Proceedings

Thales Sehn Körting and Renato Fileto (Eds.)

Dados Internacionais de Catalogação na Publicação

SI57a Simpósio Brasileiro de Geoinformática (11. : 2015: Campos do Jordão,SP)

Anais do 16º Simpósio Brasileiro de Geoinformática, Campos do Jordão, SP, 29 de novembro a 2 de dezembro de 2015. / editado por Thales Sehn Körting (INPE), Renato Fileto (UFSC). – São José dos Campos, SP: MCTI/INPE, 2015.
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. Körting, T. S. II.Fileto, R. III. Título.

CDU: 681.3.06

Preface

The Brazilian Symposium on GeoInformatics (GEOINFO) is now consolidated as the most important reference of quality research in geographic information science and related fields in Brazil. It brings together researchers, students and practitioners from several Brazilian states, and also from abroad. GEOINFO 2015 takes place at the same cosy venue in Campos do Jordão, where participants have the privilege to be close to each other most of the time, for constructively discussing ongoing research, developments, and innovative applications. This environment has helped to widen and strengthen our community, with new cooperations, advices, and joy for nurturing ideas and developing new things, while also having a good time, all together.

Past GEOINFO editions included special keynote presentations by Max Egenhofer, Gary Hunter, Andrew Frank, Roger Bivand, Mike Worboys, Werner Kuhn, Stefano Spaccapietra, Ralf Guting, Shashi Shekhar, Christopher Jones, Martin Kulldorff, Andrea Rodriguez, Max Craglia, Stephen Winter, Edzer Pebesma, Fosca Giannotti, Christian Freksa, Thomas Bittner, Markus Schneider, Helen Couclelis, Randolph W. Franklin, and Paul Brown. In this edition, we have the pleasure to receive as keynote speakers Dr. Michael Batty, from the University College London, UK, and Dr. Jan Verbesselt, from Wageningen University, The Netherlands.

This year we have experienced an expansion in the number of paper submissions. Authors of papers accepted for GEOINFO 2015 come from a higher number and a wider variety of institutions than many previous editions. In our opinion, these facts evidence the vitality and evolution of our research community. Thus, please, enjoy the papers of our technical sessions, and help to make them even better, with your questions and comments. We would like to thank all the Program Committee (PC) members, whose voluntary work was essential to ensure the quality of every accepted paper. The help and advice of previous organizers and of the GEOINFO steering committee have also been paramount for the success of the current edition. At least three PC members contributed with their review to each paper submitted to GEOINFO.

Special thanks to Antônio Miguel V. Monteiro, Vania Bogorny and Clodoveu A. Davis Jr. for gently lending their experience, sharing data, giving advice, and always having great ideas at the right times. Laercio Namikawa for his promptly last time reviews. Lúbia Vinhas for gently accepting to organize the Lightning Talks. Our warmest thanks to the many people involved in the organization and execution of the symposium, particularly our invaluable support team: Janete da Cunha, Daniela Seki and Denise Nascimento.

Finally, we would like to thank GEOINFO's supporters, the Society of Latin American Remote Sensing Specialists (SELPER), the National Council for Scientific and Technological Development (CNPq – *Conselho Nacional de Desenvolvimento Científico e Tecnológico*), the São Paulo Research Foundation (FAPESP – *Fundação de Amparo à Pesquisa do Estado de São Paulo*), and Boeing Research & Technology, identified at the symposium's Web site. The Brazilian National Institute for Space Research (INPE – *Instituto Nacional de Pesquisas Espaciais*) has provided again much of the energy and commitment required to bring together this research community, now as in the past, and continues to perform this role through their numerous research and related activities.

Florianópolis and São José dos Campos, Brazil, November, 2015.

Renato Fileto
Program Chair

Thales Sehn Körting
General Chair

Conference Committee

General Chair

Thales Sehn Körting
National Institute for Space Research, INPE

Program Chair

Renato Fileto
Federal University of Santa Catarina, UFSC

Local Organization

Daniela Seki
INPE

Denise Nascimento
INPE

Janete da Cunha
INPE

Support

SELPER - Sociedade Latino Americana de Especialistas em Sensoriamento Remoto
CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico
FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo
BOEING - Boeing Research & Technology

Program Committee

Alan Salomão, UFRJ, Brazil
Alessandra Raffaeta, Università Ca'Foscari Venezia, Italy
Alexandre Noma, UFABC, Brazil
Ana Paula Afonso, Universidade de Lisboa, Portugal
André Santanchè, UNICAMP, Brazil
Andrea Iabrudi Tavares, UFOP, Brazil
Angela Schwering, IFGI, Germany
Antônio Miguel V. Monteiro, INPE, Brazil
Armanda Rodrigues, NOVA LINCS, Portugal
Bart Kuijpers, Hasselt University, Belgium
Carla Macario, Embrapa, Brazil
Carlos Felgueiras, INPE, Brazil
Carolina Pinho, UFABC, Brazil
Chiara Renso, ISTI/CNR, Italy
Cláudio Baptista, UFCG, Brazil
Claudio Silvestri, Università Ca'Foscari Venezia, Italy
Clodoveu A. Davis Jr., UFMG, Brazil
Daniel Zanotta, IFRS, Brazil
Dieter Pfoser, George Mason University, USA
Dário Oliveira, GE Global Research, Brazil
Edzer Pebesma, University of Munster, Germany
Fabiano Morelli, INPE, Brazil
Fernando Bação, UNL, Portugal
Flávia Feitosa, UFABC, Brazil
Francisco Javier Moreno, Universidad Nacional, Colombia
Frederico Fonseca, The Pennsylvania State University, USA
Gilberto Câmara, INPE, Brazil
Gilberto Ribeiro de Queiroz, INPE, Brazil
Gilson Alexandre Ostwald Pedro da Costa, PUC-RJ, Brazil
Helen Couclelis, University of California, USA
Holger Schwarz, University of Stuttgart, Germany
Joachim Gudmundsson, The University of Sydney, Australia
Jorge Campos, UNIFACS, Brazil
José Antonio Macêdo, UFC, Brazil
João Paulo Papa, UNESP, Brazil
Jugurta Lisboa Filho, UFV, Brazil
Jussara O. Ortiz, INPE, Brazil
Karine R. Ferreira, INPE, Brazil
Lúbia Vinhas, INPE, Brazil
Laercio Namikawa, INPE, Brazil
Luciana Alvim Romani, Embrapa, Brazil
Luis Otavio Alvares, UFSC, Brazil
Marcelino P. S. Silva, UERN, Brazil
Marcus Vinicius A. Andrade, UFV, Brazil
Maria Isabel S. Escada, INPE, Brazil
Matt Duckham, University of Melbourne, Australia
Monica Wachowicz, University of New Brunswick, Canada
Mário J. Gaspar da Silva, Universidade de Lisboa, Portugal
Nikos Pelekis, University of Piraeus, Greece
Pedro R. Andrade, INPE, Brazil
Raul Q. Feitosa, PUC-RJ, Brazil
Renato Fileto, UFSC, Brazil
Ricardo R. Ciferri, UFSCAR, Brazil
Rodrigo da Silva Ferreira, PUC-RJ, Brazil
Rogério Galante Negri, UNESP, Brazil
Sergio D. Faria, UFMG, Brazil
Sergio Rosim, INPE, Brazil
Silvana Amaral, INPE, Brazil
Stephan Winter, University of Melbourne, Australia
Sérgio Costa, UFMA, Brazil
Thales Sehn Körting, INPE, Brazil
Thomas Kemper, JRC, Italy
Valéria C. Times, UFPE, Brazil
Vania Bogorny, UFSC, Brazil
W. Randolph Franklin, Rensselaer Polytechnic Institute, USA
Yannis Theodoridis, University of Piraeus, Greece

Contents

Temporal GIS and Spatiotemporal Data Sources

Karine Reis Ferreira, André Gomes de Oliveira, Antônio Miguel Vieira Monteiro, Diego Benincasa F. C. de Almeida

Image Processing Division (DPI)
National Institute for Space Research (INPE) - São José dos Campos – Brazil

karine@dpi.inpe.br, andre.oliveira@funcate.org.br, miguel@dpi.inpe.br, benincasa@dpi.inpe.be

Abstract. *The recent technological advances in geospatial data collection have created massive data sets with better spatial and temporal resolution than ever. To properly deal with these data sets, geographical information systems (GIS) must evolve to represent, access, analyze and visualize big spatiotemporal data in an efficient and integrated way. In this paper, we highlight challenges in temporal GIS development and present a proposal to overcome one of them: how to access spatiotemporal data sets from distinct kinds of data sources. Our approach uses Semantic Web techniques and is based on a data model that takes observations as basic units to represent spatiotemporal information from different application domains. We define a RDF vocabulary for describing data sources that store or provide spatiotemporal observations.*

1. Introduction

The recent technological advances in geospatial data collection, such as Earth observation and GPS satellites, have created massive data sets with better spatial and temporal resolution than ever. This scenario has motivated a challenge for Geoinformatics. We need geographical information systems (GIS) able to deal with big spatiotemporal data sets in an efficient and integrated way.

We use the term "temporal GIS" to refer to GIS that can model, access, combine, process, analyze and visualize spatiotemporal information. In the literature, there are many proposals of conceptual models to represent and handle spatiotemporal data in GIS and database systems. However, there is not yet a full-scale and comprehensive temporal GIS available (Yuan, 2009). Most existing temporal GIS technologies either are still in the research phase or are specific for certain application domain.

In this paper, we highlight challenges faced in temporal GIS development and present a proposal to overcome one of them: *how to access spatiotemporal data sets from distinct kinds of data sources*. Our approach uses Semantic Web techniques and is based on a data model that takes observations as basic units to represent spatiotemporal information from different application domains. We define a RDF vocabulary to describe spatiotemporal data sources. A preliminary proposal of this vocabulary was described as ongoing work in Ferreira et al (2014).

RDF (Resource Description Framework) is a data model for describing and connecting resources and SPARQL is a query language for RDF data sets. Both are World Wide Web Consortium (W3C) standards and are key techniques in Linked Data

and Semantic Web (Berners-Lee et al, 2001). RDF describes resources using the concepts of classes, properties, and values. The term vocabulary refers to a set of classes and properties that are defined specifically for a certain application.

1.1. Related work

In recent years, traditional GIS technologies have being extended to deal with spatiotemporal data. Gebbert and Pebesma (2014) present a field based temporal GIS, called TGRASS, based on the open source Geographic Resources Analysis Support System (GRASS). This system works with *fields* or *coverages*, following the definition of Galton (2004): *a spatial field is a mapping from spatial locations to values that may be any kind of data structures*.

Time Manager[1] is a plugin of the open source system QGIS (Quantum GIS). This plugin provides tools, such as a time slider, that allow users to animate vector and raster layers based on time attributes. It focuses on visualization of temporal layers, creating animations directly in the map window and exporting image series. Tracking Analyst is a module of the commercial ArcGIS software (ESRI, 2010). Using this module, users can add temporal data to ArcGIS, visualize it dynamically and handle it as trajectories of objects.

Peuquet and Hardisty (2010) present a web-based tool, called STempo, for visualizing and analyzing space-time event data sets and for discovering patterns from these sets. STEMgis[2] is a commercial temporal GIS to dynamically visualize and explore spatial data throughout the four dimensions of space and time.

Following this trend, we are extending the TerraLib GIS library and TerraView GIS software (Camara et al, 2008) to deal with spatiotemporal data. Differently from the previous systems cited in this section, this extension does not focus on a specific type of spatiotemporal data. TGRASS works with fields or coverages, Tracking Analyst with trajectories of objects and STempo with space-time events. Our goal is to develop a more comprehensive temporal GIS able to work with fields, trajectories as well as events. Many applications need temporal GIS that can integrate different types of spatiotemporal data.

2. Spatiotemporal data and applications

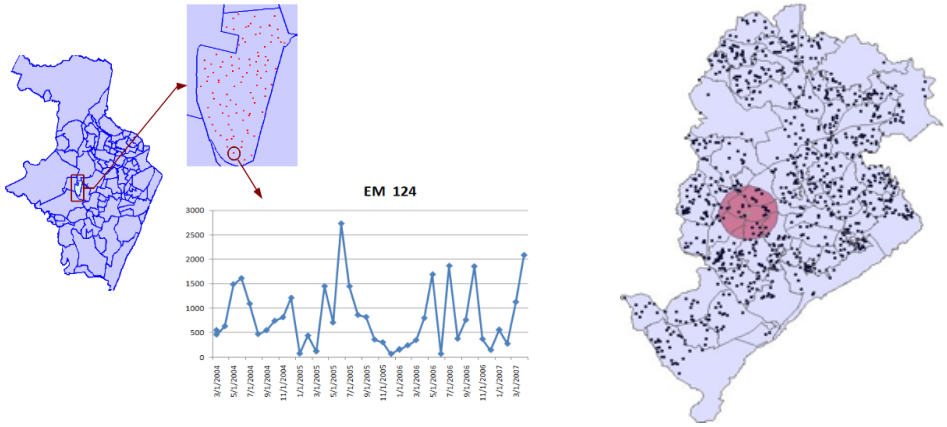
This section aims at showing the diversity of spatiotemporal data from different application domains. We present real examples of data sets from public health, location-based services, environmental and natural disaster monitoring applications.

Figure 1 (a) shows time series used in disease surveillance of dengue in the city of Recife, Brazil (Regis et al, 2009). Dengue is a viral disease transmitted by the *Aedes aegypti* mosquitoes. These mosquitoes lay their eggs in standing water; the eggs hatch in hot weather. To assess dengue risk, health services use buckets of water as egg traps. Each trap has a fixed location represented as a red point in the picture. The time series represents the number of mosquito eggs gathered weekly from an egg trap in a district

1 Available at: http://anitagraser.com/projects/time-manager/

2 Available at: http://www.discoverysoftware.co.uk/STEMgis.htm

of Recife. Figure 1 (b) presents occurrences of meningitis in Belo Horizonte city. Each event has a spatial location (black points in the picture) and a time of occurrence.



(a) (b)

Figure 1. (a) Number of mosquito eggs gathered from an egg trap in Recife, Brazil; and (b) Occurrences of meningitis in Belo Horizonte city.

Figure 2 presents routes of eight sea elephants in Antarctica. These animals were monitored by a project called MEOP - "Marine Mammal Exploring the Oceans Pole to Pole" (http://www.inpe.br/crs/pan/pesquisas/telemetria.php). The project monitored the trajectories of these animals during 3 years, shown as red lines in the figure.

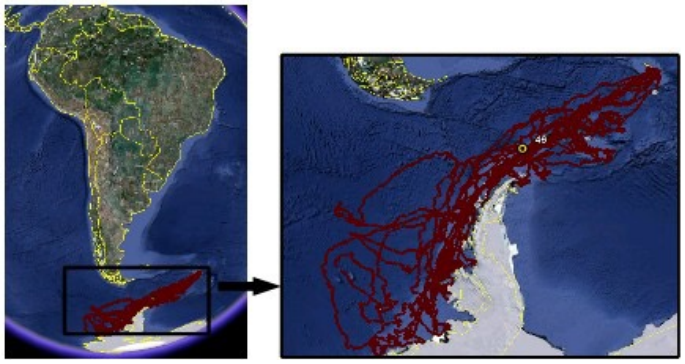


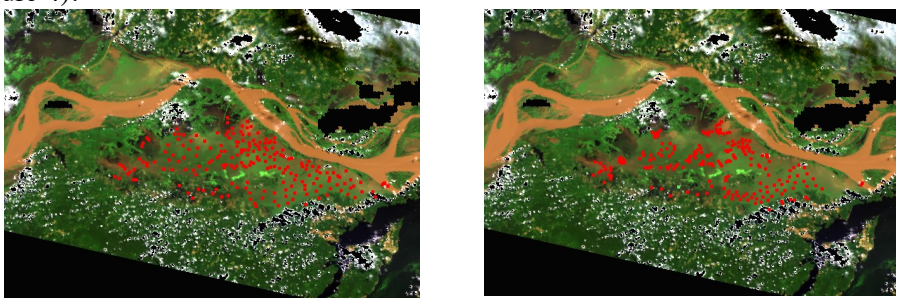
Figure 2. Trajectories of eight sea elephants in Antarctica (red lines).

Figure 3 shows a set of observations (red points) gathered in a lake of Amazon rainforest in two different months. Each observation measures the chlorophyll value, among other properties, at a specific location and time. These observations are taken monthly to analyze the variation of chlorophyll within the lake over time. Usually, a kriging interpolation function is used to estimate values at non-observed locations.

Figure 4 shows two grids; each one associated to a time. These grids contain the rain variation in the state of Rio de Janeiro during the natural disaster of 11 January 2011. Each cell contains an estimated value of precipitation, in millimeter per hour (mm/h). These grids are taken in 15-minute intervals.

To meet demands from different application, a temporal GIS has to deal with distinct types of spatiotemporal data in an integrated way. This includes: (1) time series associated to fixed spatial locations (Figure 1(a)); (2) events or occurrences that happen in a certain time and space (Figure 1(b)); (3) trajectories of moving objects (Figure 2);

(4) fields or coverages based on measures associated to spatial locations and times and on interpolation functions (Figure 3); and (5) sequence of raster or grid data over time (Figure 4).



September, 2003 November, 2003

Figure 3. Observations of chlorophyll in a lake of Amazon rainforest.

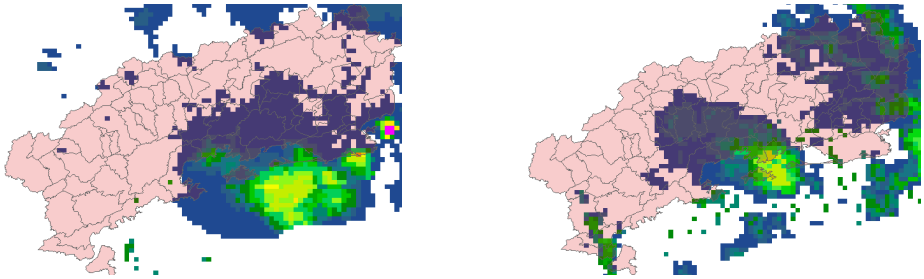


Figure 4. Precipitation grids in the state of Rio de Janeiro, Brazil, in 11 January 2011.

Each type of spatiotemporal data requires a specific set of operations. For example, we can use an interpolation function associated to Amazon lake observations in order to estimate chlorophyll values in non-measured locations and times. However, this operation does not make sense for meningitis occurrences. Besides that, the algorithm to find spatiotemporal clusters for meningitis occurrences is different from the one for trajectories. The first algorithm is suitable for independent events (Veloso, 2013) while the second has to taken into account the association between trajectories and objects.

Moreover, a temporal GIS must be able to access different kinds of spatiotemporal data sources. For instance, the egg trap time series (Figure 1(a)) are stored in a PostGIS database; the meningitis occurrences (Figure 1(b)) are available through Web Feature Service (WFS); the sea elephants trajectories (Figure 2) are stored in a Keyhole Markup Language (KML) file; the Amazon lake observations (Figure 3) are available as a set of Shapefiles and each file is associated to a specific time; and the precipitation grids (Figure 4) are stored as Geotiff files and each file is associated to a specific time.

3. Temporal GIS development

To organize the discussion about challenges in temporal GIS development, we consider a general architecture composed of four modules, as shown in Figure 5. The *Data Model* module is the core of the system. It defines the key concepts of the system that reflect on all other modules. This module contains data types, relationships, operations and rules to represent these concepts. *Data Access* is a module responsible for accessing

and querying data sets from different kinds of sources and for mapping these data sets into the concepts and types of the *Data Model*.

Data Processing and Analysis module contains methods and algorithms to process and analyze data. In most systems, users can directly interact with this module through script languages, such as Python (www.phython.org) and LUA (www.lua.org). Script languages allow users to express complex processing. *Data Presentation* module contains all elements associated to data visualization, including the system Graphical User Interfaces (GUI). It is responsible for data presentation to the user. In this paper, we focus on the two first modules, *Data Model* and *Data Access*.

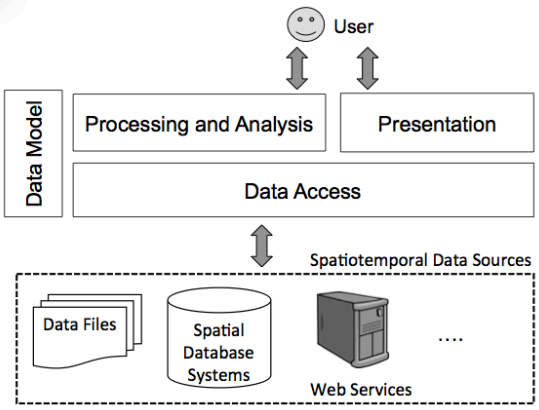


Figure 5. GIS general architecture.

3.1. Temporal GIS: Data Model

The first challenge faced in temporal GIS implementation is: *how to model spatiotemporal data*. We need a data model that defines a minimal set of data types able to represent different kinds of spatiotemporal information from distinct application domains.

In GIScience, static geospatial information is represented following well-established models and concepts. This includes the dichotomy between *object-based* and *field-based* models (Galton, 2004). Examples of long-standing concepts are vector and raster data structures, topological operators, spatial indexing, and spatial joins (Rigaux et al, 2002). Most existing GIS and spatial database systems, such as PostGIS and Oracle Spatial, are grounded on these concepts. However, there is no consensus on how to represent spatiotemporal information in computational systems.

Many existing proposals of spatiotemporal data models focus on representing the evolution of *objects* and *fields* over time. Pelekis et al. (2004) review some of these models and consider that most of them are data-specific; each one addresses a class of spatiotemporal data. Some proposals are specific for discrete changes in objects (Worboys, 1994) (Hornsby and Egenhofer, 2000), others for moving objects (Güting and Schneider, 2005) (ISO, 2008) and still others for fields or coverage (Liu et al, 2008) (OGC, 2006). However, many applications need to combine different classes of such data. For example, environmental change and natural disaster monitoring have to deal with moving objects as well as with fields. Thus, we need spatiotemporal data models as generic as possible to support such applications.

To properly capture changes in the world, representing evolution of objects and fields over time is not enough. We also need to represent events and relationships between events and objects explicitly (Worboys, 2005). Events are *occurrents* (Galton and Mizoguchi, 2009). They are individual happenings with definite beginnings and ends. The demand for models that describe events has encouraged recent research on spatiotemporal data modeling (Worboys, 2005) (Galton and Mizoguchi, 2009).

3.2. Our proposal: An Observation-Based Model for Spatiotemporal Data

We proposed a data model for spatiotemporal data and specified it using an algebraic formalism. Details about this data model and its algebra are in (Ferreira et al, 2014). Algebras describe data types and their operations in a formal way, independently of programming languages. The proposed algebra is extensible, defining data types as building blocks for other types, as shown in Figure 6.

Observations are our means to assess spatiotemporal phenomena in the real world. Recent research draws attention to the importance of using observations as a basis for designing geospatial applications (Kuhn, 2009). The proposed model takes observations as basic units for spatiotemporal data representation and allows users to create different views on the same observation set, meeting application needs.

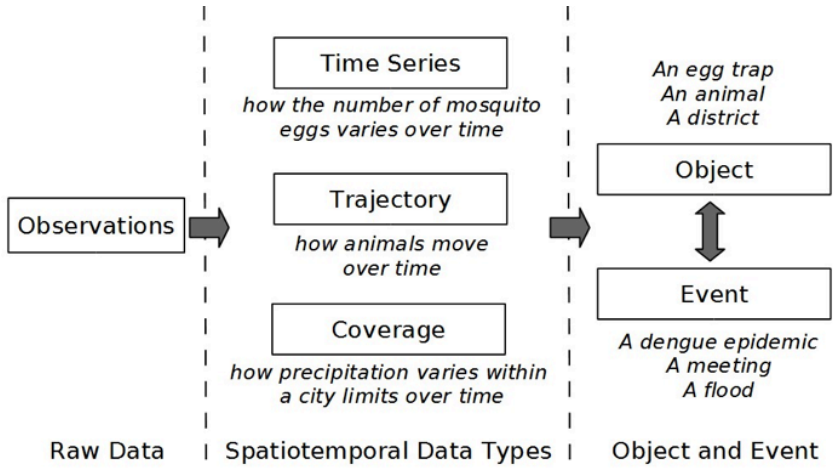


Figure 6. An Algebra for Spatiotemporal Data: From Observations To Events. Source: (Ferreira et al, 2014)

We define three spatiotemporal data types as abstractions built on *observations*: *time series*, *trajectory*, and *coverage*. A *time series* represents the variation of a property over time in a fixed location. A *trajectory* represents how locations or boundaries of an object change over time. A *coverage* represents the variation of a property in a spatial extent at a time. We also define an auxiliary type called *coverage series* that represents a time-ordered set of coverages that have the same boundary. Using these types, we can represent objects and fields that change over time as well as *events*.

We implemented all data types and operations of the proposed algebra in a new module called "TerraLib ST" in the TerraLib library. Using this module and its data types, we can represent and combine different kinds of spatiotemporal data sets from distinct application domains, such as the ones presented in section 2.

3.3. Temporal GIS: Data Access

The second challenge faced in temporal GIS implementation is: *how to access and integrate spatiotemporal data sets from different kinds of data sources*. There are not standards on how to store spatiotemporal data in spatial database systems or files as well as on how to serve such data through web services.

Since the beginning of the 2000s, the GIS community has made a serious effort towards spatial data interoperability. The International Organization for Standardization (ISO) and the Open Geospatial Consortium (OGC) have proposed standards3 to represent and store spatial information in data files and database systems as well as to serve spatial data via web services. Geography Markup Language (GML) and KML are examples of data formats proposed by OGC for spatial data interchange. Spatial extensions of traditional object-relational Database Management Systems (Spatial DBMS), such as PostGIS and Oracle Spatial, deal with vector spatial information in compliance with the OGC Simple Feature Access (SFA) specification. Regarding web services, there are standards for serving spatial data, metadata and processes, such as Web Feature Service (WFS), Web Coverage Service (WCS), Catalogue Service Web (CSW) and Web Processing Service (WPS).

The compliance with ISO and OGC standards has assured a high degree of spatial data interoperability. Many GIS tools and libraries are able to access spatial data files, databases and web services that follow these specifications. Standards are useful to promote spatial data interoperability. However, few results have been achieved regarding spatiotemporal data interoperability. Most OGC and ISO standards are related to spatial but not spatiotemporal data.

Regarding spatiotemporal data, OGC proposes a standard called Sensor Observation Service (SOS) that defines a web service interface for disseminating and querying spatiotemporal observations, sensor metadata and observed features, based on the OGC Observations and Measurements (O&M) specification (OGC, 2010). However, many data providers store and disseminate spatiotemporal information using other formats and standards, not only SOS. GIS tools must be able to access different types of spatiotemporal data sources, without forcing the use of a specific format or standard.

3.4. Our proposal: A RDF vocabulary for spatiotemporal data sources

We consider that data sources store and provide spatiotemporal observations, which are basic units for spatiotemporal data representation. A temporal GIS must access these observations from data sources and allow users to create different views on them, according to application needs.

We propose an approach to access spatiotemporal observations from different kinds of data sources using Semantic Web techniques. The central idea is to use RDF files for describing *how* spatiotemporal observations are stored and provided by data sources and SPARQL language for discovering information about these data sources. We use RDF as linked metadata files, that is, files that describe *how* data sources represent spatiotemporal observations and *links* among these data sources. We do not

[3] http://www.opengeospatial.org/standards/is

transform the spatiotemporal observations from their original data sources and formats into RDF files. Each data source has an associated RDF file and all RDF files are based on the same vocabulary, as presented in Figure 7.

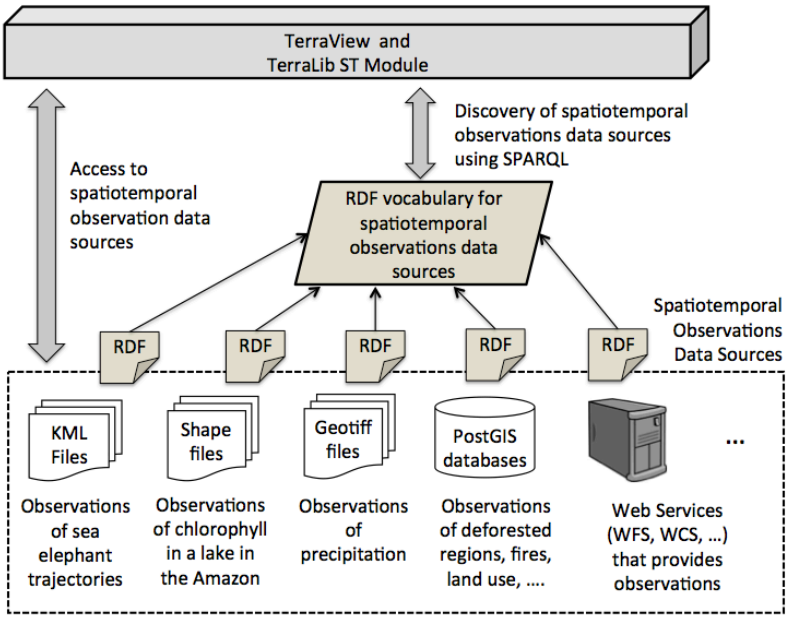


Figure 7. A proposal to access different kinds of spatiotemporal data sources.

There are many RDF vocabularies for different application domains, such as Dublin Core[4] for describing documents and FOAF[5] (Friend Of A Friend) for describing relationships among people. Examples of RDF vocabularies for describing geospatial data are W3C Basic Geo vocabulary, GeoOWL ontology, NeoGeo Vocabulary and GeoSPARQL (Battle and Kolas, 2012). GeoSPARQL is an OGC standard that defines a vocabulary for representing geospatial data in RDF and an extension to the SPARQL query language for processing geospatial data.

We define a RDF vocabulary for describing spatiotemporal observation data sources, based on OGC O&M specification, OGC GeoSPARQL vocabulary for spatial data and ISO standards for time representation (ISO, 2002) (ISO, 2004).

4. A RDF vocabulary for spatiotemporal data sources

The proposed vocabulary is described using OWL (Ontology Web Language). In this paper, we use a UML class diagram to better represent the concepts of the vocabulary, as shown in Figure 8.

A data source can have one or more spatiotemporal observation data sets. The class `STODataSourceInfo` contains information about a data source (class `DataSourceInfo`) and its spatiotemporal observation data sets (class `STODataSetInfo`). There are three types of data sources: files (class `FileDataSourceInfo`), DBMS (class `DBMSDataSourceInfo`) and web

[4] http://dublincore.org/
[5] http://xmlns.com/foaf/spec/

services (class `WSDataSourceInfo`). Each type of data source is described by a specific set of attributes. For example, to describe a DBMS data source, we have to inform its host name (`host`), port number (`port`), database name (`db_name`), a user (`user`) and its password (`password`). To describe a data source composed of files, we just need the path (`path`) of the folder where the files are. In this first version, the vocabulary accepts three types of files (shapefile, Geotiff and KML), one type of DBMS (PostGIS) and one type of web services (WFS). They are described in the enumeration `DataSourceType`.

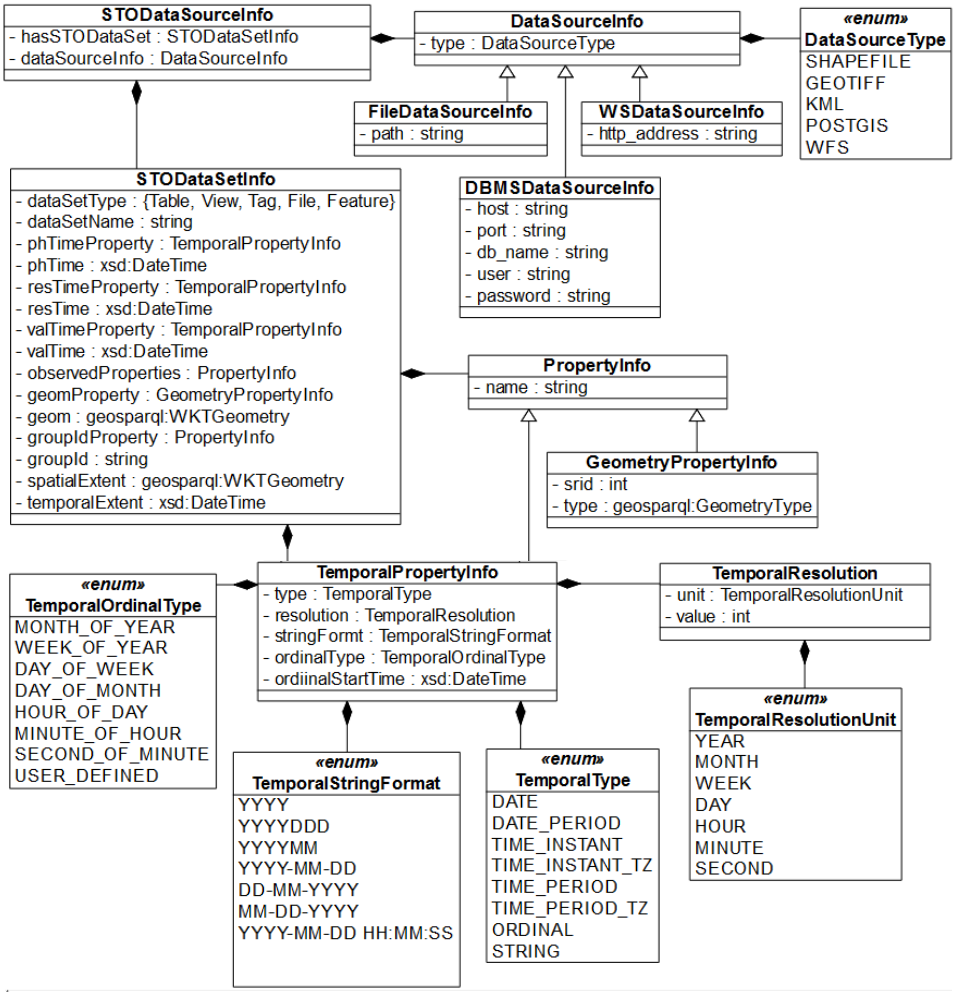


Figure 10. The RDF vocabulary for spatiotemporal observation data sources.

The class `STODataSetInfo` has information about data sets that contain spatiotemporal observations. Its two first attributes are the data set type (`dataSetType`) and name (`dataSetName`). In DBMS data sources, a data set type can be a table (`Table`) or a view (`View`); in files data sources, it can be an internal file tag (`Tag`) or a file (`File`); and in web services data sources, it is a Feature (`Feature`). If data set type is `Table`, the data set name is the table name; if type is `View`, it is the view name; if type is `Tag`, it is the tag name, and so on.

According to O&M specification (OGC, 2010), an observation has three temporal properties: *phenomenon*, *valid* and *result* time. The *phenomenon time* represents the time, instant or period, when the observation is actually measured. The *valid* time describes the time period during which the observation is available to be used. The *result* time represents the instant when the observation becomes available, typically when the observed values must be processed before being used. In the vocabulary, the phenomenon, valid and result time can be stored in data set properties (`phTimeProperty`, `valTimeProperty` and `resTimeProperty`) or be informed by the user (`phTime`, `valTime` and `resTime`).

To describe information about data set properties that store time, we define the class `TemporalPropertyInfo`. A temporal property has:

- A name (`name` of the super class `PropertyInfo`).
- A type (`type`): the types are defined in the enumeration `TemporalType` based on the ISO 19108:2002 standard (ISO, 2002). A temporal property can store instants (`DATE`, `TIME_INSTANT`, `TIME_INSTANT_TZ`) or periods (`DATE_PERIOD`, `TIME_PERIOD`, `TIME_PERIOD_TZ`) of time. Besides that, it can store times as texts (`STRING`) or as integers that indicate ordinal times (`ORDINAL`).
- A temporal resolution (`resolution` attribute represented by the class `TemporalResolution`): it indicates the time granularity that must be considered to deal with the temporal property. For example, the precipitation grids, shown in Figure 4, are taken at each 15 minutes. So, its temporal resolution unit is `MINUTE` and value is 15.
- A string format (`stringFormat`): this is necessary when the temporal property uses a textual representation of time, that is, its type is `STRING`. In this case, we need to inform what format the string is. For example, the text "01-03-2008" is ambiguous; it can represent the first day of March in 2008 or the third day of January in 2008. So, we have to inform what format it follows in order to understand its right meaning. ISO 8601:2004 (ISO, 2004) proposes some date and time format representations, such as DD-MM-YYYY or MM-DD-YYYY, and we define the enumeration `TemporalStringFormat` based on them.
- An ordinal type (`ordinalType`): this information is necessary when the temporal property stores ordinal times, that is, its type is `ORDINAL`. ISO 8601:2004 (ISO, 2004) defines some ordinal times and we define the enumeration `TemporalOrdinalType` based on them. For example, the ordinal day numbers in the week (`DAY_OF_WEEK`) mean: 1 is Monday, 2 is Tuesday, 3 is Wednesday, and so on.
- An ordinal start time (`ordinalStartTime`): this information is only necessary when the temporal property type is `ORDINAL` and the ordinal type is `USER_DEFINED`. In this case, the temporal property has user-defined ordinal numbers and so the user must inform what date and time is related to the first ordinal number.

Besides phenomenon, valid and result times, a spatiotemporal observation data set must have one or more properties that store the observed values. Users must indicate these properties through the attribute `observedProperties` in the class `STODataSetInfo`. Spatiotemporal observations are associated to spatial locations.

We use geometry types, such as point and polygon, to represent such locations. The geometries associated to observations can be stored in a data set property (`geomProperty`) or be informed by the user (`geom`). We define the class `GeometryPropertyInfo` to contain information about the property that stores geometries: its name (`name` of the super class `PropertyInfo`), its spatial reference system identifier (`srid`) and its type (`type`) whose values come from the OGC GeoSPARQL vocabulary. We also use the `WKTGeometry` type of the GeoSPARQL vocabulary to represent geometry (`geom`).

A spatiotemporal observation data set can have a property that stores identifiers that are used to group observations. Users can indicate this property through the attribute `groupIdProperty` in the class `STODataSetInfo`. Otherwise, users can give an identifier associated to all observations (`groupId`). Finally, users can inform the spatial (`spatialExtent`) and temporal (`temporalExtent`) extents of all observations. If users do not inform these extents, they are calculated taking into account all observations.

4.1. Example

To illustrate the use of the RDF vocabulary to describe spatiotemporal observation data sources, lets take the time series shown in Figure 1 (a). All measures of the mosquito egg traps are stored in a table called "`measures`" of a PostGIS database called "`dengue_surveillance`". Each row of this table is an observation in a certain time associated to an egg trap. Figure 11 shows a subset of this table and a simplified version of the RDF file to describe it, based on the vocabulary.

The RDF file has information about the PostGIS database "`dengue_surveillance`" in the tag `<DBMSDataSourceInfo>`, such as its host, port number, database name, user and password. Besides that, this file has information about the data set that contains spatiotemporal observations in the tag `<STODataSetInfo>`. In this example, the data set is the table "`measures`" indicated in tags `<dataSetType>` and `<dataSetName>`. The observation phenomenon times are stored in the table property "`m_date`" whose type is `DATE`. The measures were collected weekly. Thus its temporal resolution unit is WEEK and the value is 1. They are indicated in the tag `<phTimeProperty>`. In this example, there are not result and valid times associated to observations.

The table property "`m_number_eggs`" stores the observed values, that is, the number of mosquito eggs, as informed in tag `<observedProperty>`. The table property "`m_trap_location`" stores the spatial location of the traps and so is indicated in tag `<geomProperty>`. The identifier of each trap is stored in the table property "`m_trap_id`". All observations associated to a trap have the same trap identifier. So, this property can be used to group the observations by trap. We inform this through the tag `<groupIdProperty>`.

