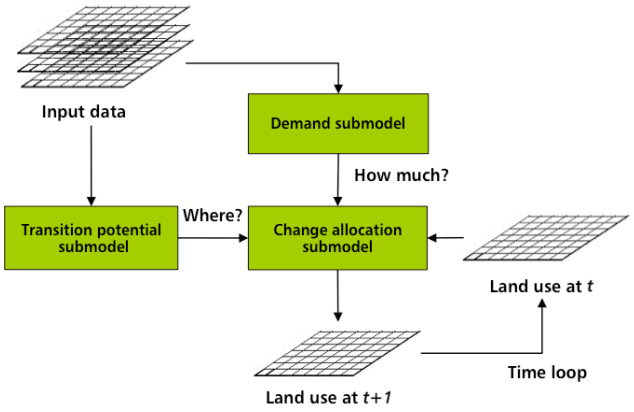


Figure 1: general view of top-down land change models. Source: Adapted from [12].

We propose a metric for evaluating goodness of fit in land change models that focuses on change and accounts for persistence. The metric takes Costanza's multiple resolution approach and restricts it to the areas of actual or projected change. This approach is consistent with the *figure of merit* metric proposed by Pontius, as discussed in section 2. Since our goal is to evaluate the potential and allocation procedures and not the correctness of the demand, the demand for change to be allocated in both cell spaces has to be the same.

We developed a metric for the case where one land cover transition is possible, e.g., from forest to deforested area. The metric can be extended to the case of multiple

land cover classes. For a single land cover transition, the metric is computed by the following equation:

$$F_w = 1 - \frac{\sum_{s=1}^{t_w} \left[\left| \sum_{i=1}^{w^2} a_{refi} - \sum_{j=1}^{w^2} a_{modj} \right| \right]_s}{2 \sum_{u=1}^{num} a_{refu}}$$

F_w = Goodness of fit at resolution w.

t_w = Number of sampling windows at resolution w.

w = Resolution (a sampling window has w^2 cells).

a_{refi} = Percent of change in land cover in cell i in the reference cell space.

a_{modj} = Change in land use/land cover in cell j in the model cell space.

i, j = Cells inside a sampling window.

u = Cells inside the cell space.

s = A sampling window.

num = Number of cells in the cell space ($t_w * w^2$)

For each window size, we get a goodness of fit metric by moving the window over the whole cell space and finding out the average value of the fit. The cell space is repeatedly traversed using sampling windows of increasing size (r). For each sampling window, we get the difference between quantity of change in the reference cell and quantity of change in the model cell space. We divide this difference by 2 to avoid double counting, since an increase in one location leads to a decrease by the same amount in another location. The error term is then summed over all windows and divided by the total change in the whole map. Subtracting it from one provides the goodness of fit.

The metric is appropriate to compare two cell spaces with the same spatial resolution and extent where the amount of change is the same in both. Using it, we find the degree of agreement between the cell spaces. The result is independent of the total area of the examined map. Therefore, including more (unchanged) cells (e.g. the cells outside the study area in a square map) in the computation does not alter the result.

4 Goodness of fit of Brazilian Amazon deforestation models

We applied the proposed goodness of fit metric to evaluated two models that try to predict deforestation in the Brazilian Amazon: The SimAmazonia model, developed by Soares-Filho et al. [3], and the model developed by Laurance et al. [4].

We evaluated model projections for the year 2011, taking the PRODES data provided by INPE (Brazilian National Space Research Institute) as the reference for observed deforestation. PRODES uses wall-to-wall mapping to get yearly data on the location and extent of the deforestation by clear cuts in the Brazilian Legal Amazon, an area of 5 million km^2. It uses remote sensing data with 20 to 30 meter resolution and produces deforestation maps in the 1:250.000 scale. Since 2003, INPE makes PRODES data freely available in the internet. The scientific community takes PRODES to be the standard reference for ground truth in Amazonia deforestation [13, 14].

SimAmazonia projects the deforestation in Amazonia in 2050, based on data from 2001. We estimated its results for 2011 using data provided by its authors. It has different submodels for 47 subregions of Amazonia, with modules considering socioeconomic factors and spatial factors (e.g. infrastructure projects). For our assessment, we took the Business-as-usual scenario (BAU) and the Governance scenario (GOV). Their main difference between BAU and GOV scenarios is the greater extent of government intervention in the latter case. The GOV scenario has more protected areas whose effectiveness is guaranteed.

The model by Laurance et al. projects deforestation in the Brazilian Amazonia in 2020 based on the data for 2000. It assumes a heavy impact of infrastructure projects that would lead to deforestation in Amazonia of 28% (optimistic scenario) or 42% (non-optimistic scenario) in 2020. The non-optimistic scenario assumes larger degraded areas close to roads and rivers and more deforestation in conservation areas. We could not get access to the original data, despite requests to the authors. Thus, we used input data and estimation methods as similar as possible to the author's description to simulate both scenarios for 2011.

We also used a neighborhood model as an example of the simplest possible land change model for Amazonia. The model has a single assumption: the potential for change in one year is the average deforestation of the neighboring cells for the previous year.

The demand for deforestation in all models is the actual total deforestation given by PRODES. The first two models originally projected higher demand compared to the PRODES estimates. We reimplemented such models in order to take into account the differences in the demand. The three models were implemented using TerraME toolkit [15]. Results for the goodness of fit at the highest resolution are shown in Figure 2. Both SimAmazonia models and the neighborhood model have goodness of fit values above 50%, which means that less than half of the demand was allocated in wrong places.

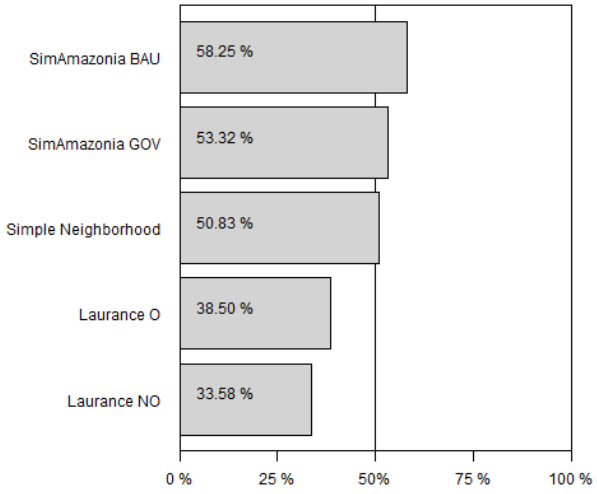


Figure 2: Bar chart of the goodness of fit at the finest resolution (pixel-wise comparison of reference and model result cell space). Laurance O and NO stand for the optimistic and the non-optimistic scenario.

Figure 3 shows the goodness of fit plotted against sampling window size. We see the differences between the model performances persist over many resolutions. The goodness of fit values increase slowly with increasing window size (note the logarithmic scale of the x axis). The steeper the increase of the fit curve, the more near-distance errors exist in the data. Near-distance errors occur when the mechanism allocates the change in a wrong location, but spatially close to the correct one. Thus, by increasing the window size, this misallocation gets smoothed out.

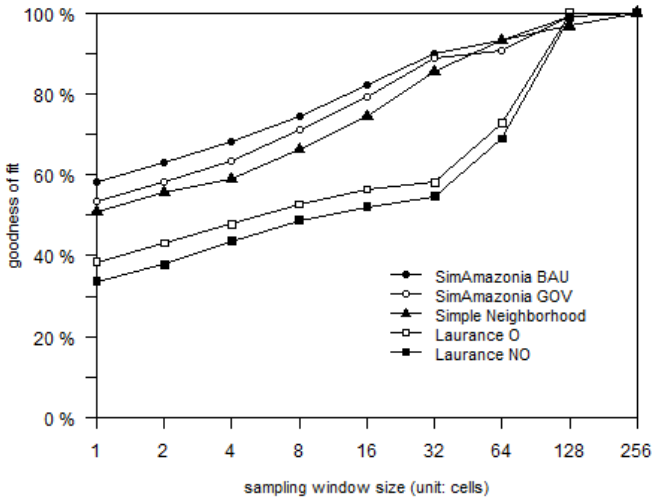


Figure 3: The goodness of fit of the different models plotted against sampling window size (logarithmic scale). The largest window is 256 by 256 cells. As it covers the whole cell space, which is 134 by 104 cells large, the goodness of fit is inevitably 100%.

The models allocate a lot of change in wrong regions. Both SimAmazonia models have a similar performance. Using a normalized demand, the allocation procedure is more realistic in the BAU scenario. The Laurance scenarios project most of the change in the wrong places. Even with sampling windows of size of 32 by 32 cells (800 by 800km), the Laurance models have a fit of only approximately 50%. The neighborhood model performs almost as well as the SimAmazonia models and much better than the Laurance models.

5 Discussion

The results presented in the previous section show that even the best model considered in our study allocates only about 60% of the change correctly. To understand the possible causes of allocation errors, we use the results of the neighborhood model. This model has a simple and restricted allocation procedure, which places all changes close to already deforested areas. We should expect that such a model reproduces the local extensions of existing areas correctly, but fails to account for new deforestation frontiers. Since the SimAmazonia models and the neighborhood model have a similar goodness of fit, we take that SimAmazonia models are not able to find out where the new frontiers of Amazonia are.

To explore further the factors that lead to modeling errors, we compare cell spaces of the deforestation as predicted by the models with the PRODES dataset in Figure 4. Because of their over-reliance on road infrastructure as the main factor for

deforestation, the Laurance models allocate much change in the wrong places. Laurance et al. consider the impact of roads in the more remote areas to be the same as those closer to the markets of Belém (a in Figure 4), Cuiabá (b) and the Brazilian Southeast. Therefore, public policies that would use these models for planning would be limited to avoiding road building. As the PRODES results show, some roads are much more relevant as drivers of deforestation than others. Capturing the relative importance of roads is thus important for models that could guide public policy making. Laurance et al. also underestimate the effectiveness of protected areas. Recent studies show that protected areas in Amazonia have very low deforestation and thus are an important part of forest protection policies [16].

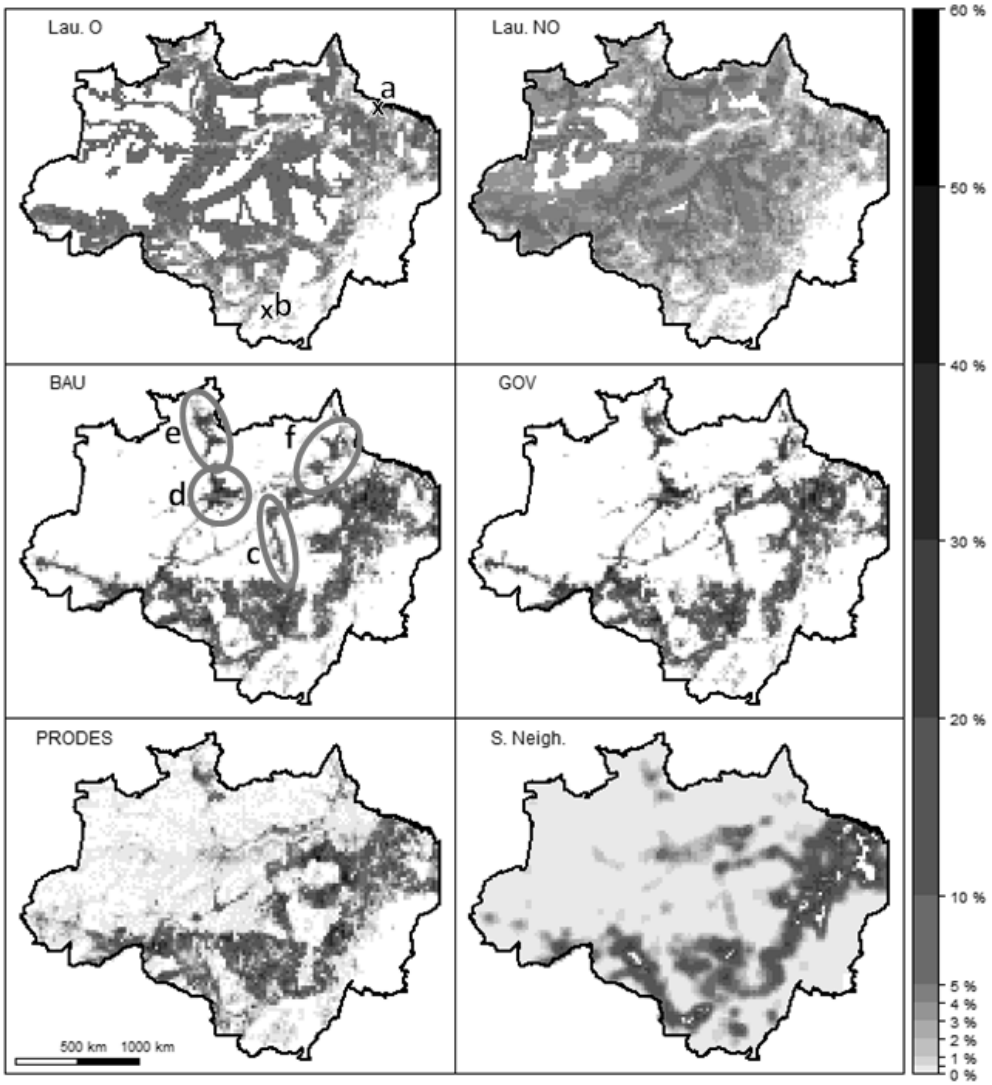


Figure 4: Maps of the area that was deforested in the years 2003-2011, according to PRODES data (lower left) and the model scenarios. The darker the cell's color, the higher the percentage of area deforested in that cell (values range from 0% to 60%). The letters a-f are explained in the text.