As presented in Figure 11, the TerraLib ST Module has two parts: (1) STDataModel that contains all data types and operations to represent spatiotemporal information based on the proposed algebra presented in section 3.2; and (2) STDataLoader that contains methods and functions to interpret RDF files that describe spatiotemporal observation data sources, to access such observations and to map them into the data types of the STDataModel.

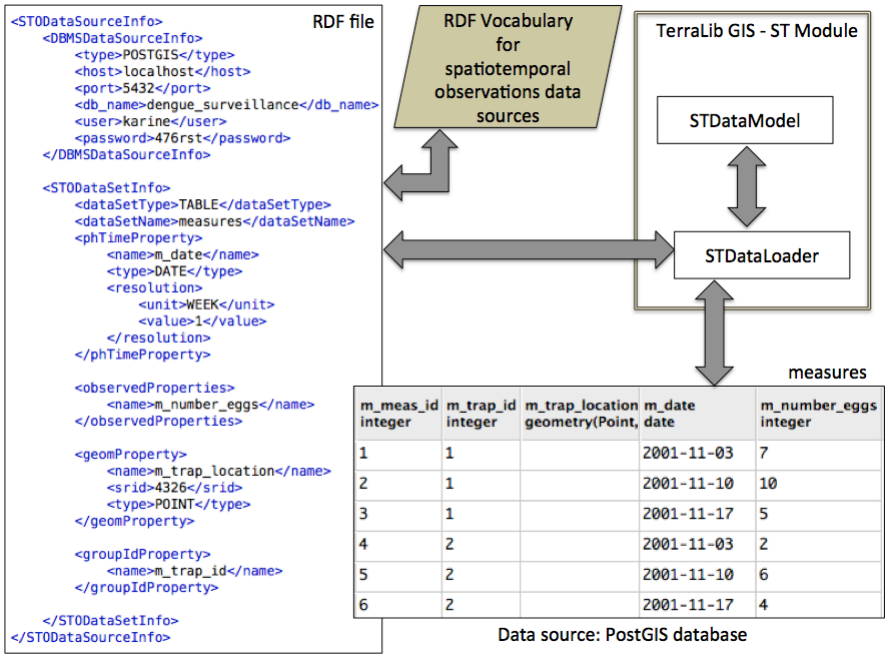


Figure 11. Example of the proposed RDF vocabulary.

5. Final remarks

This paper highlights challenges in temporal GIS development and presents a proposal to access spatiotemporal information from distinct kinds of data sources, using Semantic Web techniques. This approach consists in describing *how* data sources store spatiotemporal observations, based on a RDF vocabulary. Based on this description, temporal GIS are able to properly access and load observations stored in different formats and sources.

We implemented these proposals in the TerraLib GIS library extension to deal with spatiotemporal data; we developed a new module called "TerraLib ST Module". This module can be downloaded at www.dpi.inpe.br/terralib5. Now, we are working on a plugin for TerraView GIS software. This plugin will provide graphical user interfaces (GUI) to allow users to create the RDF files describing data sources, access spatiotemporal observation sets from these sources, handle these sets and dynamically visualize them in TerraView.

References

Battle, R., Kolas, D. (2012) "Enabling the geospatial semantic web with parliament and GeoSPARQL", Semantic Web Journal, v. 3(4).

Berners-Lee, T., Hendler, J., and Lassila, O. (2001) "The semantic web", Scientific american, v. 284(5), p. 28–37.

Camara, G., Vinhas, L., Queiroz, G. R., Ferreira, K. R., Monteiro, A. M. V., Carvalho, M. T. M., Casanova, M. A. (2008) "TerraLib: An open-source GIS library for large-scale environmental and socio-economic applications", Open Source Approaches to Spatial Data Handling. Berlin, Springer.

ESRI (2010) "Tracking Analyst Tutorial", Available at: http://help.arcgis.com/en/arcgisdesktop/10.0/pdf/tracking-analyst-tutorial.pdf

Ferreira, K. R., Almeida, D. B. F. C., Monteiro, A. M. V. (2014) "A RDF Vocabulary for Spatiotemporal Observation Data Sources", In: Proceedings of XV Brazilian Symposium on Geoinformatics (GeoInfo), Campos do Jordão, Brazil.

Ferreira, K. R., Camara, G., Monteiro, A. M. V. (2014) "An algebra for spatiotemporal data: From observations to events", Transactions in GIS, v. 18(2), p. 253–269.

Galton, A. (2004) "Fields and objects in space, time, and space-time", Spatial Cognition and Computation, v. 1, p. 39—68.

Galton, A., Mizoguchi, R. (2009) "The water falls but the waterfall does not fall: New perspectives on objects, processes and events", Applied Ontology, v. 4(2), p. 71–107

Gebbert, S., Pebesma, E. (2014) "A temporal GIS for field based environmental modeling", Environmental Modelling & Software, v. 53, p. 1–12.

Güting, R. H., Schneider, M. (2005) "Moving objects databases", San Francisco, CA: Morgan Kaufmann.

Hornsby, K., Egenhofer, M. (2000) "Identity-based change: A foundation for spatiotemporal knowledge representation", International Journal of Geographical Information Science, v. 14, n. 3, p. 207–224.

International Standard Organization (ISO) (2002) "ISO 19108:2002: Geographic information - Temporal schema", Geneva, Switzerland.

International Standard Organization (ISO) (2004) "ISO 8601:2004: Data elements and interchange formats - Representation of dates and times", Geneva, Switzerland.

International Standard Organization (ISO) (2008) "ISO 19141:2008 - Geographic information - Schema for moving features", Geneva, Switzerland.

Kuhn, W. (2009) "A functional ontology of observation and measurement", In: International Conference on GeoSpatial Semantics. Berlin, Springer LNCS, 5892.

Liu, Y., Goodchild, M. F., Guo, Q., Tian, Y., Wu, L. (2008) "Towards a general field model and its order in GIS", International Journal of Geographical Information Science, v. 22, n. 6, p. 623–643.

Open Geospatial Consortium (OGC) (2006) "OpenGIS abstract specification topic 6: Schema for coverage geometry and functions".

Open Geospatial Consortium (OGC) (2010) "Geographic Information: Observations and Measurements - OGC Abstract Specification Topic 20".

Pelekis, N., Theodoulidis, B., Kopanakis, I., Theodoridis, Y. (2004) "Literature review of spatio-temporal database models" The Knowledge Engineering Review, v. 19

Peuquet, D. J., Hardisty, F. (2010) "STempo: an interactive visualization and statistical environment for discovery and analysis of space-time patterns", Available at: http://www.geovista.psu.edu/stempo/

Regis, L., Souza, W.V., Furtado, A.F., Fonseca, C.D., Silveira, J.C., Ribeiro, P.J., Melo-Santos, M.A.V., Carvalho, M.S., Monteiro, A.M. (2009) "An Entomological surveillance system based on open spatial Information for participative Dengue control", Anais da Academia Brasileira de Ciências, v. 81.

Rigaux, P., Scholl, M., Voisard, A.: Spatial Databases with Application to GIS. Morgan Kaufman, San Francisco, (2002).

Veloso, B., Iabrudi, A., Correa T. (2013) "Towards efficient prospective detection of multiple spatio-temporal clusters", In: Proceedings of XIV Brazilian Symposium on Geoinformatics (GeoInfo), Campos do Jordão, Brazil.

Worboys, M. F. (1994) "A Unified Model for Spatial and Temporal Information", The Computer Journal, v. 37.

Worboys, M. (2005) "Event-oriented approaches to geographic phenomena", International Journal of Geographical Information Science, v. 19, p. 1–28.

Yuan, M. (2009) "Challenges and Critical Issues for Temporal GIS Research and Technologies", In: Handbook of Research on Geoinformatics, IGI Global, Hershey, p. 144–153.

Geocoding of traffic-related events from Twitter

Juan C. Salazar C, Miguel Torres-Ruiz, Clodoveu A Davis Jr., Marco Moreno-Ibarra

Instituto Politécnico Nacional, Centro de Investigación en Computación
UPALM-Zacatenco, 07738 – Mexico City – Mexico

b130126@sagitario.cic.ipn.mx, mtorres@cic.ipn.mx,
clodoveu@dcc.ufmg.br, marcomoreno@cic.ipn.mx

Abstract. *Nowadays social networks provide information with high correlation with events that are occurring in the worldwide. Twitter is a microblogging network of real time posts in which people know different classes of events such as concerts, festivals, demonstrations, etc. Other relevant topic is traffic congestions; user-generated content is useful to assist drivers in avoiding crowded areas. Therefore, this work is oriented towards following specific steps focused on improving the geocoding of traffic-related events that are associated with a number of geographic elements. Preliminary results have increased the precision and recall of locating geographic elements, achieving 85% and 83%, from a baseline of 36% and 30% respectively.*

1. Introduction

Detection of road traffic congestions is a significant problem to be solved for large cities. Urban mobility can be improved if bad conditions and accidents that occur during the day are known. Thus, the possibility to avoid congested crossroads, street segments with heavy traffic, demonstrations, lane interruptions and so on, is highly desirable.

However, it is difficult to gather traffic-related information of all places in cities throughout the day, and to carry out this task with tracking devices could probably take a long time. Therefore, to find other sources of information and know what happens in the city, social networks like Twitter are very useful. Twitter has become quite popular in different countries. The ease of sharing content through this social network encourages users to post a great amount of information regarding events that are happening in the real world [Lee et al. 2013]. Information related with traffic conditions is also very common on Twitter. When people are moving around the city, users post information about traffic-related events using their devices. As a matter of fact, there are several accounts that post exclusively information about traffic-related events. Some of these accounts are from government

agencies, official profiles of public transportation, accounts managed by radio stations and independent users. On the other hand, information in Twitter rarely includes the coordinates of where the accident happened. Usually, only few tweets contain latitude and longitude, and sometimes the location at which people tweet is not the same place where the event occurred. Tweets without coordinates are frequently ambiguous, with abbreviations, nicknames and misspellings. Furthermore, tweets are limited to 140 characters. For that reason, geocoding tweets and the relationships between them are very important challenges in particular research areas.

In this paper, we propose an approach to geocode traffic-related events from Twitter, improving the accuracy of representations by means of the number of geographic elements involved. We present a case study based on data from Mexico City. Preliminary results show a precision and recall rates of 85% and 83% respectively.

The remainder of the paper follows. Section 2 presents related work. The methodology to identify and locate geographic elements is outlined in Section 3. A description of the tweet dataset is presented in Section 4. Section 5 describes experimental results and a discussion related to the analysis. Finally, conclusions and future work are described in Section 6.

2. Related work

There are various methodologies for retrieving, processing and displaying geographic information from the web. The following approaches identify different geographic components from text. The Traffic Observatory [Ribeiro et al. 2012] proposes geocoding tweets using a gazetteer called GEODICT. This dictionary contains a collection of thoroughfare segments, street crossings, abbreviations, nicknames, neighborhoods and landmarks, along with their geographic representations. The Traffic Observatory uses exact and approximate string matching functions on gazetteer data to geolocate the streets mentioned in Twitter's stream.

Delboni et al. [2007] proposed a method to retrieve information from the web, by using natural language processing techniques, thereby recognizing positioning expressions formed by landmarks and spatial relations. Davis Jr. et al. (2011) proposed a methodology based on user relationships to infer the location of messages in Twitter. A network is created taking into account the follower-following relationships. Starting from known locations of users in the network, it infers the location of others. Working with Facebook data, Backstrom et al. (2010) showed that there is a strong connection between social relationships and geography. People that interact daily almost always live near each other, and each user has at least 10 friends with shared locations. With these assumptions, the methodology infers the most probable user location.

3. Geocoding traffic-related events from tweets

The proposed methodology involves automatic and manual steps in order to search geographic elements in Twitter (streets, neighborhoods, public transportation stations, places, and others) using a gazetteer. The initial gazetteer was obtained from GeoNames, the dataset is composed of 36,236 different streets (more than 150,000 street segments) from the Mexico City. They are in upper case, and they contain accent marks (Spanish language). They have blank rows or names with default values (*e.g.* geometries without name or with 'NO NAME' assigned) and many of them have abbreviations, such 'as', 'ave', 'st', 'rd', etc. The geocoding process consists of the following steps: (1) collect information from the tweet dataset, (2) create dictionaries and equivalent road axis names, (3) divide the gazetteer, (4) standardize, and (5) identify and locate traffic-related events. Each of these steps will be covered next.

3.1. Information from the tweet dataset

From the tweet dataset, it is important to go deeper on what people are talking about. Many tweets talk about traffic-related events, since that is the purpose of the selected accounts, but we found mentions to popular streets, common nicknames, common abbreviations, popular places and popular historical monuments. According to these findings, a script for determining the most common words in the tweet dataset was developed. The script finds the most common N-grams. A N-gram is a N-word slice of a longer statement [Cavnar et al. 1994]. From each tweet the n-grams are obtained, and the number of occurrences orders them. The most repetitive n-grams are selected (the threshold established was more than 100 mentions). Even though a N-gram includes the notion of any combination of characters or words in a sequence, in this case the script only considers contiguous slices of N-words. As an illustration, Figure 1 shows how the script works to get N-grams:

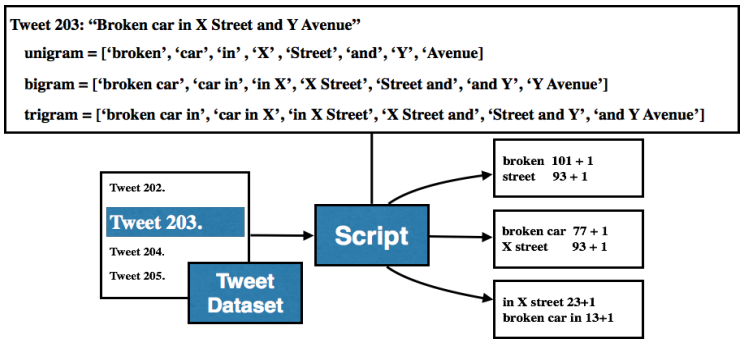


Figure 1. Creation of the lists of most frequent n-grams by the script.

From the most frequent unigram, bigram and trigram lists, we have identified by hand 456 common streets, 150 common traffic-related events, 135 common hashtags, 69 common nicknames, 65 common buildings, places and monuments, 34 common abbreviations and 26 common combinations of prepositions. Figure 2 represents all the collected information from the tweet dataset.

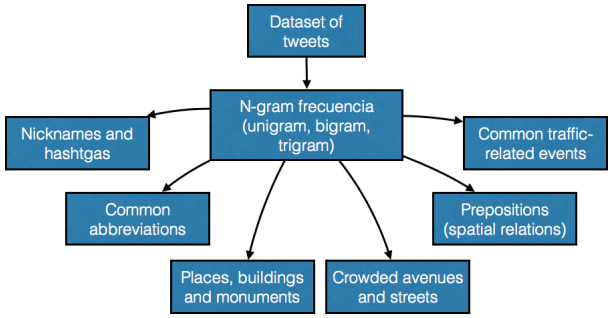


Figure 2. The collected information from the n-gram analysis.

This information was divided from the results of the frequency n-gram script. Each group of selected elements was saved in a CSV file, in order to facilitate further processing.

3.2. Dictionaries of geographic elements and equivalent road axis names

From the results obtained in the previous step, data from Open Street Map and from the National Institute of Statistics and Geography (INEGI for its acronym in Spanish) were used to generate some dictionaries meant to enrich the gazetteer (see Figure 3). The proposed dictionaries are the following: dictionary of abbreviations, dictionary of nicknames, dictionary of hashtags, dictionary of traffic-related events, dictionary of public transportation, dictionary of principal streets (only streets that appear in tweets and exist in the gazetteer), dictionary of places, buildings and monuments, and dictionary of neighborhoods. The last four dictionaries have a geographical component, which is used to spatially map the geographic features. We call them *dictionaries of geographic elements*. A dictionary of spatial relations could have been created, but we are classifying traffic-related events by using the number of geographic elements identified in tweets.

About the equivalent road axis names, it is frequent that streets can be named for more than one official name. Mexico City has 31 road axes and 2 circuits that cover more than 10 thousand kilometers of length. Axes and circuits change their names along their way when crossed with other streets. For that reason, sometimes people call them by the principal name, by its name in a certain segment (second name) or name them together (principal name + second name). However, all these options are valid, thus all alternatives must be

searched in the tweets. With the purpose of solving this issue, a dictionary of equivalent road axis names is added to the named collection.

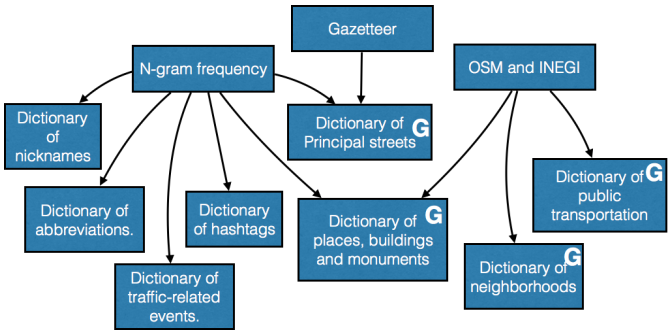


Figure 3. Creation of geographic and non-geographic dictionaries.

3.3. Division of the gazetteer

There is frequently a small group of streets that concentrate traffic-related events. Based on the N-gram frequencies, we found that only 19% of the streets in the complete gazetteer appear in tweets. In consequence, the gazetteer is split in two parts, the first part formed by the frequently named streets in tweets that exist in the gazetteer (some streets posted in tweets are outside of Mexico City, and are not considered) and the second part is composed of the remaining streets. Although the reduction does not improve the precision and recall of geocoding, the performance of the identification and location steps have increased.

3.4. Standardization

In order to improve the standardization process, dictionaries of non-geographic elements (dictionary of abbreviations, dictionary of nicknames and dictionary of hashtags) are used. In our gazetteer, the street names contained in geographic dictionaries include abbreviations, names in uppercase, names with accent marks, and even blank rows or default values. Hence, this process changes each street name to lowercase and removes accent marks (e.g. TALISMÁN ST. - talisman st.). Moreover, using the dictionary of common abbreviations, they are replaced with the complete word (talisman st. - talisman street). Finally, blank spaces and streets with default values are deleted. Other problems detected in tweets are links and mentions to other accounts, nicknames, misspellings and hashtags (e.g. http://t.co/hAN0K0WS, @OVIALCDMX, 'The angel', 'circuito interior street', #insurgentesavenue). So, in order to solve these new issues, the dictionaries of nicknames and hashtags are used to replace them with the official name in tweets (e.g. 'The angel' - 'angel of independence' and #insurgentesavenue - 'insurgentes avenue'). Links and mentions to other accounts are deleted and misspellings are not solved yet, and left for future work (using fuzzy matching or a frequent misspellings dictionary).

In addition, in both tweets and gazetteer, stop words have to be filtered out. Stop words are words that do not add meaning to the statement and they are the most common words in a language (e.g. articles, pronouns and prepositions). There is not a universal list of stop words used in natural language processing. For this case, the list of stop words has been defined by the Natural Language Toolkit Library [Bird, 2006].

3.5. Identification and location of traffic-related events

The identification of geographic elements in tweets has been carried out using all the previously described dictionaries of geographic elements: frequently named streets, uncommonly named streets (only if a considerable number of geographic elements was not found), neighborhoods, public transportation, places, buildings and monuments.

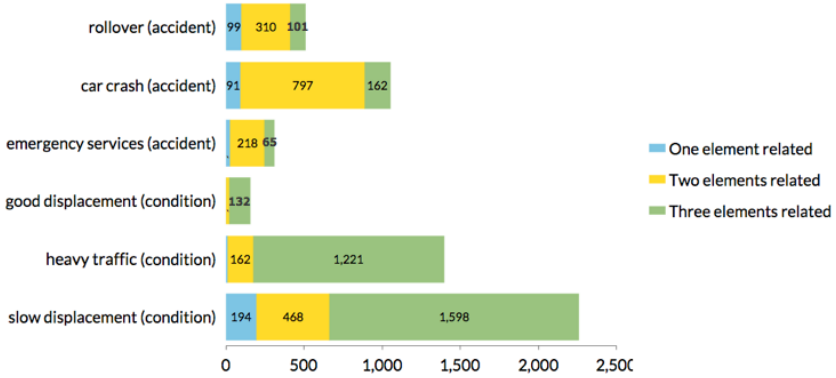


Figure 4. This bar chart shows examples of common traffic-related events that appear on tweets and the number of geographic elements detected.

Traffic-related tweets from the selected accounts frequently talk about accidents, bad or good traffic conditions. For example, traffic-related events such as accidents are described as '**car crash**', '**rollover**', '**emergency services**', bad conditions are '**slow displacement**', '**road closure**', '**settlements**', etc., and good conditions are '**still moving**', '**good displacement**' and so on. Although there is a considerable amount of advertising, questions and safety recommendations, they are easily filtered out because of the absence of any geographical element and due the short length of tweets. Based on the tweet dataset of and the N-gram frequency analysis, an accident is considered an event that happens in a certain (point) space with one or two geographic elements involved. Examples include, '**a car crash at x street and z street**', '**broken traffic signal at the intersection of x street and y street**', '**rollover in front of x subway station**', etc. The description of a bad or good condition is considered as the actual situation of a street segment. In such descriptions, commonly one, two or three geographic elements are included. For example, '**settlements on x street between y street and z street**', '**good displacement on x street from y street**

to z neighborhood', '**heavy traffic in x street on z neighborhood**', '**raining over x neighborhood**', and so on. A fragment of the lists of accidents and conditions identified in the approach are presented in Table 1. Since tweets can only contain 140 characters, it is difficult to post a mention, a link, a traffic-related event and more than three geographic elements. Therefore, we identified that the number of geographic elements included in the tweet has a strong relation with the kind of traffic-related event (see Figure 4).

Table 1. Common accidents and conditions mentioned on Twitter.

Accident	Frequency	Condition	Frequency
emergency service	378	blocked road	4377
rollover	612	still close	1053
accident	1162	heavy traffic	1423
flooding	432	slow displacement	2779
car crash	1312	road work	1225
emergency in place	508	road close	2521
broken car	1002	traffic jam	1423
Vehicular congestion	570	bumper to bumper	2246
traffic signals out of service	241	gridlock	1101

Each dictionary of geographic elements has a primitive geographic representation (point, line, polygon). Thus, the dictionary of public transportation, which is depicted by a set of points, the dictionary of streets is characterized by a set of lines and the dictionaries of neighborhoods, places, buildings and monuments are represented by polygons. Therefore, as a result of searching geographic elements from dictionaries in tweets, we obtained a collection of geographic primitive elements (see Figure 5).

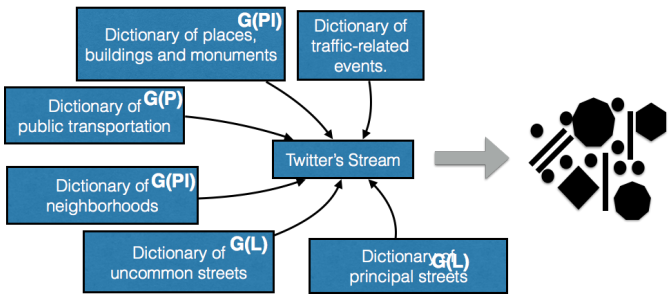


Figure 5. Result of identification process on Twitter's stream.

Assuming that there can be 1, 2 or 3 references to places in a tweet, the number of possible relations that can happen among them follows the formula for combinations with replacement (see Equation 1).

$$CR_m^n = \left(\begin{array}{c} m+n-1 \\ n \end{array} \right) = \frac{(m+n-1)!}{n!(m-1)!}, \qquad \textbf{Equation 1.}$$

Where **m** is the number of possible elements to select, in this case point, line or polygon, and **n** is the number of elements found. Therefore, for 1 element we have: [(point), (line), (polygon)], 2 elements identified: [(point, point), (point, line), (point, polygon), (line, line), (line, polygon), (polygon, polygon)], and for 3 elements identified: [(point, point, point), (point, point, line), (point, point, polygon), (point, line, line), (point, line, polygon), (point, polygon, polygon), (line, line, line), (line, line, polygon), (line, polygon. polygon), (polygon, polygon, polygon)]. Many of these relationships of the geographic primitives are not accurate with respect to the location or the relation frequency in tweets is not high, thus they can be discarded. For this preliminary research work, the following relationships were considered: [(point, line), (line, line), (point, point, line), (point, line, line), (line, line, line), (line, line, polygon)]. The relations listed above were clearly identified in the tweet dataset, the assumptions of each relationship is described as follows:

- (point) represents an accident in a public transportation station.
- (point, line) represents a condition of a street segment in front of a public transportation station.
- (line, line) represents an accident in a street intersection.
- (point, point, line) represents a condition of a street segment delimited by two public transportation stations.
- (point, line, line) represents a condition of a street segment delimited by another street and a public transportation station.
- (line, line, line) represents a condition of a street segment delimited by two streets.
- (line, line, polygon) represents a street segment delimited by a street and a place, building or historical monument.

Although it is quite probable that there are other situations involving these groups of geographic elements, the situations described above occur more frequently. In order to obtain the result of these assumptions, three spatial operations have been devised:

1. Find the street intersection.
2. Find the closest point of the geographic element (polygon or line).
3. Find the bounding box (or convex hull) of the line segment.

The spatial operations were executed using PostGIS functions such as ST_Intersection, ST_ClosestPoint, ST_Envelope and ST_ConvexHull. The result is a geometric element, where the traffic-related event took place. For example, the process to find the (line, line, line) relation is the following:

- While all the possible combinations of elements are not tested:
  - Is there intersection of A element and B element: (operation 1)
    - Save it.
- Are there two intersections?
  - Yes: Is there an element in common from the two intersections?
    - Yes: Find the bounding box (or convex hull) of the element in common delimited by the two intersections. (operation 3)
    - No: Check (line, line) relation.
  - No: Check (line, line) relation.

4. Description of the tweet dataset

Our tweet dataset contains 64,250 tweets collected over a period of six months, from July 07, 2014 until December 24, 2014, without considering retweets and posts with blank spaces. Tweets are collected from reliable Twitter profiles that correspond to known services and institutions. Such accounts have been selected considering some features: account location, account creation date, number of followers, average number of tweets posted per day, if the account belongs to a government agency and if the account has its own website (see Table 2).

Table 2. Traffic-related Twitter accounts covering Mexico City

Twitter Account	Location	Creation date	Followers	Number of tweets	Belongs to government	Website
SSPDFVIAL	Mexico City	07.14. 2010	369,115	154.65	Yes	sup.df.gob.mx
PolloVial	Mexico City	01.31.2013	667	71.91	No	No website
Trafico889	Mexico City	05.14.2009	137,099	90.54	No	siempre889.com/trafico
Alertux	Mexico City	10.16.2012	179,574	35.59	No	www.alertux.com
072AvialCDMX	Mexico City	10.20.2010	83,535	134.71	Yes	www.agu.df.gob.mx
RedVial	Mexico City	03.09.2010	63,702	44.81	No	rvial.mx

Most tweets that come from @SSPDFVIAL and @072AvialCDMX profiles belong to government agencies. The number of followers of these profiles keeps growing, and they have a specific behavior to explain traffic-related events. Hence, it is easier to geolocate their tweets. @Trafico889 and @RedVial belong to radio stations. They post information about weather and traffic conditions, and show their information in their websites. @Alertux and @PolloVial gather volunteered information and retweet information from other Twitter accounts.

5. Experiments and results

In order to measure the accuracy of this methodology, a test dataset was put together using 652 tweets geocoded by hand. We identified streets, public transportation stations, neighborhoods, places, buildings and monuments. The test dataset was compared with elements identified by our methodology, and we computed precision and recall. The methodology consists of the standardization process (this process includes the non-geographic dictionaries), the equivalent axis names and the dictionaries of geographic elements. At first, we compared with a baseline using only the gazetteer with part of the standardization process (only lowercase). Then, we compared with the baseline plus the full standardization process, and finally with the baseline plus the standardization process plus the equivalent axis names. When the system identifies all the elements of the solution, a hit is considered. When the system identifies at least one element of the solution, a partial hit is taken into consideration. Mistakes were also counted when the system did not find any element of the solution. Thus, precision and recall use true positives, true negatives and false negatives and were computed by applying Equation 2 for both cases respectively.

$$P = \frac{T_P}{T_P + F_P};\ R = \frac{T_P}{T_P + F_N} \qquad \textbf{Equation 2}$$

True positives are the geographic elements that were found by the methodology and belong to the gold standard, true negatives are geographic elements found by the methodology that do not belong to the gold standard and false negatives are geographic elements that belong to the gold standard that were not found. The precision and recall were computed for each tweet and we obtained the average of each test. Results are shown in Table 3.

In Figure 6, the behavior in Twitter and its relation with traffic-related events in the real world is shown. The number of tweets posted at 18, 19 and 20 hours is higher. This is the time of the day with the highest level of participation. Another relevant period is in the morning, around 8AM, with another peak in participation. This behavior corresponds to the rush hours in the city.

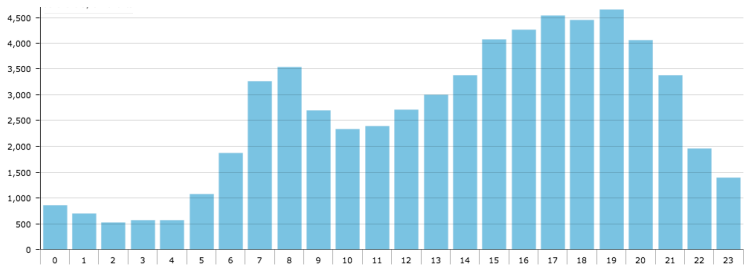


Figure 6. Relation between the behavior in Twitter and the traffic-related events.

Table 3. Results obtained by the methodology.

	Baseline	Standardization	Standardization + equivalent axis names	Standardization + equivalent axis names + dictionaries	Test Dataset
All elements found	152	152	427	456	652
At least one element found	289	388	599	608	652
Mistakes	363	264	53	44	0
Precision	0.39	0.43	0.83	0.85	1.0
Recall	0.31	0.39	0.80	0.83	1.0

As part of this work, a visualization of the results by means of a web-mapping application is presented. We have created a system that shows accidents and conditions in real time; therefore, we need two different ways to represent them (points and lines). According to that, different representations of traffic events, based on the number of geographic elements detected (see Figure 7).

Figure 7. Based on the number of geographic elements detected, image A shows events classified like accidents, therefore their representation are points, in the other hand, image B shows events that were classified as conditions, so they are represented as lines.

6. Conclusions and future work

In this paper, a novel methodology to geolocate traffic-related events in Twitter is proposed. We improve considerably the geocoding, using a gazetteer enriched with information from Twitter's stream. We also discovered how to divide traffic-related events in order to give more accurate representations, and found that the number of geographic elements in a tweet has a relation with the kind of traffic-related event. This research demonstrated that there is a relation between Twitter participation and the rush hours in the city. Geocoding of traffic-related events in Twitter or even in the web is an useful resource to know the behavior of city conditions, such as how to avoid crowded areas, assign traffic

polices or detect broken traffic lights. This information can be used for training a machine learning approach and making predictions about the city conditions throughout the day.

Future works are focused on discovering other relationships among geographic elements and finding other cases that can occur independently of the assumptions established in this work. Our approach does not consider the direction of the traffic-related event. So, it is necessary to define a method for inferring the direction where the event is located. The nearest areas outside Mexico City have a lot of urban mobility, so it is necessary to expand this methodology to the nearest areas beyond city boundaries. A temporal analysis is required to establish reasonable time duration for an accident or condition. A baseline is set up to each traffic-related event. So, a time duration *t* could be assigned to them and if the event is mention again, then the time is reset. Moreover, Twitter accounts always post with the same structure, thus a machine learning method could be implemented to learn features.

Acknowledgments

This work was partially sponsored by the Instituto Politécnico Nacional under grants 20151176, and 20151652. Additionally, we are thankful to the INEGI for the free access to its cartography, CNPq and FAPEMIG, Brazilian agencies in charge of fostering scientific development.

References

Backstrom, L., Sun, E., & Marlow, C. (2010) "Find me if you can: improving geographical prediction with social and spatial proximity." In: Proceedings of the 19th international conference on World wide web (pp. 61-70). ACM.

Bird, S. (2006) "NLTK: the natural language toolkit." In: Proceedings of the COLING/ACL on Interactive presentation sessions (pp. 69-72). Association for Computational Linguistics.

Cavnar, W. B., & Trenkle, J. M. (1994) "N-gram-based text categorization." In: Ann Arbor MI, 48113(2), 161-175.

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." In: Transactions in GIS, 15(6), 735-751.

Delboni, T., Borges, K. A., Laender, A. H., & Davis, C. A. (2007) "Semantic expansion of geographic web queries based on natural language positioning expressions." In: Transactions in GIS, 11(3), 377-397.

Lee, R., Wakamiya, S., & Sumiya, K. (2013) "Urban area characterization based on crowd behavioral lifelogs over Twitter." In: Personal and Ubiquitous Computing, 17(4), 605-620.

Ribeiro Jr, S. S., Davis Jr, C. A., Oliveira, D. R. R., Meira Jr, W., Gonçalves, T. S., & Pappa, G. L. (2012) "Traffic observatory: a system to detect and locate traffic events and conditions using Twitter", In: Proceedings of the 5th International Workshop on Location-Based Social Networks (pp. 5-11). ACM

Geographical prioritization of social network messages in near real-time using sensor data streams: an application to floods

Luiz Fernando F. G. de Assis[1], **Benjamin Herfort**[2], **Enrico Steiger**[2], **Flávio E. A. Horita**[1], **João Porto de Albuquerque**[1,2]

[1]Institute of Mathematics and Computer Science (ICMC)
University of São Paulo (USP) – São Carlos/SP – Brazil

[2]GIScience Research Group
Heidelberg University – Heidelberg, Germany

{luizffga,horita,jporto}@icmc.usp.br,
enrico.steiger@geog.uni-heidelberg.de, herfort@stud.uni-heidelberg.de

Abstract. *Social networks have been used to overcome the problem of incomplete official data, and provide a more detailed description of a disaster. However, the filtering of relevant messages on-the-fly remains challenging due to the large amount of misleading, outdated or inaccurate information. This paper presents an approach for the automated geographic prioritization of social networks messages for flood risk management based on sensor data streams. It was evaluated using data from Twitter and monitoring agencies of different countries. The results revealed that the proposed approach has a potential to identify valuable flood-related messages in near real-time.*

1. Introduction

The growing number of natural disasters, such as floods, has been leading for better preparation from vulnerable communities. In this sense, in-situ and mobile sensors are providing historical and updated information through the monitoring of environmental variables (e.g. temperature of the water or the volume of rainfall). Although these data are useful for supporting decision-making, further information is required for estimating the overall situation at an affected area [Horita et al. 2015]. Social networks like Twitter, Facebook, and Instagram, can overcome this issue either by providing information from areas which are not covered by sensors or complementing semantically the data provided by them [Starbird and Stamberger 2010, Vieweg et al. 2010, Zielinski et al. 2013, Horita et al. 2015].

Despite the fact that the combination of sensor data streams and social network messages might provide better information for supporting decision-making in critical situations like floods [Mooney and Corcoran 2011], it raises some challenges. On the one hand, the huge volume of messages shared

through social network makes difficult the identification of relevant messages, i.e. decision-makers do not want to analyse thousands of messages, they need the most valuable in order to make faster their decision-making [Vieweg et al. 2014]. On the other hand, the near real-time integration of sensor data and social network messages raises issues regarding the combination of distinct data streams (e.g. per second or per minute) and different data formats (e.g. numbers and texts) [Dolif et al. 2013].

In this context, this paper aims to present an approach for supporting flood risk management by means of the near real-time combination of social network messages and sensor data streams. It extends our previous works [Assis et al. 2015, Albuquerque et al. 2015] by adopting a workflow analysis which structures and defines an automated near real-time prioritization of social network messages based on the sensor data stream. Furthermore, it describes the formal representation of the problem statement, and makes an evaluation of the approach through case studies. In summary, the main contributions of this work are described below:

1. To define an approach to combine a sensor data stream with social network messages, aiming at identifying high value messages in near real-time for flood risk management;
2. To learn lessons from the application of the proposed approach in a real-world flood scenario in our application case study.

The remainder of this paper is structured as follows. Section 2 examines the related works. Section 3 sketches the background of the basic concepts and introduces the approach and methodology employed in this work. Section 4 describes the evaluation of the approach and their results are analyzed in Section 5. Finally, 6 draws conclusions and recommends some future works.

2. Related Works

Several applications have attempted to combine authoritative and non-authoritative data to improve the limitations of each other. Existing approaches integrate authoritative and non-authoritative data to provide location-based eventful visualization, statistical analysis and graphing capabilities in near real-time [Wan et al. 2014, Schnebele et al. 2014]. The combination of information provided by a collaborative platform (esp. Ushahidi) and sensor data collected via a wireless sensor network have been built for decision-making in flood risk management [Horita et al. 2015].

Several other studies aim at analyzing the large amount of information provided by social networks [Ediger et al. 2010, Gao et al. 2011], exploring e.g., relations between spatial information from both social network messages and knowledge about flood phenomena [Albuquerque et al. 2015]. An algorithm for monitoring social network messages (esp. tweets) and detecting

upcoming events is another eminent approach [Sakaki et al. 2010]. This algorithm classifies tweets using their keywords, number of words, and context. There are also systems for processing and analyzing social network messages in near real-time [Song and Kim 2013]. The results of its application to monitor Korean presidential elections showed that social network can support the detection and prediction in the changing of communities' behaviour. Finally, examining earliest social network messages that have produced a trend with the aim at identifying and creating a classification schema allows a categorization of messages, and thus a discovery of potential trends in near real-time [Zubiaga et al. 2015].

Although some studies have been done in the field, none of them tackles the combined use of social network messages and data collected from sensor streams in near real-time. In this manner, the processing of different data flows and data formats still pose challenges for the the use of sensor data as an alternative to support the filtering and extraction of high value social network messages in near real-time.

3. Problem Statement and Approach

3.1. Prioritization of Location-based Social Network Messages

Problem Statement. Due to the high volume of social network messages, the process of extracting relevant messages has been becoming more complex. This is because most of these messages are shared from several platforms (e.g. Flickr and Twitter) in distinct formats (e.g. photos and texts) with different data flows (e.g. per hour or per second).

Research Question. Is a sensor data stream able to support the near real-time identification of relevant social network messages in flood risk management?

Hypothesis. Given a set of catchments C, sensor data stream S and location-based social network messages M, we assume that the n-messages $M = \{m_1,...,m_n\}$ nearest to the m-flooded areas $F_{t_r} = \{f_1,...,f_m\}$ available at a time t_r tend to be more flood-related than the more distant (p-n)-messages $M = \{m_{n+1},...,m_p\}$, where n, m, p and $r \in \mathbb{N}$, and t is a timestamp. F is defined here as a time series of flooded areas. $F = \{F_{t_1}, ..., F_{t_r}\}$.