SimAmazonia captures most of the change close to existing deforested areas, but has a limited ability to predict how the frontier expands. It misses most of the deforestation around the Cuiabá-Santarém road (c) and predicted change close to

Manaus (d), in Roraima (e) and in the North of Pará (f) that did not happen. To fully understand the performance of the SimAmazonia model, a detailed analysis of its assumptions would be required, which is out of the scope of this paper. Our conjecture is that it is an effect of the 47 subregions used in the model. Breaking Amazonia into subregions is a sensible idea, since there are substantial differences inside the area. However, finding enough data to properly describe the factors that drive deforestation in each subregion is hard. Agricultural census data is available only at the municipality level. However, in Amazonia, municipalities have huge areas that make spatial allocation of driving factors extremely difficult. We presume that a better allocation of subregions in Amazonia could allow SimAmazonia to improve its goodness of fit.

6 Conclusion

This paper presents a metric to measure the quality of the transition potential and allocation methods of land change models. It uses a multi resolution method since spatial processes are scale-dependent. A model could have a low goodness of fit at detailed resolution, but could capture the general spatial pattern better than other models. The proposed metric focuses on changed areas and discards areas with no change. Thus, it prevents inflated goodness of fit values due to large fractions of the landscape remaining unchanged, as it often is the case in land change models.

We used the metric to compare several models that project deforestation in the Brazilian Amazon. Despite large quality differences between them, all the models have problems in predicting new frontiers and expansion areas. Because of this, the best model considered in this paper only performs slightly better than a simple model that predicts a cell's deforestation based on the deforestation in neighboring cells. We hope our results can motivate a new generation of deforestation models that better capture the socioeconomic factors underlying the decisions of the actors that carry out the deforestation.

7 References

1. Turner II, B., et al., *Land-Use and Land-Cover Change (LUCC): Science/Research Plan, IHDP Report No. 7*, 1995, IGBP Secretariat: Stockholm.
2. Rindfuss, R.R., et al., *Developing a science of land change: Challenges and methodological issues.* Proceedings of the National Academy of Sciences, 2004. **101**(39): p. 13976-13981.
3. Soares-Filho, B., et al., *Modelling conservation in the Amazon basin.* Nature, 2006. **440**(7083): p. 520-523.
4. Laurance, W., et al., *The future of the Brazilian Amazon.* Science, 2001. **291**: p. 438-439.
5. Costanza, R., *Model Goodness of Fit - a Multiple Resolution Procedure.* Ecological Modelling, 1989. **47**(3-4): p. 199-215.
6. Pontius Jr, R.G., E. Shusas, and M. McEachern, *Detecting important categorical land changes while accounting for persistence.* Agriculture, Ecosystems & Environment, 2004. **101**(2): p. 251-268.

7. Verburg, P.H., et al., *Modeling the Spatial Dynamics of Regional Land Use: The CLUE-S Model.* Environmental Management, 2002. **30**(3): p. 391-405.

8. Soares-Filho, B.S., G.C. Cerqueira, and C.L. Pennachin, *DINAMICA - a stochastic cellular automata model designed to simulate the landscape dynamics in an Amazonian colonization frontier.* Ecological Modelling, 2002. **154**(3): p. 217-235.

9. Pontius Jr, R.G., et al., *Comparing the input, output, and validation maps for several models of land change.* The Annals of Regional Science, 2008. **42**(1): p. 11-37.

10. Veldkamp, A. and L.O. Fresco, *CLUE: a conceptual model to study the Conversion of Land Use and its Effects.* Ecological Modelling, 1996. **85**: p. 253-270.

11. Pontius, R.G., J. Cornell, and C. Hall, *Modeling the spatial pattern of land-use change with GEOMOD2: application and validation for Costa Rica.* Agriculture, Ecosystems & Environment, 2001. **85**(1-3): p. 191-203.

12. Costa, S.S., et al. *Common Concepts to Development of the Top-Down Models of Land Changes*. In: Simpósio Brasileiro de Sensoriamento Remoto, 2009, Natal-RN.

13. Kintisch, E., *Carbon Emissions: Improved Monitoring of Rainforests Helps Pierce Haze of Deforestation.* Science, 2007. **316**(5824): p. 536-537.

14. Mithal, V., et al. *Time Series Change Detection using Segmentation: A Case Study for Land Cover Monitoring*. in *Conference on Intelligent Data Understanding, CIDU 2012*. 2012. Boulder, CO, USA.

15. Carneiro, T. G. S., et al. An extensible toolbox for modeling nature-society interactions. Environmental Modelling & Software, **46**, p. 104-117, 2013.

16. Nolte, C., et al., *Governance regime and location influence avoided deforestation success of protected areas in the Brazilian Amazon.* Proc. Natl Acad. Sc*i*, 2013, at press (doi: 1 0.1073/pnas. 1 214786110).

Developing a Framework for Modeling and Simulating Aedes aegypti and Dengue Fever Dynamics

Tiago F. M. Lima[1], Tiago G. S. Carneiro[2], Raquel M. Lana[3], Cláudia T. Codeço[3] Raian V. Maretto[4], Liliam C. C. Medeiros[5], Leandro G. Silva[1], Leonardo B. L. Santos[4], Isabel C. dos Reis[3], Flávio Codeço Coelho[6], Antônio M. V. Monteiro[4]

[1] LEDS - Laboratory of Engineering and Development of Systems
Department of Computer and Systems, Federal University of Ouro Preto (UFOP)
Ouro Preto - MG - Brazil

[2]TerraLAB - Earth System Modeling and Simulation Laboratory
Computer Science Department, Federal University of Ouro Preto (UFOP)
Ouro Preto - MG - Brazil

[3]National School of Public Health (ENSP)
Oswaldo Cruz Foundation (Fiocruz)
Rio de Janeiro - RJ - Brazil

[4]Division of Image Processing (DPI)
National Institute for Space Research (INPE)
São José dos Campos - SP - Brazil

[5]Institute of Science and Technology
Sao Paulo State University (UNESP)
São José dos Campos - SP - Brazil

[6]Getúlio Vargas Foundation (FGV)
School of Applied Mathematics
Rio de Janeiro - RJ - Brazil

tiagofml@leds.ufop.br, tiago@iceb.ufop.br, codeco@fiocruz.br

{raquelmlana, rvmaretto, liliam.castro, leandrogs99}@gmail.com

{santoslbl, izabio2005, fccoelho}@gmail.com, miguel@dpi.inpe.br

Abstract. *Dengue fever is a challenging complex transmissible disease due to its unstable temporal and spatial dynamics. Modeling is a powerful tool to understand disease dynamics and to evaluate costs, benefits and effectiveness of control strategies. In order to assist decision-makers and researchers in the evaluation and proposition of new strategies, we present DengueME, a collaborative open source platform for dengue modeling. It provides an environment for easy implementation of compartmental and individual-based models over a geographic database, combining modules describing* Aedes aegypti*'s life cycle, human demography, human mobility, urban landscape and dengue transmission.*

1. Introduction

Dengue fever is a vector borne disease characterized by a complex temporal-spatial dynamic which emerges from the interaction of multiple agents in a complex ecological and social landscape. Dissecting the contribution of specific variables to the observed dynamics as well as the effect of control strategies require modeling and simulation tools that integrate the several components of this system: ecological, social, demographic, mobility, immunological components, etc.

Several models have been developed aiming at understanding the dengue vector population dynamic and spread [Yang and Ferreira 2008] [Otero et al. 2006] [Otero et al. 2008] [Lana et al. 2011] [Lana et al. 2013], as well as the disease transmission dynamic [Santos et al. 2009] [Pinho et al. 2010] [Medeiros et al. 2011]. However, in general they represent homogeneous populations and scaling them up to several populations (metapopulation models) or introducing spatial heterogeneity demands a high dosage of efficient programming. One barrier in using these models to help in decision making is the time and effort usually required to implement and adapt these models to specific contexts, which requires parameterization with local spatial and temporal data, when data is often available at different spatial and/or temporal resolutions. A tool for modeling-based decision making should provide a variety of built-in models to attend the user as well as allow easy parameterization (eg. with meteorological data of a specific region) supporting the development of specific scenarios.

In this context, this work presents the conception and the initial state of development of the DengueME (Dengue Modeling Environment) framework, an open source tool for modeling and simulation of dengue models on geographical landscapes. DengueME is designed to simulate and compare site-specific and population-specific control strategies for dengue, offer a basic library of epidemiological and entomological spatio-temporal models that are easily configured and parameterized to attend the needs of the user and applied to real case studies. The framework seeks to provide an open environment for user contribution and sharing as well to foster the development of an active community.

2. DengueME framework overview

2.1. Framework requirements and design

DengueME was designed to attend the following requirements: (a.1) to allow efficient simulation of spatio-temporal dengue models; (a.2) to support efficient communication with geographical databases for storage of input and output data; (b) to allow easy implementation of multi-scale models as well as models based on different paradigms (compartmental, agent-based, cellular automata, hybrids); (c) to provide a common and easy language for model implementation; (d) to provide a rich library of built-in modular models which can be customized and combined to generate specific models; (e) to offer a friendly graphical user interface for creating, configuring and running built-in and user defined models; (f) to provide good documentation and a forum for communication between users; and (g) to provide standard data protocols for integration of models within the framework.

DengueME is developed over the TerraME platform [Carneiro et al. 2013], an open source programming environment for spatial dynamic modeling. This is a software architecture for building models with multiple scales that provides a rich modeling

language for the development of compartmental, individual and agent based modeling. It also provides an interface to TerraLib geographical databases, allowing direct access of models to geospatial data (http://www.terrame.org/).

Within DengueME, model implementation, parameterization and configuration use TerraML (Terra Modeling Language), a high level programming language for modeling which offer many data structures and services to support a rich description and simulation of environmental models [Carneiro et al. 2013].

As not all potential users of DengueME knows how to program in TerraML, we invested from the beginning in the design of a Visual Development Environment, which provides a graphical interface for users to configure built-in or imported models and create their own scenarios through the selection and combination of sub-models and their parameterization with data from geographic databases and tabular data. Its graphical user interface (GUI) currently provides wizards that guide users in performing the steps required for modeling 1. It allows users to select a model from a set of built-in models; to define values for the model parameters; to set the options for output visualization and storage. The user defined settings and parameters (designed scenarios) may be stored and retrieved for later use. After finishing all customization and parametrization, the GUI drives the creation of the corresponding TerraML source code.

2.2. Current built-in models

Currently, DengueME's library contains a dengue model as developed by Medeiros et al. (2011) in the form an agent-based model, and an *Aedes aegypti* population model as described in Lana et al. (2011). These models illustrate the main features of the DengueME platform, which include customization and comparison of transmission control scenarios.

The vector (entomological) model describes the population dynamics of *Ae. aegypti* with differential equations modeling the variation of the stock at each life stage (eggs, larvae, pupae and adults) under the influence of to the environmental carrying capacity and the climate [Lana et al. 2013]. The vector model may be used stand alone, or integrated with the other models.

The transmission process between humans and mosquitoes depends on the local amount of susceptible people (or mosquitoes) and infected people (or mosquitoes). Immune people act as barriers to transmission, since they absorb some of the bites from infected mosquitoes, without subsequent spread of the virus. Dengue models describe this dynamic through the classification of the stock of people and mosquitoes in states (susceptible, exposed, infected and recovered).

Future models to be implemented include a mobility model, which describes the mobility and commutation of humans and vectors through space. The spread of viruses and vectors is facilitated by the flow of individuals through the network of air and land transportation. Moreover, the urban landscape interferes in the dynamics of dengue transmission as it introduces heterogeneity in the availability of breeding sites for *Ae. aegypti*. The landscape component will provide a landscape classification model to describe the space and may be used to obtain input parameters for the models. DengueME will implement the landscape classification model suited to dengue issues developed by [Reis 2010].

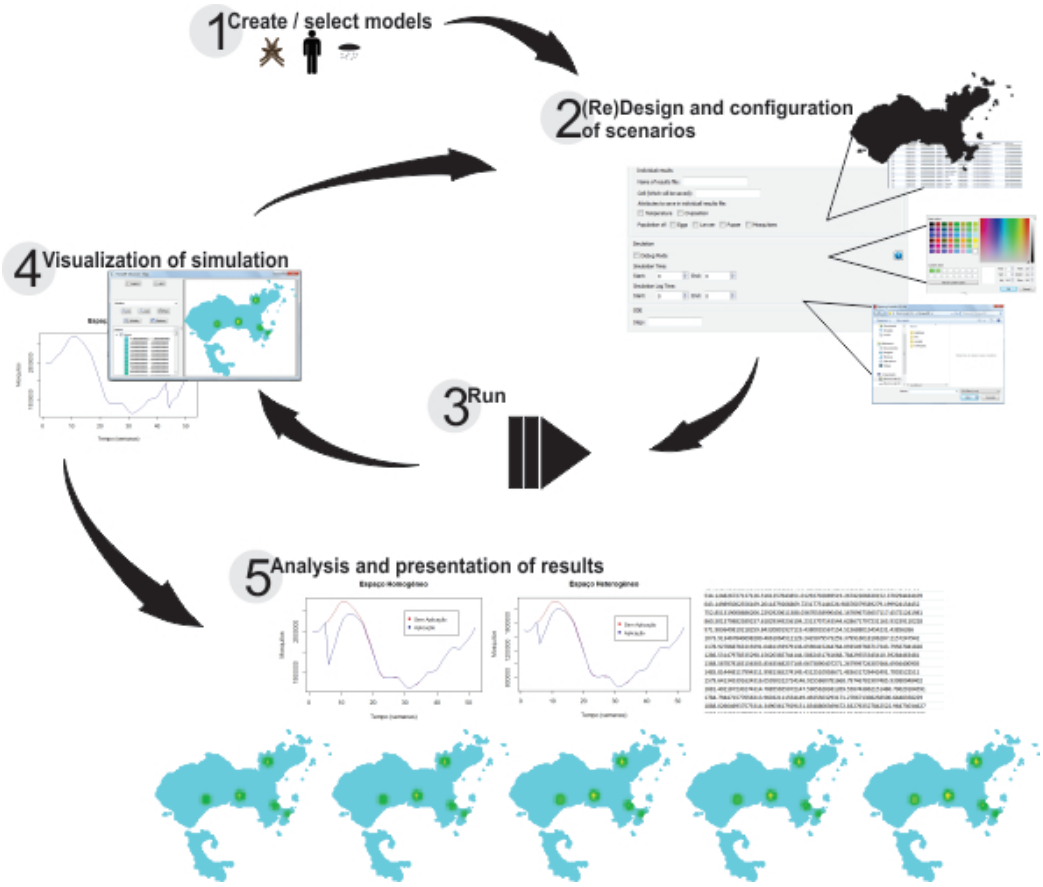


Figure 1. Diagram showing the modeling process using the DengueME Visual Development Environment.

3. Results and Final Remarks

Despite the variety of mathematical models for *Ae. aegypti* spatial dynamics and dengue transmission, there is still a need to stimulate model usage to assist the planning of dengue control interventions. Fast model reuse is still a challenge. Moreover, the application of models and tools should not require extensive experience in programming techniques or great expertise in modeling development, database management and and model-data integration (although it is required to understand the model assumptions to correctly use them).

These challenges motivated the development of the DengueME framework, which aims at supporting the modeling and simulation of the spatio-temporal dynamics of dengue and its vector in urban environments.

This work remains under development. Partial results were obtained, analyzed, and positively evaluated, motivating its continuity. Future work includes: (i) performing further framework evaluations with potential users, (ii) developing and incorporating new models into the framework, (iii) developing case studies, (iv) making a stable and documented version of the DengueME framework publicity available, (v) providing tutorials and demo applications to support users.

4. Acknowledgments

This work was partially supported by CNPq (Pronex dengue 550030/2010-7) and FAPERJ (Rede Dengue E-26/110.977/2011). We thank all members of the Pronex modeling network for their comments, ideas and support.

References

Carneiro, T. G. d. S., Andrade, P. R. d., Câmara, G., Monteiro, A. M. V., and Pereira, R. R. (2013). An extensible toolbox for modeling nature–society interactions. *Environmental Modelling & Software*, 46:104–117.

Lana, R. M., Carneiro, T. G., Honório, N. A., and Codeço, C. T. (2011). Multiscale Analysis and Modelling of Aedes aegyti Population Spatial Dynamics. *Journal of Information and Data Management*, 2(2):211.

Lana, R. M., Carneiro, T. G., Honório, N. A., and Codeço, C. T. (2013). Seasonal and nonseasonal dynamics of Aedes aegypti in Rio de Janeiro, Brazil: fitting mathematical models to trap data. *Acta Tropica*. Accepted for publication in 07/23/2013.

Medeiros, L. C. d. C., Castilho, C. A. R., Braga, C., de Souza, W. V., Regis, L., and Monteiro, A. M. V. (2011). Modeling the dynamic transmission of dengue fever: investigating disease persistence. *PLOS neglected tropical diseases*, 5(1):e942.

Otero, M., Schweigmann, N., and Solari, H. G. (2008). A stochastic spatial dynamical model for Aedes aegypti. *Bulletin of mathematical biology*, 70(5):1297–1325.

Otero, M., Solari, H. G., and Schweigmann, N. (2006). A stochastic population dynamics model for Aedes aegypti: formulation and application to a city with temperate climate. *Bulletin of Mathematical Biology*, 68(8):1945–1974.

Pinho, S. T. R. d., Ferreira, C. P., Esteva, L., Barreto, F., e Silva, V. M., and Teixeira, M. (2010). Modelling the dynamics of dengue real epidemics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1933):5679–5693.

Reis, I. C. (2010). Caracterização de paisagens urbanas heterogêneas de interesse para a vigilância e controle da dengue com o uso de sensoriamento remoto e mineração de padrões espaciais: um estudo para o Rio de Janeiro. Master in remote sensing, National Institute of Spatial Research.

Santos, L., Costa, M., Pinho, S., Andrade, R., Barreto, F., Teixeira, M., and Barreto, M. (2009). Periodic forcing in a three-level cellular automata model for a vector-transmitted disease. *Physical Review E*, 80(1):016102.

Yang, H. M. and Ferreira, C. P. (2008). Assessing the effects of vector control on dengue transmission. *Applied Mathematics and Computation*, 198(1):401–413.

A Framework for Web and Mobile Volunteered Geographic Information Applications

Clodoveu A. Davis Jr., Hugo de Souza Vellozo, Michele Brito Pinheiro

Depto. de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)
Av. Pres. Antônio Carlos, 6627 – 31.230-010 – Belo Horizonte – MG – Brasil

<clodoveu,vellozo,mibrito>dcc.ufmg.br

Abstract. *This paper proposes a framework to be used in the creation of various volunteered geographic information applications, incorporating both Web-based tools and mobile applications (apps). The framework includes the elements required to customize the information collection, while using a unified structure and interface across Web and mobile platforms. We demonstrate the use of the proposed framework in an application, named Strepitus, which gathers information on noise sources, so that users can report abusive sources of noise disturbance. The framework's elements and the components of Strepitus are presented in tandem, so that the design decisions are made clear. We observe that the framework allows for a quick development of new VGI applications, and this is important for capturing data on phenomena of current interest for the users, thus helping to maximize the number of volunteered contributions.*

1. Introduction

The quantity and variety of geographic data available on the Web for the use of the common citizen have been growing quickly. Since the introduction of Google Earth, in 2004, and of Google Maps, in 2006, there has been an increasing interest on tools that allow people to locate points of their interest and on applications that offer location-based services, such as finding urban addresses and selecting routes for vehicles. This kind of widespread interest in online maps and applications has led to an increasing demand for detail, especially in urban areas, and for currentness. People are quick to criticize a map for being outdated, not taking into consideration the cost and effort that would be required to keep huge volumes of data up to date. On the other hand, since the Web 2.0 proposal, several examples of situations in which users are allowed to contribute with the updating effort, or to supply more information, have come up, in a phenomenon usually called *Volunteered Geographic Information* (VGI) (Goodchild 2007; Goodchild 2007).

There are some online systems and Web sites that specialize in gathering user-generated spatial content. Most focus a specific theme, but overall the structure and working mechanisms of these tools is quite similar. Nevertheless, a new implementation is required for each different theme, so that parameters such as the geographic representation alternative (point, line, polygon), attributes to be collected, user identification, information validation and user feedback are adapted to the specificities of the theme and of the contributing crowd. The variety of these parameters makes it difficult to implement reusable solutions.

As a contribution to solve this problem, we propose a framework to be used in the creation of various volunteered geographic information applications, incorporating both Web-based tools and mobile applications (apps). The framework includes the elements required to customize the information collection, while allowing for a unified structure and interface across Web and mobile platforms. We demonstrate the use of the proposed framework in an application, named *Strepitus*, which gathers information on noise sources, through which users can report abusive sources of noise disturbance.

This article is organized as follows. Section 2 presents concepts and related work. Section 3 presents the framework's general architecture and components. Section 4 describes the application of the framework into the creation of Strepitus. Section 5 shows an analysis of the current version of the framework and its potential uses. Finally, Section 6 presents our final remarks and discusses further work.

2. Related Work

The mechanisms that lead people to spend time and effort to produce for free something that may or may not be useful to others are yet unclear (Budhathoki 2007; Goodchild 2007). However, the basic rule of the *wiki* mechanisms (originally from open source software development) seems to be valid for a wide range of situations: "if there are enough eyes, all *bugs* are shallow" (Raymond 1999). This means that if a dataset that is freely available on the Web is interesting for a large group of people, chances are that a reasonable share of these people would be willing to pay back the services they use through a disposition towards helping to improve the service for their peers. Such a phenomenon has been studied in Communications, where it is known as "collective action" (Bimber, Flanagin et al. 2005; Frew 2007), and directed towards public information stores, called *information commons* (Onsrud, Câmara et al. 2004).

Regarding geographic information, a similar phenomenon has been taking place in the last few years, under the generic denomination of Volunteered Geographic Information (VGI) (Goodchild 2007), but also known as spatial crowdsourcing, public participatory GIS, and other names. Starting with online maps and georeferenced image sets, such as Google Maps, Bing Maps, Yahoo Maps and others, several Web sites have been created counting on the perspective collaboration of users in the creation and/or maintenance of a niche geographic dataset. Through such mechanisms, any user, recognizing some aspect of the surrounding reality, can create an annotation, a comment, provide a place name, or connect a location to another source of complementary information. The prime example is perhaps the OpenStreetMap[1] project, with which users can contribute by editing street networks and providing an array of additional urban information. The resulting maps can be exported and used, costlessly, under an Open Database license.

Some studies (Haklay, Singleton et al. 2008; Keen 2008; Parker, May et al. 2012) show that arguments on VGI data being inferior to Professional Geographic Information (PGI) are invalid. User satisfaction derives from knowing when and how to use PGI, VGI or both. Volunteered information can be incomplete or inaccurate, but there are situations in which the only sources of detailed or thematically-relevant

[1] http://www.openstreetmap.org
[2]http://www.em.com.br/app/noticia/gerais/2012/06/07/interna_gerais,298842/denuncias-de-excesso-

information are volunteered. Therefore, VGI related projects still face numerous challenges: (1) how to effectively disseminate the data collection effort and how to motivate the largest number of people to contribute (Maué 2007), (2) how to maintain the contributors interested and active in the continuation of the effort (Coleman, Georgiadou et al. 2009; Soares and Santos 2011), (3) how to provide feedback to the contributors, generating personal or community benefits, and (4) how to validate and establish trust in collaboratively generated data (Frew 2007; Flanagin and Metzger 2008).

Our group has previously presented a generic platform for collecting and filtering volunteered geographic data (Silva and Davis Jr 2008). That initial effort was hindered by the use of proprietary software and lack of flexibility in the underlying structure. Sheppard (2012) presented a similar framework, with a high degree of reusability for VGI applications, using open source components and open standards, such as HTML5. One of the strong points of Sheppard's proposal is the use of generic code, which can run both in Web and mobile platforms. Unlike Sheppard's framework, we propose, within our framework, a set of structures that are designed to take advantage of specific hardware features in each of these environments. Besides, we use the Model-View-Controller design pattern (Reenskaug and Coplien 2009), and exercise a preference towards OGC-capable components. Therefore, our new approach, presented in the next sections, is a completely reengineered version of the previous work, adding a new architecture and new features, such as the integration of mobile platforms through the use of specific apps, while seeking a broad scope of applications on top of a simple and flexible structure.

3. Framework Architecture and Design

The proposed framework's architecture has been designed with the objective of encapsulating a basic structure for VGI applications into blocks that can be easily extended and reused in new projects. Basically, a new VGI application can be defined after some decisions are made, regarding (1) the geographic representation alternative for the collected data, (2) the required attributes, (3) the user management policies (login, history of contributions, reputation), (4) the validation policies, (5) interface elements for data inclusion and querying, and (6) user feedback, in terms of the display of the contributions, analysis of accumulated contributions, and others.

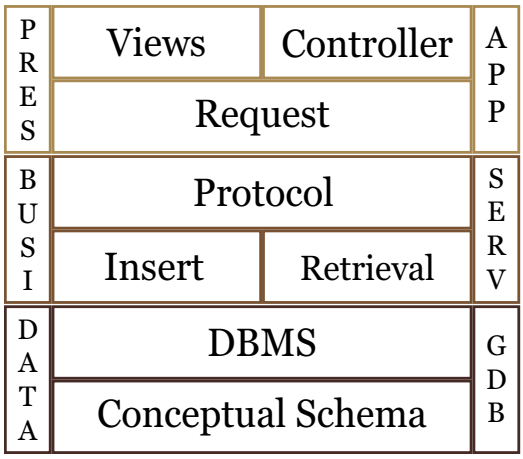


Figure 1 – General framework architecture

Based in our previous work, the framework's components are organized into three layers: Presentation, Business and Data (Figure 1). Details on the internal data components of each layer are depicted in Figure 2. The Presentation layer contains the modules that are responsible for data collection itself, implemented on multiple environments, such as Web and mobile. Currently, we have components for iOS and Android app creation. In the Business layer we concentrate services that perform the communication between the Data layer and the Presentation layer. The Data layer contains the geographic database that is responsible for storing the collected data.