Definition 1. This paper uses a set of georeferenced catchments $C = \{c_1,...,c_n\} \subseteq R^2$. Each c_j denotes a two-dimensional Euclidean space that can either contain sensors or not. A sensor data stream $S = \{s_1,...,s_m\}$ contains a set of continuous sensor data s_k = [i, v, t, p, c]. Each sensor data has an id s_k.i, a water level value s_k.v at a timestamp s_k.t, a geographic position s_k.p = (x; y) and a s.c catchment. In case a sensor data s_k contains s_k.v equal to "high" at a timestamp s_k.t, and a s_k.p within a catchment c_j, this catchment c_j become a flooded area $f_p \in F_{s_k.t} \subseteq C$. The f_p is available until that s_k.v and any other sensor value contained in the catchment c_j are not "high" anymore at a subsequent timestamp to s_k.t.

$$F = \{F_{t_1}, ..., F_{t_r}\} \tag{1}$$

$$\begin{aligned} F_{t_r} = \{f_0, ..., f_p\} \mid \exists\ s_k\ \in\ S\ and\ f_p \in F_{t_r}, \\ where\ s_k.v = \text{“}high\text{”}\ and\ s_k.c = f_p\ and\ s_k.t = t_r. \end{aligned} \tag{2}$$

Location-based social network users U = {u_1,...,u_q} produce georeferenced messages represented by M = {m_1,...,m_n}. Each message m_i = [u, t, v, g] contains a value text m_i.v, as well as is produced by an user m_i.u at a timestamp m_i.t in a geographic location m_i.g. If there is a flooded area f_p, for each new incoming message m_i, a distance d is computed by the nearest neighbour (Euclidean) distance of the m_i.g position to every element of the set F_{t_r}) = $\{f_p\}_{p=0}^{n}$ at a timestamp t_r.

Definition 2. In general, the cartesian minimum distance between two points p (message location) and p' (the nearest point contained in a flooded area) in a Euclidean space R^n is given by:

$$d(p, p') = \sqrt{\sum_{i,j=1}^{n} |p_i - p'_j|^2},\ where\ i,\ j \in \mathbb{N}. \tag{3}$$

3.2. Sensor Data Stream and Social Network Message Workflows

Given the extent of the flood phenomena, spatiotemporal characteristics of both sensor data stream and social network can be combined and more explored. Social networks messages can be used to complement sensor data stream with semantic information, while sensor data stream can add reliability to the social network messages. For this reason, this approach is designed to suit an on-the-fly prioritization for different levels of data availability that are require to ensure an effective flood risk management.

The proposed workflow is performed in a pipeline way that contains both a sensor data stream S and a social network messages stream M. In the sensor data stream part (see Figure 1), there are three possible kinds of data availability. The first alternative is to have highly qualitative information about the extent of the flood phenomena, which can be provided by Unmanned Aerial Vehicles (UAVs). Although they provide the best degree of accuracy for detecting events in near real-time, they are rarely gathered.

If no direct sensor data is able to show the existence of a flooded area f_p, thus it is analyzed if they can either provide a flood hazard area or not. Local data about the affected areas e.g., maps of risks can potentially provide this kind of information. In the last stage of the verification, if neither the flooded area f_p nor the flood hazard area are initially available, a digital elevation model (provided by an user) is used to calculate a flood hazard area. After calculating this flood hazard area, they are used as an input (along with

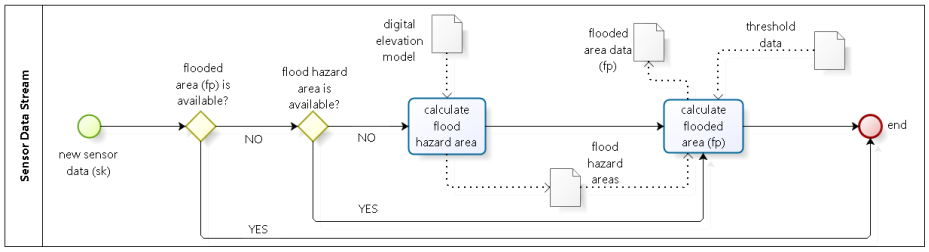


Figure 1. Sensor Data Stream Workflow.

threshold data) to calculate the flooded area f_p. The flooded area f_p in this part, is used to calculate the prioritization-based distance $P(m_j)$ when a new social network message m_j arrives (see Equation 4). In case more than one flooded area is available, the nearest flooded area distance is assign to the message prioritization.

$$p(m_j) = min(d(m_j, f_p)),\ where\ m_j \in M,\ f_p \in F\ and\ j,\ p \in \mathbb{N}. \tag{4}$$

Social network messages m_j and sensor data s_k should gathered simultaneously so that even delayed flood-related messages can be acquired. In the social network message stream part (see Figure 2), for each new incoming social network message m_j, prioritization-based distance is computed according to the nearest existing flooded area available produced by the sensor data stream part of the workflow. After calculating this prioritization-based distance, the messages are filtered to find messages that are likely to refer to a flood event. At first, our aim is to store all the messages since the specific keywords for floods might change during a flood. In this way, the filtering process can be easily adjusted without losing any flood-related messages.

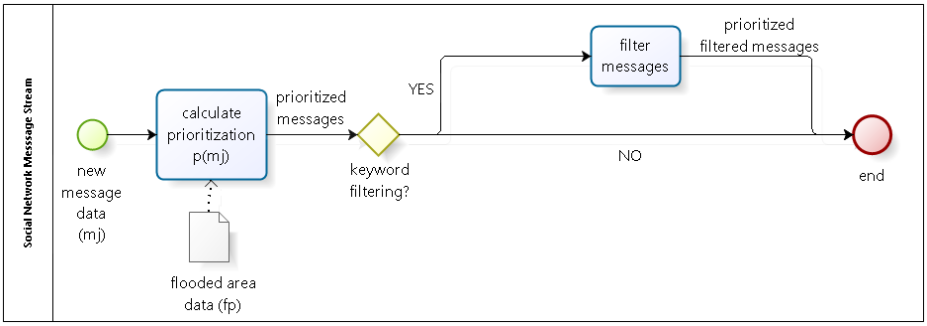


Figure 2. Social Network Message Workflow.

4. Case Studies and Experimental Setup

The approach was evaluated by means of an application case study of floods in Brazil, since it is currently taking measures to cope with and alleviate flood

situations. This set of measurements includes activities that involves pre-flood planning, managing emergency situations and post-flood recovery [Ahmad and Simonovic 2006].

Updated knowledge of river conditions plays an important role at supporting decision-making, since several technical factors can prevent this kind of information from being obtained. Countries such as Brazil where there are flash floods caused by heavy rain or the overflow of streams and narrow gullies needs this kind of management to mitigate the damage to the local infrastructure. This means that emergency agencies have to cope with the risk of human casualties and the extent of flood damage in their decision-making.

In this application case study, we considered as data source, authoritative static data (shapefiles), authoritative dynamic data (sensor data streams), and social network messages (Twitter).

4.1. Case Study: Flash Floods in Brazil

The analysis is confined to São Paulo as a single region within Brazil so that it is easier to compare and link the results of the case study. The shapefile of the State of São Paulo was produced using *geotools*[1] operations and geo-referenced data sets provided by HydroSHEDS[2]. It contains 315 small catchments. The sensor data streams was obtained from the national center for monitoring disasters and issuing warnings in Brazil (Cemaden). This agency operates by continuously installing new stations and providing their data through a Rest API.

In the State of São Paulo, Cemaden provided data from 465 stations. Each station and measurement of Cemaden are combined at the same file. The provided measurements from all the stations regarded the last four hours, considering an offset. This is the difference between the distance of the installation position and the real water level. As soon as it starts raining, the stations measure the rainfall every 10 minutes, otherwise they measure every 60 minutes. The floods in Brazil are represented at their most extreme by flash floods.

The catchments, stations and flooded areas are depicted in Figure 3, while a density map of the prioritized tweets is shown in Figure 4.

4.2. Experimental Setup

Runtime Environment: The implementation to retrieve Cemaden data was based on a simple Java toolkit for JSON[3], while tweets were retrieved using a Java library for Twitter API called Twitter4j [4]. Both of them were implemented in a pipelined fashion as a data stream. The experiments were run on a server

[1] http://www.geotools.org
[2] http://hydrosheds.cr.usgs.gov/index.php
[3] https://code.google.com/p/json-simple/
[4] http://twitter4j.org/en/index.html

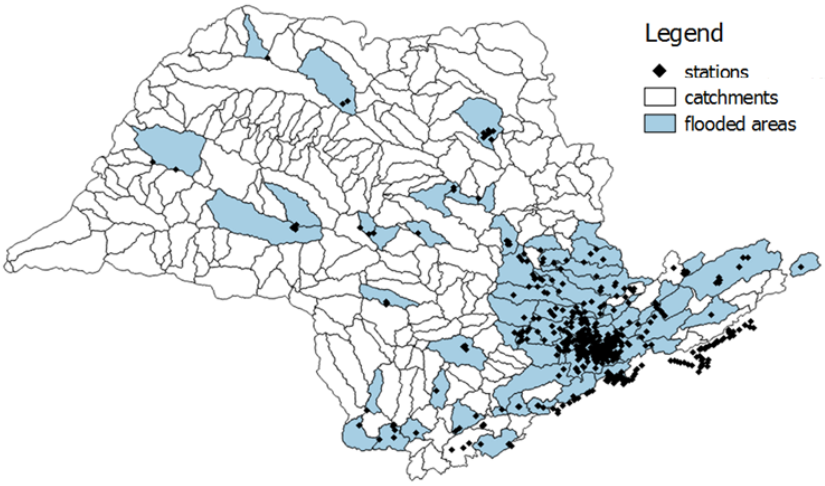


Figure 3. São Paulo - An analysis of Brazilian Stations and Catchments.

with 2GHz AMD Opteron(tm) Processor 4171 HE and 3.4 GB RAM memory running Ubuntu 12.04.5 LTS (64 bit).

Dataset: Twitter Streaming API and Cemaden Rest API were used to retrieve data in the period from April to May 2015.

5. Results and Analysis

In our case study, only 6% (68,195) of the tweets were prioritized mainly due to the flash floods. Such application case study helped to represent the scenarios when flood occur, since a large number of tweets were posted at critical moments. At first, all the flood-related tweets (403) were filtered by making a selection of the prioritized tweets (1,136,583) based on specific keywords and their synonyms. This "keyword selection" was based on the Brazilian words in the dictionary for "flood", taking into account differences of case sensitive letters without spelling mistakes. The Brazilian keywords were "enchente", "inundação" and "alagamento".

Table 1. Keyword-based filtering description of the location-based social network messages

# all the tweets	# prioritized tweets	# prioritized flood-related tweets	# prioritized non flood-related tweets
1,136,583 (100%)	68,195 (6%)	403 (0,04%)	67,792 (5.96%)

All the stations provided 284,663 measurements, which included 311 distinct stations that measured 1,030 high values. These values set for up to 59 distinct catchments areas that are considered to be flooded. A detailed description of the data provided by stations and their measurements is depicted in Table 2.

We also decided to calculate the time that our approach takes to prioritize all the tweets. The latency of the tweets was considered to be acceptable

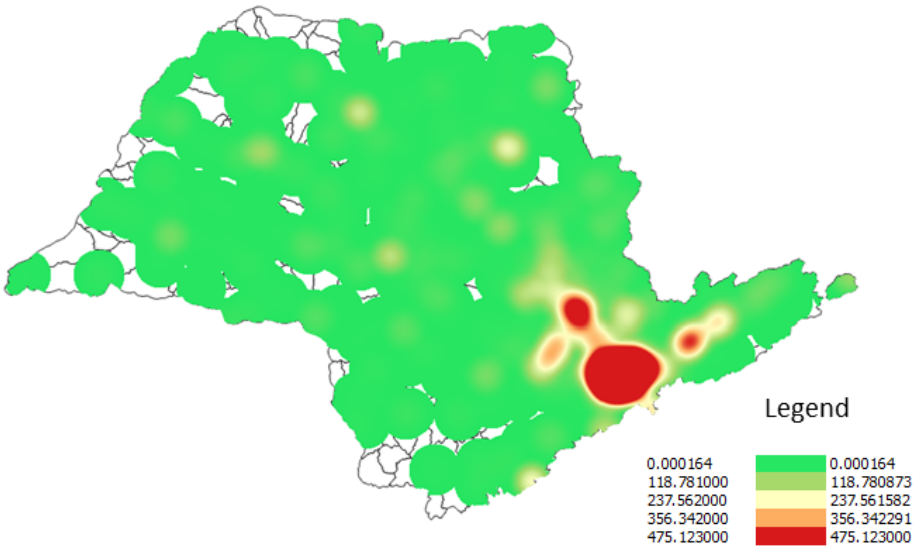


Figure 4. São Paulo - An analysis of Prioritized Tweets during periods of floods.

Table 2. Sensor measurement description of Cemaden stations.

# catchments (flooded areas)	# stations (high values)	# measurements	# all the measurements (high values)
59 (18.7%)	311 (66.9%)	284,663 (100%)	1,030 (0.0037%)

for the prioritization of social network messages for floods. This latency was the difference in time between the arrival from the Streaming API and the storage at the database. The latency was only calculated for the prioritized tweets. The processing average time per minute of all the prioritized tweets was less than one second.

Tweets containing relevant information presented not only flood-related text, but also georreferenced images that can help in flood risk management. As can be seen, exemplary prioritized tweets containing relevant messages are shown in Table 3 and Figure 5.

In addition, a statistical hypothesis test was conducted to check whether flood-related tweets are closer to hazard areas than those that are non- flood-related during floods. The Mann-Whitney U-test was employed for the samples of prioritized flood-related and non-flood related tweets. The test returns the p-value of a two-sided Wilcoxon rank sum test, which tests the null hypothesis that the distance of independent samples with different lengths from continuous distributions of flood-related and non-flood related tweets are with equal medians, against the alternative that they are not.

The p-value of 7.2940e-016 indicates a rejection of the null hypothesis of equal medians at a 5% significance level. That means, the sample of tweets which contain flood-related keywords are not equally nearer to the hazard ar-

Table 3. Examples of prioritized Tweets containing flood-related keywords without images (On Topic, Relevant).

Flood-Related Prioritized Tweets	Translation
"tá td alagado aq na marquês"	"everything is flooded here in the Marquês" (name of a place or avenue name)
"Ta alagado aqui"	"It is flooded here" (the georeference of the tweet can supplement this information)
"Nações alagada"	The "Nações (Avenue Name) is flooded"
"Alagamento na av dos Tajuras (at @AG2 Nurun in São Paulo, SP)" https://t.co/r7ECeAtiFB"	"There is a flood in Tajuras avenue" (the georeference of the tweet can suplement this information)
"@VCnoSPTV chuva de 30 minutos e alagamento na região do Brás, pra variar http://t.co/wWVqIKtz3z"	"@VCnoSPTV (Twitter account of a TV program) it has been raining for 30 minutes and the region of Bras is flooded, as always http://t.co/wWVqIKtz3z"
"5min de chuva e rua já fica alagada"	"After 5 minutes of rain, the street is already flooded" (the georeference of the tweet can supplement this information)

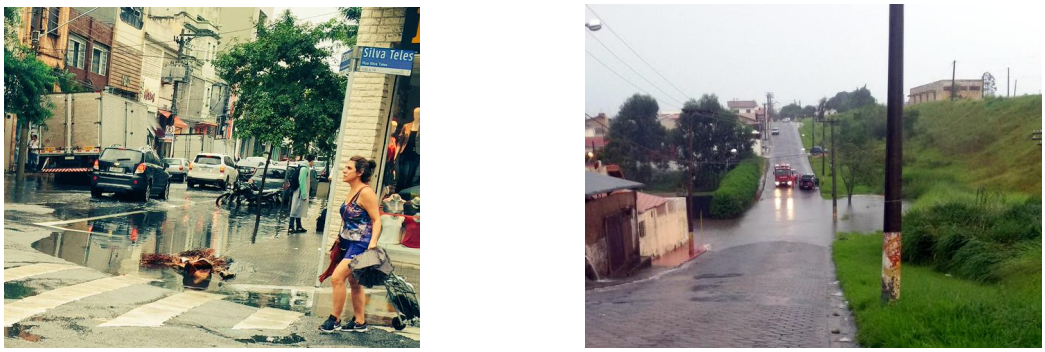


Figure 5. Examples of flood-related tweets containing images that can help in flood risk management.

eas to the ones that not contain flood-related keywords. The median distance of the sample of non- flood-related tweets was 10,905 meters away from those areas, while the sample of flood-related tweets was 3,027 meters away. Figure 6 shows the two distribution of non flood-related and flood-related tweets based on their prioritization (distance in km to flooded areas).

6. Conclusion and Discussions

This paper presents an approach for supporting flood risk management by means of a near real-time prioritization of social network messages based on sensor data streams. One case study was used for evaluating the approach. The results confirmed that the geographical relations are useful for prioritizing social network messages related to floods. They showed that there are about 3,6 times more flood-related social network messages near to flood-affected areas than non-flood-related messages. Although our approach was evaluated in a specific context of floods and using Twitter messages, it can be used to other types of disasters (e.g. droughts and landslide) and social network (e.g. Instagram and Flickr), i.e. considering images or videos instead of only texts messages.

Our approach gathered the messages per minute during a flood at an

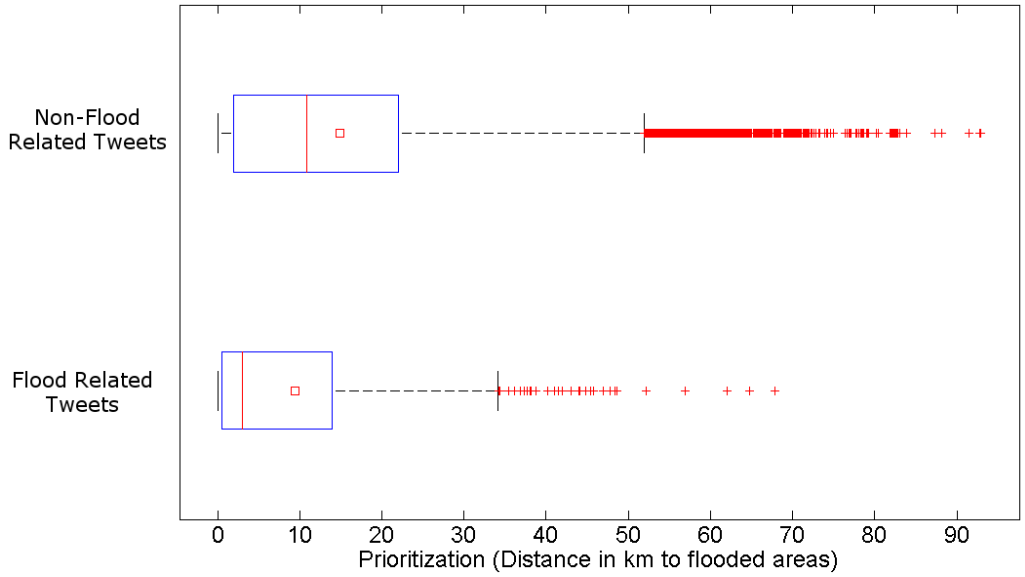


Figure 6. Median, Average and Outliers of flood-related and non flood-related tweets.

average processing time of less than one second. Given the large number of messages (time peaks should be treated as critical periods since more messages tend to be posted), such processing time to prioritize does not significantly change. This work has shown that social networks messages and sensor data streams can complement each other. Sensor data streams are accurate, dynamic, heterogeneous and continuous, although they are scarce and hard to implement and maintain. On the other hand, social network messages can enhance semantic sensor data, but their large number is not easy to handle since they can be misleading, outdated or inaccurate. Despite the lack of user experience and knowledge, social networks have been used in crisis management revealing their remarkable and positive features.

Although most of the existing approaches are still insufficient for near real-time decision-making since they fail to take note of the fact that data in disasters should be analyzed on-the-fly and automatically. Our approach searches for georeferenced social network messages using a grid 5x5 bounding box based on the catchments dimension. Although most of the messages are not flood-related (and do not contain any important keywords such as “floods” or “inundation”), they were stored at the database after first being filtered because a keyword search is arbitrary, especially for near real-time event detection.

All the social network messages located within a flooded area were prioritized with zero meters “0 m” as distance, which is the main value-based prioritization. Some of the prioritized messages have images embedded in them, which were really useful when they were geolocated because they could show

the exact situation of a particular place and sometimes helped more than simply by the words. During our analysis, a few of the total amount of available messages were both georeferenced and considered to be flood-related.

Furthermore, heavy rains might affect the connection infrastructure (e.g. cellphone services or wi-fi), which in turn may reflect on the unavailability of information sharing. Although this issue is important when dealing with social network messages, it is beyond the scope of this work. In this sense, a better time resolution and spatial distribution of the sensor measurements would improve the availability of information provided by sensors. In situations that sensors are measuring high values all the time, machine learning techniques would be an useful way to check whether the sensors are really in a flood situation or only measuring high values all the time because of its position on the river.

Future work lines should take account of using the prioritization of social network messages as one step to further filtering and classifying the quality of crowdsourcing. Besides that, it can serve as basis to improve machine learning models that consider geographical links.

References

Ahmad, S. and Simonovic, S. P. (2006). An Intelligent Decision Support System for Management of Floods. *Water Resources Management*, 20(3):391–410.

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

Assis, L. F. F. G., Herfort, B., Steiger, E., Horita, F. E. A., and ao Porto Albuquerque, J. (2015). A geographic approach for on-the-fly prioritization of social-media messages towards improving flood risk management. In *Proceedings of the 4th Brazilian Workshop on Social Network Analysis and Mining (BraSNAM)*, pages 1–12.

Dolif, G., Engelbrecht, A., Jatobá, A., da Silva, A. J. D., Gomes, J. O., Borges, M. R., Nobre, C. A., and de Carvalho, P. V. R. (2013). Resilience and brittleness in the alerta rio system: a field study about the decision-making of forecasters. *Natural hazards*, 65(3):1831–1847.

Ediger, D., Jiang, K., Riedy, J., Bader, D., Corley, C., Farber, R., and Reynolds, W. (2010). Massive social network analysis: Mining twitter for social good. In *Proceedings of the 39th International Conference on Parallel Processing (ICPP)*, pages 583–593.

Gao, H., Barbier, G., and Goolsby, R. (2011). Harnessing the crowdsourcing power of social media for disaster relief. *IEEE Intelligent Systems*, 26(3):10–14.

Horita, F. E., Albuquerque, J. P., Degrossi, L. C., Mendiondo, E. M., and Ueyama, J. (2015). Development of a spatial decision support system for flood risk management in brazil that combines volunteered geographic information with wireless sensor networks. *Computers & Geosciences*, 80:84–94.

Mooney, P. and Corcoran, P. (2011). Can Volunteered Geographic Information be a participant in eEnvironment and SDI? In *Environmental Software Systems. Frameworks of eEnvironment*, pages 115–122.

Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, pages 851–860.

Schnebele, E., Cervone, G., Kumar, S., and Waters, N. (2014). Real time estimation of the calgary floods using limited remote sensing data. *Water*, 6(2):381–398.

Song, M. and Kim, M. C. (2013). Rtˆ2m: Real-time twitter trend mining system. In *Proceedings of the 2013 International Conference on Social Intelligence and Technology*, pages 64–71.

Starbird, K. and Stamberger, J. (2010). Tweak the tweet: Leveraging microblogging proliferation with a prescriptive syntax to support citizen reporting.

Vieweg, S., Castillo, C., and Imran, M. (2014). Integrating social media communications into the rapid assessment of sudden onset disasters. *Social Informatics*, 8851:444–461.

Vieweg, S., Hughes, A. L., Starbird, K., and Palen, L. (2010). Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1079–1088. ACM.

Wan, Z., Hong, Y., Khan, S., Gourley, J., Flamig, Z., Kirschbaum, D., and Tang, G. (2014). A cloud-based global flood disaster community cyberinfrastructure: development and demonstration. *Environmental Modelling & Software*, 58:86–94.

Zielinski, A., Middleton, S. E., Tokarchuk, L., and Wang, X. (2013). Social media text mining and network analysis for decision support in natural crisis management. *Proceedings of the 10th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, pages 840–845.

Zubiaga, A., Spina, D., Martínez, R., and Fresno, V. (2015). Real-time classification of twitter trends. *Journal of the Association for Information Science and Technology*, 66(3):462–473.

Análise geográfica entre mensagens georreferenciadas de redes sociais e dados oficiais para suporte à tomada de decisões de agências de emergência

Thiago H. Poiani[1], **Flávio E. A. Horita**[1], **João Porto de Albuquerque**[1,2]

[1]Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP) – São Carlos/SP – Brasil

[2]GIScience Research Group
Heidelberg University – Heidelberg – Germany

thpoiani@usp.br, {horita,jporto}@icmc.usp.br

Abstract. *The recent damages caused by floods have called for better preparation from vulnerable communities. New data sources like in-situ sensors and social media have opened different perspectives for supporting data collection, and then improving decision-making of the emergency agencies. Therefore, this paper presents a geographical analysis of the relationship between authoritative data and georeferenced social media messages with the aim of understanding their contributions to decision-making in case of floods. The results showed a straight relationship between georeferenced social media messages and authoritative data. Furthermore, it was revealed that these messages are useful to provide information about the situation at the affected area.*

Resumo. *Os recentes danos causados pelas inundações chamam a atenção para uma melhor preparação das comunidades vulneráveis. Novas fontes de dados como sensores estáticos e mídia social abriram diferentes perspectivas para auxiliar na coleta de dados e, assim, melhorar a tomada de decisões das agências de emergência. Este artigo apresenta uma análise geográfica do relacionamento entre dados oficiais e mensagens georreferenciadas de mídias sociais com o objetivo de entender suas contribuições para a tomada de decisões em inundações. Os resultados mostraram uma forte relação entre mensagens georreferenciadas de mídias sociais e dados oficiais. Além disso, tais mensagens também podem prover informações úteis sobre a situação na área afetada.*

1. Introdução

Inundações são perigos naturais hidrológicos recorrentes em diversas regiões do Brasil e que mais afetaram pessoas e causaram mortes entre o período de 2004 e 2014 [Guha-Sapir et al. 2015]. Para servir como suporte aos desastres naturais no país, o Ministério da Ciência, Tecnologia e Inovação criou, em 2011, o Centro Nacional de Monitoramento e Alertas de Desastres Naturais (CEMADEN)[1]. Pluviômetros instalados em áreas de risco de inundação são monitorados por essa agência de emergência, coletando dados climáticos que dão suporte para tomada de decisões.

[1]http://www.cemaden.gov.br/

O acúmulo de informações de fontes de dados distintas auxilia a formulação de estratégias de gestão de risco de inundações. Mídia social é uma fonte com potencial de uso devido à grande quantidade de informações geográficas voluntárias (VGI) distribuída em um tempo curto por *sensores humanos* [Goodchild 2007].

O objetivo desse artigo é apresentar uma análise geográfica da relação entre dados oficiais e mensagens georreferenciadas de mídias sociais. Para isso, são utilizados dados de sensores pluviométricos do CEMADEN e mensagens coletadas no Twitter. A partir disso, espera-se, além de entender as contribuições das mensagens de mídias sociais, identificar novos locais relatados por sensores humanos que não são monitorados para auxiliar na tomada de decisões das agências de emergências no caso de inundações.

O restante desse artigo está organizado da seguinte forma: na Seção 2 descreve-se a fundamentação teórica e alguns trabalhos relacionados. Na Seção 3 estão as técnicas e metodologias utilizadas nessa pesquisa. Na Seção 4 são apresentados os resultados. Por fim, a Seção 5 apresenta a conclusão e sugere trabalhos futuros.

2. Gestão de Risco de Inundações e Mídias Sociais para Desastres

No Brasil, os problemas de inundações são recorrentes. No período de 2004 a 2014, esses perigos naturais causaram mais dano do que outros tipos de eventos, como secas e escorregamentos de terra [Guha-Sapir et al. 2015]. Nesse contexto, a gestão de risco de inundações se mostra uma importante solução para minimizar os impactos sociais, financeiros e ambientais. Suas atividades podem ser agrupadas em três fases [Ahmad and Simonovic 2006]: (1) Planejamento pré-inundação; (2) Gestão de emergência; e, (3) Recuperação pós-inundação. Em todas estas fases, a coleta de informações é fundamental no suporte às atividades dos tomadores de decisão [Ahmad and Simonovic 2006].

Neste sentido, plataformas de mídia social como Twitter, Facebook e Instagram, permitem aos usuários o compartilhamento de suas informações com outras pessoas através da rede social. Por meio destas plataformas, torna-se possível analisar atividades diárias e, com isso, prever possíveis movimentações sociais. Alguns exemplos de pesquisas voltadas para o campo de desastres visam apoiar à tomada de decisões [Vieweg et al. 2014], auxiliar na predição de eventos [MacEachren et al. 2011] e aumentar o conhecimento situacional [Starbird et al. 2010]. Outro grupo de pesquisa busca analisar as contribuições para a integração de informações de mídias socias e dados oficiais. [Croitoru et al. 2013] revelam a existência de uma relação entre o espaço, rede social e eventos, que pode render na compreensão do comportamento de uma comunidade. [Albuquerque et al. 2015] demonstram que mensagens de redes sociais mais próximas ao evento natural podem possuir mais informações úteis sobre o desastre.

Apesar de tratar da integração de dados oficiais e mensagens de mídias sociais, muitas das pesquisas anteriores falham em utilizar esses dados como forma de filtrar mensagens de mídias sociais. Esta combinação poderia auxiliar na descoberta de conhecimento relevante e, assim, prover mais informações para melhorar a tomada de decisões na gestão de risco de inundações.

3. Metodologia

Esta pesquisa tem como objetivo analisar a relação geográfica entre dados oficiais e mensagens georreferenciadas de mídias sociais. Dessa forma, ela busca responder a seguinte pergunta de pesquisa: *PP) Dados oficiais podem auxiliar na identificação de novas áreas de inundação por meio da análise de mensagens georreferenciadas de mídias sociais?*

Para isso, essa Seção descreve os passos realizados para o desenvolvimento das análises qualitativas e quantitativas, tendo como estudo de caso o estado de São Paulo por possuir uma grande densidade populacional, com 166,25 habitantes por quilômetro quadrado [Instituto Brasileiro de Geografia e Estatística 2010], e 367 sensores pluviométricos monitorados pelo CEMADEN.

3.1. Análise qualitativa

A análise qualitativa é responsável pela classificação de mensagens publicadas na rede social Twitter no período de 7 a 31 de maio de 2015.

Para a coleta de mensagens, foi usado o serviço Twitter Streaming API[2] que permite uma coleta contínua utilizando filtragem por localização feita por um *bounding box*, uma área limite definida por um polígono através das posições geográficas de seus vértices. Um *bounding box* que abrange todo o estado de São Paulo foi determinado como: -53.11 (longitude mínima), -25.48 (latitude mínima), -44.16 (longitude máxima), -19.78 (latitude máxima). A partir disso, as mensagens recebidas foram armazenadas em uma base de dados não relacional orientada a documentos.

A análise dos dados necessitou que os *tweets* fossem normalizados, mantendo assim apenas as propriedades essenciais para a análise de conteúdo: identificador, hora de criação, texto e dados geográficos. Para os *tweets* que não possuíam geolocalização, a propriedade ”dados geográficos”foi definida com valor nulo.

Para a extração dos dados, foram considerados apenas *tweets* que possuíam georreferência do local de envio e mensagens com determinados termos relevantes para a pesquisa. Foram determinadas palavras-chave para evitar que conteúdo irrelevante fosse retornado. Após alguns testes pilotos para definir quais seriam os termos mais relevantes, os seguintes termos foram escolhidos: *chuva*, *chuvisco*, *água*, *garoa*, *nuvem*, *tempestade*, *temporal*, *dilúvio*, *alagamento*, *inundação*, *enchente*. Dessa forma, os *tweets* foram extraídos da base de dados a partir do mecanismo de consulta *full-text search*, que permite o retorno de mensagens que possuem as palavras-chave determinadas e termos similares.

Por fim, essas mensagens foram lidas e classificadas em categorias de acordo com o seu conteúdo. Mensagens sem relação com a proposta do estudo foram classificadas como ”fora do contexto”. Publicações com relação foram classificadas como ”dentro do contexto”, porém as mensagens mais relevantes, que possuíam informações temporais e geográficas, foram classificadas também como ”relevante”. Vale ressaltar também que foi realizado um processamento adicional com base nas coordenadas dos limites de São Paulo para garantir a inclusão de *tweets* apenas do estado.

[2]https://dev.twitter.com/streaming

3.2. Análise quantitativa

A análise quantitativa é responsável por identificar novas áreas de riscos de inundação através da combinação da análise dos *tweets* e dos locais das estações pluviométricas do Centro Nacional de Monitoramento e Alertas de Desastres Naturais.

As medições das estações pluviométricas estão disponíveis através da área de download do Mapa Interativo da Rede Observacional para Monitoramento de Risco de Desastres Naturais[3].

Os pluviômetros da área realizam medições a cada 10 minutos quando ocorre chuva contínua, caso contrário, de hora em hora. O arquivo transferido é uma planilha composta por dados dos pluviômetros, com identificador, coordenadas geográficas, hora da medição e volume de chuva. Para esta pesquisa, o documento do mês de maio e do estado de São Paulo foi utilizado.

A maior medição de chuva registrada no período analisado ocorreu em Campos do Jordão, atingindo um valor de 55,4 no dia 13/05 às 02h30. Contudo, o segundo maior valor é 28,4, registrado em Caieiras no dia 10/05 às 20h30. Portanto, essa medição de Campos do Jordão será considerada como um *outlier*, sendo removida da análise.

4. Resultados

No período estudado, foram coletados 1.589.549 *tweets* apenas com o filtro de *bounding box*. Adicionando os filtros de palavras-chave e georreferência, foram retornados 4.171 *tweets*. Com a remoção das mensagens que estavam fora dos limites do estado de São Paulo, foram totalizados 3.037 *tweets* para a análise. A partir da extração, os *tweets* foram classificados, atingindo uma quantidade de 1.614 mensagens fora do contexto, 1.423 dentro do contexto e, dentre estas, 1.181 relevantes para a pesquisa.

Com base na análise dos *tweets*, foi possível identificar dias com picos de publicações, em que a quantidade de mensagens dentro do contexto da pesquisa foi maior que as mensagens fora do assunto (Figura 1). Para investigar se o aumento da quantidade de mensagens relevantes está relacionado aos dias que ocorreram precipitações ou chuvas, foi necessário a análise das medições das estações pluviométricas.

Durante o período analisado, foram realizadas 403.046 medições nas estações pluviométricas. Para uma análise mais consistente dos dias e locais que registraram precipitações, os dados foram filtrados com volume de chuva maior que 0, chegando a uma quantidade de 56.032 medições. Na Figura 2 está representada a quantidade total de medições e as medições com volume de chuva por dia.

Para determinar se é possível identificar novas áreas de risco de inundação a partir da análise de mídia social combinada com pluviômetros, foi realizada uma análise dos dias 10 e 31, por representarem os maiores picos de atividades em ambos os gráficos.

Na Figura 3 está representada a disposição entre os locais de envio de *tweets* relevantes (pontos vermelhos) e as estações pluviométricas (clusters e marcadores azuis) que realizaram medições em 10 de maio. Com essa sobreposição, é possível identificar que a maioria dos locais que os *tweets* foram enviados relatando chuvas possuem pluviômetros próximos, como as regiões de Santos, São Paulo e Ribeirão Preto. Contudo, ainda assim

[3]http://www.cemaden.gov.br/mapainterativo

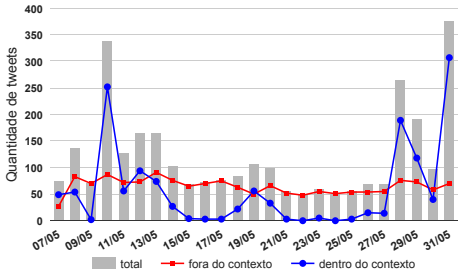


Figura 1. Quantidade de tweets classificados no período analisado

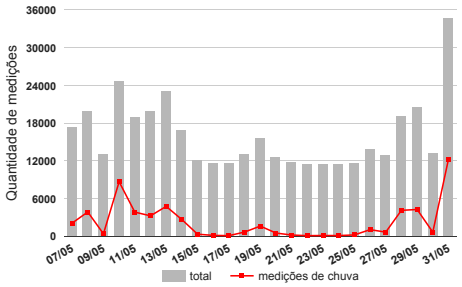


Figura 2. Quantidade de medições dos pluviômetros no período analisado

existem locais sem pluviômetros em que humanos agiram como sensores, sendo possível identificar possíveis novas áreas de risco, como na região de Ibitinga, Araçatuba e Birigui.

Na Figura 4 está apresentado a disposição entre sensores humanos e pluviômetros que realizaram medições no dia 31 de maio. Com essa sobreposição, é possível identificar que os principais *tweets* georreferenciados estão próximos de estações pluviométricas, com poucas exceções, como Presidente Prudente, Assis e Poços de Caldas.

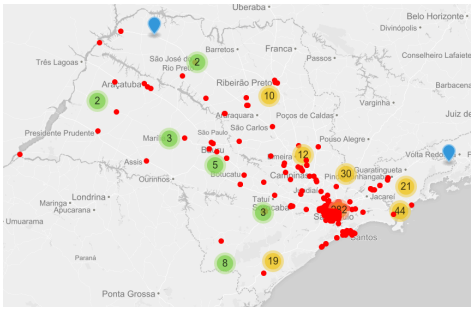


Figura 3. Disposição entre *tweets* e estações pluviométricas no dia 10 de maio

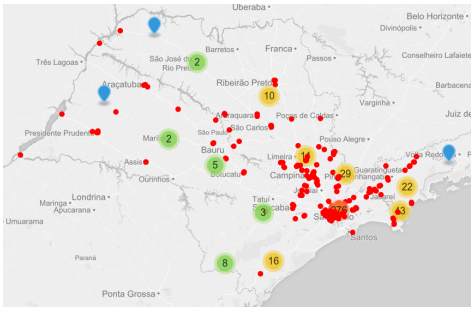


Figura 4. Disposição entre *tweets* e estações pluviométricas no dia 31 de maio

5. Conclusão

Nesse trabalho foi realizada uma análise quali-quantitativa para investigar se mensagens georreferenciadas de mídias sociais contêm informação útil para identificar novas áreas de risco de inundação. No período analisado, foram detectados picos de atividades com alta concentração de publicações de mensagens e medições de chuvas pelas estações pluviométricas. Com a análise do conteúdo de mensagens georreferenciadas relevantes, identificamos que os autores escrevem informações climáticas da região na qual se encontram, além de informar possíveis áreas de risco de alagamento. Dessa forma, pode-se afirmar que dados oficiais podem ser utilizados para auxiliar na filtragem de mensagens de mídias sociais e, assim, permitir a descoberta de informação relevante. Essa análise também serviria como uma etapa de preparação na gestão do risco de inundações, pois com uma grande concentração de mensagens sobre um mesmo evento, torna-se possível localizar novas áreas de risco.

Como trabalhos futuros, recomenda-se a elaboração de mapas de vulnerabilidade de inundação baseados em informações de redes sociais, comparando e avaliando com o mapa de vulnerabilidade da Agência Nacional das Águas (ANA)[4]. Uma análise geoestatística dos dados coletas (por exemplo, indicadores locais de associação espacial) mostrou-se necessária, sendo então adicionada nos próximos artigos. Além disso, tanto a criação de modelos para identificação de mudanças climáticas a partir de análise de redes sociais, quanto a automação das etapas de coleta e categorização de mensagens de mídias sociais são áreas promissas para trabalhos futuros.

Agradecimentos

THP agradece ao CNPq (130153/2015-0) e FAPESP (2015/05929-3) pelo apoio financeiro. FEAH e JPA agradecem a CAPES (Edital Pró-alertas 24/2014). FEAH agradece o suporte financeiro do CNPq (202453/2014-6). JPA agradece a CAPES (88887.091744/2014-01) e Heidelberg University (Excellence Initiative II / Action 7) por apoiar a sua contribuição à essa pesquisa.

Referências

Ahmad, S. and Simonovic, S. P. (2006). An Intelligent Decision Support System for Management of Floods. *Water Resources Management*, 20(3):391–410.