The Data layer is defined within a spatial DBMS, after a conceptual modeling for the VGI application. The conceptual schema for the resulting database has been standardized for the framework, using a generic schema with three groups of objects: *users*, *contributions* and *contribution assessment*. In the *users* group, classes called User, LogUser and UserType are included. The first manages user accounts, including login names and passwords. The second records user activity in the system, for further analysis and for defining user trust and reputation. The third supports user classification into groups that are meaningful for the application. Functions define configuration alternatives such as (1) modes of user identification to access the system or to contribute, (2) raising users to moderator status, (3) the policies for establishing the user's reputation in the system, among others.

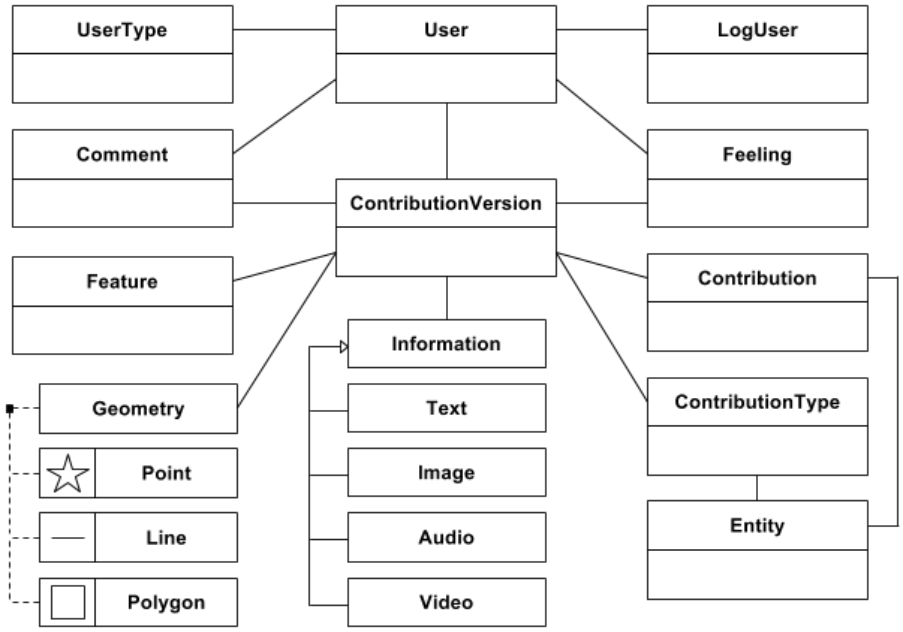


Figure 2 – Class diagram for the framework's generic database

In the *contributions* group, five classes are defined, namely *Contribution*, *ContributionVersion*, *ContributionType*, *Entity* and *Feature*. Each contribution can be edited to include further textual information, or associated to image, audio or video data. Contributions can be represented using points, lines or polygons. All versions of a contribution are stored, so that unmoderated contributions can be distinguished from ones that have undergone some revision. The *Entity* class allows for the definition of multithematic contributions, i.e., applications in which the user can provide data on

more than one theme. Furthermore, multiple types of contributions can exist for a given entity, so that variations in the definition of entities can be accommodated.

The third group of classes encompasses the ones that contain data on the assessment of user contributions. There are many validation strategies for VGI data, including peer validation and moderation. The *Feeling* class records peer opinions (rebuttals or confirmations, for instance). Additional textual data are recorded by the Comments class. Naturally, the validation mechanism and policies for filtering out inadequate contributions vary according to the data collection effort, and our intention with the framework is to provide a range of options for the creation of new applications.

Services in the Business layer are divided info three main components: *Protocol*, *Insertion and Retrieval Services*, and *Access Drivers*. The first component defines how the service will be available to the applications, i.e., the network protocol to be used in the exchange of messages, and their format. This is important for the separation of the Presentation layer, so that it can be implemented in multiple platforms. The *Insertion and Retrieval* services can be customized in order to accommodate the requirements of the applications and to respect constraint imposed by the database management system. The *Access Drivers* vary according to the DBMS used in the Data layer and to the computational environment for the implementation of the services.

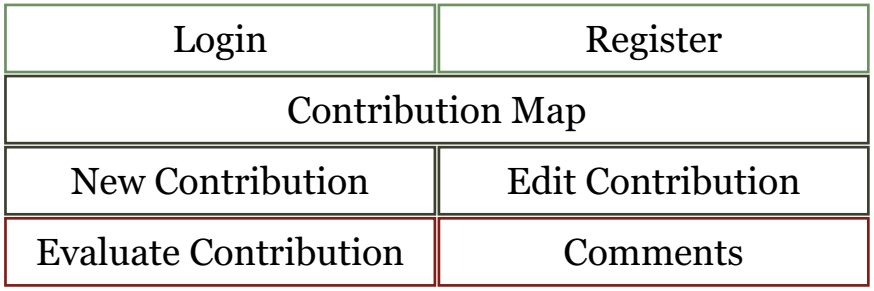


Figure 3 – Application view details

Finally, the Presentation layer implements the applications' user interface, which can be described using a Model-View-Controller design pattern. According to the pattern, the applications are defined as a set of *Views* that supply information to the *Controllers*. These, in turn, process the data supplied by the users along with data received by the Request module (our *Model* component, details below), which works as the input and output of application requests for the services. In the Views set, three subgroups have been defined, related to the groups of classes mentioned in the Data layer. Views for login and user registry supply and receive information from the first subgroup of data classes, implementing user creation and application access code. Views for the contributions in the map, attribute input and editing are also defined, and related to the second group of data classes. Finally, views for assessing contributions and comments input are coordinated with the third group of data classes.

Controller components are used to pre-process information. Such components can use, along with user-supplied data, information from the environment in which they have been developed. This is very common in mobile applications, in which the hardware is responsible for supplying data on the device's geographic location, using GPS or A-GPS hardware. Likewise, Web applications can return a geographic

coordinate derived from a mouse click, by inverting the projection transformation used to display a map or georeferenced image.

Our Model component is called Request, because its primary function is to send and receive information to the services. This module is able to handle one or more services, depending on the complexity of the application and on the resources that are available in the implementation environment.

4. Strepitus: a VGI application for noise mapping

4.1 Motivation

Living in large cities is getting more and more stressful, due to daily modern life problems. Activities that generate environmental impact are sometimes beyond the control of governmental authorities. The proximity of such impact-generating activities and housing districts produces a rising number of conflicts, often generating discussions that end up in police reports or even law suits.

In this context, disturbances caused by excessive noise levels are contrast with the need for a relative silence in resting periods. The Brazilian standard NBR-10151 establishes a limit of 60 dB for nightly noise in predominantly industrial areas, and 50 dB in residential areas. These limits are often exceeded in large cities. In Belo Horizonte, the number of complaints to governmental authorities about excessive noise has risen by 30% in 2012, relatively to 2011[2]. In São Paulo, a survey determined that about 20% of calls to the police over weekends are noise-related[3]. In Rio de Janeiro, noise pollution operations take place in the early hours of the day from Friday to Sunday, leading to police arrests[4]. However, the environmental laws and the action of the authorities are not powerful enough to keep the problems from happening, sometimes because of the difficulty to produce evidence on a problem that is geographically dispersed, does not occur all the time, and is more intense in out-of-office hours.

Inspired by the intensity and pervasiveness of noise complaints, we used the framework described in the last section to create a VGI application (named *Strepitus*) that is able to receive input from citizens on excessive noise levels. Strepitus was designed to let users indicate the location affected by a noise source, adding data on the sound intensity as an attribute. Since regulations have different limits for indoor and outdoor environments, the user also indicates whether the noise level has been recorded indoors or outdoors.

In mobile platforms, Strepitus is able to estimate the noise level using the device's microphone. Users are required to log in to the system, creating an account if necessary. After logging in, they can record a contribution. In the Web application, the user must indicate her location over a base map, and then fill out the attributes (noise level, indoors/outdoors) on the screen. In the mobile apps the position is obtained from

[2]http://www.em.com.br/app/noticia/gerais/2012/06/07/interna_gerais,298842/denuncias-de-excesso-debarulho-sobem-38-em-bh.shtml

[3] http://www.estadao.com.br/noticias/impresso,policiais-vao-fiscalizar-lei-do-silencio-,1010397,0.htm

[4]http://g1.globo.com/ma/maranhao/noticia/2012/05/em-cinco-meses-combate-poluicao-sonora-japrendeu-69-em-sao-luis.html

the device's operating system location functions, and the noise level is determined from the microphone interface.

The next subsections present the mapping from the framework to an actual VGI tool, along with elements from its operation.

4.2 Architecture

Along the lines of the proposed framework, Strepitus implements the three previously mentioned layers: Presentation, Business and Data. In the Data layer, a central database server has been configured to receive and maintain all necessary data. The Business layer includes a Web service provider and a metadata catalog server, which offer data access that goes beyond the need of the VGI application itself, organizing access to the underlying data tables. In the Presentation layer, two mobile applications have been developed, for the iOS and Android operating systems, along with a Web application.

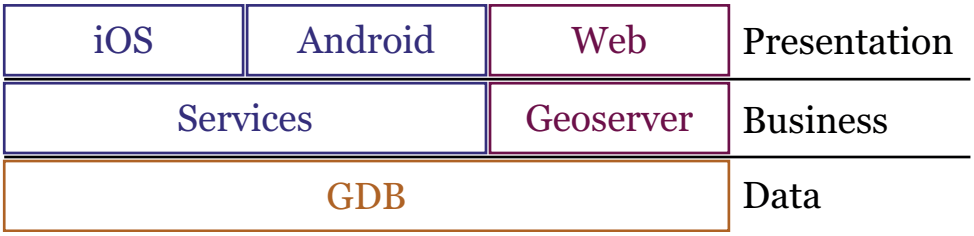


Figure 4 – Strepitus architecture

The generalized data schema described in Section 3 has been adapted so that the User, LogUser and ContributionType classes are defined for Strepitus. The ContributionType class holds information on noise sources, which correspond to classes of contributions in the framework. This is actually a table that holds typical sound intensity for common sources that serve as references to the users (Table 1).

Table 1. Noise sources and corresponding sound intensities

Noise source (Contribution type)	Typical intensity (dB)
Library	40
Conversation at home	50
Air conditioner	60
Dog bark	65
Vacuum cleaner	70
Classroom	75
Large volume of street traffic	80
Construction	100
Live show/concert	115
Sporting event	120
Party	120
Pneumatic hammer	130
Jet engine	135

For Strepitus, we merged the ContributionVersion, Contribution and Feature classes, storing all the volunteered data in the same structure. This alternative applies to situations in which editing is not allowed, and the attribute set is simple. In Strepitus,

values are not editable, since the noise level is typically captured by the hardware and events have a short time span.

In the Business layer, the framework was used to support the creation of a Web service running under PHP. This service runs from the same server that hosts the database server, but it can be directed to a separate machine, depending on the expected volume of user requests. According to the framework's architecture, HTTP is used as the message transmission protocol, and messages are encoded using JSON. The service offers mediation in the insertion of new users, contributions and application accesses, as well as for data recovery on the existence of users and on contributions that must be presented over a base map. For the access drivers, we used PHP libraries that support connections to PostGreSQL (PostGIS). Geoserver is also included in the Business layer, serving as a tool that provides data to be displayed along with application maps. As an application server, Geoserver supports connections to several different data sources and is able to publish them as OGC Web services (Percivall 2003). It also provides a metadata catalog. Using this component, additional layers can be presented over the base map, combining data from various distinct sources.

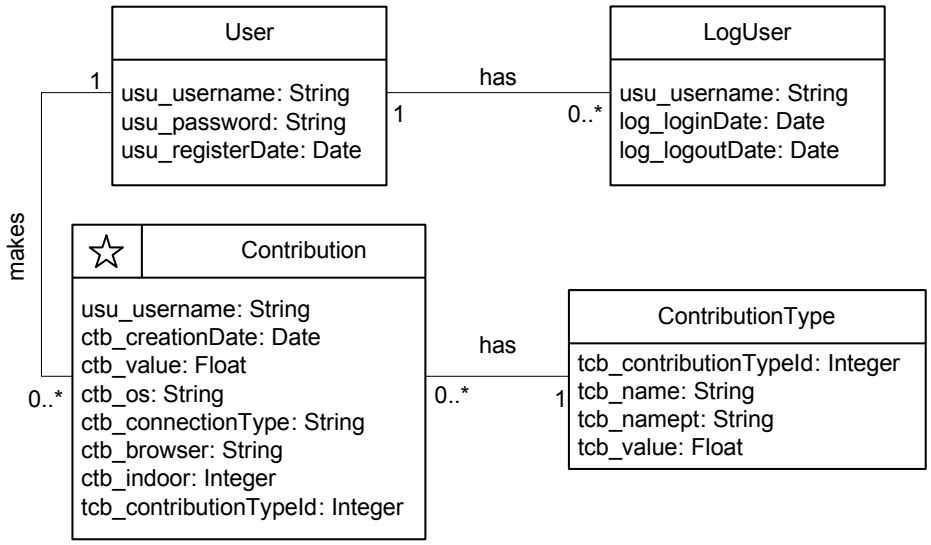


Figure 5 – Strepitus' database class diagram

In the Presentation layer, along with a Web application, mobile apps have been developed (Figures 1 and 2). Mobile development environments generally suggest the use of the MVC design pattern, so that the user interface does not have to be interrupted during data requests, and can therefore maintain processes dedicated to view updating. For the mobile apps, iOS and Android platforms were selected, due to their current popularity. The same set of user interface functions was developed for each platform, including login and registering screens, contributions map, contribution data collection, configurations, and help information. Special Controller modules have also been developed for each mobile platform, in order to get position data and noise intensity from the corresponding hardware components of the mobile computers. Finally, there are also framework components to send user requests to the database server, using the previously defined protocol, mentioned earlier.