Albuquerque, J. P., Herfort, B., Brenning, A., and Zipf, A. (2015). A Geographic Approach for Combining Social Media and Authoritative Data Towards Identifying Useful Information for Disaster Management. *International Journal of Geographical Information Science*, pages 1–23.

Croitoru, A., Crooks, A., Radzikowski, J., and Stefanidis, A. (2013). Geosocial Gauge: a System Prototype for Knowledge Discovery from Social Media. *International Journal of Geographical Information Science*, 27(12):2483–2508.

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

Guha-Sapir, D., Below, R., and Hoyois, P. (2015). EM-DAT: International Disaster Database. Université catholique de Louvain.

Instituto Brasileiro de Geografia e Estatística (2010). Censo Demográfico. Disponível em: http://www.censo2010.ibge.gov.br/sinopse/index.php?dados=10&uf=00. Acesso em: 22 out.

MacEachren, A. M., Robinson, A. C., Jaiswal, A., Pezanowski, S., Savelyev, A., Blanford, J., and Mitra, P. (2011). Geo-twitter analytics: Applications in crisis management. In *25th International Cartographic Conference*, pages 3–8.

Starbird, K., Palen, L., Hughtes, A. L., and Vieweg, S. (2010). Chatter on The Red: What Hazards Threat Reveals about the Social Life of Microblogged Information. *Proceedings of the 2010 ACM conference on Computer Supported Cooperative Work (CSCW)*, pages 241–250.

Vieweg, S., Castillo, C., and Imran, M. (2014). Integrating Social Media Communications into the Rapid Assessment of Sudden Onset Disasters. *Social Informatics*, 8851:444–461.

[4]http://www2.snirh.gov.br/home/item.html?id=cf201bd9b2c540fa951b0619006eb2af

HydroGraph: Exploring Geographic Data in Graph Databases

Jaudete Daltio[1,2], **Claudia M. Bauzer Medeiros**[2]

[1]Institute of Computing - UNICAMP, Campinas – SP – Brazil

[2]Brazilian Agricultural Research Corporation's - EMBRAPA, Brazil

{jaudete, cmbm}@ic.unicamp.br

Abstract. *Water becomes, every day, more scarce. Reliable information about volume and quality in each watershed is important to management and proper planning of their use. Data-intensive science is being increasingly needed in this context. Associated analysis processes require handling the drainage network that represents a watershed. This paper presents an ongoing work that explores geographic watershed data using graph databases – a scalable and flexible kind of NoSQL databases. The Brazilian Watershed database is used as a case study. The mapping between geographic and graph models is based on the natural network that emerges from the topological relationships among geographic entities.*

1. Introduction and Motivation

During the last decade, the volumes of data that are being stored have increased massively. This has been called the "industrial revolution of data", and directly affected the world of science. Nowadays, the available data volume easily outpaces the speed with which it can be analyzed and understood [Fry 2004]. Computer science has thus become a key element in scientific research.

This phenomenon, known as eScience, is characterized by conducting joint research in computer science and other fields to support the whole research cycle, from collection and mining of data to visual representation and data sharing. It encompasses techniques and technologies for data-intensive science, the new paradigm for scientific exploration [Hey et al. 2009].

Besides the huge volume, the so-called "big data" carries many heterogeneity levels – including provenance, quality, structure and semantics. To try to deal with these requirements, new database models and technologies emerge aiming at scalability, availability and flexibility. The term *NoSQL* was coined to describe a broad class of databases characterized by non-adherence to properties of traditional relational databases [Hecht and Jablonski 2011]. It encompasses different attempts to propose data models to solve a particular data management issue.

Geospatial big data (i.e., big data with a geographic location component) faces even more challenges – it requires specific storage, retrieval, processing and analysis mechanisms [Amirian et al. 2013]. In addition, it demands improved tools to handle knowledge discovery tasks.

The more widely accepted kinds of NoSQL databases include key-value, document, column-family and graph models. Of these, graph databases are the most suitable

choice to handle geospatial big data [Amirian et al. 2014]. Indeed, graphs are the only data structure that natively deals with highly connected data, without extra index structures or joins. No index lookups are needed for traversing data, since every node has links to its neighbors. Besides, in GIS, topological relationships play an important role. These relationships can be naturally modeled with graphs, providing flexibility in traversing geospatial data based on diverse aspects.

Geospatial data about water resources fits these graph connectivity criteria – e.g., watersheds or drainage networks. Owing to the shortage of drinking water, reliable information about volume and quality in each watershed is important for management and proper planning of their use. A watershed is usually represented as drainage network, with confluences, start and end points connected by *drainage stretches* (the network edges).

This paper presents an ongoing work that explores geospatial watershed data taking advantage of graph databases. The goal is to show that this scenario provides additional opportunities for knowledge discovery tasks through classical graph algorithms. The Brazilian Watershed database is used as a case study. The mapping between geospatial and graph models is based on the natural network that emerges from the topological relationships among geographic entities.

The rest of this paper is organized as follows. Section 2 contains a brief description of the main concepts involved and gives an overview of the Brazilian Watershed relational database. Section 3 presents the process of loading watersheds to a graph database and presents results of important and recurrent queries over watersheds. Some research challenges involved are presented in section 4. Finally, section 5 presents conclusions and ongoing work.

2. Research Scenario and Theoretical Foundations

2.1. Brazilian Water Resources Database

Brazil is a privileged country in the water-shortage scenario: it holds 12% of the world total and the largest reserve of fresh water on Earth [Brebbia and Popov 2011]. Its distribution, however, is uneven across the country. Amazonas, for instance, is the state with the largest watershed and one of the less populous in Brazil. Furthermore, some rivers are being contaminated by waste of illegal mining activities (such as mercury), agricultural pesticides, domestic and industrial sewage leak and garbage.

Reliable information about volume and quality in water resources is extremely important to management and proper planning of their use. To this end, the Brazilian Federal Government approved in 1997 the National Water Law [Brazil 1997] aiming to adopt modern principles of management of water resources and created in 2000 the National Water Agency (ANA), legally responsible for accomplishing this goal and ensuring the sustainable use of fresh water.

To organize the required data and support management tasks, ANA adopts the watershed classification proposed by Otto Pfafstetter [Pfafstetter 1989], constructing a database that covers the entire country, named *Brazilian Ottocoded Watershed*. This database represents the hydrography as a drainage network: a set of drainage points and stretches. This network is represented as a binary tree-graph, connected and acyclic,

whose edges – the drainage stretches – go from the leaves to the root, i.e., upstream to downstream.

The Brazilian drainage network is composed by 620.280 drainage points (vertices, in graph terms) and 620.279 drainage stretches (edges). Drainage points represent diverse geographic entities:

(i) a watercourse start point, usually a spring or water source;
(ii) a watercourse end point, usually a river mouth;
(iii) a stream mouth point, which flows into the sea; and
(iv) the shoreline start or end point, two reference points in the coast (one of each) that delimit the shoreline line, being the integrating elements of the entire drainage system.

The first three kinds of drainage points can be seen in Figure 1. The degree of a drainage point represents its valence, value 1 represents start or end points and value 3 represents confluences.

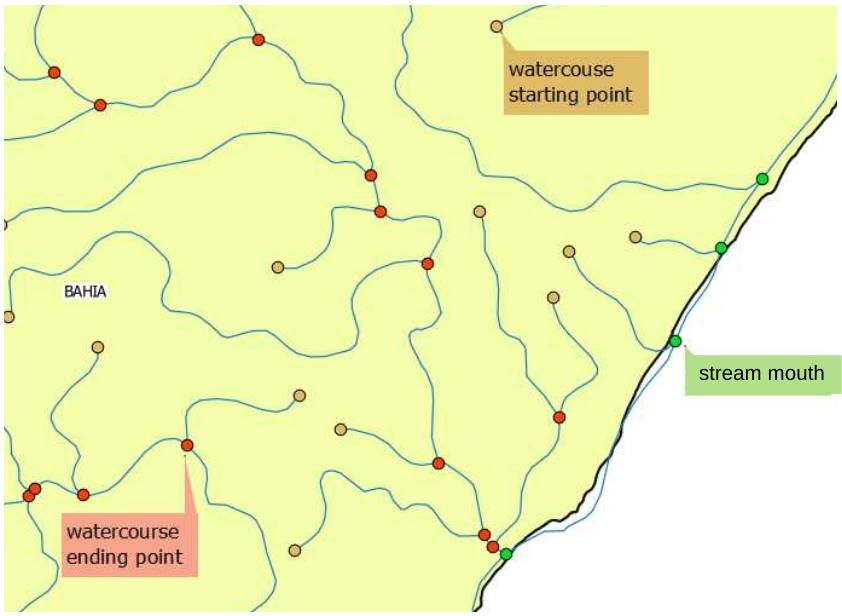


Figure 1. Kinds of Points in Drainage Network

The drainage stretches, on the other hand, represent only one geographic entity: the connection between two drainage points. Each stretch has two important attributes: (i) the hydronym, i.e., the name of the water body to which it belongs; and (ii) the hydrographic catchment area, which represents its importance in the drainage network – higher values indicate critical stretches with large areas of water catchment.

2.1.1. Cartographic Aspects

The scale of the Brazilian drainage network varies according to the cartographic mapping used as base in each geographic region, as shown in Figure 2. The Brazilian official

cartography, projected in the WGS84 Spatial Reference [1] is the start point of the mapping process. The steps of the hydrographic vectorization comprise the representation of each watercourse as a one-line entity, and identification of their crossing areas as start, end or confluence points. Digital elevation models (such as SRTM - Shuttle Radar Topography Mission [2]) are usually applied in the process of layout refine.

Research on specific watersheds is funded according to their strategic or economic importance, thus generating more detailed data in some regions. Figure 2 shows part of the drainage stretches in three scales: 1:1.000.000 (the majority of Brazilian watersheds), 1:250.000 (river Paraiba do Sul) and 1:50.000 (basin of rivers Piracicaba, Capivari and Jundiai) [3]. The latter, for instance, supplies one of Brazil's most populated regions and is the target of several studies, headed by the "PCJ Consortium". This consortium is composed by a group of cities and companies concerned about planning and financial support actions towards the recovery of water sources and raising societal awareness about the importance of watersource issues.

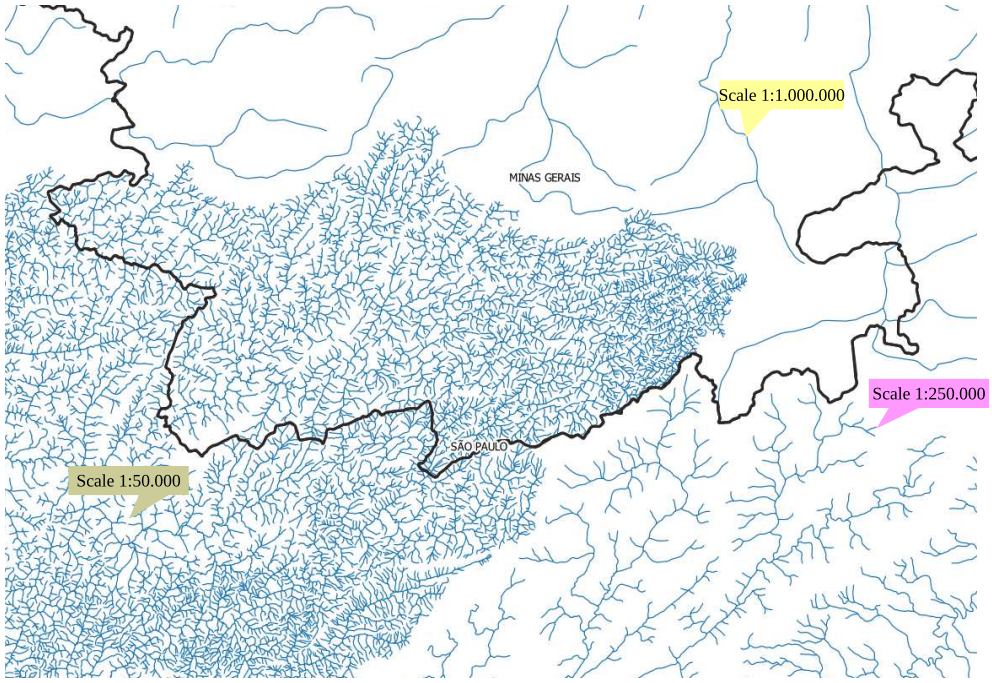


Figure 2. Different Drainage Stretch Scales in Drainage Network

The cartographic representation of the drainage network provides an important input to territorial analyses, i.e., when it is necessary to overlay the hydrographic data with other layers (using the geospatial information as the integrating component), in an attempt to understand some spatial phenomenon.

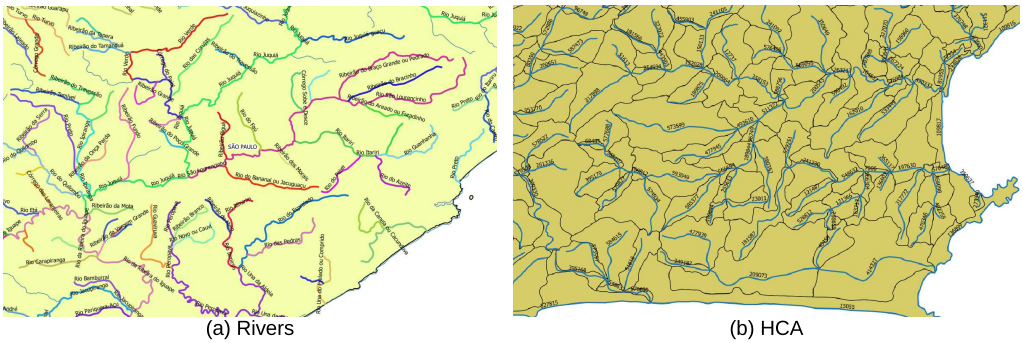
[1] spatialreference.org/ref/epsg/4326

[2] www2.jpl.nasa.gov/srtm

[3] Metadata available in: http://metadados.ana.gov.br/geonetwork/srv/pt/main.home?uuid=7bb15389-1016-4d5b-9480-5f1acdadd0f5

2.1.2. Logical Elements

There are at least four important logical elements in the Brazilian water resources database: hydronyms, hydrographic catchment areas, watersheds and main watercourses. The hydronym is an immutable attribute associated with each drainage stretch that indicates the logical element commonly known as "river". A river is composed by all drainage stretches that are connected and have the same hydronym. Figure 3 (a) partially shows the drainage network under this perspective.



(a) Rivers (b) HCA

Figure 3. (a) Rivers: continuous drainage stretches with the same hydronym and (b) HCA: drainage stretches and their hydrographic catchment area

The other three elements are computed. Every time that the drainage network is updated these elements have to be recalculated. Updates occur for instance during some cartographic refinement process (more accurate scales) or to reflect human actions (e.g., by river transposition or construction of artificial channels). Updates do not occur very often. Thus, if the algorithms that construct the network are well defined, it is possible to materialize network elements, and update them whenever necessary.

The hydrographic catchment area (HCA) is a drainage stretch attribute, represented as a polygon, that delimits the water catchment area of the stretch. This delimitation is highly influenced by relief, given it influence in the water flow. Although HCA is a geospatial attribute, as shown in Figure 3 (b), only its area is relevant in most analyzes.

Watersheds and watercourses are two correlated elements – one is used to determine the other in a recursive way. A watershed is the logical element that delimits a drainage system channel. It is the official territorial unit for the management of water resources adopted by ANA. Unlike a basin – that refers only to where the water passes through – a watershed comprises the entire area that separates different water flowing. Every watershed has a main watercourse.

ANA adopts the Otto Pfafstetter Coding System [Pfafstetter 1989](ottocode) to define the watershed division process and watercourse identification. Each digit in the ottocode embeds a context about the stream (the main river or inter-basin, for instance). The main watercourse of a watershed is a set of connected drainage stretches selected by a traversal in the sub drainage network. It is constructed by selecting, in every confluence, the stretch with the largest hydrographic catchment accumulated area upstream (from the mouth to the spring). Following the watercourse layout, the watershed can be split in a set of sub-watersheds and the ottocode allows retrieving their hierarchical relations.

A $n - level$ watershed has a code with n digits. Figure 4 illustrates one step of this methodology: 4 (a) shows the drainage network of the watershed *Rio Trombetas* and its main watercourse, which has the ottocode 454 (level 3). Figure 4 (a) shows the 9 new watersheds created (level 4) by applying recursively the same methodology. The original code 454 is held as prefix to new watershed codes. More details about this methodology can be found in [Pfafstetter 1989].

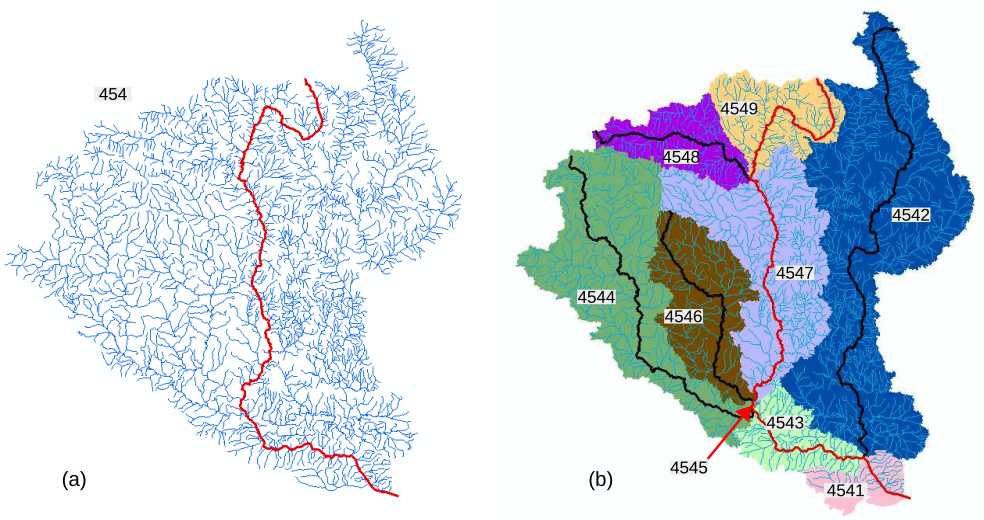


Figure 4. Otto Pfafstetter methodology

As can be seen, there are many studies that can take advantage of the network structure of this database and its logical preprocessed elements, even without considering geospatial aspects. Graph algorithms can be used, for instance, to ensure the network consistency or even to determine the main watercourse in a watershed; the latter can be found through a traversal algorithm in a subset of the drainage network, using higher HCA values as the navigation criterion.

2.2. Graph Data Management Paradigm

The graph data management paradigm is characterized by using graphs (or their generalizations) as data models and graph-based operations to express data manipulation. It is relationship driven, as opposed to the relational data model which requires the use of foreign keys and joins to infer connections between data items. Graph databases are usually adopted to represent data sets where relations among data and the data itself are at the same importance level [Angles and Gutierrez 2008]. Graph data models appeared in the 90's; nevertheless, only in the past few years have they been applied to information management systems, propelled by the rise of social networks such as Facebook and Twitter.

The formal foundation of all graph data models is based on variations on the mathematical definition of a graph. In its simplest form, a graph G is a data structure composed by a pair (V, E), where V is a finite non empty set of vertices and E is a finite set of edges connecting pairs of vertices. On top of this basic layer, several graph data structures were proposed by the database community, attempt to improve expressiveness, representing data in a better (and less ambiguous) way, such as property

graph (or attributed graph) [Rodriguez and Neubauer 2010, Robinson et al. 2013], hypernode [Levene and Loizou 1995] and RDF graph [Bonstrom et al. 2003].

Considering the edges, a graph can be directed (i.e., there is a tail and head to each edge); single relational or multi-relational (i.e., multiple relationships can exist between two vertices). The connection structure affects the traversal. An edge can have different meanings, such as attributes, hierarchies or neighborhood relations. Despite their flexibility and efficient management of heavily linked data, there is no consensual data structure and query language for graph databases.

One of the most popular graph structures is the property graph (or attributed graph) [Rodriguez and Neubauer 2010, Robinson et al. 2013]. It tries to arrange vertex and edge features in a flexible structure through key-value pairs (e.g., type, label or direction).

3. Implementation

3.1. Original Relational Database: pgHydro

The pgHydro project [4] – developed by ANA and started in 2012 – aims to implement a spatial relational database to manage the hydrographic objects that compose the Brazilian Water Resources database [Teixeira et al. 2013]. It encompasses tables, constraints and views, and a set of stored procedures to ensure data consistency and to process routine calculations. The conceptual model of pgHydro is illustrated in Figure 5.

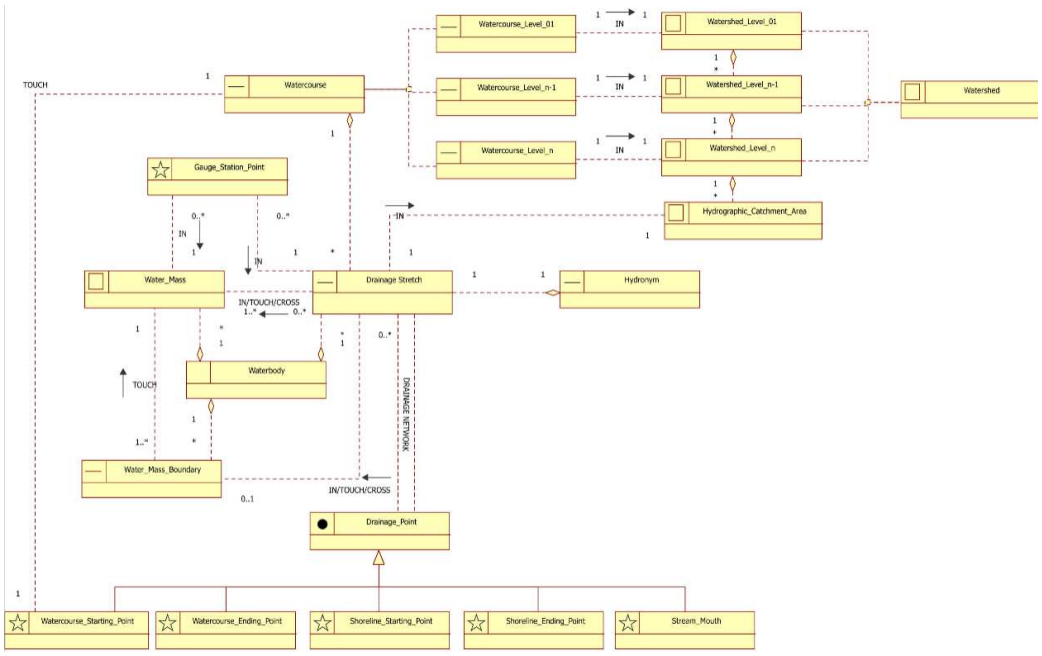


Figure 5. PgHydro Database Conceptual Model

PgHydro was implemented in PostGIS/PostgreSQL and a Python interface. PgHydro is a free and open source project and is available for companies and organizations with

[4] pghydro.org

an interest in management and decision making in water resources. More spatial analysis can be done using GIS, such as ArcGIS [5] or QuantumGIS [6].

3.2. Proposal Graph Database: HydroGraph

We have transformed ANA relational database (the drainage network) into a graph database, here denoted by G_{Hydro} (partially illustrated in Figure 6), keeping the same basic structure of vertices (the drainage points) and edges (the drainage stretches). This data model makes easier to understand the drainage network as it really is: a binary tree-graph, connected and acyclic, whose edges go from the leaves to the root.

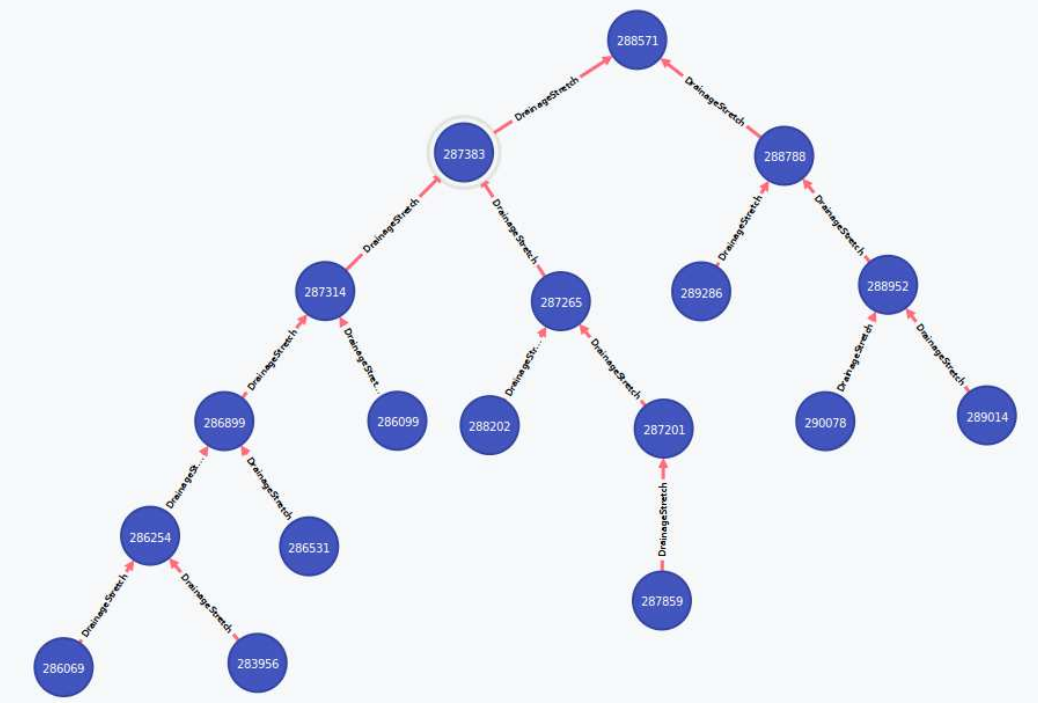


Figure 6. G_{Hydro}: Brazilian Drainage Network as a Graph Database

The graph database chosen was Neo4j [7] – a labeled property multi-graph [Robinson et al. 2013]. Every edge must have a relationship type, and there is no restrictions about the number of edges between two nodes. Both vertices and edges can have properties (key-value pairs) and index mechanism. Neo4j implements a native disk-based storage manager for graphs, a framework for graph traversals and an object-oriented API for Java. It is an open source project and it is nowadays the most popular graph database [8].

The creation and population of G_{Hydro} were done through **LOAD CSV** command – a load engine provided by Neo4j. The input could be a local or a remote classical CSV

[5] www.arcgis.com
[6] www.qgis.org
[7] neo4j.com
[8] According to DB-Engines Ranking of Graph DBMS (accessed on September, 2015) [db-engines.com/en/ranking/graph+dbms]

file – containing a header and a set of lines in which each line represents a record, and the line is a set of fields separated by comma. The CSV files were extracted from the PostgreSQL database using the **COPY** command [9]. Figure 7 shows some of the LOAD CSV commands that giving rise to G_{Hydro} (commands (i) to drainage points and (iii) to drainage stretches). Commands (ii) and (iv) ensure the integrity constraint of unique values for all the identifiers.

(i)
```
LOAD CSV WITH HEADERS FROM "file:<path>/drainage_point.csv" AS line
CREATE (p:DrainagePoint {id:toInt(line.id), valence:toInt(line.valence), geom:line.geom})
```

(ii)
```
$ CREATE CONSTRAINT ON (point:DrainagePoint) ASSERT point.id IS UNIQUE
```

(iii)
```
$ LOAD CSV WITH HEADERS FROM "file:<path>/drainage_stretch.csv" AS line match
  (source:DrainagePoint { id: toInt(line.drs_drp_pk_sourcenode)}),
  (target:DrainagePoint { id: toInt(line.drs_drp_pk_targetnode)}) CREATE (source)-
  [:DrainageStretch { id: toInt(line.drs_pk), upstreamstretch: toInt
  (line.drs_drs_pk_upstreamstretch), downstreamstretch: toInt
  (line.drs_drs_pk_downstreamstretch), distancetosea: line.drs_nu_distancetosea,
  distancetowatercourse: line.drs_nu_distancetowatercourse, waterbodyoriginal:
  line.drs_nm_waterbodyoriginal, length: line.drs_gm_length, hca:
  line.drs_hca_pk,  upstreamarea: line.drs_nu_upstreamarea, wtc: line.drs_wtc_pk,
  hdr: line.drs_hdr_pk, geom: line.st_astext, domain:line.drs_wtc_ds_domain
  }]->(target)
```

(iv)
```
$ CREATE CONSTRAINT ON (stretch:DrainageStretch) ASSERT stretch.id IS UNIQUE
```

Figure 7. LOAD CSV commands

The **LOAD CSV** command is based on Cypher syntax, the graph query language available on Neo4j [Robinson et al. 2013]. Cypher is a pattern oriented, declarative query language. It has two kinds of query structures: a read and a write query structure. The pattern representation is inspired by traditional graph representation of circles and arrows. Vertex patterns are represented in parenthesis; and edge patterns in brackets between hyphens, one of which with a right angle bracket to indicate the edge direction. For example, the expression **(a)-[r:RELATED]->(b)** is interpreted as two vertex patterns **a** and **b** and one edge pattern **r**, type **RELATED**, that starts on vertex **a** and ends in vertex **b**.

3.3. PgHydro Functions

The most important functions of pgHydro are:

1. To validate drainage network consistent;
2. To define the direction of water flow;
3. To apply Otto Pfafstetter's watershed coding system;
4. To select the set of upstream/downstream stretches;
5. To calculate the upstream hydrographic/downstream catchment area.

As can be readily seen, most of these functions can be solved applying to graph algorithms on G_{Hydro}. Execute these tasks over relational databases would require many join operations – one of the most computationally expensive processes in SQL databases. Another possibility would be to build an in-memory network representation on top of the

[9] www.postgresql.org/docs/9.2/static/sql-copy.html

relational storage model and to use APIs and programming languages. Graph databases exempt the need of intermediate models from storage to application logic layer.

Consistency tests over the drainage network concern mainly two aspects: connectivity of all stretches and the binary tree structure. In graph terms – considering G_{Hydro} implementation – we can apply the connected component analysis solution. A connected component in a graph G is a subgraph H of G in which, for each pair of vertices u and v, there is a path connecting u and v. If more than one connected component is found in G_{Hydro}, the database is inconsistent. The binary tree structure, on the other hand, is checked selecting all vertices whose degree value are different from 1 (start or end points) or 3 (confluences).

The selection of the upstream stretches can be done applying to Depth-First Search, starting on the stretch of interest and ending on the watershed root. To calculate the upstream hydrographic catchment area, we sum the HCA from each drainage stretch returned in the previous selection. The same approach can be applied to downstream stretches, using the opposite navigation direction and aggregating all subtrees.

The calculation of the Otto Pfafstetter watershed coding is a more complex task, but it is still a graph traversal. The base task is to define the main watercourse. Here, unlike the previous computations we need to establish graph traversal criteria on each node: selecting, at every confluence, the stretch with the largest HCA accumulated upstream.

Among all these functions, only the definition of water flow direction is actually a GIS task and depends on the geospacial information. This calculation involves solving equations that examine the relationship among several variables such as stream length, water depth, resistance of the surface and relief.

4. Research Challenges

There are at least three important challenges involved in our approach. The first is related to the incompleteness of graph data models. According to the classical definition, a complete data model should be composed by three main elements: (i) data structure types, (ii) operators to retrieve or derive data and (iii) integrity rules to define consistent the database states [Codd 1980]. Related work on graph data models shows that they are incomplete concerning least one of these aspects. Most of them concern only data structures – hypergraphs, RDF or property graphs. Others describe only query languages or APIs to manipulate or retrieve data. There are few attempts to discuss consistency or ACID properties over graph data models. This scenario hampers the formalizing of a complete graph data model. Besides, most implementations of graph databases do not adhere to the theoretical models.

Second, traditional Relational Database Management Systems (RDBMS) are the most mature solution to data persistence and usually the best option when strong consistency is required. Besides, there are many spatial extensions over RDBMS current used as foundation to geospatial systems and services. Therefore, in some cases there is need for the coexistence of both models – relational and graph – dividing tasks of management and analysis according to their specialties. This requires the development of hybrid architecture to enable the integration of relational and graph databases, as proposed by [Cavoto and Santanche 2015].

Finally, the task of network-driven analysis is not completely solved once the graph database is available. The graph data design (i.e., which data is represented as vertices, which is represented as edges and what kind of properties they have) can streamline or even render non-viably the extraction of topological or graph properties. There is no simple way to crossing through different designs in graph databases. This challenge is also goal of our research, as described in [Daltio and Medeiros 2014]. The idea is to specify and implement an extension of the concept of view (from relational databases) to graph database, thereby allowing managing and analyze a graph database under arbitrary perspectives. Consider this specific database, it would be possible to explore not only the drainage network, but also the network among the logical elements – rivers, watersheds and watercourses.

5. Conclusions

This paper presented our ongoing work to construct a graph database infrastructure to support analysis operations on the Brazilian Water Resources database. Our research shows the importance of graph driven analysis over the drainage network, rather than the computationally expensive process of relational databases for such analysis. It was presented the G_{Hydro} – a version of the original relational database implemented on Neo4j, composed by 620.280 drainage points (vertices) and 620.279 drainage stretches (edges).

Our research takes advantage of graph structures to model and navigate through relationships across the network and its logical elements – watersheds and watercourses. This helps analysts' work in analysis and forecast. However, given the complexity of geospatial data – mainly on big data proportions – there is still no single solution to solve all persistence, management and analysis issues. Hybrid architecture approaches seem to be the most flexible and complete choice.

Acknowledgment

Work partially financed by FAPESP/Cepid in Computational Engineering and Sciences (2013/08293-7), the Microsoft Research FAPESP Virtual Institute (NavScales project), FAPESP-PRONEX (eScience project), INCT in Web Science, and individual grants from CNPq. The authors thank Alexandre de Amorim Teixeira, geoprocessing expert of National Water Agency (ANA), for his help in clarifying watershed data analysis and relationship issues.

References

Amirian, P., Basiri, A., and Winstanley, A. (2013). Efficient online sharing of geospatial big data using nosql xml databases. In *Proceedings of the 2013 Fourth International Conference on Computing for Geospatial Research and Application*, COMGEO '13, pages 152–, Washington, DC, USA. IEEE Computer Society.

Amirian, P., Basiri, A., and Winstanley, A. (2014). Evaluation of data management systems for geospatial big data. In *Computational Science and Its Applications - ICCSA 2014*, volume 8583 of *Lecture Notes in Computer Science*, pages 678–690. Springer International Publishing.

Angles, R. and Gutierrez, C. (2008). Survey of graph database models. *ACM Comput. Surv.*, 40(1):1:1–1:39.

Bonstrom, V., Hinze, A., and Schweppe, H. (2003). Storing rdf as a graph. In *Web Congress, 2003. Proceedings. First Latin American*, pages 27–36.

Brazil (1997). Water resources federal (9.433). Federal Register [of] Federative Republic of Brazil, Executive Branch. Section 1, p. 470.

Brebbia, C. and Popov, V. (2011). *Water Resources Management VI*. WIT transactions on ecology and the environment. WIT Press.

Cavoto, P. and Santanche, A. (2015). Fishgraph: A network-driven data analysis. In *Proceedings of the 11th IEEE International Conference on eScience*, pages 1–10, Munich, Germany.

Codd, E. F. (1980). Data models in database management. *SIGMOD Rec.*, 11(2):112–114.

Daltio, J. and Medeiros, C. B. (2014). Handling multiple foci in graph databases. In Switzerland, S. I. P., editor, *Lecture Notes in Bioinformatics (LNBI) - Proceedings of 10th International Conference on Data Integration in the Life Sciences*, volume 8574, pages 58–65, Lisboa, Portugal.

Fry, B. J. (2004). *Computational Information Design*. PhD thesis, MIT. AAI0806331.

Hecht, R. and Jablonski, S. (2011). Nosql evaluation: A use case oriented survey. In *Cloud and Service Computing (CSC), 2011 International Conference on*, pages 336–341.

Hey, T., Tansley, S., and Tolle, K., editors (2009). *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, Redmond, Washington.

Levene, M. and Loizou, G. (1995). A graph-based data model and its ramifications. *IEEE Trans. Knowl. Data Eng.*, 7(5):809–823.

Pfafstetter, O. (1989). Watershed classification: coding methodology (in portuguese). National Department of Sanitation Construction. Rio de Janeiro, RJ.

Robinson, I., Webber, J., and Eifrem, E. (2013). *Graph Databases*. O'Reilly Media, Incorporated.

Rodriguez, M. A. and Neubauer, P. (2010). Constructions from dots and lines. *Bulletin of the American Society for Information Science and Technology*, 36(6):35–41.

Teixeira, A. A., Silva, A. M., Molleri, G. S. F., Ferreira, F. V., and Borelli, A. J. (2013). Pghydro - hydrographic objects in geographical database (in portugueses). In *Proceedings of the 2013 Brazilian Symposium on Water Resources*, pages 1–8.

Efficient Algorithms to Discover Flock Patterns in Trajectories*

Pedro Sena Tanaka[1], Marcos R. Vieira[2], Daniel S. Kaster[1]

[1]Department of Computer Science – University of Londrina

[2]Big Data Lab – Hitachi America, Ltd. – R&D

pedro.stanaka@gmail.com, marcos.vieira@hal.hitachi.com, dskaster@uel.br

Abstract. *The wide and increasing availability of GPS and location tracking devices has led to research advances in discovering spatiotemporal pattern in very large trajectory datasets. The overall objective of such patterns is to discover how spatial relationships between moving objects (e.g., vehicles, people, animals) behave over time. Among such patterns, one interesting and important category is flock pattern. This pattern is defined as a set of moving objects with minimum size that stay together within a maximum distance for a continuous period of time. Typical application examples are monitoring and surveillance, where they both rely on efficiently identifying groups of suspicious people/vehicles in large spatiotemporal streaming data. Previous works proposed polynomial-time algorithms to the flock pattern problem with fixed time duration. In this paper, we improve on previous work by applying a plane sweeping technique and inverted indexes. Plane sweeping technique speeds up the detection of candidate groups in a particular timestamp, while inverted indexes are employed to quickly compare candidate disks across timestamps. Using a variety of large real-world trajectory datasets we show that our proposed methods are efficient compared to state-of-the-art solution. In our experiments we show our proposed methods are up to 46x faster than previous solution.*

1. Introduction

The widespread use of location-based devices (e.g., GPS, RFID, mobile devices) and services (e.g., Swarm[1], Waze[2]) in the past few years are the main two factors of the exponential growth of dataset in the form of trajectories. A trajectory represents a sequence of recorded locations over time for a moving object. An interesting aspect of trajectories is not only that the availability of large historical spatiotemporal data is increasing in recent years, but also the rapid expansion of online services providing spatiotemporal streaming data. One example of such services is AccuTracking[3], which helps large retailer and shipping companies (e.g., United States Postal Service (USPS)) to online track large vehicle fleets around the world. Other examples are applications that provide location-based services to end-users, like Foursquare[4] and Waze, which both have millions of users reporting their location activities over time.

There is an increasing need to find more efficient algorithms that can analyze the continuous growth of historical and spatiotemporal streaming data. Nevertheless,

*This work has been supported by a CAPES scholarship.

[1]www.swarmapp.com

[2]www.waze.com

[3]www.accutracking.com

[4]www.foursquare.com

discovering spatiotemporal patterns in large volumes of spatiotemporal streaming data is a very challenging task. This is due the fact that spatiotemporal patterns are generally defined as how the spatial relationships among moving objects evolve over time. Although this analyze is computational expensive, it may reveal common behaviors among the observed moving objects in a period of time (e.g., migration patterns among wild animals, traffic patterns in road networks, suspicious activities in urban areas). In the past few years, several spatiotemporal patterns were proposed, each of which describing a different kind of behavior among moving objects. Examples of such patterns include density-based patterns, such as group [Wang et al. 2006, Li et al. 2013], swarm [Li et al. 2010] and convoys [Jeung et al. 2008] patterns, and distance-based patterns, such as flock pattern [Benkert et al. 2008, Gudmundsson and van Kreveld 2006, Vieira et al. 2009, Romero 2011], which is the subject of our work.

Flock pattern is defined by a set of minimum number of objects that are "spatially close" together for a time duration. A few proposals on flock pattern focus on discovering *maximal length* (or *duration*) flocks (e.g., [Gudmundsson and van Kreveld 2006, Arimura and Takagi 2014]), which returns the minimum set of μ objects moving enclosed by a disk with diameter ϵ for the *longest timespan*. Other works provide solutions to find flocks with *fixed time duration*, i.e., same as above but for *at least* δ time instants. Methods to discover flocks with fixed time duration can be classified as: **(a) offline methods**, e.g., [Al-Naymat et al. 2007, Romero 2011], which require the entire dataset be available beforehand in order to map/compute statistics of the entire dataset; and **(b) online methods** that report results as soon as they are discovered, thus they can deal with spatiotemporal streaming data. The online flock pattern detection is important and has a wide range of applications, from real-time monitoring suspicious activities to observing animal behaviors. The state-of-the-art work on reporting flock patterns with fixed time duration is [Vieira et al. 2009], which presents a baseline algorithm, called Basic Flock Evaluation (BFE), and also several heuristics to improve the baseline algorithm.

This paper presents novel methods to significantly extend the BFE algorithm, named: **(1) plane sweeping technique** to quickly detect candidate flock groups in a particular time instant; **(2) binary signatures** to allow reducing the number of set comparisons, and thus pruning subsets of candidates in a given time instant; and **(3) inverted index** to quickly check when a candidate disk in the current timestamp follows any partial flock in a previously time instant. To the best of our knowledge, it is the first time that our proposed methods have been applied to the flock pattern problem.

We evaluated our proposed methods using several real-world datasets with respect to the BFE baseline method. Our experiments showed our proposed methods consistently outperformed the baseline method: our methods achieved up to **46x speedup** in the experiments. An important observation is that our proposed methods could leverage the heuristics used to extend BFE [Vieira et al. 2009] to potentially achieve even higher performance improvements. We are exploring these extensions as part of our future work.

The remainder of this paper is organized as follows: Section 2 presents the basic theory on flock pattern and plane sweeping technique; Section 3 describes our new proposed algorithms to find flock patterns, which combines plane sweeping technique and inverted index; Section 4 presents extensive performance evaluation of our proposed algorithms and previous work; and Section 5 concludes this paper.

2. Background

2.1. Flock Pattern Problem

Since the work that first presented the flock pattern problem [Laube and Imfeld 2002], several others have followed addressing variations of the problem. Common to all those variations, the flock pattern is the problem of identifying all sets of trajectories that stay "close together" during a time period. This pattern enforces that there must be no pair of elements in a flock which are "farther" from each other than a given distance threshold during the flock's lifespan. The property of closeness can be depicted by a disk of a given diameter ϵ that covers all objects belonging to a flock in all timestamps during a period. This flock problem is known as a disk-based spatiotemporal pattern.

Figure 1 shows an example of flock pattern with two instances, each one involving three moving objects. One instance is formed by trajectories with identifiers $\{T_1, T_2, T_3\}$ covered by the disks labeled according to their centers $\{c_1^1, c_1^2, c_1^3\}$, where each c_i^j is the center of a disk of the flock i in the timestamp j. The second flock answer is composed by trajectories $\{T_4, T_5, T_6\}$ covered by disks centered on $\{c_2^2, c_2^3, c_2^4\}$.

It is important to note two properties in flock pattern: (1) in order to cover all trajectories, the center of the disks can freely move in the spatial domain; and (2) centers do not necessarily coincide with an object location in a specific time instant. These two properties make the problem of discovering flock patterns more challenging, as there may be infinite spatial positions to place the center of a disk at each time instance [Vieira et al. 2009].

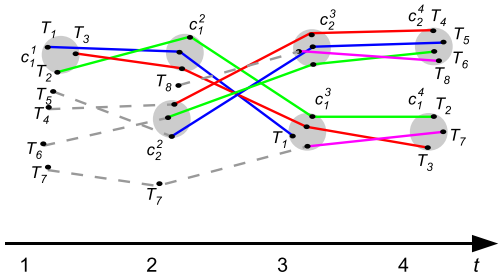


Figure 1. Example of flock pattern: flocks $\{T_1, T_2, T_3\}$ and $\{T_4, T_5, T_6\}$ in the timestamps 2 to 4 and 3 to 5, respectively (adapted from [Vieira et al. 2009]).

[Gudmundsson et al. 2004] defines flock pattern problem by a set of trajectories moving in the same direction for a *single timestamp*. Clearly, this definition cannot find more complex and interesting flock patterns that evolve over time, Other variations have followed by introducing a pattern that happen during a *time window*, which can be either of *maximal size* [Romero 2011, Geng et al. 2014, Arimura and Takagi 2014] or *fixed size* [Benkert et al. 2008, Vieira et al. 2009]. In our work we use the latter definition, where the lifespan of the pattern is fixed. Following the work of [Vieira et al. 2009], a flock pattern is formally defined as:

Definition 1 (Flock pattern) *Let $\mathcal{T}$ be a set of trajectories, $\mu > 1$ be a minimum number of trajectories ($\mu \in \mathbb{N}$), $\epsilon > 0$ be a distance threshold regarding a distance metric d ($\epsilon \in$*

$\mathbb{R}^+$) *and* $\delta > 1$ *be a minimum number of time instances* ($\delta \in \mathbb{N}$). *A* ***Flock***(μ, ϵ, δ) *pattern reports a set* $\mathcal{F}$ *containing all the flocks* f_k*, which are sets of maximal size such that: for each* $f_k \in \mathcal{F}$*, the number of trajectories is greater than or equal to* μ *and there exist* δ *consecutive timestamps* $t_j, \ldots, t_{j+\delta-1}$ *in which there is a disk of center* $c_k^{t_i}$ *and diameter* ϵ *that covers all trajectories of* $f_k^{t_i}$*, which is the flock* f_k *in time* t_i*,* $j \leq i \leq j + \delta - 1$.

Several different solutions have been proposed to the problem as in Definition 1. For instance, [Benkert et al. 2008] proposed to first map polylines representing trajectories to points in a high dimensional space, and then to search for flocks in the mapped space. [Al-Naymat et al. 2007] employs dimensionality reduction techniques to report approximate answers. These two approaches both have the drawback of being limited to analyze only historical data. [Vieira et al. 2009] is the first work to propose algorithms for detecting flock patterns in streaming data using only spatiotemporal search techniques. In this work, all proposed algorithms were based in a common algorithm, called BFE (Basic Flock Evaluation). They are considered the state-of-art solution for reporting flock patterns with fixed time duration.

Similar to Definition 1 is the *Maximal Length (or Duration) Flocks*, which does not have a defined time window (i.e., no definition of δ parameter). [Turdukulov et al. 2014, Romero 2011] proposed an algorithm to report maximal duration flocks based on a frequent pattern mining approach [Agrawal and Srikant 1994]. This approach transforms the input historical trajectory dataset to a transactional one, which is then used in the LCM algorithm (Linear time Closed itemset Miner) [Uno et al. 2005]. A second class of algorithm is Flock Pattern Miner (FPM) [Arimura and Takagi 2014, Geng et al. 2014], which uses a depth-first search (DFS) to find all-maximal duration flocks. These algorithms were the first to address the problem of enumerating the maximal duration flocks in polynomial delay. The approach used by this class of algorithms is to use DFS to check all time instants for a particular trajectory. Nonetheless, all previous approaches described here are limited to historical datasets. On the other hand, our proposed methods can handle both historical and spatiotemporal streaming data.

2.2. Basic Flock Evaluation Algorithm

As aforementioned, there may have infinite possible spatial locations to place disk centers in a time instant. This makes the task of finding candidate disks very challenging. In order to overcome this problem, [Vieira et al. 2009] introduced a theorem that reduces the search space to finding candidate flock disks as follows:

Theorem 1 *If for a given time instance* t_i *there exists a point in the space* $c_k^{t_i}$ *such that,* $\forall T_j \in f_k$*,* $d(p_j^{t_i}, c_k^{t_i}) \leqslant \epsilon/2$*, then there exists another point in the space* $c_k'^{t_i}$ *such that* $d(p_j^{t_i}, c_k'^{t_i}) \leqslant \epsilon/2$ *and there are at least trajectories* $T_a \in f_k$ *and* $T_b \in f_k$ *such that* $\forall T_j \in \{T_a, T_b\}, d(p_j^{t_i}, c_k'^{t_i}) = \epsilon/2$.

Theorem 1 states that if there is a disk with center $c_k^{t_i}$ and diameter ϵ that covers all trajectories in a flock f_k, and then there is another disk with different center $c_k'^{t_i}$ that also covers all the trajectories of f_k. This theorem affects considerably the search space for flock patterns as it limits the locations inside the spatial domain where it is necessary to search for flocks. For a dataset of $|\mathcal{T}|$ trajectories, there are $|\mathcal{T}|^2$ possible combinations of pairs of points in a time instant. For each pair there are (at most) two disks with radius $\epsilon/2$ that have these two points in their circumferences.

Based on Theorem 1 it was proposed the Basic Flock Evaluation (BFE) algorithm. This algorithm is the simplest amongst five presented in [Vieira et al. 2009]. The other four algorithms enhance the BFE algorithm with different heuristics to, potentially, speed up their running time. In summary, the BFE algorithm operates in the following three steps:

(1) Find flock disks: this step generates a set of candidate disks, where each disk defines a flock pattern. For each time instant, this step searches for pairs of points that qualify to generate disks. Since this step has a running time of $O(n^2)$ (i.e., all possible pairs of point combinations at any given time instance), BFE employs a grid-based index to speed up the candidate disks and flock detection. The grid-based index is based on fixed-size cells of ϵ distance. Each object in the dataset is mapped to one cell in the index based on the objects' locations, thus the index size and performance depend on the spatial distribution of the dataset. In the search phase, each cell $g_{x,y}$ is evaluated to find pair of points that are within ϵ distant to each other. To find pairs of points, each point in cell $g_{x,y}$ is matched to every other point located in cell $g_{x,y}$, as well as points in the nine adjacent cells to cell $g_{x,y}$ (i.e., cells that may have points within ϵ distance to the point being evaluated). Each pair of points within ϵ distance is used to compute (at most) two disks whose circumferences intersect exactly in the pair of points. Afterwards, the algorithm simply counts the number of points within the disks, and then filtering out disks with less than μ entities;

(2) Filter candidate disks: since the first step finds *all* flock disks in a particular time instant, the second step is to keep only disks containing *maximal* sets of trajectories. A naive approach to select *maximal* sets is to compare every possible pair of disks. However, this approach has running time quadratic to the number of disks, without considering computational expensive set intersection operations. In order to avoid set intersection operations, BFE only compares pair of disks that intersects each other in the space domain;

(3) Set join between consecutive timestamps: the result from the second step is a set of flocks for one time instant. Thus, in order to find pattern flock with a time duration, the result set has to be "merged" with results from previous time instant. Differently from previous step, joining sets between consecutive timestamps cannot employ topological relations to reduce the search space. Therefore, this last step is computationally expensive when joining very large sets. After the joining phase, the results that could not be joined are discarded, and the ones with lifespan of δ are reported as flocks. The current "active" flocks (valid until the current timestamp) are maintained and further used in the next iteration (step 1 for the next timestamp).

2.3. Plane Sweeping Technique

Plane sweep technique was first introduced to detect intersection between line segments in the plane [Shamos and Hoey 1976]. Since then, this technique has proven to be important to reduce computational complexity in large variety of problems, mainly in computational geometry area. The main idea of plane sweeping is to use a "sweeping line" (generally vertical) throughout the plane, i.e., scanning the plane from left-to-right in x-axis. The "sweeping" process continues until a condition is met (e.g., line intersects with a point). Whenever this event happens, then geometric operations are performed on the points that prompted the event. Note that the operations are performed on a reduced set of points

closer to the sweeping line. This process of sweeping ends when all dataset points are swept by the line.

There are several examples in which the plane sweeping technique allows reducing the time complexity of algorithms. A classic example is the problem of finding closest-pairs in a dataset, which can be solved with a naive approach in $O(n^2)$, but in $O(n.\log(n))$ with plane sweeping [Hinrichs et al. 1988]. In our work we use plane sweeping technique in a similar fashion to the closest-pair problem, as detailed in the next section.

3. Proposed Methods

We now describe our proposed methods that employ plane sweeping technique and binary signatures to improve the search for disks on a specific timestamp. We then detail how to further improve our proposed methods by employing inverted indexes in the spatiotemporal join phase to reduce the number of set intersection operations.

3.1. Plane Sweep-based Disk Detection

As previously described the BFE algorithm (and its extensions) first constructs a grid-based index, and then generates candidate disks for each timestamp. This process of building and searching the index can be time consuming. Thus, to reduce this cost we propose a new approach based on plane sweeping to find flock disks without index construction. Our proposed approach is described in Algorithm 1.

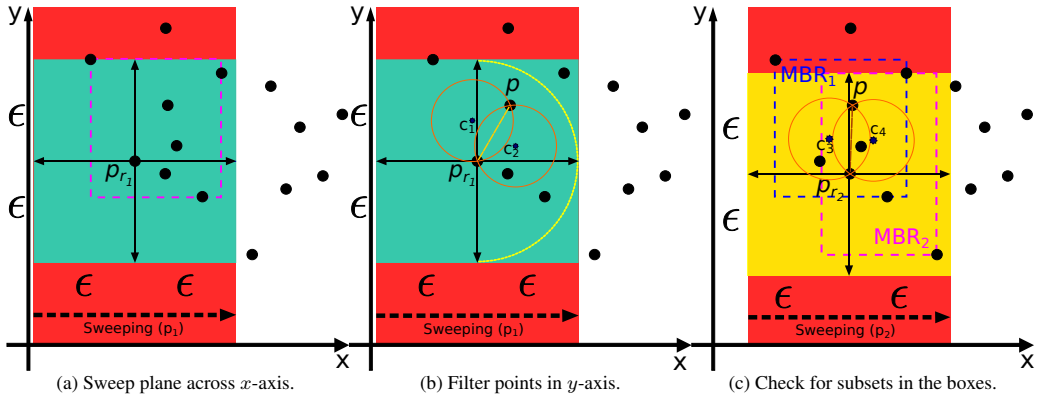
Algorithm 1: Find candidate disks with plane sweeping technique

Input: $\mathcal{T}[t_i]$: positions in timestamp t_i, sorted by x-axis values, ϵ: flock diameter, μ: minimum size of flock
Output: $\mathcal{C}$: candidate disks for timestamp t_i, $\mathcal{B}$: active boxes in timestamp t_i

```
1  C ← ∅, B ← ∅
2  foreach p_r ∈ T[t_i] do // analyze elements in increasing x-values
3      P ← ∅ // list of elements of current box defined by p_r
4      foreach p_s ∈ T[t_i] : | p_s.x − p_r.x | ⩽ ε do // test only elements inside 2ε x-band
5          if |p_s.y − p_r.y| ⩽ ε then // check if p_s is inside 2ε y-band
6              P ← P ∪ p_s // add element p_s to box
7      foreach p ∈ P : p.x ⩾ p_r.x do (// elements inside right half of box
8          if dist(p_r, p) ⩽ ε then // calculate pair distance
9              let {c_1, c_2} be disks defined by {p_r, p} and radius ε/2
10             foreach c ∈ {c_1, c_2} do
11                 if |c ∩ P| ⩾ μ then // check the number of entries in disk
12                     C ← C ∪ c // add c to candidate disks
13                     B ← B ∪ box(p_r) // add box to active boxes
14 return C, B
```

Algorithm 1 first sweeps the plane (from left to right in x-axis) using a band of size 2ϵ along the x-axis centered at a point p_r (red box on Figure 2(a)). The algorithm selects all the points inside the band that are in the range $[p_r.y - \epsilon, p_r.y + \epsilon]$ (blue box on Figure 2). These steps are presented in lines 2–6 of Algorithm 1.

After selecting the points in the $2\epsilon \times 2\epsilon$ box defined by p_r, we then check for pairs of points that qualify for new flock disks (refer to Theorem 1). Thus, we generate disks defined by p_r and any point p inside the right half of box such that the distance between p_r and p is at most ϵ (yellow-dotted semicircle in Figure 2(b)). Points in the left half of box were checked in previous steps. If a candidate disk contains at least μ entities inside it, then the underlying entity set is reported as a candidate set and the box is set



(a) Sweep plane across x-axis. (b) Filter points in y-axis. (c) Check for subsets in the boxes.

Figure 2. Steps needed to find disks in one timestamp.

active in the timestamp. Every active box is represented through the Minimum Bounding Rectangle (MBR) enclosing its elements (dotted rectangle in Figure 2(a)). These last steps are illustrated in Figure 2(b) and represented by lines 7–13 of Algorithm 1.

3.2. Signature-based Candidate Set Filtering

The next step after the candidate sets of a particular timestamp were detected is to check for disks that are subsets or supersets. Similar to BFE algorithm, we are interested in finding only *maximal* sets of flocks. The BFE algorithm uses only spatial properties of disks to accelerate the maximal set detection. Here we employ a different approach: we use spatial properties of boxes (instead of disks as in BFE) and binary signatures to prune subsets without executing (expensive) set intersection operations.

The process of filtering out candidate sets is shown on Algorithm 2. The spatial relationship between boxes is given by their MBRs. Each box has information about all candidate sets that belong to it. The step of filtering check boxes that hold at least one candidate set. Algorithm 2 begins by iterating on each active box (i.e., a box that has a candidate set) in the *current* timestamp, and checks if there is intersection between the MBR of a box and the MBRs of boxes near it. In this case, it is necessary to check whether there is duplicate or subsets between these boxes (Figure 2(c)).

Before performing the set intersection operation, we propose to apply a second filtering step using binary signatures (lines 10–18 of Algorithm 2). As entities get inserted in a candidate set, a set of hash functions are used to generate a signature for it. Starting from a two-byte zero signature (0000 0000 0000 0000), the resulting signature is computed by executing the hash functions in sequence for every identifier in the set. Each hash function maps an object identifier to a position inside the signature (*bucket*): it sets a number of 1-bit according to a given identifier. The primitives of performing subset queries using binary signatures are described in details in [Goel and Gupta 2010]. In this work a set of Bloom filters is used to represent subsets of a universe, and then these filters, which essentially are binary vectors, are used to check for subsets amongst the sets. It is well-known that Bloom filters can generate false-positives, but most importantly, no false-negatives. This is the reason that, after we check one disk is subset of another using binary signatures, we still need to verify using set intersection whether the results is a false-negative (lines 13 and 16 of Algorithm 2).

Algorithm 2: Filter out disks which are subsets

```
1  Algorithm FilterCandidates(B)
       Input: B: active boxes of timestamp t_i, sorted by x-axis values
       Output: C: final set of disks for timestamp t_i
2      C ← ∅
3      for j ← 0 to j ⩽ |B| do
4          for k ← j + 1 to k ⩽ |B| do
5              if IntersectsWith(B[j], B[k]) then
6                  foreach c ∈ B[j].disks do
7                      C ← InsertDisk(C, c)
8              else // No intersection.
9                  break

10 Procedure InsertDisk(C, c)
       Input: C: set of disks, c: new disk
11     foreach d ∈ C do
12         if c.sign ∧ d.sign = c.sign then // c can be a subset of d
13             if |d ∩ c| = |c| then // Remove chance of false-positive
14                 return C // No need to insert c
15         else if c.sign ∧ d.sign = d.sign then // d can be a subset of c
16             if |c ∩ d| = |d| then // Remove chance of false-positive
17                 C ← C \ d // Remove d
18     return C ∪ c
```

Figure 3 illustrates the process of filtering through binary signatures. Suppose that a disk contains the objects identified by $\{3, 5, 7\}$, and then the disk signature is identical to the third line of Figure 3. H_1 and H_2 refer, respectively, to the SpookyHash[5] and MurMurHash[6], which are fast hashes implementations and with good non-linearity (measured by avalanche criterion[7]. Note that, since the signature size is very small, some collisions may happen between the hash functions (bits represented in purple in Figure 3). Now, suppose we want to avoid performing a set intersection operation to determine if a disk is subset/superset of another disk in case this is surely false. In order to achieve this, we apply a logical **and** operation between the two signatures from the disks. If the result of the operation is equal to one of the operands, and then this operand may be a subset of the other. Otherwise, we can surely say that no disk is a superset of the other. For instance, consider two sets containing the objects identified by $\{3, 5, 7, 8\}$ and $\{3, 5, 7\}$. The signature of set $\{3, 5, 7, 8\}$ is given by the last line of Figure 3. The result of $\{3, 5, 7\}.binSignature \wedge \{3, 5, 7, 8\}.binSignature$ is 1000 0011 0100 0100, which is exactly the signature for $\{3, 5, 7\}$, indicates it may be a subset of $\{3, 5, 7, 8\}$. As mentioned before, this approach is subject to false positives, thus it is necessary to perform a set intersection operation as a post-processing step. Nevertheless, this step should eliminate many false-negatives depending on the hash functions chosen.

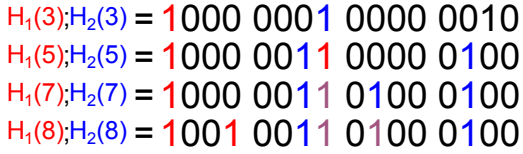


Figure 3. Process of generating binary signatures.

[5] burtleburtle.net/bob/hash/spooky.html
[6] Murmurhash 2.0: sites.google.com/site/murmurhash
[7] floodyberry.com/noncryptohashzoo

3.3. Inverted Index-based Join Between Sets of Consecutive Timestamps

After all disks for a given time instant are found, we then need to join them with others from the previous time instant. One straightforward way to join disks is to process all disks from one time instant with the other time instant, and then check for the joining condition (i.e., if two disks have at least μ objects in common). However, this process is computational expensive since we have to do for all timestamps and all disks found in the previous step. Instead, here we propose the use of inverted indexes to speed up the step of joining disks across two consecutive time instants.

Inverted index is a well-known method employed to index documents and then efficiently search for terms in the index [Zobel and Moffat 2006]. Usually, an inverted index has a list of keys that are the common terms appearing in all the documents. Each item, in turn, has a set of document identifiers where the term represented by that item appeared. In our particular problem, inverted index can be employed to search only disks from previous time instant (t_{i-1}) that have at least μ object in common with the disk being processed of current time instant (t_i). When a new disk from t_i is processed, we use the set of OID's that belongs to the disk as the query elements Q_s to the inverted index. The query condition is that a document (or disk) is returned if it has at least μ terms (or objects) in common with the query set. Now, suppose we have $n \in \mathbb{N}$ disks in t_{i-1} and $m \in \mathbb{N}$ in t_i, and the average number of objects in each disk is $l; l \in \mathbb{N}, l < \mu$. If we have to compare all disks from t_i with the ones in t_{i-1}, and then we have a time complexity of $O(n.m.l)$. However, our approach can drastically reduce this time complexity on most cases, as shown in the experimental evaluation.

4. Experimental Evaluation

In order to verify the efficiency of our proposed methods, we performed an extensive experimental evaluation with several real-world spatiotemporal datasets. The datasets used in our experiments are: *Trucks*, *Buses*, *Cars*, and *Caribous*. *Trucks* dataset has 112,203 GPS locations generated by 276 moving trucks, while *Buses* has 66,096 locations generated by 145 buses, both of them were collected in the metropolitan area of Athens, Greece[8]. *Cars* contains 134,264 locations collected from 183 private cars moving in Copenhagen, Denmark[9]. *Caribous* is generated from the migration movements of 43 caribous in northwest states of Canada, with a total of 15,796 locations.

We used the BFE as baseline approach, which is the original implementation described in [Vieira et al. 2009]. As for our proposed methods, we tested four new methods that contain combinations of our three proposed techniques. The main idea is to evaluate which technique works better in which conditions. In summary, we performed experiments with the following five methods:

1. **BFE:** The original BFE algorithm;
2. **BFI:** BFE with inverted index to join disks;
3. **PSW:** BFE with plane sweeping method (grid-based index is absented);
4. **PSB:** PSW (as described above) with binary signatures to accelerate the step of finding disks with same time instants;
5. **PSI:** PSB (as described above) with inverted index to prune flock disks.

[8] chorochronos.datastories.org/

[9] www.daisy.aau.dk

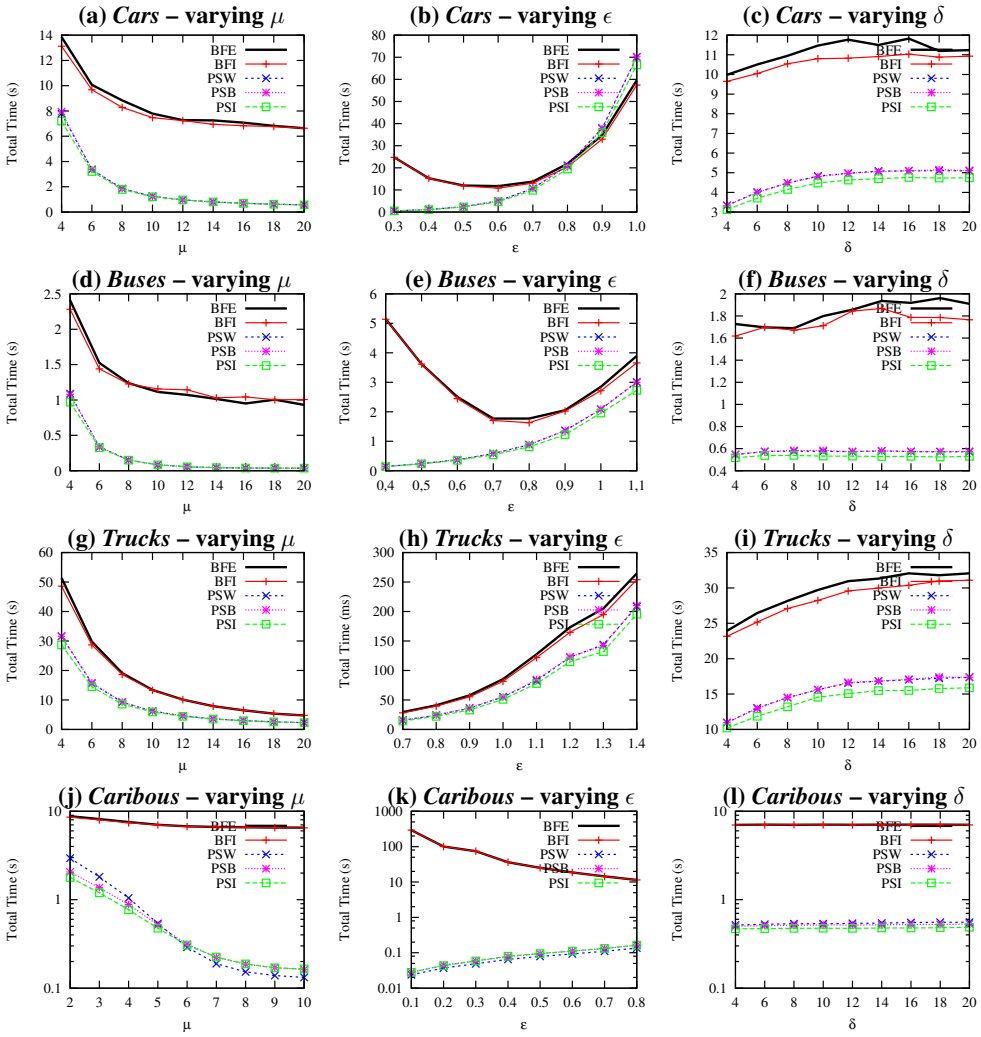


Figure 4. Experiments with *Cars* (1st row), *Buses* (2nd row), *Trucks* (3rd row) and *Caribous* (4th row) datasets when varying μ (cardinality), ϵ (distance diameter) and δ (time duration).

All tested methods were implemented in C++ and compiled with GCC v4.9. We ran our experiments in an Intel Core 2 Quad@2.83GHz CPU, 4GB@1333MHz RAM and 500GB@3Gb/s HDD. In order to test different scenarios, all three flock pattern parameters were tested with different range of values. Figure 4 shows the experimental results when increasing values for the parameters μ, ϵ and δ, respectively, in the first, second and third column. All plots show the total running time (in seconds) needed to evaluate the *entire* dataset. In order to enhance the differences in performance of all methods, all three plots for the *Caribous* are in logarithmic scale.

The first observation from the results is that as we increase μ, or decrease ϵ or δ, the running time required to report flock patterns decreases. This behavior is expected, since the parameters of flock patterns became more selective and, thus, the bookkeeping costs related to maintain candidate sets. In general, the best results were from methods

that employ the plane sweeping technique. This is due the fact the BFE method needs to build a grid-based index before actually starts to search for disks. Moreover, the grid-based index is highly dependent on the spatial data distribution, e.g., when the data is skewed not only the index is bigger but it also takes more time to be constructed.

In general all the extensions proposed in this paper perform better than the baseline approach BFE. The only exception for the previous statement is for large values of disks diameter (ϵ) for *Cars* dataset (Figure 4(b)). The performance of our proposed methods was worse than BFE and BFI. The reason for this result is that, when the disks diameter increases the number of cells in the grid-based index used by BFE is reduced, which makes its performance to be more efficient. Further analyzing the results, it is possible to conclude that neither inverted index nor binary signatures presented substantial performance gain. The best performance gain between these two methods is with the inverted index (see plots in the third column of Figure 4, when varying δ). The behavior for the binary signatures is expected, since the BFE method is very optimized to use spatial properties of disks, and thus avoid performing unnecessary set intersection operations [Vieira et al. 2009]. This is the main reason we did not include the results for using binary signatures with BFE (e.g., the gain in performance is (almost) none).

In the *Caribous* dataset we see that when increasing μ values, the use of the plane sweeping technique has a bigger impact on performance than in the other datasets. This is due the fact this dataset contains trajectories that are highly correlated to each other (i.e., they are clustered in space over time). In other words, when μ parameter is increased the total number of boxes are drastically reduced, since those boxes probably will not have enough μ entries. Compared to the BFE method, its performance is stable mainly because it has to create indexes and it cannot avoid the process of generating and checking disks.

5. Conclusion

The broad usage of location devices has aroused the interest in studying patterns that portray collaborative behaviors in spatiotemporal data (e.g., groups, swarm, flocks). Related works have focused on the problem of finding maximal duration flocks or providing off-line solutions to report flock patterns with *fixed time* duration. Nevertheless, several real-world applications demand online solutions to this problem. In this work we proposed the application of plane sweeping technique to help accelerate the process of finding disks defining flocks in one time instant. Next, we employed the use of binary signatures and inverted index to reduce the amount of set operations necessary to detect flocks. Our extensive experimental evaluation using real-world datasets show considerable speedups compared to the baseline approach. Our plane sweeping approach achieved the best results regardless varying the three parameters that define flock patterns: μ, ϵ and δ. Our findings have a major impact not only for our proposed approach, but also for other methods that share common primitives similar to BFE (e.g., [Romero 2011]).

References

[Agrawal and Srikant 1994] Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proc. of the VLDB*, pages 487–499.

[Al-Naymat et al. 2007] Al-Naymat, G., Chawla, S., and Gudmundsson, J. (2007). Dimensionality reduction for long duration and complex spatio-temporal queries. In *Proc. of the ACM SAC*, pages 393–397.

[Arimura and Takagi 2014] Arimura, H. and Takagi, T. (2014). Finding All Maximal Duration Flock Patterns in High-dimensional Trajectories. *Manuscript, DCS, IST, Hokkaido University, Apr*.

[Benkert et al. 2008] Benkert, M., Gudmundsson, J., Hübner, F., and Wolle, T. (2008). Reporting flock patterns. *Computational Geometry*, 41(3):111–125.

[Geng et al. 2014] Geng, X., Takagi, T., Arimura, H., and Uno, T. (2014). Enumeration of complete set of flock patterns in trajectories. In *Proc. ACM SIGSPATIAL Workshop on GeoStreaming*, pages 53–61.

[Goel and Gupta 2010] Goel, A. and Gupta, P. (2010). Small subset queries and bloom filters using ternary associative memories, with applications. In *ACM SIGMETRICS Perform. Eval. Rev.*, pages 143–154.

[Gudmundsson and van Kreveld 2006] Gudmundsson, J. and van Kreveld, M. (2006). Computing longest duration flocks in trajectory data. In *Proc. of the ACM GIS*, pages 35–42.

[Gudmundsson et al. 2004] Gudmundsson, J., van Kreveld, M., and Speckmann, B. (2004). Efficient Detection of Motion Patterns in Spatio-temporal Data Sets. In *Proc. of the ACM GIS*, pages 250–257.

[Hinrichs et al. 1988] Hinrichs, K., Nievergelt, J., and Schorn, P. (1988). Plane-sweep solves the closest pair problem elegantly. *Inf. Process. Lett.*, 26(5):255–261.

[Jeung et al. 2008] Jeung, H., Yiu, M. L., Zhou, X., Jensen, C. S., and Shen, H. T. (2008). Discovery of convoys in trajectory databases. *Proc. of the VLDB Endowment*, 1(1):1068–1080.

[Laube and Imfeld 2002] Laube, P. and Imfeld, S. (2002). Analyzing relative motion within groups of trackable moving point objects. In *GIScience*, volume 2478 of *LNCS*, pages 132–144.

[Li et al. 2013] Li, X., Ceikute, V., Jensen, C. S., and Tan, K.-L. (2013). Effective Online Group Discovery in Trajectory Databases. *IEEE Trans. Knowl. Data Eng.*, 25(12):2752–2766.

[Li et al. 2010] Li, Z., Ding, B., Han, J., and Kays, R. (2010). Swarm: Mining relaxed temporal moving object clusters. *Proc. of the VLDB Endowment*, 3(1-2):723–734.

[Romero 2011] Romero, A. (2011). Mining moving flock patterns in large spatio-temporal datasets using a frequent pattern mining approach. Master's thesis, University of Twente.

[Shamos and Hoey 1976] Shamos, M. I. and Hoey, D. (1976). Geometric intersection problems. In *Proc. of the IEEE FOCS*, pages 208–215.

[Turdukulov et al. 2014] Turdukulov, U. et al. (2014). Visual mining of moving flock patterns in large spatio-temporal data sets using a frequent pattern approach. *Int'l J. of Geog. Inf. Sci.*, 28(10):2013–2029.

[Uno et al. 2005] Uno, T., Kiyomi, M., and Arimura, H. (2005). LCM Ver.3: Collaboration of Array, Bitmap and Prefix Tree for Frequent Itemset Mining. In *Proc. of the Int'l Workshop on OSDM*, pages 77–86.

[Vieira et al. 2009] Vieira, M. R., Bakalov, P., and Tsotras, V. J. (2009). On-line discovery of flock patterns in spatio-temporal data. In *Proc. of the ACM SIGSPATIAL*, pages 286–295.

[Wang et al. 2006] Wang, Y., Lim, E. P., and Hwang, S. Y. (2006). Efficient mining of group patterns from user movement data. *Data Knowl. Eng.*, 57(3):240–282.

[Zobel and Moffat 2006] Zobel, J. and Moffat, A. (2006). Inverted files for text search engines. *ACM Comput. Surv.*, 38(2):6.

Inferring Relationships from Trajectory Data

Areli Andreia dos Santos[1] **, Andre Salvaro Furtado**[1]**,**
Luis Otavio Alvares[1]**, Nikos Pelekis**[2]**, Vania Bogorny**[1]

[1]Informatics and Statistics Department – Universidade Federal de Santa Catarina (UFSC)
Florianopolis – SC – Brazil

[2]Department of Informatics – University of Piraeus (UPRC)
Piraeus – Attica – Greece

Abstract. *Devices like smart phones and GPS navigators are very popular nowadays. These equipments can save the location of an object with an associated time, generating a new kind of data, called trajectories of moving objects. With these data it is possible to discover several interesting patterns, among which is the interaction between individuals, allowing to infer their relationship. This work addresses the discovery of relationship degree between moving objects based on their encounters. To calculate the relationship degree we propose different measures based on frequency, duration, and area of the encounters. These measures were evaluated in experiments with a running example and real trajectory data, and show that the method correctly infers relationships.*

1. Introduction and Motivation

The price reduction of mobile devices such as GPS and mobile phones, as well as advances in satellite and wireless sensor technologies, has enabled a significant increase in the use of these mechanisms. These devices allow recording people's movement. Accordingly, any individual who carries a mobile device, while moving, generates a trace, in which each time point corresponds to a location in space. This trace is called trajectory of the moving object. There are several works dealing with such data, as the one that describes avoidance of trajectories [Alvares et al. 2011], chasing [de Lucca Siqueira and Bogorny 2011], outliers [de Aquino et al. 2013], flocks, leadership, convergence, and encounter [Laube et al. 2005].

Although there are numerous works on patterns in trajectories, only a few address the encounter/meeting patterns, and even less works infer friendship relationships from trajectories or consider encounters for relationship inference. The first work to define encounter was [Laube et al. 2005], where encounters happen when a set of objects have points in a specific given radius. [Gudmundsson et al. 2007] defined the encounter pattern with a minimum number of entities inside a given radius. [Bak et al. 2012] proposed an algorithm to detect encounters between two trajectories, where all points that are close in space and time are connected forming a line. The work of Bak focuses on visual analysis of encounters. The most formal definition of encounter is given by [Dodge et al. 2008], which defines encounter as a convergence where objects arrive at a place at the same time.

Existing works do only define the concept of encounter, and have neither go deeper in the encounter pattern analysis nor use them for relationship inference among moving objects. The inference of relationships is an important issue for several application domains. In biology, for example, we can discover how much time the pandas A and B

stayed together in the last summer, and which areas they visited together and alone. For investigative applications, we can verify the total time that a group of individuals stayed close in the last month, and how much time two objects Y and Z of this group stayed together and which areas they visited with a bigger group of objects. We strongly believe that the relationship of objects is directly related to amount of time they spend together. For instance, a married couple stays more time together than a couple that is dating, and the couples, in turn, stay more time together than a couple of friends.

To infer relationships from encounter patterns is not a trivial task. Let us consider the example shown in Figure 1, where Louis met Marie and then both met John, and after they met Susan, at day 5th May. For this day we have four encounters, each one with duration of one hour. The first encounter is between Louis and Marie (from 9 to 10). The second encounter is between Louis, Marie and John (from 10 to 11). The third encounter is between Louis and Marie alone (from 11 to 12), and the fourth encounter is between Louis, Marie and Susan (from 12 to 13). This example illustrates three important things when we reason about measuring the relationship degree based on encounters: the number of objects present on each encounter, the frequency, and the duration. In this example, still considering 5th of May, the duration of the encounter is higher for Louis and Marie, corresponding to 4 hours in total. So we can consider that Louis and Marie have a stronger relationship degree among each other than with John and Susan. When considering the encounter of Louis and Marie alone, their encounter has 2 hours of duration. Considering the whole time they stayed together at 5th of May, the duration is 4 hours. This example shows that it is very complex to analyze encounters and relationships between moving objects, and that we must analyze every different encounter and with all different objects.