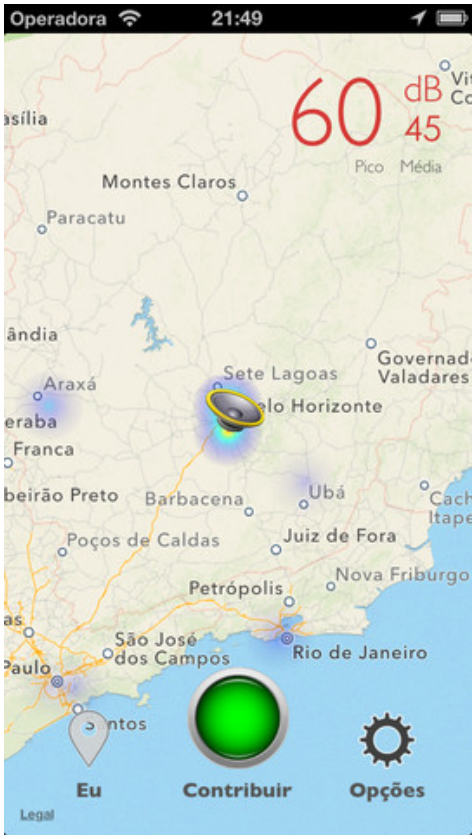


Figure 1 –iOS app

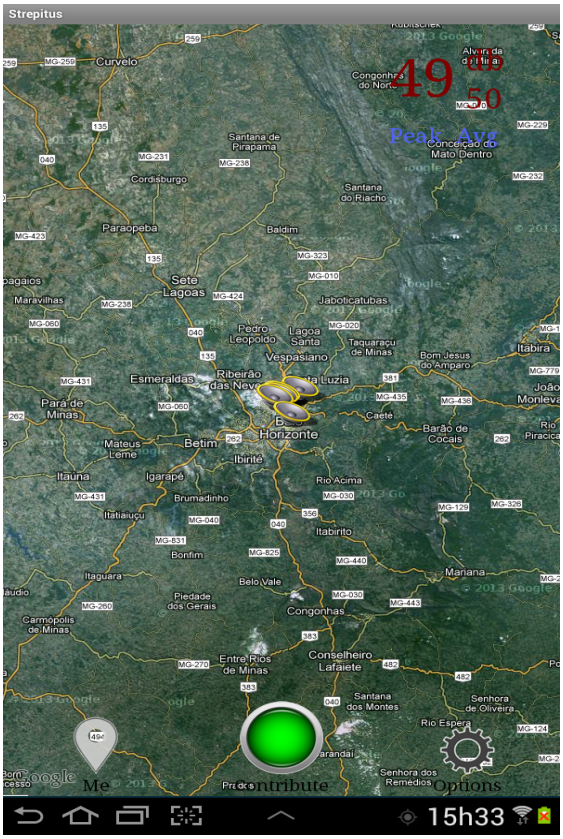


Figure 2 –Android app

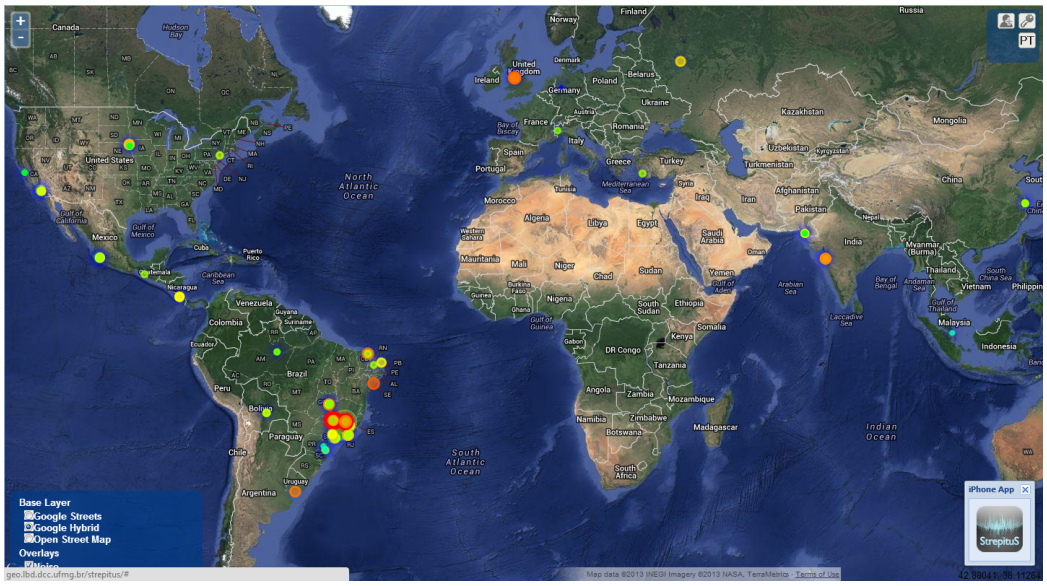


Figure 3 – Web application

The Web application (Figure 3) was developed using PHP and JavaScript, also using libraries such as OpenLayers, ExtJS and GeoExt. The user interface functions

developed for the mobile platforms were adapted to smaller windows, instead of using up all the screen space. Since the Web application runs in machines that typically are not equipped to measure noise, the user is provided with a reference scale of noise sources, so that the most approximate perceived noise level can be selected for the contribution (Table 1). The main Controller module used in the Web application is the one in which the user indicates the geographic position for the contribution. Among the Model components, the one responsible for requesting base map layers uses a communications protocol dedicated to OGC Web services.

5. Conclusions and Future Work

This paper presented a framework for the creation of VGI applications, capable of coordinating contributions from Web and mobile apps. Our framework includes all the components required to implement various VGI applications, aiming to reduce effort of preparing and publishing new VGI themes. In fact, our intention is to be able to quickly design and launch new VGI applications directed at current events, so that more people can be motivated to contribute.

Some components are still needed for the framework. In the near future, we intend to incorporate a user identification and login function that allows for the use of social network IDs, as in the case of Facebook and Twitter. Some integration with these social media is also sought, so that a contributor can simultaneously provide data and invite her friends to participate. We are also investing in a wider array of validation functions, in order to allow for user-based confirmation of peer-posted contributions. There is also space for additional visualization functions, to be used to provide visual feedback to users as to the universe of contributions and their geographic distribution.

As to Strepitus, the application is currently online, and the iOS and Android apps can be downloaded from the respective app stores. This application allowed us not only to test the ideas that led to the framework, but also to learn about the publication process for today's mobile apps. Since its publication, Strepitus has recorded contributions from all over the world, although only a small number of them, since no publicity campaign was made for its use. As previously mentioned, user motivation is part of the most relevant challenges for VGI research, and it is probably much easier to gain popular support to themes that are currently under intensive coverage by the media – and for that reason we intend to be able to generate VGI applications very quickly from the proposed framework.

Acknowledgments

The authors acknowledge the support by CNPq (560027/2010-9, 308678/2012-5) and FAPEMIG (CEX-PPM-00518/13), Brazilian agencies in charge of fostering research and development.

References

Bimber, B., A. J. Flanagin, et al. (2005). "Reconceptualizing collective action in the contemporary media environment." Communication Theory **15**(4): 365-388.

Budhathoki, N. R. (2007). Reconceptualization of user is essential to expand the voluntary creation and supply of spatial information. Workshop on Volunteered Geographic Information, Santa Barbara, California, USA.

Coleman, D. J., Y. Georgiadou, et al. (2009). "Volunteered geographic information: the nature and motivation of produsers." International Journal of Spatial Data Infrastructures Research **4**: 332-358.

Flanagin, A. J. and M. J. Metzger (2008). "The credibility of volunteered geographic information." GeoJournal **72**(3): 137-148.

Frew, J. (2007). Provenance and volunteered geographic information. Workshop on Volunteered Geographic Information, Santa Barbara, California, USA.

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

Goodchild, M. F. (2007). "Citizens as Voluntary Sensors: Spatial Data Infrastructure in the World of Web 2.0." International Journal of Spatial Data Infrastructures Research **2**: 24-32.

Haklay, M., A. Singleton, et al. (2008). "Web mapping 2.0: The neogeography of the GeoWeb." Geography Compass **2**(6): 2011-2039.

Keen, A. (2008). The Cult of the Amateur: How blogs, MySpace, YouTube, and the rest of today's user-generated media are destroying our economy, our culture, and our values, Random House Digital.

Maué, P. (2007). Reputation as a tool to ensure validity of VGI. Workshop on Volunteered Geographic Information, Santa Barbara, California, USA.

Onsrud, H., G. Câmara, et al. (2004). Public Commons of Geographic Data: research and development challenges. GIScience 2004 (LNCS 3234). M. J. Egenhofer, C. Freksa and H. J. Miller. Berlin/Heidelberg, Springer-Verlag**:** 223-238.

Parker, C. J., A. May, et al. (2012). "The role of VGI and PGI in supporting outdoor activities." Applied Ergonomics(In press).

Percivall, G. (2003). OpenGIS Reference Model, Open Geospatial Consortium, Inc.

Raymond, E. (1999). "The Cathedral and the Bazaar." from Available at http://www.tuxedo.org/~esr/writings/cathedral-bazaar/.

Reenskaug, T. and J. O. Coplien (2009) "The DCI Architecture: A New Vision of Object-Oriented Programming."

Sheppard, S. A. (2012). wq: A Modular Framework for Collecting, Storing, and Utilizing Experiential VGI. 1st ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information (GeoCrowd'12). Redondo Beach, CA**:** 62-69.

Silva, J. C. T. and C. A. Davis Jr (2008). Um framework para coleta e filtragem de dados geográficos fornecidos voluntariamente. X Brazilian Symposium on GeoInformatics (GeoInfo 2008), Rio de Janeiro (RJ), Sociedade Brasileira de Computação.

Soares, M. D. and R. Santos (2011). Ciência cidadã: o envolvimento popular em atividades científicas. Ciência Hoje. **47:** 38-43.

Indexing Vague Regions in Spatial Data Warehouses

Thiago Luís Lopes Siqueira[1,2], João Celso Santos de Oliveira[1], Valéria Cesário Times[3] Cristina Dutra de Aguiar Ciferri[4], Ricardo Rodrigues Ciferri[1]

[1] Department of Computer Science, Federal University of São Carlos, UFSCar, 13.565-905, São Carlos, SP, Brazil

[2] São Paulo Federal Institute of Education, Science and Technology, IFSP, 13.565-905, São Carlos, SP, Brazil.

[3] Informatics Center, Federal University of Pernambuco, UFPE, 50.670-901, Recife, PE, Brazil

[4]Department of Computer Science, University of São Paulo, USP, 13.560-970, São Carlos, SP, Brazil

prof.thiago@ifsp.edu.br, joaocelso@comp.ufscar.br, vct@cin.ufpe.br
cdac@icmc.usp.br, ricardo@dc.ufscar.br

Abstract. *A vague spatial data warehouse allows multidimensional queries with spatial predicates to support the analysis of business scores related to vague spatial data, addressing real world phenomena characterized by inexact locations or indeterminate boundaries. However, vague spatial data are usually represented and stored as multiple geometries and impair the query processing performance. In this paper, we introduce an index called VSB-index to improve the query processing performance in vague spatial data warehouses, focusing on range queries and vague regions. We also conduct an experimental evaluation demonstrating that our VSB-index provided remarkable performance gains up to 97% over existing solutions.*

1. Introduction

Decision-making support has gained the attention of researchers of Geographic Information System (GIS), Data Warehouse (DW) and Online Analytical Processing (OLAP). Fast, flexible, and multidimensional ways for spatial data analysis are provided by Spatial OLAP tools that query a Spatial Data Warehouse (SDW), which is a subject-oriented, integrated, time-variant, voluminous, non-volatile and multidimensional database that mainly stores crisp spatial data as vector (e.g. political boundaries) and their descriptive attributes (conventional data) [Bimonte et al. 2010]. A fact denotes the scores of business activities through numeric measures or spatial measures, while dimensions hold conventional attributes and spatial attributes that contextualize values of measures. Usually, spatial range queries concerning ad hoc spatial query windows select specific spatial objects stored in the SDW, e.g. intersection range query (IRQ) [Siqueira et al. 2012a]. The performance of query processing is a critical issue in SDW and motivates the design of indices to reduce the elapsed time to join huge tables, process spatial predicates and aggregate voluminous data [Papadias et al. 2001; Siqueira et al. 2012a].

Mainly, SDWs store crisp spatial data. On the other hand, several real world phenomena are affected by spatial vagueness, which is one kind of spatial data

imperfection concerning the difficulty of distinguishing an object shape from its neighborhood. As a result, it is not possible to be sure if the parts of a spatial object belong completely or partially to it or not. Exact models based on objects represent spatial vagueness reusing well-known crisp spatial data models and extending the theory of spatial data types and spatial predicates [Pauly and Schneider 2010].

A vague region *r* has a known extent and a broad boundary comprising a two-dimensional zone surrounding the known extent, instead of a one-dimensional line with minimal thickness. According to Pauly and Schneider (2010), *r* is a pair of crisp regions: the *kernel* and the *conjecture*. The *kernel* represents the known extent and is the determinate part of *r*, while the *conjecture* represents the broad boundary and is the vague part of *r*. The interior of the kernel and the interior of the conjecture are disjoint. If *p* is a point and belongs to the kernel, then it *certainly* belongs to *r*. However, if *p* belongs to the conjecture, then *p possibly* belongs to *r*. If *p* does not belong to the kernel and neither to the conjecture, then p does not belong to *r*.