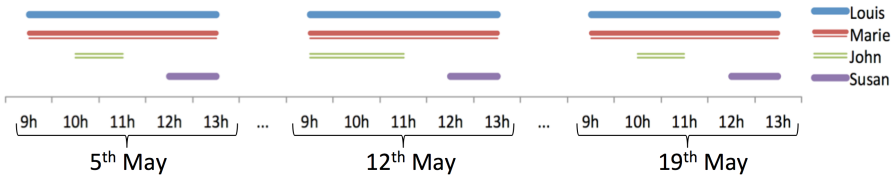


Figure 1. Temporal Representation of encounters between the same individuals

In this paper we propose a new definition of level-wise encounter patterns. From these encounters we propose an algorithm to infer relationships, called MORE (Moving Objects Relationship inference from Encounters). The main contribution of our approach is the definition of level-wise encounter patterns and the inference of moving object relationships, considering their encounters, the frequency of the encounters, the duration, and the encounter area. In this work we go one step further to existing approaches, which do only detect encounters from trajectory data and do not infer objects relationships, making the following contributions: (i) define encounter patterns as well as the encounter area from trajectory data; (ii) define different measures to compute the relationship of individuals based on the frequency, the duration, and the area of their encounters; (iii) propose an efficient algorithm to infer the relationship degree of individuals from their trajectories.

The rest of the paper is organized as follows. In section 2 we summarize the related work. In Section 3 we present the main definitions and the algorithm $MORE$. In

Section 4 we present preliminary experiments to evaluate the algorithm, and finally, the conclusions are presented on Section 5.

2. Related Work

In this section we summarize the works that define encounters and flocks, and since there are only a few works that infer relationships from GPS trajectories, we present some methods that infer relationships from other types of spatio-temporal data, as phone logs/calls and social networks.

2.1. Encounters

Among the few existing studies in the literature that directly address the encounter between trajectories we can highlight the work of [Laube et al. 2005], which defines a set of patterns considering geographic and temporal aspects. Between these patterns are flock and encounter. The flock pattern can be detected from the REMO (RElative Motion Pattern) matrix, which relates the time and the direction of objects movements. When objects are moving in the same direction at the same time, and within a certain area, they present a flock behavior. The encounter pattern cannot be detected from the REMO matrix, because objects come from different directions until they meet. To detect encounters, [Laube et al. 2005] proposed a division of the space in cells of a given size, and the trajectories that intersect the same cells in a similar time have an encounter pattern.

Gudmundsson in [Gudmundsson et al. 2007] defines encounter as a minimal number of objects m that "stay" inside an area of size r at a certain time. A flock is defined as a set of m objects that "move" inside a spatio-temporal cylinder of size r for a certain time, and objects may leave or enter the flock. In this work both encounter and flock are considered as different patterns, while in our proposal we assume that objects that are together, either stopped or moving, are having an encounter. Indeed, while in [Gudmundsson et al. 2007] objects enter and leave a flock, we create a *different* encounter every time the group changes.

Bak in [Bak et al. 2012] proposed an algorithm to detect encounters focused on visual analysis. The main idea of the algorithm is to connect with a line two points of different objects that are close in space and time. The user can vary the space and time threshold, visualizing the connected dots and evaluating the results.

In our work, we consider as an encounter the whole time that the objects remain together, does not matter if they are stopped or moving. Usually, a whole encounter is separated in two patterns to fit in the definitions of encounter and flock. For example, a walk with a friend followed by a visit to a restaurant is separated in two parts: the walk is a flock pattern and the staying in a restaurant, an encounter.

The works about encounters and flocks focused only on detecting the pattern, not on the inference of relationships between level-wise objects, which is the objective of our work. Indeed, the area of the encounters and flocks is not defined.

2.2. Relationship Inference from GSM and Social Network

Before describing these works we must highlight that the task of relationship inference in social network is only an affine topic, and it is more trivial since much information

about friendship is available in the data. In the domain of social network analysis there are several works that try to infer the relationship between pairs of objects. Some of these works use information of phone logs and calls like [Eagle et al. 2009], others use only social network data where the user makes a check-in at some place, or shares a geo-tagged picture [Crandall et al. 2010], [Pham et al. 2013] and [Wang et al. 2014].

A friendship network structure based on mobile phone data was proposed in [Eagle et al. 2009]. A group of students answered if another member of the group is a friend or not. In a first step, each pair of objects is classified as "Reciprocal Friends", "Non-Reciprocal Friends" and "Reciprocal Non-Friends". Then, the friendship relations are inferred using proximity information of the mobile phones, based on phone logs, calls, and Bluetooth connections. Finally, the proximity of these persons is evaluated based on the day of the week and the kind of place the objects met. The output is a friendship network, where each node is an individual and the edge corresponds to a score based on two factors: the first one is based on the proximity at place of work, and the second is based on the proximity outside the work environment.

Crandall, in [Crandall et al. 2010], proposed a model to infer social ties between pairs of users, where spatio-temporal co-occurrences are detected based on shared photos of the site Flickr. First, the space is divided in cells, then if both users shared a photo within t days in the same cell, a co-occurrence is detected. Finally, the number of different cells visited by the same pair is counted. Based on the number of different co-locations between the pair of objects a probabilistic model is used to compute the friendship.

A model to infer social strength called EBM (Entropy-Based Model) was proposed by Pham in [Pham et al. 2013]. It analyses information from the social network Gowalla, that allows the users to make check-ins when they are in a known place. First, the co-occurrences are computed for each pair of objects. The co-occurrences are coincident check-ins of the objects, considering space and time. Finally, the relationship is calculated considering the entropy of the places and the frequency of the co-occurrences.

Another model to infer relationship strength between pairs of users based on check-ins from location-based social networks was recently proposed by Wang in [Wang et al. 2014]. This approach is based on personal, global, and temporal factors. The personal factor considers an individual user's probability to visit a certain location. The global factor captures the popularity of a location to the general public. The temporal factor considers the time gaps between consecutive meeting events.

Even though check-ins from social networks are useful to infer relationships between people, when we want to measure the relationship among animals or people which do not use social networks (e.g. house arrest criminals), GPS trajectories are the most appropriate data.

2.3. Relationships in Trajectories of Moving Objects

Only a few works use GPS data to infer relationships, like [Brilhante et al. 2012], that infers relationships among places. Because GPS data are more complex and lack in relationship information, [Brilhante et al. 2012] use summarized trajectories, i.e., stops or stay points to reduce the complexity. In our approach we use raw trajectories, first computing encounters/flocks and from these encounters propose a method to infer relationships.

Brilhante [Brilhante et al. 2012] proposed a methodology to discover communities of interesting places, using as input the trajectories of moving objects and known POIs (Point of Interest). The first step is to detect stops at the given POIs. If a group of trajectories has short stops on a pair of POIs, these POIs are connected. This work does not infer relationships among users, only between points of interest.

In summary, none of the related work infer a relationship degree based on encounters of groups of multiple objects, and none of them consider area, duration and frequency of encounter as measures to determine the relationship degree.

3. Main Definitions and the Proposed Algorithm

In this section we first present the main concepts to define an encounter pattern, the encounter area, and the relationship degree between objects (Section 3.1). In Section 3.2 we present the algorithm $MORE$ (Moving Objects Relationship inference from on Encounters), an algorithm to infer the relationship degree between a group of objects.

3.1. Main Concepts

We start our definitions with the well known concepts of *point*, *trajectory*, and *subtrajectory*, inspired by the definitions presented in [de Lucca Siqueira and Bogorny 2011] and [Bogorny et al. 2014].

Definition 1 *Point. A point p is a tuple (x,y,t), where x and y are geographic coordinates that represent a position in space and t is the timestamp in which the point was collected.*

A trajectory is an ordered list of points that correspond to the position of the object in space at a time, as presented in Definition 2.

Definition 2 *Trajectory. A trajectory $T_o = \langle p_1, p_2, p_3, ..., p_n \rangle$ is an ordered list, where o is the object identifier, $p_j = (x_j, y_j, t_j)$ and $t_1 < t_2 < t_3 < ... < t_n$.*

It is well known that several trajectory patterns do not hold for an entire trajectory, but only in a trajectory part. For the encounter pattern it is not different. Two trajectories may not be together during all their life, but only in parts of their movements, and this parts are called subtrajectories. The definition of subtrajectory is given in Definition 3.

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 $p_i \in T$ and $k + l \leq n$.*

[Laube et al. 2005], [Gudmundsson et al. 2007] and [Dodge et al. 2008] define an encounter as a set of objects that are close in space and time. Our definition for encounter is a bit different, where we compute encounters of every object in the database in relation to other objects which stay close in space and time, for a minimal amount for time. For example, in Figure 1, we compute the encounters between Louis and Marie, between Louis, Marie and John, and between Louis, Marie and Susan. We do not require a minimal number of objects, since we are, in fact, interested in all possible encounters of any number of two or more objects.

In this work we do not distinguish stationary and moving encounters, since we want to know when two or more objects stay together. Definition 4 presents the concept of encounter. For the sake of simplicity, in the following we will restrict the definitions for two moving objects, but note that the generalization to more than two objects is straightforward.

Definition 4 *Encounter. Let $T_1 = \langle p_1, p_2, p_3, ..., p_n \rangle$ and $T_2 = \langle q_1, q_2, q_3, ..., q_m \rangle$ be two trajectories. Let $s_1 = \langle p_a, p_{a+1}, ..., p_{a+u} \rangle$ and $s_2 = \langle q_b, q_{b+1}, ..., p_{b+v} \rangle$ be two subtrajetories of T_1 and T_2, respectively. T_1 and T_2 have an encounter at two maximal subtrajectories s_1 and s_2 w.r.t a spatial threshold Δ_d, a temporal threshold Δ_t and a minimum duration $minTime$ IIF the following conditions hold:*

- $\forall p_i \in s_1, \exists q_j \in s_2 |$ *spatialDist(p_i,q_j)* $< \Delta_d \wedge$ *temporalDist(p_i,q_j)* $< \Delta_t$
- $\forall q_j \in s_2, \exists p_i \in s_1 |$ *spatialDist(q_j,p_i)* $< \Delta_d \wedge$ *temporalDist(q_j,p_i)* $< \Delta_t$
- *(min(p_{a+u}.t,q_{b+v}.t) - max(p_a.t,q_b.t)) > minTime*

where the functions $spatialDist()$ and $temporalDist()$ compute, respectively, the Euclidean distance and the temporal distance between the points p_i and q_j.

To the best of our knowledge, there are no works in the literature which take into account the area where two or more objects have an encounter to infer relationships. In this work, as we want to measure the relationship degree for all combinations (sets) of objects, just assuming that objects should stay together (close) in space for a certain amount of time is not enough. However, if we consider meetings at different places, we reduce the fact that they meet only by coincidence. For instance, objects that leave in a nearby area and work at a nearby place (e.g. in a shopping center), will be detected as having encounters, even if these encounters represent a coincidence. Although the coincidence may generate several encounters, we cannot ignore them, because for several applications, mainly for security, objects that came close in space and time may have a contact, and this contact should be considered for relationship inference. However, by considering that these objects that have frequent encounters at the same places (e.g. nearby homes and working place) also have encounters in different areas, the probability of coincidence will be reduced, and the confidence that these objects know each other will increase the relationship degree.

We define encounter area as the union of the subtrajectories of all trajectories involved in the encounter. More formally,

Definition 5 *Encounter area. Let e be an encounter between the subtrajectories s_1 and s_2, w.r.t Δ_d, Δ_t and $minTime$. The encounter area a_e is given by the formula:*

$$a_e = buffer(makeLine(s_1), \Delta_d/2) \cup buffer(makeLine(s_2), \Delta_d/2) \quad (1)$$

where $makeline()$ is a function that transforms a set of points of a subtrajectory s in a line and $buffer()$ is a function that builds a polygon of size $\Delta_d/2$ around s.

To take into account the area of an encounter, hereafter we refer to encounter as a tuple e = (O, $beginTime$, $endTime$, a), where O is the set of objects involved in the encounter, $beginTime$ and $endTime$ are, respectively, the begin and end time of the encounter, and a is the spatial area of the encounter.

To define the relationship degree between two or more objects based on their encounters, we consider three main criteria: the *frequency* of the encounters, the *duration*, and the *different areas* where the encounters take place.

The frequency reveals how many times two or more objects meet.

Definition 6 *Frequency-Based Relationship Degree. Let $DB = e_1, e_2, ..., e_n$ be a set of encounters w.r.t. Δ_d, Δ_t and $minTime$, of all sets of moving objects in a trajectory database. Let $E(o_i, o_j)$ denote the set of all encounters between any objects o_i and o_j. The frequency- based relationship degree between a pair (o_1, o_2) is given by:*

$$R_f(o_1, o_2) = \frac{|E(o_1, o_2)|}{max(|E(o_i, o_j)|)} \tag{2}$$

where $|X|$ represents the cardinality of X.

The duration of an encounter tells how much time two objects spend together. We assume that the higher the duration of an encounter is, the higher will be the relationship between the objects. The duration based relationship degree is given in Definition 7.

Definition 7 *Duration-Based Relationship Degree. Let $DB = e_1, e_2, ..., e_n$ be a set of encounters w.r.t. Δ_d, Δ_t and $minTime$, of all sets of moving objects in a trajectory database. Let $E(o_i, o_j)$ denote the set of all encounters between o_1 and o_2. The duration based relationship degree between o_1 and o_2 is:*

$$R_d(o_1, o_2) = \frac{\sum_{z=1}^{z=|E(o_1,o_2)|}(endTime_z - beginTime_z)}{max\left(\sum_{z=1}^{z=|E(o_i,o_j)|}(endTime_z - beginTime_z)\right)} \tag{3}$$

The first idea when we think about defining a relationship degree between a group of several objects is to use the duration and frequency. However, in downtown areas we can find different objects close to each other in space and time. In these cases, people who get the same bus everyday together could have a strong relationship, when they not even know each other. To reduce this problem we define that a group that has encounters in different areas has a higher relationship degree. The more different areas two or more objects have an encounter, the higher is the probability that the objects know each other. Therefore, we define the area-based relationship according to Definition 8.

Definition 8 *Area-Based Relationship Degree. Let $DB = e_1, e_2, ..., e_n$ be a set of encounters w.r.t. Δ_d, Δ_t and $minTime$, of all sets of moving objects in a trajectory database. Let $E(o_1, o_2)$ denote the set of all encounters between o_1 and o_2. Let $A(o_1, o_2) = \{a_1, a_2, ..., a_r\}|a_1 \cap a_2 \cap ... \cap = \emptyset$ be the set of different encounter areas between o_1 and o_2. The area-based relationship degree between o_1 and o_2 is:*

$$R_a(o_1, o_2) = \frac{|A(o_1, o_2)|}{max(|A(o_i, o_j)|)} \tag{4}$$

Considering duration, frequency, and encounter area, the final relationship degree between two or more objects is computed by the sum of the degrees, as shown in Definition 9.

Definition 9 *Relationship Degree. Let $DB = e_1, e_2, ..., e_n$ be a set of encounters w.r.t. Δ_d, Δ_t and $minTime$, of all sets of moving objects in a trajectory database. Let $E(o_i, o_j)$ denote the set of all encounters between o_1 and o_2. The final relationship degree between o_1 and o_2 is computed as:*

$$R(o_1, o_2) = (R_f(o_1, o_2) + R_d(o_1, o_2) + R_a(o_1, o_2))/3 \tag{5}$$

In the following section we present an algorithm to infer the relationship degree between moving objects, called MORE (Moving Objects Relationship inference from Encounters)

3.2. MORE (Moving Objects Relationship inference from Encounters)

The input of the algorithm $MORE$, shown in Listing 3, is: a set of trajectories T, the time tolerance Δ_t, the distance threshold Δ_d, and the minimum time for detecting an encounter $minTime$. The output is a list with the relationship degree of the moving objects R.

Listing 1. MORE Algorithm

```
1  Algorithm MORE
2  Input: T //set of trajectories
3       Δt //Time Tolerance
4       Δd //Distance Threshold
5       minTime //Minimum Encounter Tolerance
6  Output: R //list of objects, with their relationship degree
7
8  E = BeingTogether(T, Δt, Δd, minTime)
9  E = computeArea(E, Δd)
10 encountersPerObjects = retrieveEncountersPerObjects(E)
11 max_f = getMaxFrequence(encountersPerObjects)
12 max_d = getMaxDuration(encountersPerObjects)
13 max_a = getMaxDiffAreasCount(encountersPerObjects)
14 for each set of objects o ∈ encountersPerObjects.values do
15    R_f = getFrequencyOf(encountersPerObjects.get(o)) / max_f
16    R_d = sumDurationsOf(encountersPerObjects.get(o)) / max_d
17    R_a = getDistinctsAreasOf(encountersPerObjects.get(o)) / max_a
18    result.R = (R_f + R_d + R_a)/3
19    R.put(e.O, result)
20 end for
21 return R
```

The first step is to compute the encounters (according to Definition 4), using the function $BeingTogether()$ (line 8). Since we have the set of encounters E, it is possible to compute the encounter area (according to Definition 5), using the function $computeArea()$ (line 9). Two encounter areas are considered as only one if their intersection is higher than 75%.

In order to compute the frequency, the duration, and the distinct areas of the encounters, the algorithm transforms the set of encounters E into a list that contains each different group of objects with their respective encounters (line 10). Figure 2 shows this transformation for the encounters between Marie ($o1$), Louis ($o2$), John ($o3$) and Susan ($o4$), according to Figure 1. Since we have the groups of objects and their respective

id	O	area	beginTime	endTime
e_1	{1,2}	a_1	5th May 9h	5th May 10h
e_2	{1,2,3}	a_2	5th May 10h	5th May 11h
e_3	{1,2}	a_3	5th May 11h	5th May 12h
e_4	{1,2,4}	a_4	5th May 12h	5th May 13h
e_5	{1,2,3}	a_5	12th May 9h	12th May 11h
e_6	{1,2}	a_6	12th May 11h	12th May 12h
e_7	{1,2,4}	a_7	12th May 12h	12th May 13h
e_8	{1,2}	a_8	19th May 9h	19th May 10h
e_9	{1,2,3}	a_9	19th May 10h	19th May 11h
e_{10}	{1,2}	a_{10}	19th May 11h	19th May 12h
e_{11}	{1,2,4}	a_{11}	19th May 12h	19th May 13h

oids	encounters
{1,2}	{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}}
{1,2,3}	{e_2, e_5, e_9}
{1,2,4}	{e_4, e_7, e_{11}}

Figure 2. (left) list of encounters and (right) encounters grouped by objects

encounters, the next step of the algorithm is to get the maximum values for the frequency (line 11), duration (line 12), and the distinct area of the encounters (line 13). Then, for

each group of objects (line 14) the algorithm computes the frequency (line 15), the duration (line 16) and the area of the encounters (line 17) according to Definitions 6, 7 and 8, respectively. Finally, the relationship degree is computed (line 18) and added to a list (line 19).

The complexity of the algorithm $MORE$ is given by the complexity of the function to transform the encounter list $retrieveEncountersPerObjects$ plus the number of different sets of objects, represented by the variable $encountersPerObjects$. In the iteration are computed frequency, duration, and different areas for each different group of objects, having a cost of $O * m$, where O is the size of the transformed list and m is the number of encounters.

4. Preliminary Experiments

In this section we present two preliminary experiments: a running example and a real trajectory dataset where the encounters are known.

4.1. MORE applied on a Running Example

For a better understanding of the relationship inference we apply the $MORE$ algorithm over the example shown in Figure 1. The output is illustrated on Table 1, which is sorted in descending order by R, forming a relationship degree rank. Table 1 shows frequency, duration, and number of different encounter areas between Marie, Louis, John and Susan.

Table 1. Relationship Measures for the running example

O	$frequency$	$duration$	$area$	R_f	R_d	R_a	R
{Marie, Louis}	3	12	3	1	1	1	1
{Marie, Louis, John}	3	4	3	1	0.333	1	0.778
{Marie, Louis, Susan}	3	3	3	1	0.25	1	0.75

According to Figure 1, Marie and Louis stayed together from 9h to 13h on each one of the three days, therefore the frequency of this group is equal to 3 and the duration is 12h. Marie, Louis and Susan, stayed together for one hour on 5th of May (from 12h to 13h), one hour on 12th of May and another hour on 19th of May, hence the total duration is 3 hours. Assuming that, in this example, the objects met at three different places, each group has 3 different encounter areas.

Observing the example in Figure 1, the group with the highest relationship during the period between 5th of May to 19th of May was Marie and Louis. Notice that the relationship degree between the objects Marie, Louis and John is a bit higher than between the objects Marie, Louis and Susan. This is because there was one encounter (Figure 1) at 12th of May between objects Marie, Louis, John with duration of 2 hours (from 9h to 11h), while all others had the same one hour of duration.

4.2. UFSC dataset

On August 13th, 2015, a group of eleven volunteers walked around the UFSC campus to simulate encounters, generating a dataset with 29329 points. The seven simulated encounters are represented by rectangles in different colors in Table 2 and are visually represented in Figure 3. The participants received a smartphone, a map, and the instructions to visit three different places during the time described in Table 2 (from 17:40 to 18:20). The visits, with around 10 minutes of duration each, happened at different places

as shown in Table 2. For instance objects 1, 2, and 3 have only one encounter alone, at $Place_1$ and in the path to $Place_7$, where they met object 4, generating a new encounter with four objects. Objects 7 and 8, for instance, have two different encounters at $Place_3$ (from 17:40 to 17:50 and 18:10 to 18:20). The detected encounters are illustrated in Figure 3. All simulated encounters were correctly detected by the algorithm considering Δ_t = 10 s, Δ_d = 15 m, $minTime$ = 5 min.

Table 2. Encounters at UFSC

oid	**1st place** [17:40, 17:50]	⇒	**2nd place** [17:55, 18:05]	⇒	**3rd place** [18:10, 18:20]
1	$Place_1$	⇒	$Place_7$	⇒	$Place_9$
2	$Place_1$	⇒	$Place_7$	⇒	$Place_9$
3	$Place_1$	⇒	$Place_7$	⇒	$Place_9$
4	$Place_6$	⇒	$Place_7$	⇒	$Place_9$
5	$Place_2$	⇒	$Place_4$	⇒	$Place_9$
6	$Place_2$	⇒	$Place_4$	⇒	$Place_9$
7	$Place_3$	⇒	$Place_8$	⇒	$Place_3$
8	$Place_3$	⇒	$Place_6$	⇒	$Place_3$
9	$Place_4$	⇒	$Place_9$	⇒	$Place_6$
10	$Place_4$	⇒	$Place_9$	⇒	$Place_5$
11	$Place_5$	⇒	$Place_3$	⇒	$Place_4$

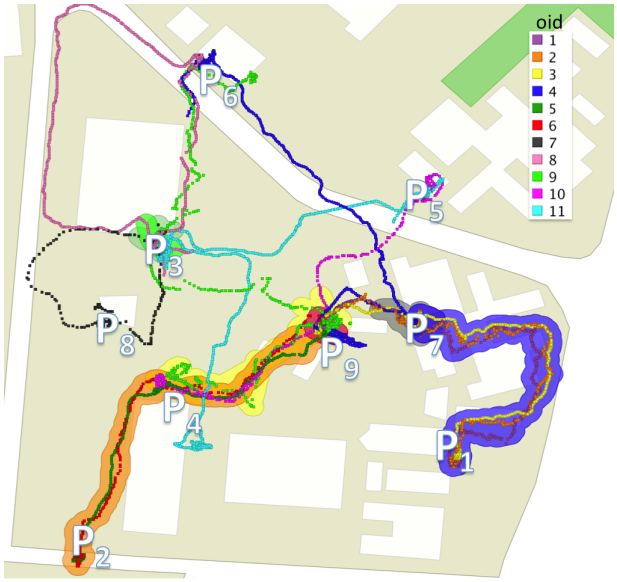


Figure 3. Trajectories and Encounters at UFSC

Table 3. Relationship Degrees between objects at UFSC

O	$frequency$	$duration$	$area$	R_f	R_d	R_a	R
{1,2,3}	1	35.1	3	0.5	0.992	1	0.831
{5,6}	1	35.4	2	0.5	1	0.667	0.722
{7,8}	2	24.3	1	1	0.687	0.333	0.673
{1,2,3,4}	1	19.9	2	0.5	0.562	0.667	0.576
{10,9}	1	27.5	1	0.5	0.779	0.333	0.537
{1,2,3,4,5,6}	1	8.4	1	0.5	0.238	0.333	0.357

Table 3 shows the relationship degree for each group of objects, and is sorted in descending order of R. As can be seen in Table 3, objects 1, 2 and 3 have the highest relationship degree (0.831). Those objects stayed together during all the experiment and their encounters happened at three different areas. Objects 5 and 6 also stayed together

during all the experiment. However, they are the second in the rank because their encounters happened at only two different areas. As can be seen in Table 2, while objects 5 and 6 visited $Place_2$ and $Place_4$ they stayed together alone, generating only one encounter in this case.

Although objects 7 and 8 have the highest frequency of the experiment, they are only the third group of the rank. This happens because although they had two encounters, they were at the same place ($Place_3$), so having only one encounter area.

The group of objects 1, 2, 3, 4, 5, and 6 had the lowest relationship degree (0.357), because this group had only one encounter, and this encounter had the lowest duration of the experiment. Object 11 (see Table 2) did not have any encounter during the experiment, so it has no relationship.

This experiment showed one of the key advantages of the $MORE$ algorithm over related work: the inference of a relationship degree between moving objects. Independently of the size of the group, if they had an encounter, their relationship degree will be calculated. Even in a small dataset it is possible to understand the relevance of this information. The knowledge of relationship allows the understanding of which objects are the most related inside larger groups.

Further experiments, with larger datasets and comparing the $MORE$ algorithm with related work will be conducted in an extended version of this paper.

5. Conclusion and Future Work

In the last two decades, there was a popularization of different GPS-enabled devices, that allow recording the moving objects location. Consequently, there was an increase in the amount of mobility data generated from these devices. To best of our knowledge, none of existing works in the literature proposed the inference of relationship degree between multiple objects based on their encounter patterns extracted from trajectories.

In this work we proposed $MORE$, a new algorithm to compute the relationship degree of a level-wise moving objects. This algorithm is based on the new proposed definitions of encounter and encounter area. The algorithm considers the encounter duration, the frequency of encounters, and the different encounter areas. $MORE$ presents conceptual advantages over related work, such as the possibility to infer the relationship degree between multiple objects. We evaluated the proposed method with a running example and performed an experimental study with real trajectory data in a simulated scenario, where the encounters were known. The results of the experiment showed that our method was able to identify relationships between pairs and groups of objects.

In the proposed approach we use raw trajectory data without considering semantic information. However, as future ongoing work, we are investigating new measures to ensure the value of a relationship degree among a group of objects.

6. Acknowledgments

The work on which this article is based was supported by EU project FP7-PEOPLE SEEK (No. 295179). We would also thank CAPES and CNPQ for the partial support of this research and all the volunteers for their help in the data collection.

References

Alvares, L. O., Loy, A. M., Renso, C., and Bogorny, V. (2011). An algorithm to identify avoidance behavior in moving object trajectories. *Journal of the Brazilian Computer Society*, 17(3):193–203.

Bak, P., Marder, M., Harary, S., Yaeli, A., and Ship, H. J. (2012). Scalable detection of spatiotemporal encounters in historical movement data. In *Computer Graphics Forum*, volume 31, pages 915–924. Wiley Online Library.

Bogorny, V., Renso, C., Aquino, A. R., Lucca Siqueira, F., and Alvares, L. O. (2014). Constant–a conceptual data model for semantic trajectories of moving objects. *Transactions in GIS*, 18(1):66–88.

Brilhante, I. R., Berlingerio, M., Trasarti, R., Renso, C., de Macedo, J. A. F., and Casanova, M. A. (2012). Cometogether: discovering communities of places in mobility data. In *Mobile Data Management (MDM), 2012 IEEE 13th International Conference on*, pages 268–273. IEEE.

Crandall, D. J., Backstrom, L., Cosley, D., Suri, S., Huttenlocher, D., and Kleinberg, J. (2010). Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Sciences*, 107(52):22436–22441.

de Aquino, A. R., Alvares, L. O., Renso, C., and Bogorny, V. (2013). Towards semantic trajectory outlier detection. In *GeoInfo*, pages 115–126.

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

Dodge, S., Weibel, R., and Lautenschütz, A.-K. (2008). Towards a taxonomy of movement patterns. *Information visualization*, 7(3-4):240–252.

Eagle, N., Pentland, A., and Lazer, D. (2009). Inferring friendship network structure by using mobile phone data. In *Proceedings of the National Academy of Sciences (PNAS)*, pages 15274–15278. PNAS.

Gudmundsson, J., van Kreveld, M., and Speckmann, B. (2007). Efficient detection of patterns in 2d trajectories of moving points. *Geoinformatica*, 11(2):195–215.

Laube, P., van Kreveld, M., and Imfeld, S. (2005). Finding remo - detecting relative motion patterns in geospatial lifelines. In *Computer Graphics Forum*, volume 31, pages 915–924.

Pham, H., Shahabi, C., and Liu, Y. (2013). Ebm: an entropy-based model to infer social strength from spatiotemporal data. In *Proceedings of the 2013 international conference on Management of data*, pages 265–276. ACM.

Wang, H., Li, Z., and Lee, W.-C. (2014). Pgt: Measuring mobility relationship using personal, global and temporal factors. In *Data Mining, 2014. ICDM'14. International Conference on*, pages 570–579. IEEE.

Prediction of Destinations and Routes in Urban Trips with Automated Identification of Place Types and Stay Points

Francisco Dantas N. Neto[1,2], Cláudio de Souza Baptista[1], Cláudio E. C. Campelo[1]

[1]Systems and Computing Department, Federal University of Campina Grande (UFCG)
Zip Code 58.429-900 – Campina Grande – PB – Brazil

[2]Federal Institute of Education, Science and Technology of Paraíba (IFPB)
Zip Code 58.432-300 – Campina Grande – PB – Brazil.

dantas.nobre@ifpb.edu.br, {baptista,campelo}@dsc.ufcg.edu.br

Abstract. *Predicting the destination and the route that someone is likely to take is useful for various purposes, such as to prevent people from going through congested routes. Most of existing approaches to this prediction problem only consider geographic patterns within their models, although this appears to be not enough for creating a robust predictor. This paper proposes an approach to improving the task of predicting route and destination which makes use of further semantic information associated with destinations and routes, apart from location patterns. Our model does not require user's active interaction and is able to automatically identify stay points (i.e., places users visit) and type of places. We evaluated our model with real world data collected from users' smartphones and obtained promising results.*

1. Introduction

Thanks to the possibility of gathering geographic position with current *smartphones* (since they have built-in GPS device embedded), the number of location-aware systems have increased considerably. There are several benefits that location-aware systems can provide to users for helping their daily routine, such as indicating Points of Interest (POIs) around their current location, by considering their preferences, and then displaying on a map the best path to reach a POI selected by the user.

Systems that provide such location-based services are commercially used nowadays. However, there are many other topics related to location-aware systems that are still under investigation. Of particular interest in this work is the task of automatically discovering the type of place a user is located (such as "home" or "work") [Alvares et al. 2007]; and the prediction of routes and destinations [Simmons et al. 2006].

A previous step of automatically discovering the type of a place, it is the task of identifying *stay points*, i.e., the geographic region where a user is stopped. This geographic region is composed by a centroid point and a radius that associate a GPS point to the stay point. The importance of identifying *stay points* is related to the possibility of analyzing the behavior of a user in visiting specific places, enabling to understand the semantics of a place to a certain user. Identify stay points can be achieved using spatial clustering techniques, such as *DBSCAN*, *OPTICS* or *K-Means* [Aggarwal and Reddy 2013]. K-Means algorithm is a *distance-based* method, i.e., it is necessary a parameter that defines the number of clusters previously. DBSCAN and OPTICS algorithms are *density-based* methods, where the number of clusters is identified on demand [Tork

2012]. Thus, for the model proposed by this work, *density-based* methods are more suitable, since we do not know previously how many stay points might be created.

When the stay points are identified, the next step is to identify the type of places. The task of automatically discovering the type of a place may be facilitated by the use of APIs services which return a POI given a certain location, such as *Google Places*[1] and *Foursquare*[2]. However, this task is not trivial as it seems, since a user might be at a restaurant for *leisure*, and another might be at the same restaurant for *working*. Thus, this discernment is one of the challenging that needs to be addressed. Therefore, gathering further information, such as day of the week and duration that a user spent in a place, can help understand the relationship between users and locations.

At the moment that a vehicle starts to move, predicting the destination and route is useful in several contexts. For instance, by having this information, along with real-time traffic data, a computational system could suggest the user to take a detour, because the route commonly used is jammed. Furthermore, it is also possible to suggest POIs, such as a bakery or a market located along the route to the user's destination. A remarkable feature of predicting is that both points of interest and less jammed routes could be suggested without an active user participation in the process, which could improve the daily use of this kind of system. Thus, by just starting the trip, the system should be capable of predicting the destination and the path.

There are two important observations related to user displacements that we empirically have identified:

- **People's daily driving follows a pattern**. Workday activities often include trips to work, to home, or to a leisure activity (e.g., beach, restaurant). Even in vacation times, people use to repeat certain trips, such as visits to some Shopping Center. Furthermore, for a significant number of daily trips, it can be observed repetitions of the paths traveled. For example, people tend to always take the same route to go from home to work. Thus, if the place of departure and the destination of a user are known, it is possible to estimate the path the user is likely to take.
- **Trips occurs at similar times**: Besides the repetition of trips (i.e., origin, destination and route), it can be observed a pattern of times and the days of the week in which the trips occur. Hence, it is reasonable to assume that certain contextual information, such as day of the week and time, could be useful variables to improve the destination prediction.

Given a set of GPS points, our model identifies the stay points, infer the type of place that a user is located, partition all the trips which users travelled, associate each GPS point to a road segment, which is called *map matching* technique [Quddus and Noland 2006], and predicts the destination and the remaining path. For route and destination prediction, we propose Prediction by Partial Matching (PPM) technique as the core of our model, which was originally conceived for the data compression context. Summarizing, the main contributions of the model proposed by this work are as follow:

[1] https://developers.google.com/places/

[2] https://developer.foursquare.com/

- Identify stay points and type of places automatically, with support of APIs services, such as *Google Places* and *Foursquare*;
- Enrich trajectories semantically, by the use of contextual information, improving the task of understand the behavior of users' displacement;
- Predict real-time route and destination as soon as user starts a trip, apart from the type of place prediction.

The experiment carried out in this work was focused on individuals who use the vehicle for personal transportations only, instead of those who use it as work, as is the case of taxi drivers. The route database was created from real displacements, captured by using an application installed into *smartphones* of the participants of this work. From the GPS points collected, information such as day of the week and departure time related to the points was also obtained, for helping to improve the model.

The rest of this paper is organized as follows. Section 2 addresses related works. Section 3 presents our developed approach. The collected data and experimental results are discussed in section 4. Finally, the last section concludes the paper and discusses future work.

2. Related Work

There are many works that can be found in the literature concerning the problem of short-term and long-term prediction of destination and routes, and several different techniques have been proposed. Simmons et al. (2006) used the Hidden Markov Model (HMM) and contextual information (day of the week, time and speed of the vehicle) in a corpus of 46 trips in the Michigan area, in the United States. The rate of correct predictions was of 98%. Nevertheless, only 5% of the transitions from one segment to another occurred in intersections between streets, while the other 95% were connected to only one other road segment, which reduces the difficulty in the prediction of the next segment. For the 5% of transitions occurred in corners, the rate of correct predictions was between 70% and 80%. In Krumm's (2008) work, the focus of his model is in predicting short-term, i.e., only next segments, instead destination prediction. His model uses Markov model for prediction, and after observing the last 10 segments traveled by a user, it is possible to predict the next one with 90% accuracy. For predicting the next 10 segments the accuracy rate decrease to 50%. In contrast with Krumm's work, our model predict both route and destination, instead of only the next road segments.

Froehlich and Krumm (2008) use a closest match algorithm, that identifies the similarity between an ongoing route and a route performed in the past, and, if they are similar, the remaining path and destination are predicted. They do not use *map matching* technique, which considerably increase the volume of data that they work. Tiwiri et al. (2012) use a similar methodology for predicting routes and destination as proposed by Froehlich and Krumm (2008). However, Tiwiri et al. (2012) perform map matching, and showed a reduction in the size of data worked, apart from a progress in the performance of the predictive algorithm. The works of Froehlich and Krumm (2008) and Tiwiri et al. (2012) have reached about 40% of accuracy rate in prediction. The PPM algorithm has already presented encouraging results in the work of Burbey and Martin (2008), which is also concerned with the prediction of future location. The training approach considers the time the users arrive at places, the amount of time they stay at those places, and their

current location. The results present 92% accuracy. A main difference between Burbey and Martin (2008) work's and ours is that we consider route prediction, and uses automatic semantic identification of places.

Knowledge discovery techniques, such as association rules, have already been used as an approach to the prediction problem. When a vehicle starts to move, an association rule is obtained for the moving object (according to the streets it passes by). Then a pattern matching function searches for the set of segments of the path traveled in a paths tree. In Morzy (2006), a version of the *Apriori* algorithm is used to generate the association rules. Tanaka et al. (2009) present a hybrid method of predicting destination. Their hybrid method is capable of changing the approach to predicting the destination according to the type of road.

In location-aware systems, semantic information is the action of linking contextual data about geographical places with raw position data collected [Parent et al. 2013]. Thus, a cluster where many geographic points are located can be useful for identifying pattern of displacements, but limited for identifying the reason why the person stays in such place. Thus, semantic information can enrich a trajectory with information such as name and type of place. Ying et al. (2011) are among the pioneers in considering semantic data for improving place prediction. The data that they collected are from both GPS and cell tower signals. For creating semantic tags, they populate the geographic semantic information database (GSID), which contains semantic information from *Google Maps*[3]. Their system comprises two modules: one offline, which is responsible for tagging the semantic locations; and another online, which is responsible for a real time location prediction. A limitation of this procedure relates to updating of the information. Ying et al. (2014) improved their previous work with item recommendations, i.e., when the system identifies that a person should stay in some place, it can suggests some items that are sold at that establishment.

Lung et al. (2014) developed a model for predicting destinations and for detecting the transportation mode. They use *Google Maps API* to search for a location, and enrich the trajectory. Their prediction model, which is based on Hidden Markov Model, was tested with real world data, and an accuracy rate of 68.3% was obtained for identifying the next location. Cao et al. (2010) proposed a model that first identifies the *stay points.* When the object remains stationary for a long period of time at the same place, a *stay point* can be identified. Then, they try to tag that place retrieving the name and type of place from the *Yellow Pages*. They do not perform location prediction, but they create a ranking for the most visited locations.

Our work differs from works that only use geographical information because we also consider semantic information for enriching the trajectories. We are not only interested in identifying the patterns of movements, but also in understanding the reason why the user is at a certain place. The difference between our work and the work of Ying et al. (2014) and Lung et al. (2014) is that we predict not only destination, but also the route user will pass.

Table 1 demonstrates the works most related to ours, and summarizes them by the following features: if the type of place is automatically identified; whether both route and

[3] https://www.google.com.br/maps

destination (or place) are predicted (or one of them); the method applied for route and destination prediction; the accuracy rate. Each line represents one work analyzed.

Table 1: Summary of works most related to ours

Authors	Identify type of place auto?	Route and Destination Predition?	Method for Prediction	Accuracy Rate
Simmons et al. (2006)	No	Both	Hidden Markov Model	95% / 70-80%
Krumm (2008)	No	Segment	Markov Model	90%
Burbey and Martin (2008)	No	Place / Destination	PPM	92%
Tiwari et al. (2012)	No	Both	Closest Match Algorithm	40%
Mazhelis (2011)	No	Both	Longest Common Subsequence	87%
Ying et al. (2011)	Yes	Place / Destination	Partial Matching and Longest Common Sequence	53% - 68%
Monreale et al. (2009)	No	Place / Destination	Prefix Tree Pattern Matching	~54%
Froehlich and Krumm (2008)	No	Place / Destination	Closest Match Algorithm	40%
Lung et al. (2014)	Yes	Place / Destination	Hidden Markov Model	68.3%

It can be noticed that a few works draw attention to join semantic information with geographic location. Most of the papers that we encountered in the literature only consider geographical information for predicting route and destination. The exploration of geographic semantic information can be an important feature to improve the prediction.