Although vague SDWs store both vague spatial data and crisp spatial data, and allow multidimensional and spatial analysis of business scores regarding spatial vagueness, the design and use of SDWs supporting vague spatial data, i.e. vague SDWs, are still in infancy [Siqueira et al. 2012b; Edoh-Alove et al. 2013]. Furthermore, little attention has been devoted to the experimental evaluation of query processing performance in vague SDWs and to the investigation of the cost to process spatial predicates against vague spatial data represented as multiple geometries. Such investigation could aid designers to improve the performance of their systems.

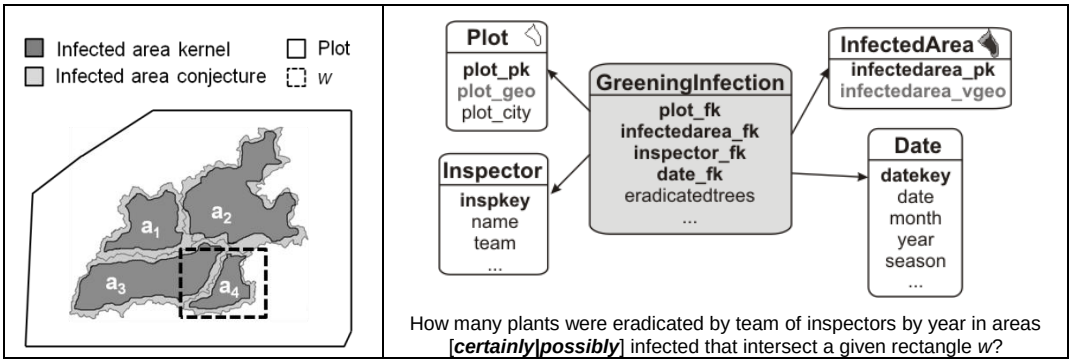
Motivated by the challenge of improving the query processing performance in vague SDWs, in this paper, we provide the following contributions: (i) an experimental evaluation of indices implemented by DBMSs and developed for SDWs to verify if they offer a reasonable query processing performance in vague SDW; (ii) a progressive approximation called MIP, which drastically reduces the cost of the spatial predicate resolution; and (iii) the proposal of an index called Vague Spatial Bitmap Index (VSB-index) to efficiently process multidimensional queries extended with spatial predicates concerning vague regions in SDWs.

This paper is organized as follows. Section 2 summarizes a case study, Section 3 surveys related work, Section 4 reports a performance evaluation of DBMS and Indices for SDW, Section 5 describes the VSB-index, Section 6 reports the experimental evaluation of the VSB-index and Section 7 concludes the paper.

2. Case Study: Greening

The following case study of a real problem in agriculture increases the motivation of this work. Greening is a serious disease that infects citrus and impairs the industry. It is caused by a bacterium transmitted by an insect. As there is not a cure so far, its control is done by visual inspection and immediate eradication of the infected plant by the roots. Temporal and spatial patterns of distribution of greening in the field at different scales, as plots and cities, are crucial to reduce the rate of failures in visual inspections [Silva et al. 2011]. Every area infected by greening is a vague region with broad boundaries as shown in Figure 1a. The kernel is the extent where plants were infected, while the conjecture is the broad boundary where the insect possibly transmitted the bacterium to plants.

Therefore, the vague SDW depicted in Figure 1b was built according to Siqueira et al. (2012b). *GreeningInfection* is a fact table referencing dimension tables and holding the numeric measure of eradicated plants. *Inspector* and *Date* are conventional dimension tables. *Plot* is a crisp spatial dimension table with the crisp spatial attribute *plot_geo* of type polygon. *InfectedArea* is a vague spatial measure pushed in a vague spatial dimension table with the vague spatial attribute *infectedarea_vgeo* of type multipolygon. Areas *certainly* infected are fetched as their kernels intersect the spatial query window *w*, e.g. a_3 and a_4 in Figure 1a. More coarsely, areas *possibly* infected are fetched as their conjectures intersect *w*, e.g. a_1, a_2, a_3 and a_4 in Figure 1a. These queries are more relevant to aid reducing failures in visual inspections than queries specified by exact models, e.g. Pauly and Schneider (2010), which compare a vague region to other vague regions of the dataset.



(a) Vague SDW schema **(b) Vague spatial data distribution**

Figure 1. A vague SDW on greening infection

3. Related Work

Currently, spatial predicates are solved in vague SDWs using indices of DBMSs, which are suitable for crisp spatial data and use the MBR as conservative approximation, e.g. R-tree [Guttman 1984] and GiST [Aoki 1997]. The filter step of the spatial predicate resolution uses strictly one conservative approximation: the MBR. Differently from our VSB-index, they do not perform a multistep spatial predicate resolution [Brinkhoff et al. 1993], i.e. they do not use a progressive approximation in addition to the MBR in order to identify answers already in the filter step to reduce the cost of the refinement step.

A bitmap join index on the attribute *C* of a dimension table indicates the set of rows in the fact table to be joined with a certain value of *C* [O'Neil and Graefe 1995]. Although the bitmap join index avoids joining huge tables in DWs, it cannot solve spatial predicates. In SDWs, the aR-tree [Papadias et al. 2001], the SB-index and the HSB-index [Siqueira 2012b] are capable to process spatial predicates, conventional predicates, aggregation and sorting. They use a single conservative approximation on crisp spatial data: the MBR. As a result, they produce identical sets of candidates to be processed in the refinement step. The aR-tree and the HSB-index have hierarchical data structures and a tree-based search in the filter step (similar to the R-tree's), while the SB-index has a sequential data structure and a sequential search in the filter step (similar to our VSB-index'). Conventional predicates, aggregation and sorting are processed by the aR-tree manipulating multidimensional arrays, while the SB-index, the HSB-index and our VSB-index reuse bitmap join indices to process them.

Some indices for vague spatial data focus on probability density functions and field data [Tao et al. 2005; Zinn et al. 2007]. On the other hand, the VSB-index addresses vector data. The vague R-tree is an index for vague regions based on the R-tree [Petry et al. 2007]. Its intermediate nodes maintain a pair of entries per cluster of vague regions. Entry *O* holds a MBR circumscribing the MBRs of the clustered vague regions, while entry *I* holds a MBR circumscribing the MBRs of kernels of the clustered vague regions. Progressive approximations were not addressed and only algorithms for point queries were designed, differently from our VSB-index that uses progressive approximations and tackles range queries. Besides, the vague R-tree was not assessed through an experimental evaluation, differently from the VSB-index.

4. Performance Evaluation of DBMS and Indices for SDW

In this section we conduct an experimental evaluation to demonstrate that indices implemented in DBMSs and indices for SDW are not suitable to manipulate vague regions. Section 4.1 addresses the experimental setup and Section 4.2 describes results.

4.1 Experimental Setup

Regarding the dataset, we processed real polygons of the rural census of the Brazilian Institute of Geography and Statistics to create the vague SDW shown in Figure 1b with 302,357 multipolygons in the attribute *infectedarea_vgeo*. The kernel was a negative buffer on the real polygon, while the conjecture was the convex hull of the real polygon minus the real polygon. The data generator of the Star Schema Benchmark produced conventional data for the other tables with scale factor 10 (60 million facts).

The workload was based on the query shown in Figure 2 (adapted from the Star Schema Benchmark) that assesses IRQ as spatial predicate, testing the rectangular ad hoc spatial query window *w* against a high cardinality attribute (e.g. infectedarea_vgeo). We performed 10 consecutive queries using disjoint spatial query windows, flushing the system cache between them, and gathered the average elapsed time. Given a spatial attribute of cardinality *c*, and a spatial query window *w* that retrieves *n* objects, the selectivity was $n \div c$. We used the following selectivity values for the spatial predicate: 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.02, 0.03 and 0.04. We assume that a user of SOLAP tool would hardly select more than 12,000 vague regions to be retrieved and displayed. Time reduction measured how much a configuration was more efficient than another. Configurations are described as follows.

Complying with the logical design methods of Siqueira et al. (2012b), configuration *DBMS1* assessed the DBMS processing the query transcribed in Figure 2 with *TABLE=InfectedArea* and *ATTRIBUTE=infectedarea_vgeo* over the schema shown in Figure 1b. A GiST index was built on infectedarea_vgeo to aid the processing of the spatial predicate on multiple geometries (multipolygons).

The SB-index and the aR-tree were evaluated due to their efficiency in SDWs. They were implemented in C/C++ and the disk page size was set to 8 KB. The filter step comprised index scan, while the refinement step accessed multipolygons of *infectedarea_vgeo*. The SB-index built bitmap join indices on attributes *team*, *year*, *infectedarea_pk* and *eradicatedtrees* using FastBit (https://sdm.lbl.gov/fastbit/), while

the aR-tree built a 1,000×7×302,357 3-dimensional array based on the cardinalities of *team*, *year*, *infectedarea_pk*, respectively, to store values of the measure *eradicatedtrees*.

In contrast to the logical design methods of Siqueira et al. (2012b), we created configuration *DBMS2* to assess the DBMS using a schema similar to that shown in Figure 1b, but replacing the table InfectedArea by table *InfectedArea2*, whose attributes were *infectedarea_pk*, *infectedarea_ker_geo* of type polygon storing the kernel of the vague region and *infectedarea_outb_geo* of type polygon storing the outer boundary of the conjecture of the vague region. Note that the outer boundary of the conjecture encompasses the vague region. The query transcribed in Figure 2 had the parameters set to *TABLE=InfectedArea2* and *ATTRIBUTE=infectedarea_outb_geo*. GiST indices were built on *infectedarea_ker_geo* and *infectedarea_outb_geo* to aid processing the spatial predicate on simple polygons instead of multiple polygons (DBMS1).

The platform was a computer with a 3.2 GHz Pentium D processor, 8 GB of main memory, a 7200 RPM SATA 320 GB hard disk with 8 MB of cache, Linux CentOS 6, PostgreSQL 9.2 and PostGIS 2.0.1.

```
SELECT team, year, SUM(eradicatedtrees) FROM Inspector, Date, GreeningInfection, TABLE
WHERE inspkey=inspector_fk AND datekey=date_fk AND infectedarea_fk=infectedarea_pk
AND team='XY' AND INTERSECTS(ATTRIBUTE,w)
GROUP BY team, year ORDER BY team, year
```

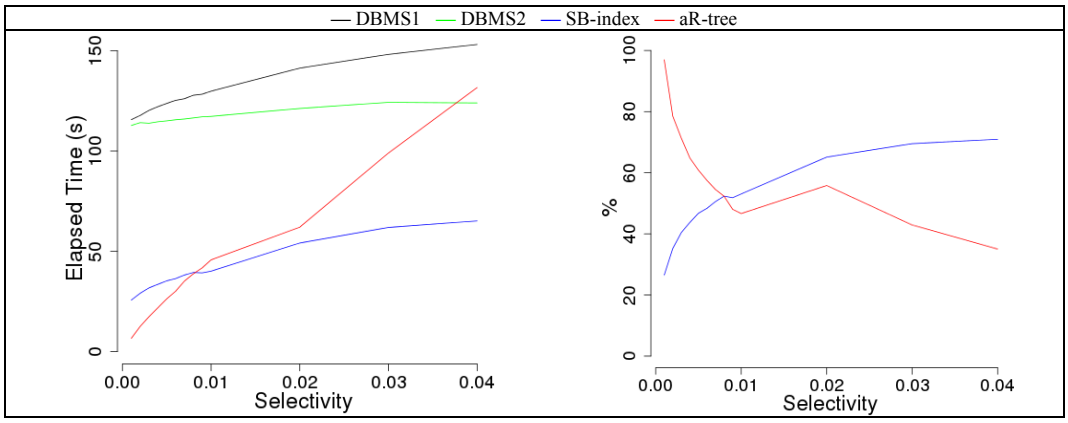
Figure 2. Querying the vague SDW shown in Figure 1b

4.2 Results

Figure 3a reports the elapsed time to process queries using the configurations DBMS1, DBMS2 and SB-index previously described. Clearly, as higher the selectivity value was, greater was the time spent to process queries in all configurations. Furthermore, the separation of the vague spatial attribute of type multipolygon in two attributes of type polygon benefited the query processing performance, since DBMS2 spent less time than DBMS1 to process queries. Such performance finding opposes the logical design methods stated by Siqueira et al. (2012b) and indicates that designers should separate vague regions in a vague SDW stored in the DBMS as we have done in Section 4.1.

However, both DBMS1 and DBMS2 had prohibitive query response times and indicated the necessity of using indices to provide a better performance. The aR-tree greatly overcame the other configurations for selectivity values less than 0.01, while the SB-index overcame the other configurations for higher selectivity values. Yet, both indices spent prohibitive times to process queries with increasing values of selectivity.

To identify the bottleneck in the query processing performance of the indices for SDW, we measured the time spent on each phase of its query processing algorithm. The fraction to resolve the spatial predicate is reported in Figure 3b. We concluded that the resolution of spatial predicates against vague regions was a costly step to process queries in vague SDWs. For the SB-index, such cost augmented as the selectivity of the spatial predicate increased. For instance, it was less than 30% for the selectivity 0.001 and greater than 70% for the selectivity 0.04. Clearly, the SB-index does not offer mechanisms to reduce the cost of the spatial predicate resolution against vague regions. Conversely, such cost decreased in the aR-tree for increasing values of selectivity since the manipulation of the multidimensional array imposed a larger overhead.