3. The PredRoute Prediction Model

This section describes our predictive model. First, we formally introduce important concepts used along this paper: *route*, *partial route*, *remaining route, stay point, contextual information* and *trajectory model*. These definitions are stated below.

- A *route R* comprises a sequence of segments $(S_1,S_2,S_3,\ldots,S_n,\ n > 0)$, i.e., $R = (S_1,S_2,S_3,\ldots,S_n)$, with $n > 0$ and S_i representing the i^{th} road segment of a route;
- Each *road segment*, or just *segment*, has exactly two geographic points $(P_{i1},P_{i2},P_{i3},\ldots,P_{ik},\ k > 1$ and $1 \leq i \leq n)$, i.e., $S_i = (P_{i1},P_{i2},P_{i3},\ldots P_{ik})$, with $k > 1$, and P_{ik} representing the k^{th} point on the i^{th} road segment. A point (x, y) represents a geographic coordinate (latitude, longitude);
- A *partial route T* represents a subset of segments of a route R $(S_1,S_2,S_3,\ldots,S_m,\ 1 \leq m < n)$, i.e., $T = (S_1,S_2,S_3,\ldots,S_m)$, with $1 \leq m < n$;
- A *remaining route F* $(S_{m+1},S_{m+2},\ldots S_{m+p},S_n,\ m + p + 1 \leq n)$ represents the predicted subset of segments to a certain destination, i.e., $F = (S_{m+1},S_{m+2},\ldots,S_{m+p}S_n)$, with $m + p + 1 \leq n$. Figure 1 depicts the concepts of *route, partial route, remaining route* and *road segments*;

- We consider many variables as *contextual information*, among them: *day of the week* of the departure, which is represented by an integer (0 = Sunday, 1 = Monday, ..., 6 = Saturday); the *time interval* of departure which is represented by an integer that corresponds to an interval i between two times (0 for $0 < i \leq 1$; 1 for $1 < i \leq 2$; ...; 23 for $23 < i \leq 24$); *origin* and *destination*, which represents, respectively, the place of origin and the place of destination of a route; *type of place*, which represents the type of location that a user remains. The possible values for the variable *type of place* in our work are home, work, other, sports, education, leisure and unknown;
- A *stay point, cluster* or *stop*, is a geographic area which represents a place that a user spent a time interval greater than a *threshold D*. The value for *D* considered in our work is 10 minutes. For finding out the time interval that a user spent in a cluster, it is necessary that the GPS points are ordered by timestamp, and that the distance between consecutive points are less than *X* meters. The value for *X* considered in our work is 40 meters. Both values for *D* (10 minutes) and *X* (40 meters) were empirically defined;
- A *trajectory model* comprises a list of road segments and contextual information.

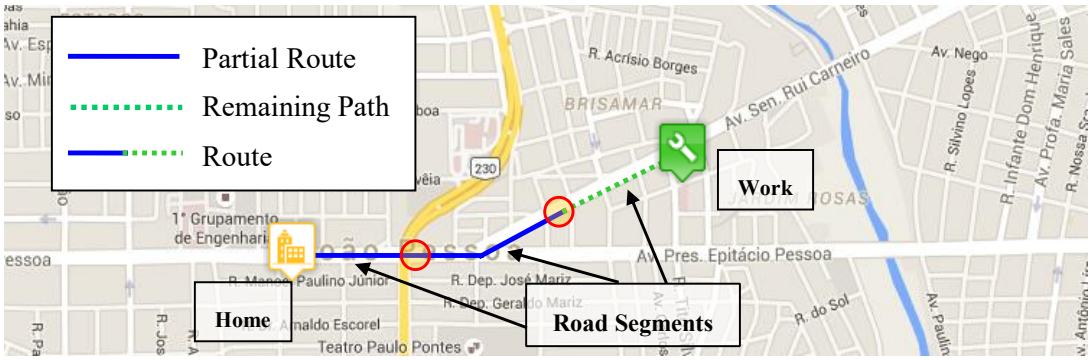


Figure 1: Definition of route (or trip), partial route, remaining path and road segments

3.1. Prediction by Partial Matching

The Prediction by Partial Matching (PPM) algorithm is a sophisticated method for data compression based on statistical models, and is among the most efficient techniques concerned with compression without loss of information [Salomon 2004]. The key idea of this method is the use of an adaptive symbolic model in a finite context. That is, a probability is assigned to a symbol not based on its frequency in the information source, but on its frequency in the context formed by the last *n* characters. For each order of, there is a table of symbols, which is updated for each new symbol codified.

PPM has some features which can be useful in classification and prediction tasks, since it has the capability of rapidly elaborating a symbols tree, adapted to the information source. The symbols tree is called a PPM symbols tree, or simply PPM tree. Further details about the behavior of PPM, including a step by step of an example and the creation of the PPM tree, can be found in Nobre Neto et al. (2015). Because of the features and behavior of PPM, we use it as the core of our model for predicting route and destination.

3.2. Identifying Stay Points

An important step of our predictive model is the process of identifying stay points automatically, which is based on *clustering* techniques. An stay point comprises a centroid point (latitude, longitude) and a radius of 40 meters, and it is created when the object remains stationary inside this area more than 10 minutes. The algorithm of identifying stay points proposed by this work is based on DBSCAN [Ester et al. 1996], a *density-based* algorithm for clustering spatial points [Tork 2012]. Algorithm 1 details the procedure for creating the stay points. The algorithm takes as input a list of users (line 2). For each user (line 6), the algorithm retrieves the set of GPS points ordered by timestamp, which represents the trajectories performed by that specific user (line 7). From those data, the clusters are extracted (line 8). For creating of stay points from GPS points, it is necessary that a user remains stationary for a minimum of 10 minutes, and the distance between the points may not be superior 40 meters. When the stay points are identified, they are associated with the current user (line 9). Then, based on the stay points recently created and on the set of GPS points, the algorithm calculates the routes performed by the user (line 10). Afterwards, the map matching procedure is performed, which associate a geographic point (latitude, longitude) with road segments (line 11). The advantage of doing map matching is that the data to be handled by our model is reduced [Tiwiri et al. 2012]. The output of the algorithm is the same list of users, however containing information about their stay points and the routes performed (in terms of road segments).

Algorithm 1: Procedure for spatial clustering creation

```
1   INPUT
2     users       // List of users for creating spatial clustering points
3   OUTPUT
4     users       // List of users updated, with their respectively list of clusters
5   METHOD
6     FOR EACH users as anUser DO
7       gpsPoints = anUser.getGpsPointsOrderedByTimestamp();
8       clusters  = extractClustersFromGpsPoints(gpsPoints);
9       anUser.clusters = clusters;
10      anUser.trips = extractTripsFromClustersAndGpsPoints(clusters, gpsPoints);
11      anUser.tipsRoad = mapMatchPointsRoad(anUser.trips);
12  // End of FOR EACH
```

It is important to notice that our methodology for identifying stay points does not involve any procedure for identifying the type of place. Up to this moment, we just identify the length of time a user remains stationary in a stay point and the time the user reached the destination. Thus, we are dealing only with geographical data.

3.3. Type of Places Identification

Our approach to automatically identifying type of places of the stay points is detailed in Algorithm 2. This algorithm takes as input a list of users with their respective stay points, as showed in the procedure of Algorithm 1 (line 2). For each stay point of each user (lines 6 and 7), the algorithm retrieves contextual information (the day of the week, the time interval and the length of time remained stationary in the stay point) (line

8). Then, external services API (*Google Places*, *Foursquare* and *Factual*[4]) are online queried for reverse geocoding the stay point (centroid point), gathering information about the POIs around it (lines 9-12). The information collected of the POIs include the name, type of place, the distance between the stay point and the POI. After that, the algorithm identifies the nearest POI among the three retrieved to the stay point (line 13). Then, the type of POI is retrieved, and mapped to the types of location that our model considers (line 14). For instance, if the POI chosen was from *Foursquare* service, and his type is *Restaurant*, then our inference engine might identifies whether the type of place of the stay point is for *Leisure* or for *Work*. The inference engine considers the contextual information retrieved related to the stay point that the person remains stationary to discover the type of place (line 14). Our inference procedure works as follows:

- **Home**, if a user spends more than 10 hours at a 90% of the days;
- **Work**, if a user spends between six and eight hours at a location, and there are some days of the week that the user does not go to that place;
- **Leisure**, if a user goes to a place that he/she does not go frequently, and spends between two and four hours;
- **Sports**, if the type of the POI retrieved is related with sports (such as "gym", "soccer", "football"), and user spends between one and two hours;
- **Education**, if the type of the POI retrieved is related with education (such as "library", "university", "high school") , and user spends between two and four hours certain days of the week;
- **Other**, when the user is supposed to be sorting things out and spends between ten and sixty minutes at a place;
- **Unknown,** if none of the types of place above has occurred.

Algorithm 2: Procedure for automatically type of places identification

```
1   INPUT
2     users       // List of users with their respectively clusters
3   OUTPUT
4     users       // List of users updated, with their the clusters enriched with semantic
5   METHOD
6     FOR EACH users as anUser DO
7         FOR EACH anUser.stayPoint as stayPoint DO
8           Info = getContextualInformation(stayPoint);
9           centroidPoint = getClusterLocation(stayPoint);
10          googleInfo = getGooglePlaceInfo(centroidPoint);
11          foursquareInfo = getFoursquareInfo(centroidPoint);
12          factualInfo = getFactualInfo(centroidPoint);
13          serviceChosen = getNearestPOI(googleInfo, foursquareInfo, factualInfo);
14          stayPoint.placeType = inferType(serviceChosen, info, centroidPoint);
15  // End of both FOR EACH
```

3.4. Route and Destination Prediction

This sections is divided into two, which describes the details about the training and testing stage.

[4] http://factual.com/

3.4.1. Training Stage

The training stage consists of creating our predictive model for route and destination for each participant of the experiment. Therefore, the predictive model of a given user is personalized, that is, it will not be influenced by the trajectories performed by another user.

The procedure for training our predictive model is presented in Algorithm 3. The algorithm takes as input a list of users, which contains information about displacements, stay points visited by the users and user identification (line 2). The output of the algorithm is a list of users with their respectively trajectory models created (line 4). Regarding the execution of the algorithm, for each map matched route (at this moment a route is a list of road segments) from each user (lines 6 and 7), the exact location and road segments of origin and destination are gathered (line 8). Then, contextual information is retrieved from the route, which are the day of the week, the time interval of departure and the type of location of the origin and destination of stay points (line 9). Such route information is then used to create the PPM tree (line 10). The next step (line 11) consists of creating a trajectory model from all of these information captured between lines 8 and 10. If this trajectory model already exists (i.e., the model has already stored this trajectory), then a counter is incremented (lines 12 and 13). This can occur in case of a user has several equal displacements, such as home to work. Otherwise, the trajectory model is stored for the first time (lines 14 and 15).

Algorithm 3: Procedure for training stage

```
1   INPUT
2     users       // List of users for creating spatial clustering points
3   OUTPUT
4     users       // List of users updated, with their respectively trajectory models
5   METHOD
6     FOR EACH users as anUser DO
7       FOR EACH anUser.tripRoad as route DO
8         POIs    =  getOriginAndDestinationLocation(route);
9         contextualInfo = getContextualInformation(route);
10        ppm-tree = routeToPPMTree(route);
11        traject-model = createModel(POIs, contextualInfo, ppm-tree);
12        IF (anUser.existTrajectory(traject-model)) THEN
13            anUser.incrementCount(traject-model);
14        ELSE
15            anUser.store(traject-model);
16  // End of both FOR EACH
```

3.4.2. Testing Stage

The testing stage consists in obtaining the rates of correct predictions of the users destination and route, from the moment their trip starts. A test in the context of our work is to predict the geographic destination and route of a user ongoing displacement, and to predict the type of place that a user is going. The routes used in the training stage are not used in the testing stage. Therefore, we apply cross-validation in our tests, partitioning the corpus of routes for training to the corpus of routes for testing.

Algorithm 4 details the procedure for executing tests. The algorithm takes as input the object user, the list of GPS points along with timestamp of an ongoing route and

contextual information, which are day of the week, type of stay point of the origin and origin, (lines 2-4). First, the algorithm retrieves trajectory models that have similar contextual information with the ongoing route, such as the day of the week, the time interval of departure, the stay point of departure and the type of the stay point of the origin (line 9). Then, the algorithm performs a map matching with the list of GPS points of trip (line 10). The route performed so far is compressed with all PPM trees of the retrieved trajectories model (line 12), in order to obtain the trajectory model with the best compression ratio (lines 13-15). The compression generates a Compression Rate (CR), which is the division of the clean file with the codified file. Nobre Neto et al. (2015) provides further details about this compression rate. The output of this algorithm is the best selected trajectory model for the ongoing trip, which contains information about the remaining path (road segments), the destination and the stay point of destination (line 6). Thus, with this information, we provide for the final user the stay point and the type of the stay point that he or she is going, besides the route that will be performed.

Algorithm 4: Procedure for testing stage

```
1   INPUT
2     user        // User that is an ongoing route
3     trip       // List of GPS points along with timestamp info of an ongoing trip
4     contextualInfo // Contextual information: day of week, type of place origin, origin
5   OUTPUT
6     selected-trajectory-model   // A trajectory-model predicted
7   METHOD
8     max-compression-rate = -1
9     trajectories-model = user.getTrajectoriesModel(trip, contextualInfo);
10    routeMapMatched = mapMatchPointsRoad(trip);
11    FOR EACH trajectories-model as aModel DO
12        curr-comp-rate = compress(aModel, routeMapMatched);
13        IF (cur-comp-rate > max-compression-rate) THEN
14            max-compression-rate = cur-compression-rate
15            selected-trajectory-model = aModel
16  // End of FOR EACH
```

4. Experimental Evaluation

This sections explains the data selected for the testing stage, and presents the results obtained from our model.

4.1. Data Selection

The data used in this work were obtained from people living in the cities of João Pessoa and Campina Grande, both in the State of Paraíba (Brazil). We selected eight participants for installing into their *smartphones* an application for capturing their position. The application can use both wireless network and GPS device of the *smartphone*. If a user is located in an indoor place, which possess Wi-Fi, then this type of resource is used for gathering the location. In an outdoor location, the 3G (if enabled) or GPS device of *smartphone* was used. The participants were oriented to let the application executing, since it can send data to the server automatically. More than 300.000 GPS points were collected from the *smartphones* of the participants, which represents a total of 156 routes. Thus, an average of 19.5 routes per user were generated. The data were collected for users that possess completely different habits, during one month.

4.2. Results

As mentioned in section 3.4.2, cross-validation (90% of data for training and 10% for testing) was performed in this work. From the route to be tested, our model derives three new ones, the first with 15% of the route, the second with 50% and the third with 85%. This is important for discovering if the prediction accuracy increases or remains the same when the route is getting near from destination.

Table 1 summarizes the results obtained. There are two results considered in this work: one about route and destination prediction (RDP), which considers only geographic movements; and the other that is type of place prediction (TPP), which considers semantic information. For each result, there are three columns, representing the progress of the route to be tested. With 15% of the route performed, the accuracy rate for RDP was 39.2%, while TPP have 60.7% of correct rate. Testing 50% of the route, the accuracy rate of RDP increases to 45.96%, while TPP reached 62.9%. When the route has 85% of the segments travelled, RDP has an accuracy rate of 46.02%, and TPP reaches 62.9%.

Table 1: Accuracy rate according to the percentage of an ongoing partial route

	Route and Destination Prediction (RDP)			Type of Place Prediction (TPP)		
	15%	**50%**	**85%**	**15%**	**50%**	**85%**
Accuracy Rate	39.2%	45.96%	46.02%	60.7%	62.9%	62.9%

Our tests were performed on a computer equipped with a Core i7-4500 CPU, 16GB of RAM and 1TB of Hard Disk, and about one second have been spent for predicting route, destination and the type of place.

5. Conclusions and Future Work

The model proposed by this work is for predicting both destination and routes, apart from the type of location. In the tests performed, where our algorithm uses cross-validation, it was possible to obtain that the model has a better accuracy rate for predicting the type of place of the destination compared to the route and destination prediction, which considers only geographic displacements. Thus, even that the algorithm predicted wrong geographic destinations, it was possible that the type of place predicted might be correct. Differently from many works, we incorporate semantic information in our predictive model. The daily use of our model might be really useful, because it is not necessary an active user interaction and a good performance was obtained of the execution.

For further work, we intend to predict if a person is getting away from a destination that we initially predicted, that is, instead of predicting a new destination based on historical displacement, we will try to discover if the user is going to a place that he had never visited before. This will be possible because our model is considering semantic information. Another planned improvement is to expand the type of places that we consider, and develop an *Application* for implementing the model proposed by us.

References

Aggarwal, C. C. and Reddy, C. K. (2013), Data Clustering: Algorithms and Applications. Chapman and Hall/CRC. Boca Raton, FL.

Alvares, L. O., Bogorny, V., Palma, A., Kuijpers, B., Moelanes, B., and Macedo, J. A. F. (2007) "Towards Semantic Trajectory Knowledge Discovery", In: Technical Report, Hassel University.

Burbey, I. and Martin, T. L. (2008) "Predicting Future Locations Using Prediction-by-Partial-Match", In: Proc. 1st ACM MELT, pages 1-6.

Cao, X., Cong, G., and Jensen, C. S. (2010) "Mining Significant Locations From GPS Data", In: Proceeding of VLDB, Vol. 3, No. 1, pages 1009-1020.

Ester, M., Kriegel, H., Jorg, Xu, C. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise", In: Proc. 2nd Int. Conf. on KDD.

Froehlich, J. And Krumm, J. (2008) "Route Prediction From Trip Observations", In: Society of Automotive Engineers (SAE).

Krumm, J. (2008) "A Markov Model for Driver Turn Prediction", In: SAE.

Lung H-Y., Chung C-H., and Dai, B-R. (2014) "Predicting Locations of Mobile Users Based on Behavior Semantic Mining", In: Trends and Applications in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, Vol. 8643.

Morzy, M. (2006) "Prediction of moving object location based on frequent trajectories", In: ISCIS, p. 583-592, Springer.

Nobre Neto, F. D., Baptista, C. S., and Campelo, C. E. C. (2015) "Predicting Routes and Destinations of Urban Trips Using PPM Method", In: 7th SBCUP, Brazil.

Quddus, M. A., and Noland R. B. (2006) "A High Accuracy Fuzzy Logic Based Map Matching Algorithm for Road Transportation", In: Journal of Intelligent Transportation Systems.

Salomon, D. (2004), Data Compression: The Complete Reference. Springer, 3rd Edition, New York, NY.

Simmons, R., Browing, B., Yilu, Z. and Sadekar, V. (2006) "Learning to Predict Driver Route and Destination Intent", In: Intelligent Transportation Systems Conference.

Tanaka, K., Kihino, Y., Terada, T., and Nishio, S. (2009) "A Destination Prediction Method Using Driving Contexts and Trajectory for Car Navigation Systems", In: ACM symposium on Applied Computing, pages 190-195.

Tiwari, V. S., Chaturvedi, S., and Arya, A. (2013) "Route Prediction Using Trip Observations and Map Matching", In: IEEE IACC.

Tork, H. F. (2012) "Spatio-temporal Clustering Methods Classification", In: DSIE – Doctoral Symposium in Informatics Engineering.

Parent, C., Spaccatietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiana, M. L., Gkoulalas-Divanis, A., Macedo, J., Peledis, N., Theodoridis, Y. and Yan, Z. (2013) "Semantic Trajectories Modeling and Analysis", In: ACM Computing Surveys, Vol. 45, No. 4, Article 42.

Ying, J. J-C., Lee, W-C., Weng, T-C. and Tseng V. S. (2011) "Semantic Trajectory Mining for Location Prediction", In: Proceedings of the 19th ACM SIGSPATIAL GIS.

Ying, J. J-C., Chen H-C., Lin, K. W., Lu, E. H-C., Tseng V. S., Tsai, H-W., Cheng, K. H. and Lin, S-C. (2014) "Semantic trajectory-based high utility item recommendation system", In: Expert Syst. Appl. 41(10): 4762-4776.

Optimization of Taxi Cabs Assignment in Geographical Location-based Systems

Abilio A. M. de Oliveira[1], Matheus P. Souza[1], Marconi de A. Pereira[1], Felipe A. L. Reis[2], Paulo E. M. Almeida[2], Eder J. Silva[1], Daniel S. Crepalde[1]

[1]Departamento de Tec. em Eng. Civil, Computação e Humanidades
Universidade Federal de São João Del-Rei (UFSJ) - Ouro Branco, MG - Brazil

[2]Departamento de Computação
Centro Federal de Educação Tecnológica de MG (CEFET-MG) - Belo Horizonte, MG - Brazil

Abstract. *In this paper, different approaches are evaluated to assign taxi cabs to customers in geographical location-based systems. The main purpose of this work is to identify the solution in which all current customers are met in an acceptable time, however minimizing the distance traveled by existing free taxi cabs. Two aspects are considered: 1) the method to calculate the distance between vehicles and customers; and 2) a vehicle assignment strategy. The methods to calculate the distance between vehicles and customers are: a GPS-based routing (a shortest path algorithm) and the Euclidean distance. On the other hand, as vehicle assignment approaches, the considered strategies are: a greedy algorithm, which assigns each vehicle to the closest customer, and an optimization algorithm, which assigns vehicles considering the whole scenario, minimizing the global distance traveled by taxi cabs to meet the customers. This last strategy considers an optimization model in such a way that the calls are not readily answered. In this case, a short waiting window is implemented, where the calls are stored and then the optimization algorithm is executed, in order to minimize the required distance and to meet all current customers. The combination of the two methods of distance calculation and the two vehicle assignment strategies formed four possible approaches, which are evaluated in a realistic simulator. Results show that the approach which uses the shortest path algorithm and an optimization algorithm reduces the average service time by up to 27.59%, and the average distance traveled by up to 45.79%.*

1. Introduction

Taxi services are an alternative urban transportation mode which offers convenience and speed in relation to public transport. It is noticed that taxi service is not efficient because much of the time, about 50%, they are empty [Reis et al. 2011]. On the other hand, some passengers complain that some calls are missed. In Belo Horizonte, for example, about 10% to 15% of customers, who prefer the convenience of ordering a taxi by phone, give up because of the long wait for the service [Castelo-Branco 2012].

To make this service more efficient, several studies are being conducted and new attribution methods based on geographical location are being adopted [Reis et al. 2011]. These methods use GPS (Global Positioning System) to locate taxis and customers and propose a better way to assign each vehicle to attend the existing calls. Usually, two aspects are explored by these approaches:

1. The procedure used to estimate the distance traveled by a taxi to achieve a customer. Two very common possibilities are the Euclidean distance, and the shortest path between points. It is important to note that the evaluation of the distance between the taxi and the customer can be treated in a more elaborate way, taking also into account the route that the taxi will go, considering factors that complicate the taxi displacement, such as speed limits, traffic jams, among others;
2. The vehicle assignment strategy. A common vehicle assignment method is based on a greedy approach, where the taxi cab which is closest to a customer is designated to take the call. The intent is to minimize the passenger waiting time and to minimize the idle taxi time. Moreover, as all greedy heuristic, this solution can be a good one in the local space of solutions but can be a poor choice when considering global solution space. Thus, the problem can be modeled as an optimization problem, where it is wished to minimize the idle taxi time, treating the passenger waiting time as a restriction of this problem.

This paper aims to explore these two aspects, in such a way to measure the effectiveness of the most common ways currently adopted to estimate distance and to assign taxi cabs to customers. Also, this work intends to propose an optimization approach to solve this problem, to objectively evaluate its performance by means of micro-simulation and to draw some practical conclusion about the experiments undertaken.

The remaining of this article is organized as follows: in Section 2 a historical background is revised, and some study cases are briefly presented. In Section 3, the tools and methods used to simulate and evaluate the discussed approaches are detailed. A practical experiment and some simulation approaches are presented in Section 4. Results and discussions about the experiments are shown in Section 5. Finally, in Section 6, some conclusions are presented along with some future work suggestions.

2. Related Work

The use of mobile applications is increasing the productivity of resources, especially in the context of smart cities [Steenbruggen et al. 2015]. A constantly growing demand, combined with an increasing traffic complexity, made the task to identify the best taxi to attend a call became very hard. This problem has been studied for some time around the world, showing that the problem is not exclusive to big cities, emphasizing the need for a solution to improve this picture.

In this sense, [Xu et al. 2005] describe a scenario in Shanghai, where taxis which joined Dazhong Company were modified and received GPS trackers, to indicate their location in real time. Customer requests arrive at a call center, which uses the trackers to know the nearest free taxi to answer each call, since the customer's location is also known. Then, the taxi drivers, within a certain range, respond to the call accepting it or not, via a button located on the equipment installed in the vehicles. Each consumer is then informed about how long it will take for his/her demand be met. The control of occupied and free taxi cabs and their distribution throughout the city is made in the dispatch center of the company itself (DZDC - *Dazhong Dispatching Center*), which provides a higher speed in attendance.

In the city of Singapore, the system that handles the taxi service is the Automatic Vehicle Location and Dispatch System (AVLDS). This system, like the one used in Shang-

hai, consists of taxi cabs sending their positions to a central station. Customers calls arrive at this central, which congregates all taxi companies in the city. The system assigns the closest vehicle to each customer and waits around 10 seconds for the taxi driver to accept or decline the call. If he/she declines, then the system performs a new search for taxis using a new request for that service [Liao 2009][Yang and Wong 1998].

In both cities, the adoption of the mentioned systems provided a reduction of 16% to 32% on traveled distances without a passenger, and a reduction of 15 to 30 minutes in the customer average waiting time. These numbers were obtained considering the scenario with the traditional method, where the call center sends a broadcasting message, and a taxi driver who believes he/she is closer to the passenger offers to take the call.

[Liu et al. 2015] directed their work to highlight the importance of communication between various taxi drivers, to predict where and when a customer will need a taxi. The past route information can be used in a machine learning process, combined to a optimization process in order to reduce the cabs idle time.

Conversely, [Santos and Xavier 2015] and [Jung et al. 2015] present a different practice that helps to reduce the idle traveled distance, using a cab sharing method (ride sharing). Each vehicle has a known location and all cabs are connected to a wireless network, so that the customers can find out if a vehicle will pass through the desired route. Then, they can share the same taxi cab with existing passengers. Therefore, this practice consists in helping both the driver and the customers, reducing both the global cost of travel and the overall traveled distances.

3. Tools and Methods

3.1. Tools

MATLAB (acronym for MATrix LABoratory) is a multi-purpose tool focused on numerical modeling, having matrices as its fundamental data structure. SUMO (Simulation of Urban MObility) is a simulation platform for discrete event traffic, microscopic, using continuous space and presenting inter- and multi-modal capabilities [Krajzewicz et al. 2012]. SUMO was chosen because it is open source software, very popular in works focused on traffic simulations.

The relationship between SUMO and MATLAB is based on TraCI4Matlab [Acosta et al. 2015], which is an API (Application Programming Interface) developed in MATLAB. TraCI4Matlab is an implementation of TRACI (Traffic Control Interface) protocol, which allows for interactions between MATLAB and SUMO in a client-server scenario. TraCI4Matlab can be used to implement vehicles control, traffic lights timing, intersections monitoring, among other applications of traffic for SUMO simulations.

3.2. Assignment Problem

Linear Programming Problems (LP) are those which present a linear objective function and linear constraints, regarding the existing design variables [Taha 2008]. Certain special cases of LP, such as those involving network flow and multi-commodity flow, are considered so important that they have generated several researches and specialized algorithms in order to solve them.

There is a special case of LP which is called the "Transportation Problem" [Taha 2008]. A common mathematical modeling approach to this case can be seen in equation (1), where m indicates the origins, n destinations, c_{ij} the cost per transport unit and x_{ij} the amount sent from i to j.

Minimize:

$$z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} * x_{ij}. \tag{1}$$

The solution of a Transportation Problem consists of determining x_{ij} values which will minimize the total cost of transportation and satisfy all constraints of demand and supply.

There is a special case of transportation problems, called "Designation Problem" (DP) [Taha 2008], which can be used in this context. The objective here is to systematically assign an available taxi cab to attend a customer, meeting some criteria z. A mathematical model to solve this problem can be seen in equation (2) [Goldbarg and Luna 2005].

Minimize:

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} * x_{ij}, \tag{2}$$

subject to:

$$\sum_{j=1}^{n} x_{ij} = 1, \ i = 1, ..., n; \tag{3}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \ j = 1, ..., n; \tag{4}$$

for:

$$x_{ij} \in \{0, 1\} \ \ \forall \, i = 1, ..., n; j = 1, ..., n. \tag{5}$$

DP can be understood as the problem of allocating n producing cells to n tasks at a cost c_{ij}. If i receive the call to given j, x_{ij} turns 1, indicating the allocation of the call, otherwise, if i is not intended for the specific j, x_{ij} gets 0 [Goldbarg and Luna 2005]. Therefore, each variable x_{ij} is binary.

The solution of a DP can be obtained using the Hungarian Algorithm (HA) [Jonker and Volgenant 1986]. It works by analyzing a matrix consisting of j lines, indicating the number of calls to be answered, and i columns, representing each taxi cab available for service at any given cycle time.

In the context handled in this paper, the Assignment Problem is used to identify which vehicle will met each customer call. To do this, it is necessary to calculate the distance (or cost) of each taxi to meet each customer. It can be done by a classic shortest path algorithm, like the Dijkstra algorithm [Dijkstra 1959]. For all free taxi cabs, the shortest path to met the passenger is calculated; then, HA considers the costs of these paths to identify which is the best solution, considering the whole scenario and trying to minimize the criteria z.

In order to provide realistic calculations, Traci4Matlab library, coupled with SUMO, is used. It is important to note that these calculations go beyond to simply calculating the distance between sources and destinations. SUMO allows for a realistic simulation of the scene, where traffic signals, different speed limits, as well as the existence of other vehicles traveling on the streets, are simultaneously considered. This effect is similar to those GPS applications for smartphones, such as Waze[1] and Tomtom[2], which use traffic information to calculate the best route to reach a given destination.

3.3. Methods

It is possible to classify a taxi service in four ways, according to the presence or absence of information about geographical locations of taxis and passengers [Reis et al. 2011]. The first one, a "Random Searching" mode, occurs when a customer waits in a random location for taxis moving around on the way. This situation is characterized by the ignorance of positions of both, taxis and customers.

A second mode, "Fixed Stop", occurs when customers go after a taxi stop point. It is characterized by the knowledge of taxi cab positions, however the customer's initial position is unknown. The third mode is the "Broadcasting", which uses the geographical location of customers, but the positions of taxi cabs is unknown by the central control system. In an example of this mode, a customer calls the central service and the customer's location information is passed on by radio to the taxi drivers. The first driver to accept the request is confirmed by the central to pick up the customer. Since the position information of the available taxi cabs is unknown, this method does not guarantee that the closest driver or the most suitable vehicle is assigned to the customer.

The last mode is represented by "GPS-based models", which can identify the geographical position of both, customers and taxis. These GPS methods are distinguished from each other through the algorithms used to define the taxi driver which will be responsible for picking up each passenger. The geographical location of the actors involved underlies the criteria and costs used by the algorithms to choose the attendant to each request. Cost is a factor that determines the difficulty of a taxi driver to answer a call. Basically, two methods can be used: the euclidean distance and the routing distance between each customer and the available vehicles (the distance that each taxi driver must travel through streets and avenues to reach the customer). The following sections explain some features of GPS-based algorithms, exploring both the distance calculation methods between actors, as the vehicle assignment strategy.

3.3.1. Greedy Algorithm based on Euclidean Distance

This allocation method has the advantage of low computational cost required, compared to the next methods, due to the simplicity of the calculations involved. The operating principle is to calculate the distance between two points, and then finding the taxi cab which has the lowest linear distance from the customer, calculated through Equation (6):

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \tag{6}$$

[1]https://www.waze.com/

[2]http://www.tomtom.com/

where x_1 and y_1 are the coordinates of the customer and x_2 and y_2 are the coordinates of an available taxi cab.

For every customer call, the system checks what is the best candidate, using therefore a greedy approach. As can be seen in the Figure 1, not always the result of this method is the actual best. Due to the traffic structure of the cities, the euclidean distance is not equal to the distance to be traveled to reach the destination. However, this is a solution to be analyzed, since it has a very large computational gain when compared to other methods based on shortest path algorithms.

In the case presented in Figure 1, Taxi 1 is very close to Customer 1, considering the linear (Euclidean) distance. However, it must go through four different streets to meet the customer. On the other hand, Taxi 2 is more distant from the customer, considering the linear distance. Although, only three streets separate the taxi cab and the customer. This scenario is a small sample of the complexity that the context (traffic jams, traffic flow, speed limits) may transform the calculation of the cost in a non-trivial scenario.

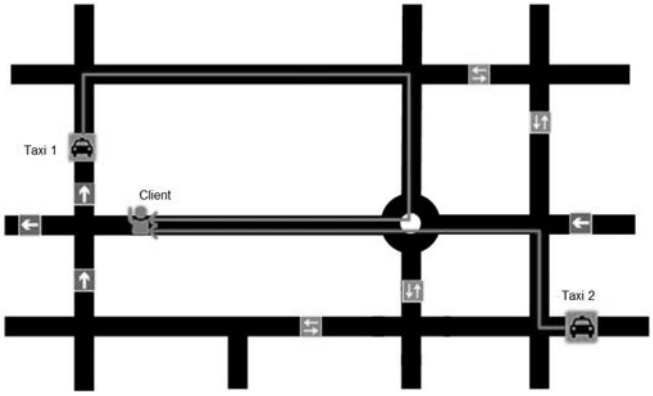


Figure 1. Problem of the Euclidean distance in the assignment of taxi cabs [Reis et al. 2011].

3.3.2. Greedy Algorithm based on Shortest Path

This variation of the previous greedy algorithm aims to partially repair the deficiency pointed by Figure 1. The operating principle is to calculate the actual distance, finding the taxi cab which has the lowest driving distance to the customer.

This method still uses a greedy approach, since, for each customer call, a taxi cab which has the shortest driving distance to the customer is selected to do the service. The following methods to be detailed, on the other hand, try to consider a global view of the scenario. In some way, they effectively try to minimize the total distance that vehicles have to travel through the streets, without a passenger, before reaching the customers.

3.3.3. Hungarian Algorithm based on Euclidean Distance

The Hungarian Algorithm (HA) is a popular procedure for solving assignment problems. In this work, it analyzes the matrix formed by customer calls collected in a time window of 50 seconds and the taxi cabs available to attend, so that the total cost of the chosen solution pairs is minimized. So, this is a very different approach from Greedy Algorithm. The cost factor used, as this model's name implies, is the Euclidean distance between customers and taxi cabs.

If the number of calls is less than the number of available vehicles, some taxis are not included in the search, but all calls are answered in such a way that the total cost is the lowest possible. The same goes for the opposite case, when the number of taxis is lower. Some calls are not answered on that iteration, always looking to find a solution which brings the lowest overall cost.

The evidence that HA determines the best assignment among taxi drivers and customers is given by the Great Allocation Theorem and the König Theorem [Konig 1931]. The Great Allocation Theorem says that if a real number is added or subtracted from all entries of a row or column in a cost matrix, then an optimal assignment for the resulting cost matrix is also an optimal assignment for the original cost matrix [Kuhn 1955].

3.3.4. Hungarian Algorithm based on Shortest Path

In this approach, the cost calculation used in HA is based on the actual distance to be traveled by the taxi cab to meet a customer. The calculation consists on the analysis of a matrix formed by available taxis and customer calls, on a time window of 50 seconds. Thus, the algorithm minimizes the total cost of taxi assignments. This optimization approach provides an improvement in the previous algorithm, since now the actual distance to be traveled by each driver is considered.

The use of HA does not imply a significant rise in processing time when comparing this approach with greedy algorithms. Its major issue is the need for a set of calls to improve the results, which demands the time window. The value of 50 seconds was chosen with the intention to avoid leading to excessive idleness of the vehicles, and conversely, a big amount of calls to be answered, keeping a trade-off duration during the processing of customer calls.

4. Experiment and Simulations

The four algorithms presented in Section 3.3 were implemented in practice: Greedy Algorithm based on Euclidean Distance (GED), Greedy Algorithm based on Shortest Path (GSP), Hungarian Algorithm based on Euclidean Distance (HED), Hungarian Algorithm based on Shortest Path (HSP). Simulations for each algorithm were carried out under the same conditions, considering a same number and an equivalent geographical distribution of taxi cabs and other vehicles. Likewise, customer requests are located in the same places and amount for all simulated algorithm. The choice of each taxi to meet the requests followed the criteria for each algorithm. Spent time and distances traveled by taxi cabs to service the requests are then made available by SUMO to MATLAB, allowing fair comparison among the methods.

The simulation environment tried to be as realistic as possible, considering two-way and one-way streets, traffic lights and other vehicles traveling freely. The streets and avenues cover an area of 54 km^2 (see Figure 2). Border roads were designated as rapid traffic routes, having a speed limit of 80 km/h. Vertical roads were considered arterial roads with a limit of 60 km/h and horizontal roads were considered collector roads with speed limit of 40 km/h.

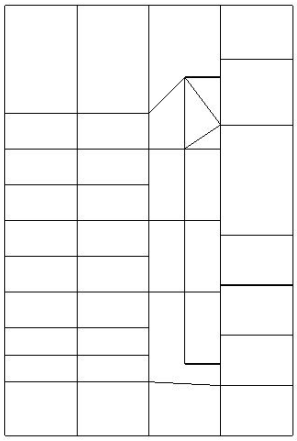


Figure 2. Simulation environment in SUMO.

The amount and types of vehicles which formed the traffic flow were set in proportion to the environmental area of the city of Belo Horizonte [Castro 2014] [IBGE 2014]. The routes for all private (non taxi) vehicles were randomly determined. The simulation included: 9,333 private cars, 3,443 motorcycles, 155 buses and 450 taxi cabs, among which 225 were green (available) and 225 red (busy). The number of taxi cabs was defined according to the average value of cabs into each 0.12 km^2 in the city of Belo Horizonte [Reis et al. 2011].

Calls and taxi cabs were distributed uniformly and randomly throughout the network. The moments of occurrence of calls were also random and those are kept the same for all simulations. In order to better represent the dynamics of the system, taxi cabs have a specific area of initial actuation which is altered according to assignments to answer calls. Routes were created by vehicle, by Dijkstra least-cost algorithm, already implemented in an internal function of the simulator, after taxi-customer assignments were determined.

The relationship between the amount of available and occupied taxis is kept constant throughout the simulation, which keeps the tests with no tendency of increase or decrease of the taxi supply relative to the number of customer calls. This steady state ensures that waiting times for services keep up uniform, directly related to the amount of time the taxis run free, without customers [Yang and Wong 1998]. However, in actual life, for each day of the week, time, region, events, weather, etc., there are variations in the number of requests. So, for this set of simulations, it is assumed to be part of a day, with a constant number of requests, giving a better scenario for comparisons.

Another feature, not included on the simulations, were the concurrency requests. There is the possibility, in an actual environment, of a taxi driver having its status changed at the same time a call processing is initiated, making this driver unavailable for those

particular calls being processed.