(a) Total elapsed time **(b) Fraction of the spatial predicate**

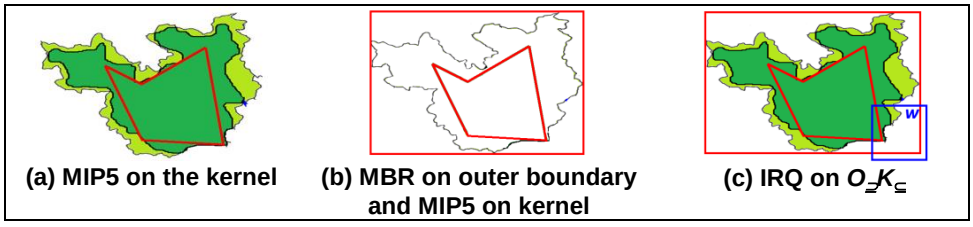
Figure 3. Results of DBMS and Indices for SDW

5. The Vague Spatial Bitmap Index

To propose the VSB-index, some design choices were made as follows, based on the previous discussions. We prioritized a multistep resolution of the spatial predicate and then created a specific progressive approximation to be used in the filter step and reduce the cost of the refinement step. We have chosen range queries as the spatial predicates to be supported by the VSB-index, initially, to satisfy the case study requirements. Since queries issued on a vague SDW process not only spatial predicates, but also conventional predicates, aggregation and sorting, the latter three are processed by bitmap join indices that are commonly used in DWs. This section details the proposal of the VSB-index and is organized as follows. Section 4.1 introduces the progressive approximation MIP. Section 4.2 defines the data structure of the VSB-index. Section 4.3 addresses the building operation of the VSB-index. Section 4.4 focuses on the VSB-index query processing.

5.1. Maximum Area Inscribed Polygon

The Maximum Area Inscribed Polygon (MIP) is a progressive approximation consisting of a polygon with x vertices. The number of vertices is the suffix, e.g. MIP5 for x=5. We define MIP to be applied specially on vague regions, to improve the resolution of spatial predicates in query processing. Figure 4a shows a vague region, its conjecture (light green), its kernel (dark green) and a MIP5 on its kernel (red contour). The outer boundary of the conjecture encompasses the vague region and therefore a MIP5 on the kernel is also a subset of the outer boundary of the conjecture, as shown in Figure 4b.



(a) MIP5 on the kernel **(b) MBR on outer boundary and MIP5 on kernel** **(c) IRQ on $O_{\supseteq}K_{\subseteq}$**

Figure 4. Vague region, approximations and query

5.2. Data Structure

The VSB-index for vague regions is an array whose entries have the type *vrbitvector* (vague region bit-vector), comprising: (i) one key value *pk*; (ii) one mandatory conservative approximation $O_{\supseteq}$ on the outer boundary of the conjecture; (iii) one optional progressive approximation $O_{\subseteq}$ on the outer boundary of the conjecture; (iv) one optional conservative approximation $K_{\supseteq}$ on the kernel; (v) one optional progressive approximation $K_{\subseteq}$ on the kernel; and (vi) a pointer *ptr* to the bitvector of the key value in a bitmap join index. The nomenclature considers the conservative approximation $_{\supseteq}$ as a superset of the vague region, and the progressive approximation $_{\subseteq}$ as a subset of the vague region. All feasible configurations for the VSB-index are listed in Table 1. The conservative approximation $O_{\supseteq}$ is mandatory to enable the query processing since the outer boundary of the conjecture encompasses the vague region. Except $O_{\supseteq}$, the other approximations are optional and allow flexible data structures and query processing algorithms (Section 5.4). We encourage using MIP as progressive approximations $O_{\subseteq}$ and $K_{\subseteq}$.

5.3. Building Operation

The building operation of the VSB-index issues a SQL query selecting the primary key and the spatial attribute from the spatial dimension table and sorting the results in ascending order based on the primary key. For each row retrieved, the approximations of the vague region are calculated and copied together with the key value into one entry of an array in the main memory. When the array becomes full, it is written to a disk page of the index file. After processing all rows and writing all disk pages, a bitmap join index is built on primary key values. Since the entries of the VSB-index are sorted by the primary key values, VSB-index[i] refers to the bitvector B[i] of the bitmap join index.

The size of a VSB-index entry, in bytes, is $s = sizeof(int) + sizeof(O_{\supseteq}) + sizeof(O_{\subseteq}) + sizeof(K_{\supseteq}) + sizeof(K_{\subseteq})$. Each disk page with l bytes maintains $L = l \; DIV \; s$ index entries. Some unused bytes $U = c \; MOD \; L$, where c is the cardinality of the indexed vague spatial attribute, are left between different disk pages to avoid fragmented entries and prevent two disk accesses to obtain a single entry. There is also a header disk page to store metadata. Then, $A = 1 + c \; DIV \; L + y$ disk pages are required to store the VSB-index, where y=0, if $c \; MOD \; L = 0$; and y=1, otherwise. Besides, A disk accesses are required to build the index file. Table 1 exemplifies values of s, L and A for MIP5, l=8192 bytes and c=302,357.

Table 1. Index entry sizes in bytes (*s*), number of entries per disk page (*L*) and number of disk pages used to store the index file (*A*).

	$O_{\supseteq}O_{\subseteq}K_{\supseteq}K_{\subseteq}$	$O_{\supseteq}O_{\subseteq}K_{\supseteq}$	$O_{\supseteq}O_{\subseteq}K_{\subseteq}$	$O_{\supseteq}O_{\subseteq}$	$O_{\supseteq}K_{\supseteq}K_{\subseteq}$	$O_{\supseteq}K_{\supseteq}$	$O_{\supseteq}K_{\subseteq}$	$O_{\supseteq}$
s	228	148	196	116	148	68	116	36
L	35	55	41	70	55	120	70	227
A	8640	5499	7376	4321	5499	2521	4321	1333

5.4. Query Processing

The range queries supported by the VSB-index are the following. Let w be an iso-oriented rectangle called ad hoc spatial query window and S be a set of vague regions. An $IRQ_{poss}(w,S)$ concerns an intersection that is *possibly* true and retrieves vague regions in S whose outer boundary of the conjecture intersects w. Conversely, an $IRQ_{cert}(w,S)$ concerns

an intersection that is *certainly* true and retrieves vague regions in S whose kernel intersects w. CRQ_{poss} and CRQ_{cert} are defined analogously for the relationship of containment. ERQ_{poss} and ERQ_{cert} are defined analogously for enclosure ("inside of").

The VSB-index query processing firstly performs a filter step as a sequential scan on the index file which requires A disk accesses (Section 5.3). The functions that execute such filter step are function *f1* detailed in Algorithm 1 and function *f2* detailed in Algorithm 2. They produce candidates and answers of the spatial predicate and store them in their proper sets in the main memory, i.e. *setCandidates* and *setAnswers*, respectively.

Function *f1* performs a sequential scan over the index file (lines 2-7), which retrieves each disk page (line 3) and temporarily stores it in the main memory (line 4). Function *get* obtains the conservative approximation of every entry transferred to main memory (line 6). Such conservative approximation is $O_{\supseteq}$ or $K_{\supseteq}$, depending on the parameter passed, and is tested against the ad hoc spatial query window (line 6). If the spatial relationship is satisfied, the entry's primary key value is appended to a set (line 7). Finally, the index file is closed (line 8). The aforementioned set might be the set of candidates or the set of answers, depending on the parameter passed.

To identify answers already in the filter step, function *f2* performs a sequential scan that firstly tests the conservative approximation and secondly tests the progressive approximation. For each entry (lines 5-10), if the spatial relationship is satisfied for both the conservative and progressive approximations, the entry is considered an answer and its primary key value is stored in the set of answers (lines 6-8). However, if only the conservative approximation satisfies the spatial relationship, the entry is considered a candidate and its primary key value is stored in the set of candidates (line 10). According to the calls, *f2* is particularly useful for IRQ_{poss} when $O_{\supseteq}$ and $O_{\subseteq}$ are available, and for IRQ_{cert} when $K_{\supseteq}$ and $K_{\subseteq}$ are available. Also, $K_{\subseteq}$ can be used to fetch results when querying IRQ_{poss}, as well as $O_{\supseteq}$ can be used to indicate candidates when querying IRQ_{cert}.

Algorithm 1: f1(R, w, conservative, set, idx, L)

Input: parameters described in Table 2

Declarations: page, array

Output: a set of candidates or a set of answers of the spatial predicate

```
1   open (idx)
2   while not (eof(idx)) do
3      read (idx, page)
4      copy (page, array)
5      for i ← 0 to L do
6         if R(w, get(array[i],conservative))
7            append(set, array[i].pk)
8   close(idx)
```

Algorithm 2: f2(R, w, conservative, progressive, setCandidates, setAnswers, idx, L)

Input: parameters described in Table 2

Declarations: page, array

Output: the set of candidates and the set of answers of the spatial predicate

```
1   open (idx)
2      while not (eof(idx)) do
3        read (idx, page)
4        copy (page, array)
5        for i ← 0 to L do
6          if R(w, get(array[i],conservative))
7             if R(w, get(array[i],progressive))
8              append(setAnswers, array[i].pk)
9             else
10             append(setCandidates, array[i].pk)
11  close(idx)
```

The filter step is a call to an adequate function based on a decision regarding the spatial predicate to evaluate and which approximations are available among $O_{\supseteq}$, $O_{\subseteq}$, $K_{\supseteq}$

and $K_{\subseteq}$. There are a total of 48 situations, but in this paper we focus on IRQ_{cert} and IRQ_{poss} which are solved by *f1* or *f2*. The 16 calls to process IRQ_{cert} and IRQ_{poss} are listed in Table 3. For instance, *f2* is called to process both IRQ_{cert} and IRQ_{poss} for $O_{\supseteq}K_{\subseteq}$ and adds the vague region shown in Figure 4c to the set of answers already in the filter step.

After the filter step, the refinement step is performed using the DBMS and its results are recorded in *setAnswers*. Further, a key-matching produces a string with a conventional predicate based on primary key values of those vague regions that satisfy the spatial predicate. Such string replaces the spatial predicate of the query submitted to the vague SDW. For instance, "INTERSECTS…" in Figure 2 is replaced by "(infectedarea_ pk=10 OR infectedarea_pk=15)", where 10 and 15 are key values of vague regions that satisfy the spatial predicate. Finally, the rewritten query is solved by efficient bitmap join indices that avoid joining huge SDW tables and provide the query answer.

Table 2. Parameters of Algorithms 1 and 2.

Parameter	Description
conservative	Indicator for $O_{\supseteq}$ or $K_{\supseteq}$
conservativeK	Indicator for $K_{\supseteq}$
conservativeO	Indicator for $O_{\subseteq}$
idx	The VSB-index file
L	The maximum number of index entries that a disk page can hold
progressive	Indicator for $O_{\subseteq}$ or $K_{\subseteq}$
progressiveK	Indicator for $K_{\subseteq}$
progressiveO	Indicator for $O_{\subseteq}$
pk	The primary key attribute of *table*
R	The spatial relationship (intersection, containment or enclosure)
setAnswers	The set of answers of the spatial predicate
setCandidates	The set of candidates (possible answers) of the spatial predicate
table	The vague spatial dimension table queried
vsa	The vague spatial attribute of *table*
w	The ad hoc spatial query window

Table 3. Calls made to functions *f1* and *f2* by configuration of the VSB-index.

Configuration	Function calls of IRQ_{cert} and IRQ_{poss} (R=intersection)
$O_{\supseteq}O_{\subseteq}K_{\supseteq}K_{\subseteq}$	**IRQ_{cert}:** `f2 (R, w, conservativeK, progressiveK, setCandidates, setAnswers, idx, L)` **IRQ_{poss}:** `f2 (R, w, conservativeO, progressiveO, setCandidates, setAnswers, idx, L)`
$O_{\supseteq}O_{\subseteq}K_{\supseteq}$	**IRQ_{cert}:** `f1(R, w, conservativeK, setCandidates, idx, L)` **IRQ_{poss}:** `f2 (R, w, conservativeO, progressiveO, setCandidates, setAnswers, idx, L)`
$O_{\supseteq}O_{\subseteq}K_{\subseteq}$	**IRQ_{cert}:** `f2 (R, w, conservativeO, progressiveK, setCandidates, setAnswers, idx, L)` **IRQ_{poss}:** `f2 (R, w, conservativeO, progressiveO, setCandidates, setAnswers, idx, L)`
$O_{\supseteq}O_{\subseteq}$	**IRQ_{cert}:** `f1(R, w, conservativeO, setCandidates, idx, L)` **IRQ_{poss}:** `f2 (R, w, conservativeO, progressiveO, setCandidates, setAnswers, idx, L)`
$O_{\supseteq}K_{\supseteq}K_{\subseteq}$	**IRQ_{cert}:** `f2 (R, w, conservativeK, progressiveK, setCandidates, setAnswers, idx, L)` **IRQ_{poss}:** `f2 (R, w, conservativeO, progressiveK, setCandidates, setAnswers, idx, L)`
$O_{\supseteq}K_{\supseteq}$	**IRQ_{cert}:** `f1(R, w, conservativeK, setCandidates, idx, L)` **IRQ_{poss}:** `f1(R, w, conservativeO, setCandidates, idx, L)`
$O_{\supseteq}K_{\subseteq}$	**IRQ_{cert}:** `f2 (R, w, conservativeO, progressiveK, setCandidates, setAnswers, idx, L)` **IRQ_{poss}:** `f2 (R, w, conservativeO, progressiveK, setCandidates, setAnswers, idx, L)`
$O_{\supseteq}$	**IRQ_{cert}:** `f1(R, w, conservativeO, setCandidates, idx, L)` **IRQ_{poss}:** `f1(R, w, conservativeO, setCandidates, idx, L)`

6. Experimental Evaluation of the VSB-index

This section reports the remarkable performance of the VSB-index. The experimental setup is described in Section 5.1, the building operation and the storage requirements are discussed in Section 5.2, and IRQ_{poss} and IRQ_{cert} are tackled in Section 5.3.

6.1 Experimental Setup

We extended the experimental setup described in Section 4.1 as follows. Five configurations of the VSB-index were reported because their results were more notable: $O_{\supseteq}$, $O_{\supseteq}K_{\supseteq}$, $O_{\supseteq}K_{\subseteq}$, $O_{\supseteq}O_{\subseteq}$ and $O_{\supseteq}O_{\subseteq}K_{\supseteq}K_{\subseteq}$. We used MBR as conservative approximation and MIP5 as progressive approximation – 5 vertices by analogy with 5C from Brinkhoff et al. (1993). The VSB-index was implemented in C/C++ and the disk page size was set to 8 KB. MIP5 was built using the CGAL, Computational Geometry Algorithms Library (http://www.cgal.org) version 4.0.2 and the method CGAL::maximum_area_inscribed _k_gon_2. The method uses monotone matrix search [Aggarwal et al. 1987] and has a worst case running time of $O(x \times n + n \times \log n)$, where n is the number of vertices provided as input to build the MIP and x is the number of vertices of the output.

6.2 Time spent and storage requirements to build

Figure 5a reports the elapsed time to build the sequential file of the VSB-index for each configuration and separates: (i) the time spent to extract the boundary of the kernel or the outer boundary of the conjecture; and (ii) the time spent to build the approximations and store them in disk. The configurations that hold only MBRs were built in shorter time, i.e. $O_{\supseteq}$ and $O_{\supseteq}K_{\supseteq}$. Also, the overhead to build the MIP5 on the outer boundary of the conjecture was significantly lower than the overhead to build the MIP5 on the kernel. Then, the time spent to build the configuration $O_{\supseteq}O_{\subseteq}$ was shorter than to build the configurations $O_{\supseteq}K_{\subseteq}$ and $O_{\supseteq}O_{\subseteq}K_{\supseteq}K_{\subseteq}$. In fact, the boundary of the kernel is a negative buffer and has more vertices than the outer boundary of the conjecture that is a convex hull (Section 4.1). Therefore, the high number of vertices of the kernel impaired the calculation of the MIP5 for configurations $O_{\supseteq}K_{\subseteq}$ and $O_{\supseteq}O_{\subseteq}K_{\supseteq}K_{\subseteq}$. The storage requirements for the VSB-index are detailed in Figure 5b. As expected, configurations that hold more approximations also require more storage space (Table 1). Considering that bitmap join indices occupied 4GB (Section 4.1), then the VSB-index' sequential file added at least 0.25% ($O_{\supseteq}$) up to at most 1.7% ($O_{\supseteq}O_{\subseteq}K_{\supseteq}K_{\subseteq}$) to storage requirements.

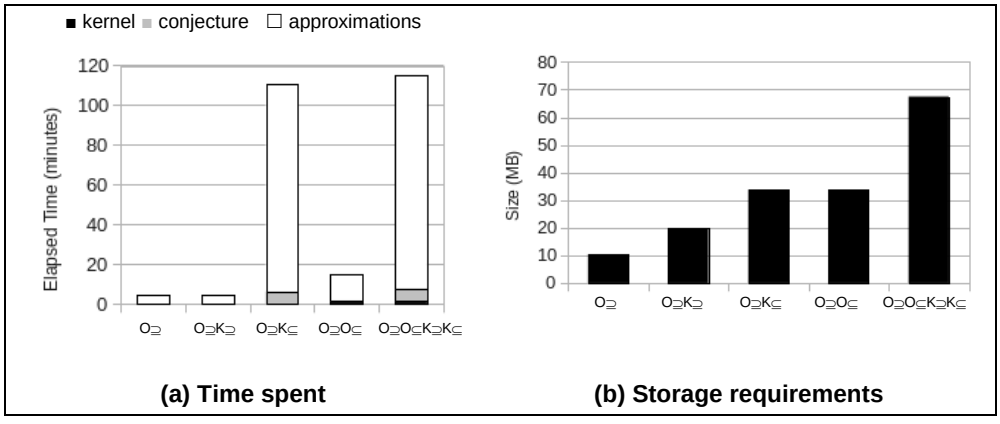


Figure 5. Results to build the VSB-index.

6.3 IRQ_{poss} and IRQ_{cert}

Instead of examining the time to process a complete query in the vague SDW (Section 3), this section focuses the resolution of the spatial predicate, motivated by the results discussed in Section 4.2. As SB-index' and aR-tree's performances were similar to the results achieved by the configuration $O_{\supseteq}$, we reported them together. A very low selectivity value of 0.0001 was included. Figure 6a shows the results of IRQ_{poss}. The configuration $O_{\supseteq}O_{\subseteq}$ outperformed the other configurations because since MIP5 allows identifying answers of the spatial predicate in the filter step. Although the configuration $O_{\supseteq}O_{\subseteq}K_{\supseteq}K_{\subseteq}$ produces the same set of candidates, given by the same call to function *f2*, more disk accesses are performed in the filter step due to a larger index entry size in bytes required to store four approximations (Table 1). Even though the query assessed the outer boundary of the conjecture, the progressive approximation MIP5 on the kernel of $O_{\supseteq}K_{\subseteq}$ provided a shorter query response time than configuration $O_{\supseteq}O_{\subseteq}K_{\supseteq}K_{\subseteq}$ that has the MIP5 on the outer boundary of the conjecture. Both configurations $O_{\supseteq}$ and $O_{\supseteq}K_{\supseteq}$ do not maintain a progressive approximation and therefore were severely impaired because they did not identify answers in the filter step and had a costly refinement step. Although the IRQ was issued over the outer boundary of the conjecture, a progressive approximation on the kernel improved the query processing performance, as the configuration $O_{\supseteq}K_{\subseteq}$ provided a time reduction of at least 23% and at most 97% over SB-index and aR-tree ($O_{\supseteq}$), for selectivity values 0.0001 and 0.04, respectively.

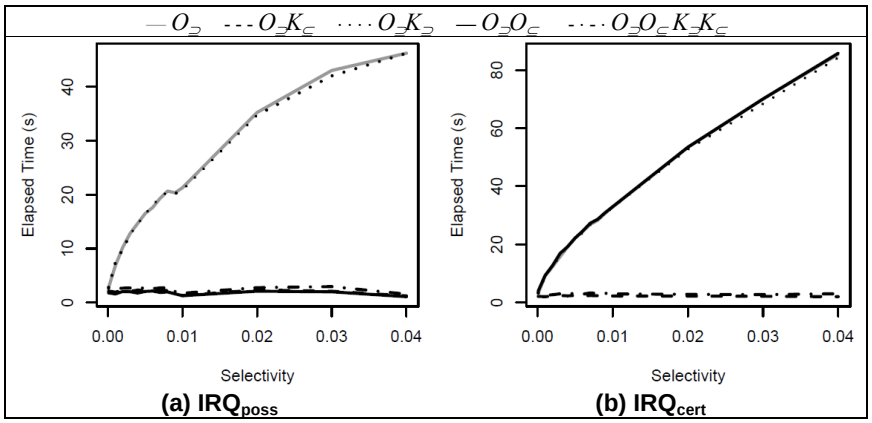


Figure 6. Results for IRQ_{poss} and IRQ_{cert}.

As for the VSB-index, Figure 6b shows the results of IRQ_{cert}. Curves for configurations $O_{\supseteq}$, $O_{\supseteq}K_{\supseteq}$ and $O_{\supseteq}O_{\subseteq}$ overlap each other. These configurations do not hold a MIP5 on the kernel and therefore had worst performances. As for the configuration $O_{\supseteq}O_{\subseteq}$, the progressive approximation the outer boundary of the conjecture could not improve the query processing performance of IRQ_{cert}, since it is not possible to obtain answers by calling the function *f1*. On the other hand, configurations $O_{\supseteq}O_{\subseteq}K_{\supseteq}K_{\subseteq}$ and $O_{\supseteq}K_{\subseteq}$ provided shorter query response times because they hold a MIP5 on the kernel and therefore can identify answers in the filter step. Again, the configuration $O_{\supseteq}O_{\subseteq}K_{\supseteq}K_{\subseteq}$ was impaired by its larger entry size that lead to more disk accesses to perform the filter step than the configuration $O_{\supseteq}K_{\subseteq}$. The configuration $O_{\supseteq}K_{\subseteq}$ provided a time reduction of at least 36% and at most 97% over SB-index and aR-tree ($O_{\supseteq}$), for selectivity values 0.0001 and 0.04, respectively. Notably, configuration $O_{\supseteq}K_{\subseteq}$ efficiently processed IRQ_{cert} and IRQ_{poss}.

7. Conclusions and Future Work

In this paper, we identified the lack of an index for vague SDW and the bottleneck to process vague spatial data using indices designed for crisp SDW. We also have gone one step forward and introduced the VSB-index to provide efficient processing of multidimensional queries extended with range queries against vague regions in vague SDWs. The VSB-index has a flexible data structure that enables the use of multiple approximations such that its query processing algorithm fits according to them. We also presented the progressive approximation MIP used by the VSB-index to reduce the cost of the refinement step in the spatial predicate resolution. An experimental evaluation corroborated the efficiency of the VSB-index that had remarkable performance gains up to 97% over existing solutions. Also, a study case was described to reinforce the feasibility of using our VSB-index in real applications. We are currently evaluating containment range queries and enclosure range queries. As future work, we intend to extend the VSB-index to support nearest neighbor queries and spatial joins, to index other data types as vague points and vague lines, and to enable SOLAP operations as roll-up and drill-down.

Acknowledgements. This work has been supported by FAPESP, CNPq, CAPES, INEP and FINEP. The 2nd author thanks PIBIC/CNPq/UFSCar for the undergraduate scholarship.

References

Aggarwal, A., Klawe, M. M., Moran, S., Shor, P. W., Wilber, R. 1987. Geometric applications of a matrix-searching algorithm. Algorithmica, 2, 195-208

Aoki, P.M. 1997. Generalizing "Search" in Generalized Search Trees. In ICDE, 380-389

Bimonte, S., Tchounikine, A., Miquel, M., Pinet, F. 2010. When Spatial Analysis Meets OLAP: Multidimensional Model and Operators. IJDWM, 6, 4, 33-60

Brinkhoff, T., Kriegel, H. P., Schneider, R. 1993. Comparison of Approximations of Complex Objects Used for Approximation-based Query Processing in Spatial Database Systems. In ICDE, 40-49

Edoh-Alove, E., Bimonte, S. Pinet, F., Bédard, Y. 2013. Exploiting Spatial Vagueness in Spatial OLAP: Towards a New Hybrid Risk-Aware Design Approach. In AGILE, 4p.

Guttman, A. 1984. R-Trees: A Dynamic Index Structure for Spatial Searching. ACM SIGMOD Record, 14, 2, 47-57

O'Neil, P., Graefe, G. 1995. Multi-Table Joins Through Bitmapped Join Indices. ACM SIGMOD Record, 24, 3, 8-11

Papadias, D., Kalnis, P., Zhang, J., Tao, Y. 2001. Efficient OLAP Operations in Spatial Data Warehouses. In SSTD, 443-459

Pauly, A., Schneider, M. 2010. VASA: An algebra for vague spatial data in databases. Inf. Syst., 35, 1, 111-138

Petry, F., Ladner, R., Somodevilla, M. 2007. Indexing Implementation for Vague Spatial Regions with R-trees and Grid Files. In: A. Morris, S. Kokhan, Geographic Uncertainty in Environmental Security, 187-199

Siqueira, T. L. L., Ciferri, C. D. A., Times, V. C., Ciferri, R. R. 2012a. The SB-index and the HSB-Index: efficient indices for spatial data warehouses. Geoinformatica, 16, 1, 165-205

Siqueira, T.L.L., Ciferri, C.D.A., Times, V.C., Ciferri, R. 2012b. Towards Vague Geographic Data Warehouses. In GIScience, 173-186

Silva, D.C.P., Posadas A., Jorge, L.A.C., Inamasu, R.Y. Paiva, M.S.V. 2011. Geração de mapas de incidência de greening utilizando técnicas Wavelets-Multifractais. In: R.Y. Inamasu et al. (Eds), Agricultura de Precisão: um novo olhar, EMBRAPA Instrumentação, 82-86 (Portuguese).

Tao, Y., Cheng, R., Xiao, X., Ngai, W., Kao, B., Prabhakar, S. 2005. Indexing multi-dimensional uncertain data with arbitrary probability density functions. In VLDB, 922-933

Zinn, D., Bosch, J., Gertz, M. 2007. Modeling and Querying Vague Spatial Objects Using Shapelets. In VLDB, 567-578

Index of authors