To statistically validate the results, the number of simulations conducted for each scenario was equal to 30, with a duration time of 4 hours, totaling 120 hours (or 5 days), to comply with Central Limit Theorem. On average, a taxi cab serves 15 calls per day, working for 13 hours (two or more drivers can use the same vehicle, in complementary time [Lopes 2014]). Therefore, for each simulation of 4 hours, a number of approximately 4 calls per taxi was randomly generated and distributed for both, instant of time and occurrence position, totaling 54,000 requests simulated for each algorithm.

The requests were generated until a predetermined final time was achieved, and the same conditions were repeated for each simulation and for each algorithm, considering place, calling time and traffic, so ensuring that the same factors were present under the same conditions, for each algorithm. Therefore, the algorithms defined the best taxi cab for each request, and the values of the waiting time and the distance were extracted and evaluated.

In our experiments, each simulation can be objectively described as follows:

- Vehicles are randomly distributed on the map: private cars, motorcycles and taxis. In each second, vehicles can enter to and exit from the network, simulating the traffic flow. However, the number of taxi cabs remains constant throughout the simulation;
- Half of the taxi cabs remain free and the other half occupied;
- Requests are made and the simulation is independently carried out in 4 scenarios, one for each algorithm. Each algorithm determines the best taxis, according to its own criteria, to meet each customer;
- A taxi cab state changes from free to busy and it is classified as *ongoing service*, avoiding to be assigned to other calls. A minimum cost algorithm determines the best route taken by the driver to the customer;
- To keep the ratio of free and busy taxis, when a taxi is assigned, another taxi currently classified as *ongoing service* becomes free;
- Once a taxi meets a customer, the distance and the traveled time spent to arrive to the passenger's location are stored.

5. Results and Discussions

The four algorithms were tested considering the same conditions for the simulation in SUMO, as presented before. The results could then be analyzed with assurance that comparisons are fair. As can be seen in Table 1, average and standard deviations of both attendance **time** and the total **distance** traveled were calculated for each algorithm, considering all 54,000 simulated calls divided into a total of 30 simulations performed for each algorithm.

The first result was the average waiting time for each call, which was measured from the time that the request was made by the client until the instant that a taxi arrived. For algorithms based on the Euclidean distance, it was observed that instantaneous distribution algorithm (greedy) was approximately 7.98% more efficient in attendance time, but the Hungarian Algorithm was 1.09% more efficient at average traveled distance to meet the customer. Variation of the average values for both, time and traveled distance, were very close. It was observed also that, when using the Euclidean distance as a cost criterion

in the algorithms, the implementation of the Hungarian Algorithm does not become more efficient, not generating significant results.

Table 1. Attendance Time and Distance (Average $\pm$ Standard Deviation).

	Time	Distance(Km)
GED	4min36sec $\pm$ 5min48sec	3.3718 $\pm$ 3.0467
HED	5min $\pm$ 5min51sec	3.3352 $\pm$ 3.0531
GSP	2min28sec $\pm$ 2min11sec	2.0459 $\pm$ 1.9868
HSP	1min47sec $\pm$ 1min38sec	1.1090 $\pm$ 1.3777

The highlight is due to the algorithms based on the actual distance to be traveled between customers and taxi cabs. The Greedy Algorithm, the worst based on the shortest path, showed up a gain of 46.48% in the average attendance time, when compared to GED, and a gain of 38.66% in average distance traveled, when compared to HED.

Now, the proposed algorithm to optimize the taxi service process, using Hungarian Algorithm (HA), presented an average attendance time of 27.59% lower than Greedy Algorithm (GA). As for the total distance traveled in attendance, HA presented an even better result, approximately 45.79% below. Another point to note is the stability of responses observed through standard deviation estimations. HA showed up a variance around 25.10% lower than GA in service time and a variance 30.66% lower than AG, when comparing the average distances traveled.

Considering the algorithms based on shortest path, these results mean an average gain of 0.9369 Km for HA in relation to GA in a single request of attendance. As 54,000 calls were simulated, the total gain was around 50,620 kilometers relative to GA. The result becomes even more significant considering, for example, the city of Belo Horizonte, where around 96,000 calls happen in a day [Reis et al. 2011]. In this scenario, HA would imply a gain in relation to GA by approximately 89,991 kilometers daily.

Figure 3 shows another view of the obtained results, detailing the sum of the attendance times for calls. The algorithms based on the Euclidean distance presented a big difference when compared to algorithms that use actual shortest path. GED, however, showed up a total gain of 21,400 minutes compared to HED. In the graphic of distance traveled, shown in Figure 4, the discrepancy between the lines of the algorithms based on the Euclidean distance is smaller, with a lead of 2,000 kilometers for Hungarian algorithm (HED).

The algorithms based on the shortest path stood out. The total attendance time of GSP is similar to the attendance time of 29 thousand requests of GED. GSP exceeded HSP in 36,680 minutes. The total time of HSP service (to 54 thousand calls) is similar to the attendance time of 39 thousand requests of GSP.

Even more significant are the results found for the traveled distance (Figure 4). The total traveled distance by taxi cabs to all calls of GSP amounts to the same distance of HED in approximately 32,860 calls. HSP again achieved the best result. It was smaller than GSP in 50,620 kilometers. Its total distance is similar to the distance of GSP in 29,300 calls. So, HSP would save 50,620 kilometers of travel, which is equivalent to 24,700 requests.

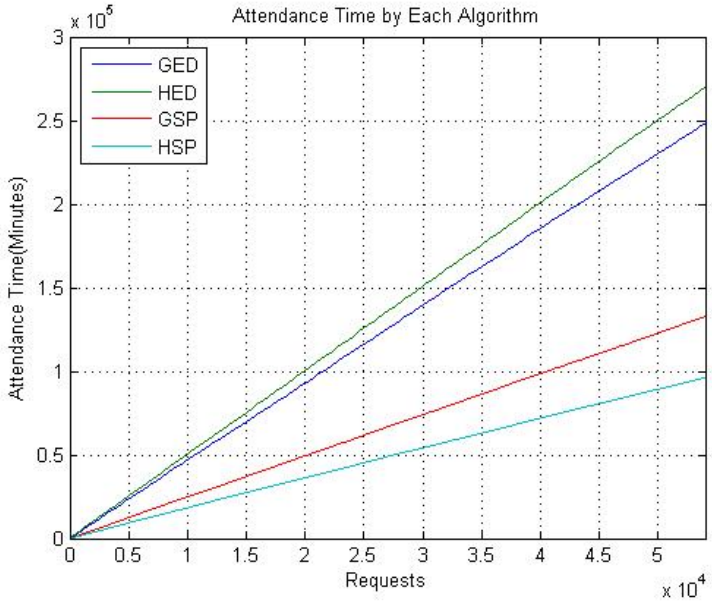


Figure 3. Sum of the attendance time by each algorithm.

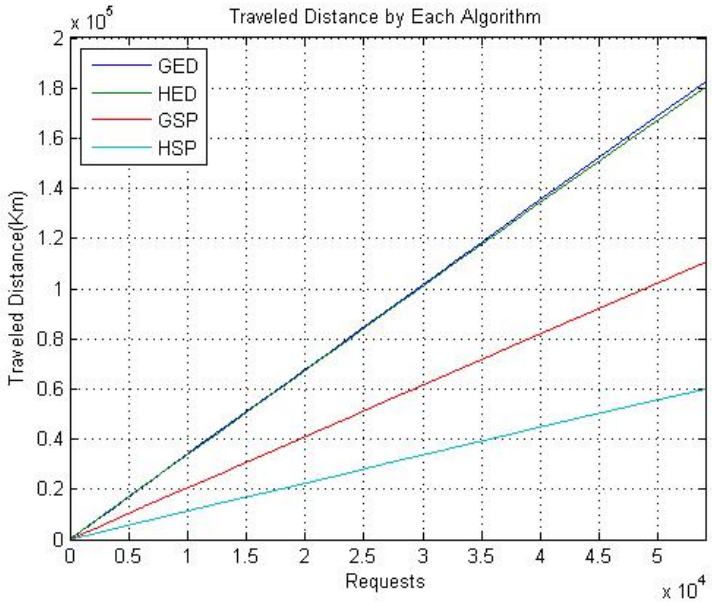


Figure 4. Sum of the traveled distance by each algorithm.

6. Conclusion

The experiments included the choice of taxis from the Greedy Algorithm based on the Euclidean Distance (GED), the Greedy Algorithm based on Shortest Path (GSP), the Hungarian Algorithm based on Euclidean Distance (HED), and the Hungarian Algorithm based on Shortest Path (HSP). GED and GSP can be considered the most popular algorithms, because they resemble the most used form of order a taxi: either via radio or via

smartphone, the vehicle assignment systems try to identify an available taxi cab which is closest to the customer, so that it can answer the call. The use of an optimization algorithm, like HA, for the taxi cab assignment process achieved positive results (when using actual distance) when compared to instantaneous selection algorithm, based on greedy strategies.

Simulations were based on a map with some complexity presented in a small and a medium city (different speed limits, traffic lights, two-way and one-way streets, and so on). Additionally, the scenery simulation represented normal hours of service, with usually lower demand than the number of available taxis. Also, the time window used was arbitrarily chosen to be 50 seconds, and in non ordinary days, such as rainy days or special events (concerts, sports), this interval could have a different value. It is expected that in more complex maps, the difference between the algorithms evaluated become even more significant.

Current results are promising, since the possibility of involving the Designation Problem with different quantity and a more complex distribution of calls. This work directed the research focus on taxis but those algorithms can have applications in different situations, as fleets of private firms and even on autonomous cars research field.

Acknowledgments

The Authors would like to thank CAPES Foundation, under Grant 10224-12-2, for the financial support.

References

Acosta, A., Espinosa, J., and Espinosa, J. (2015). Traci4matlab: Enabling the integration of the sumo road traffic simulator and matlab® through a software re-engineering process. In Behrisch, M. and Weber, M., editors, *Modeling Mobility with Open Data*, Lecture Notes in Mobility, pages 155–170. Springer International Publishing.

Castelo-Branco, A. (2012). Demora no atendimento de táxi em bh leva 15% dos passageiros a cancelar pedido. [Online; Access Date: 5 julho 2014].

Castro, C. M. d. (2014). Viaduto da floresta tem 40mil carros por dia, diz pesquisa origem destino. Website. Disponível em: http://g1.globo.com/minas-gerais/noticia/2012/10/viaduto-da-floresta-tem-40-mil-carros-por-dia-diz-pesquisa-origem-destino.html.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Goldbarg, M. C. and Luna, H. P. L. (2005). *Otimização combinatória e programação linear: modelos e algoritmos*. Elsevier, Rio de Janeiro, RJ, Brasil.

IBGE (2014). Área territorial de bh. Website. Disponível em: http://www.ibge.gov.br/home/geociencias/areaterritorial/area.php?nome=Belo

Jonker, R. and Volgenant, T. (1986). Improving the hungarian assignment algorithm. *Operations Research Letters*, 5(4):171–175.

Jung, J., Jayakrishnan, R., and Park, J. (2015). Dynamic shared-taxi dispatch algorithm with hybrid-simulated annealing. *Computer-Aided Civil and Infrastructure Engineering*.

Konig, D. (1931). Gráfok és mátrixok. matematikai és fizikai lapok, 38.

Krajzewicz, D., Erdmann, J., Behrisch, M., and Bieker, L. (2012). Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

Liao, Z. (2009). Real-time taxi dispatching using global positioning systems. *Communications of ACM*, 46(5):81–83.

Liu, S., Wang, S., Liu, C., and Krishnan, R. (2015). Understanding taxi drivers' routing choices from spatial and social traces. *Frontiers of Computer Science*, 9(2):200–209.

Lopes, V. (2014). Bhtrans fecha o cerco ao táxi ocioso. Website. Disponível em: http://www.em.com.br/app/noticia/gerais/2013/02/15/interna_gerais,350571/bhtrans-fecha-o-cerco-ao-taxi-ocioso.shtml.

Reis, F. A. L., de Arruda Pereira, M., and Almeida, P. E. M. (2011). Location-based dispatch to reduce the waiting time for taxi services.

Santos, D. and Xavier, E. (2015). Taxi and ride sharing: A dynamic dial-a-ride problem with money as an incentive. *Expert Systems with Applications*, 42(19):6728–6737.

Steenbruggen, J., Tranos, E., and Nijkamp, P. (2015). Data from mobile phone operators: A tool for smarter cities? *Telecommunications Policy*, 39(3-4):335–346. New empirical approaches to telecommunications economics: Opportunities and challenges Mobile phone data and geographic modelling.

Taha, H. A. (2008). *Pesquisa Operacional: uma visão geral*. Person Prentice Hall, São Paulo, SP, Brasil.

Xu, Z., Yuan, Y., Jin, H., and Ling, H. (2005). Investigating the value of location information in taxi dispatching services: A case study of dazhong taxi. *Pacific Asia Conference on Information Systems*.

Yang, H. and Wong, S. (1998). A network model of urban taxi services. *Transportation Research Part B: Methodological*, 32(4):235–246.

Spatial visualization of job inaccessibility to identify transport related social exclusion

Pedro Logiodice[1], Renato Arbex[1], Diego Tomasiello[1], Mariana Giannotti[1]

[1]Departamento de Engenharia de Transportes da Escola Politécnica – Universidade de São Paulo (USP)
Caixa Postal 61548 – São Paulo – SP – Brazil

{pedro.logiodice, renatoarbex , diegobt86 , mariana.giannotti}@usp.br

Abstract. An evaluation of the spatial distribution of job accessibility in São Paulo was conducted through the correlation of work trips demand and socioeconomic data of public transport users. It was proposed a gravitational job inaccessibility index (GJI) that identifies higher volumes of home-based work trips with longer durations. Therefore, a characterization of the daily home-based work displacements of low-income population was obtained. The results show that, in general, half of the public transport users spend at least 2h30min daily in commutes. Finally, the GJI spatialization reveals the gravity force of production and attraction of work trips for the low-income population, as well as the city segregation.

1. Introduction

In the recent 21th century, the world witnessed an important milestone: for the first time in history the urban population overcame the rural population. Brazil, in turn, had gone thru this demographic shift 40 years ago, and, now, virtually 85% of the Brazilians live in urban areas [IBGE, 2010]. São Paulo, the main economic pole of Latin America, as well as many others cities in the developing countries [Davis, 2007], faced an extremely accelerated urbanization process: in 120 years it grew from a small city of around 65 thousand inhabitants to a huge urban conglomerate of more than 11,2 million people [IBGE, 2015].

This urban growth was fast and unplanned, mainly from center to peripheries [Carvalho, 2014]. The downtown area became economically developed, with a great variety of jobs and services, whereas the outskirts, the areas farther from the city center, grew progressively with lower-income population and lack of formal economic opportunities, as well as poor infrastructure [Villaça, 2011].

This spatial inequality of opportunities created an enormous transport demand. Today there are around 44 million daily trips with public and private transport in the SPMA [São Paulo metropolitan area]. Historically, in order to answer the increasing transportation demand, governments adopted policies that prioritized the use of private motorized transports instead of public transportation [Carvalho, 2014]. This increase in the use of private transportation is considered to be the main responsible for the urban mobility collapse [Silva Dias, 2014]. According to the São Paulo Origin

Destination (OD) Metro Survey conducted in 2007 [Metrô SP, 2015], the average travel time in the city is 2h42min and for 20% of the population it is higher than 4h.

A central issue in the transport research is the need to quantify and measure accessibilities inequalities [Bocarejo S. e Oviedo H., 2012]. Fortunately the advances on GIS, and Information, Communication and Positioning technologies - such as GPS systems - are nowadays used to monitor buses, increasing the amount of spatial data available, thus there is an enormous potential to use this information to understand the mobility patterns of the cities, as well as the demands of public transports [Yuan *et al*., 2012]. Hence, the GIS became a powerful tool system that can be used for both analyzing urban mobility and evaluating public policy, in order to democratize the public spaces [Salonen *et al*., 2014; Mavoa *et al*., 2012].

This paper uses public transport data to characterize daily home-based work displacements of the low-income population in São Paulo megacity and proposes a gravitational job inaccessibility index (GJI) to identify higher volumes of home-based work trips with longer durations. After this introduction a brief bibliographic review from the main topics is presented, followed by methods description, results discussions and conclusions remarks.

2. Job Accessibility as a mean to analyse social exclusion

Accessibility Definitions

Hansen (1959) defined accessibility as "the potential of opportunities for interaction", in other words, the measurement of the capacity of an individual to access a service, job or location. Thus, the level of accessibility relates closely to the development of the area used in a given location. Around a decade after Hansen's definition on the subject, Ingram (1971) complemented Hansen by defining accessibility as "the degree of interconnection with all other points on the same surface". This degree of interconnection is directly related to the capacity of a certain transport infrastructure to enable the displacement between those areas by overcoming the distance between them in a certain amount of time.

Therefore, accessibility could be understood as a measure in which exchange opportunities can be reached, considering the magnitude and the quality of each activity. This potential spectrum of social and economics interactions is inherent as part of the relative location advantage of a certain area, as well as the displacement conditions of the individual [Hansen, 1959; Ingram, 1971; Handy e Niemeier, 1997; Spiekermann e Wegener, 2006; Mouette, 1998; Spiekermann e Neubauer, 2002; Goto, 2000].

Poverty and Social Exclusion

Although the concepts of poverty and social exclusion are often used together, they are not the same, and it is important to distinguish them in order to understand these concepts in a deeper approach [Sposati, 1998; Church *et al*., 2000].

The definition of poverty is not clear because, besides being a multidimensional phenomenon, it varies according to the social, economic, politic, religious, cultural, and even geographic context. In general, it intimately relates to deprivation of basic needs such as nutrition, health, freedom, dignity and human rights [Sindzingre, 2005]. In order to characterize poverty from a quantitative perspective, it should be considered that numbers related to lack of goods overshadow the "politic core of poverty", that is to say, "*being poor isn't just not to possess, but mainly to be prevented to possess, what makes it more an issue of being than an issue of possessing*" [Demo, 1993].

Thereby, poverty does not exist as an state, but as a situation; situation of lack of opportunities to access services (health and education), urban infrastructure (basic sanitation and drinking water access), culture and justice [Maricato, 2003].

Social exclusion, in turn, consists in the lack of accessibility to jobs and services, being by distance, lack of transportation, social or economic reasons or any other reason [Church *et al*., 2000]. Therefore, social exclusion has, as well, a cultural and ethnic aspect, and is related to discrimination, stigmatization, loss of bonds, to the abandonment and to the fraying of the coexistent relationships [Sposati, 1998].

Kenyon *et al*. (2002) define social exclusion as:

> […] the unique interplay of a number of factors, whose consequence is the denial of access, to an individual or group, to the opportunity to participate in the social and political life of the community, resulting not only in diminished material and non-material quality of life, but also in tempered life chances, choices and reduced citizenship.

Sposati (2000) considers that a utopic referential of social inclusion would be guaranteed with seven fields: autonomy, life quality, human development, equity, citizenship, democracy and happiness.

Therefore, poverty is intimately related to resource, distribution, income and purchase power deprivations, while social exclusion is related to the (lack of) social, participation, coexistent relationships and citizenship.

Accessibility: a tool to tackle poverty and social exclusion

Economically, transport is a vital intermediate good that facilitates the production of a final good, as well as services, and its inefficiency inhibits the cities sustainable growth. Socially, it represents the physical opportunity for the population to access job, health, education and public equipment, required for the society's wellbeing. Furthermore, the lack of accessibility is considered to be the main cause of social exclusion of low-income urban areas [World Bank, 2002].

Kenyon et al. (2002) understand mobility-related exclusion as:

> […] the process by which people are prevented from participating in the economic, political and social life of the community because of reduced accessibility to opportunities, services and social networks, due, in whole or in part, to insufficient mobility in a society and environment built around the assumption of high mobility.

Church *et al*. (2000) discuss a framework that relates social exclusion and transport: physical exclusion, geographical exclusion, exclusion from facilities, economic exclusion, time-based exclusion, fear-based exclusion and space exclusion.

Besides this close relation, few studies relate social exclusion to transport. Most of the researches are limited to identify origins and causes in issues such as job market, housing or social inequality, but rarely take it to the account of transport as being one of the main factors. In general, these studies hardly consider detailed geographical factors, for instance, the relation between residence and desired activities locations, and the required transportation capacity from one to another [Church *et al*., 2000].

Hence, there are few researches that seek specifically to understand the travel needs of residents in areas with high levels of social exclusion, in particular the accessibility guaranteed by available transport infrastructure framework [Church et al., 2000; Bocarejo S. e Oviedo H., 2012]. The present work evaluates the spatial distribution of job accessibility through the correlation of work trips demand and socioeconomic data of public transport users in São Paulo megacity.

3. Methodology

Data

In order to achieve the aforementioned objectives, this study was based on the following data:

- Georeferenced Smart card transactions data of the São Paulo's public transport (2013);
- São Paulo's bus services average travel times calculated from GPS records of public transport (2015);
- Metro OD Survey of São Paulo (2007);
- Demographic Census of 2010 - Brazilian Institute of Statistic and Geography (IBGE);
- Annual Social Information Report of 2012 (RAIS), which contains information regarding workplaces, e.g. wage and job location.

The smartcards data analysis is from August 12th of 2013. It has information of all trips within this day, of all the users of the public transport network of São Paulo, with approximately 12 million transactions of around 4.5 million smart card holders. The smart card share use in São Paulo city is around 96% of all transactions.

In order to select only trips related to job activity, the focus of present paper, trips were filtered considering the time interval from consecutive transactions. Two trips with a time difference superior than 7h were selected as job related. The 7h parameter, called here by permanence time, equivalent to a work day period, is similar to the methodology presented by Munizaga *et al*. (2014). In addition, it was assumed the "destiny transaction" as the location where the transaction immediately after the longer permanence time occurred; and the "origin transaction" as the first of the day made by certain user. According to the hypothetic trip illustration showed below (Figure 1):

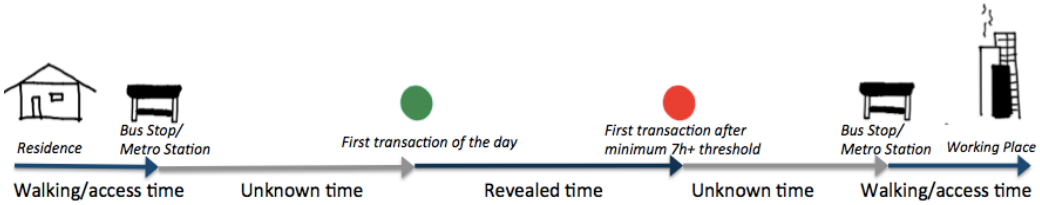


Figure 1: Schematic resume of the concepts

Where:

- Revealed time: Time difference between the transaction that occurred before the time interval of 7h+ hours and the transaction that occurred after this time period.
- Walking/access time: Time needed for the user to access the bus stop or metro station. This segment of the trip is unknown in smart card transaction records, and thus the walking distance is from the centroid of its related traffic zone to the bus stop will be used.
- Unknown time: As some few users don't tap immediately after boarding the bus, there is an unknown time gap between the boarding in the bus stop and the real observed transaction from smart card database. The authors are conducting an ongoing research to better study this issue in order to reduce this trip origin uncertainty.

From the GTFS data processed it was developed a network at a GIS environment, with the real velocity of the public transport system in São Paulo. From this network, a time matrix of the trips for each OD pair was calculated, for real travel time values of the average business day at 7 am.

4. Methods

In a GIS environment a spatial analysis was made in order to relate jobs supply georeferenced data, smart card inferred origin and destination locations and socio demographic data from census, as described in figure 2.

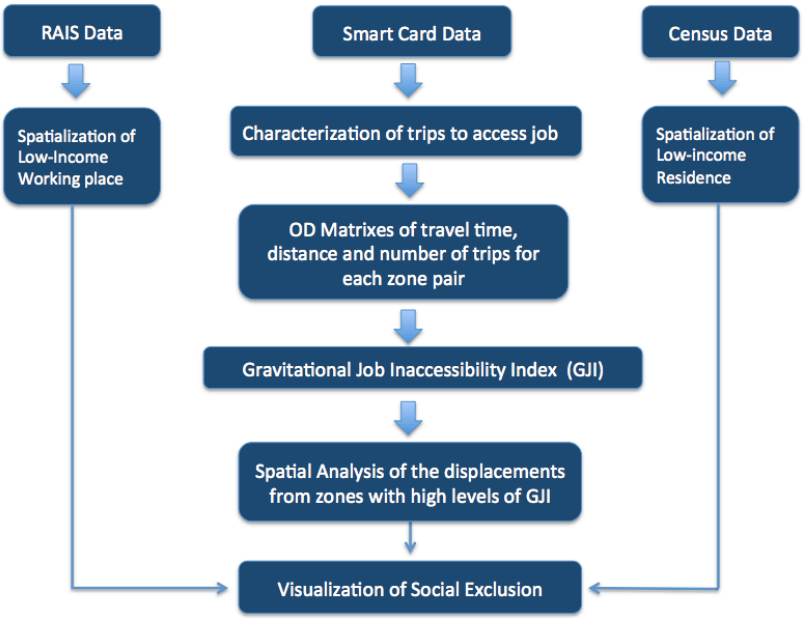


Figure 2: Proposed methods.

Gravitational job inaccessibility index (GJI)

A main issue in the transport research is the development of accessibility analytic tools that can quantify and identify access inequalities, evaluate existing projects and likewise prioritize needs [Bocarejo S. e Oviedo H., 2012].

Therefore, the objective of the gravitational job inaccessibility index (GJI), proposed in this paper, is to identify zones with high production of long trips. Thereby, this index is in directly proportional to the trip duration and to the number of trips produced in certain zones, and inversely proportional to the trip speed:

$$I_i^n = \frac{b_i}{B}\sum_{j=1}^{n}\frac{t_{ij}}{v_{ij}} = \frac{b_i}{B}\sum_{j=1}^{n}\frac{t_{ij}^2}{d_{ij}}$$

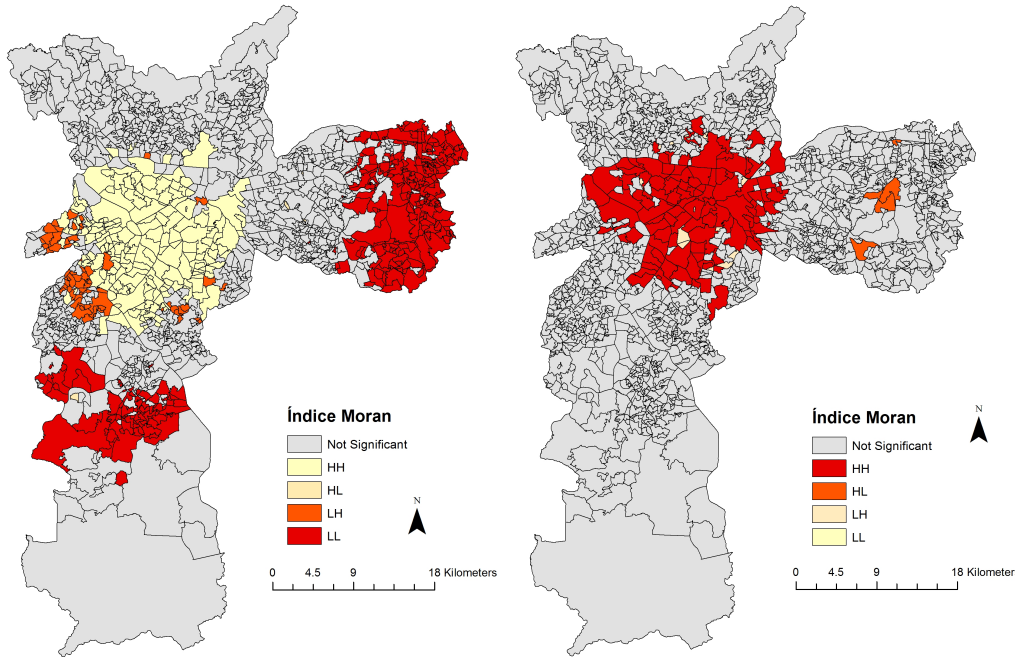
Where: n= number of OD zones; i= zone of trip production; j= zone of trip destination; v= speed; b= number of smart card transaction for this OD pair; t= trip duration between zone centroids; d= length between centroids; e B= total number of smart card transactions.

5. Results and Analyses

Spatial analysis of low-income population and its working places

A Moran spatial correlation and two Local Indicator of Spatial Association (LISA) maps were prepared from the average income data in São Paulo (figure 3a) and the workplaces with salary till two minimum wages (figure 3b). The Moran index obtained for the average income was 0,615 and for the working places were 0,620, indicating in both cases a positive spatial correlation.

In the average income Lisa map (figure 3a) it is possible to observe that there are low-income clusters in the east and south of São Paulo, as well as a high-income cluster in the central areas. In contrast, the Lisa map of working places with salary under two minimum wages (figure 3b) show a huge cluster in the central area of the city and virtually no significant clustering in the zones of low income residences.



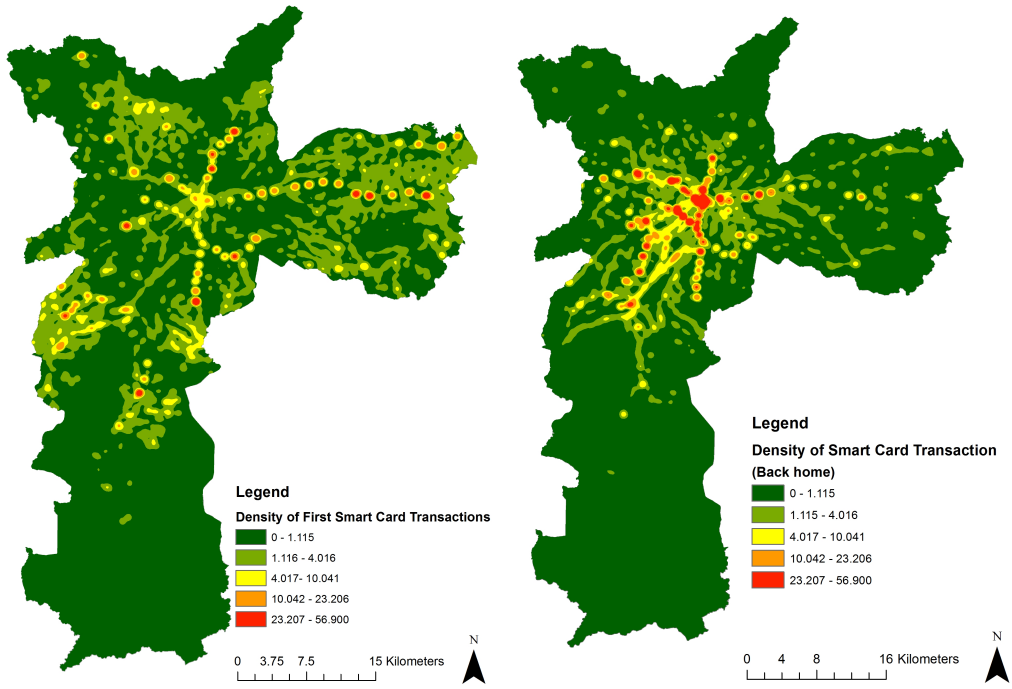
(a) **(b)**

Figure 3: Local Indicator Spatial Association - LISA map [(a) Average income in São Paulo/ (b) Working places with salaries under two minimum wages].

Spatial analysis of smart card transaction records

The Kernel estimator maps showed in figure 4 represents the spatial distributions of all the first smart card transactions (4a) and the first return home transaction (4b) made by

around 1.8 million public transport users whose trips are assumed as work trips (those trips with permanence time higher than 7h between two single transactions). It is possible to notice that there are some relevant aspects of this spatial distribution. Firstly, the high density of first transaction in metro and train stations evidences the users' behavior of tapping the smart card only near alighting for the first transfer, fact that brings the origin uncertainty discussed previously. Secondly, the high density of transactions in terminals of the metro and train lines (figure 4a) demonstrates the gravity force that the rail transport network has over the areas with low accessibility indexes, as the São Paulo public transport network has a trunk-feeder system in those areas, concentrating bus lines to metro and rail network stations. Thirdly, it is also noticed (figure 4b) the high density of first transaction to residency (return) in the center of the city, evidencing the monocentric structure of São Paulo.



(a) (b)

Figure 4: Kernel estimator of the first transaction (a) and the first return home transaction (b)

Spatial analysis of the gravitational job inaccessibility index (GJI)

The thematic map of figure 5 was generated from the gravitational job inaccessibility index (GJI). It is clear that, in general, central zones have a lower GJI than the peripheral, what is consistent with the respective transport infrastructure and workplaces characteristics. In addition, it is noticeable that virtually all the zones with high GJI are also outliers of low-income represented in the map of figure 5. Another aspect that is worth discussing is the influence of São Paulo metropolitan area (SPMA). Thereby many users start their journey in a neighboring city, which is in some cases even farther from the job opportunities of the central areas. That is in part

responsible for the spatial pattern of peripheral zones with high GJI showed in figure 5. Far zones with low GJI indicate that the number of users is relatively lower.

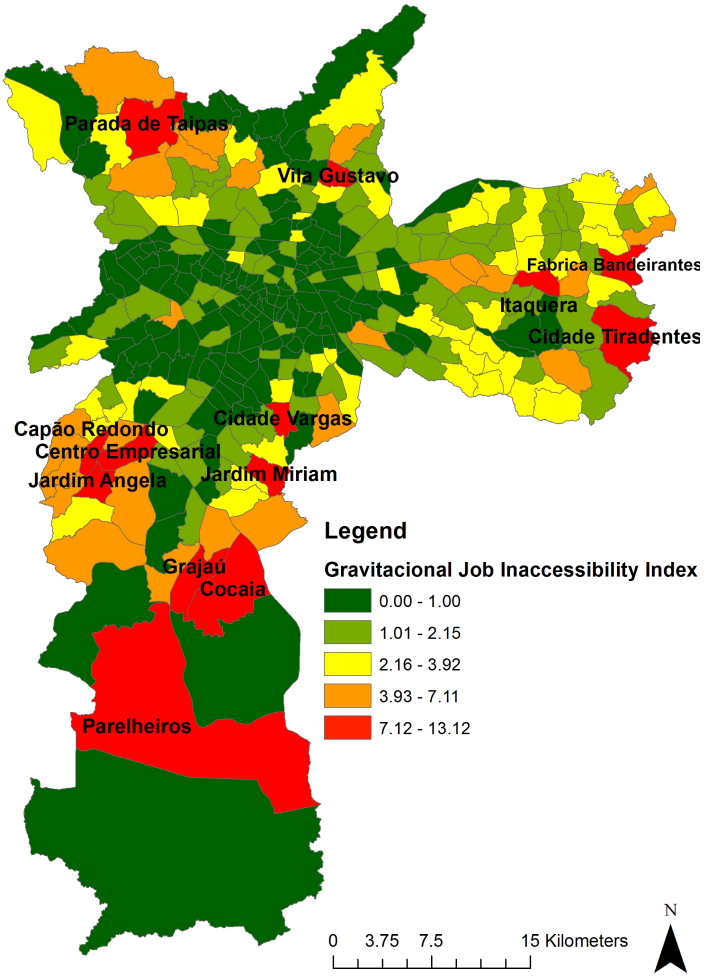
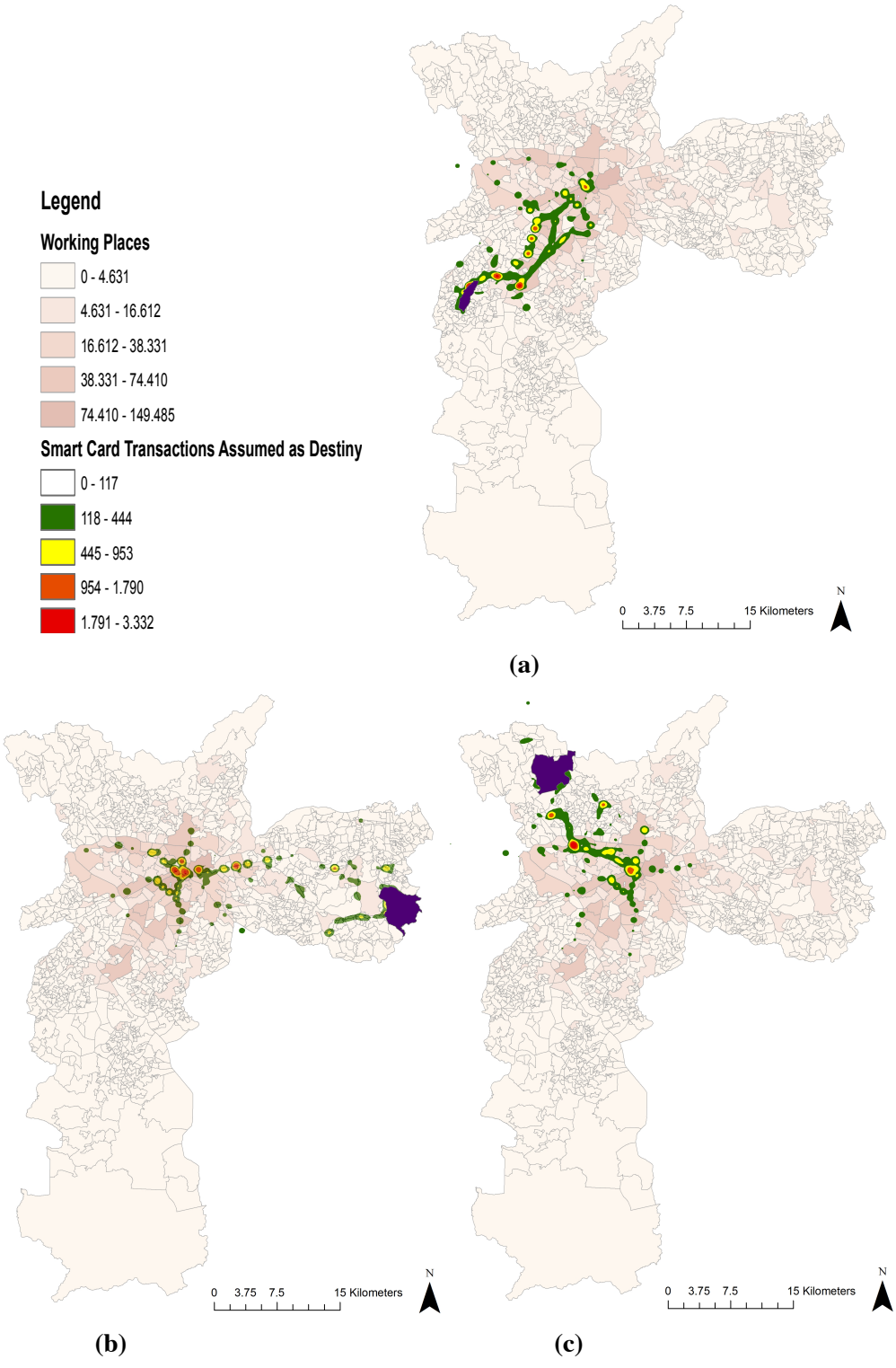


Figure 5: Gravitational job inaccessibility index map.

Following the maps presented in figure 6 illustrates the gravity forces that the working places exert over the zones with high GJI, which are, in general, lacking job opportunities. It is also interesting to observe the magnitude of the attraction force that the central area exerts over even really distant zones. In addition, it is noticeable that the trip destination of the south areas users are more distributed to multiple poles of attraction in the south region, whereas in the east the displacements are lengthier and the main destination areas are more concentrated in the downtown region of the city, and highly concentrated around metro stations (figure 4).


Legend
Working Places
0 - 4.631
4.631 - 16.612
16.612 - 38.331
38.331 - 74.410
74.410 - 149.485
Smart Card Transactions Assumed as Destiny
0 - 117
118 - 444
445 - 953
954 - 1.790
1.791 - 3.332
0 3.75 7.5 15 Kilometers
N
(a)
(b)
(c)

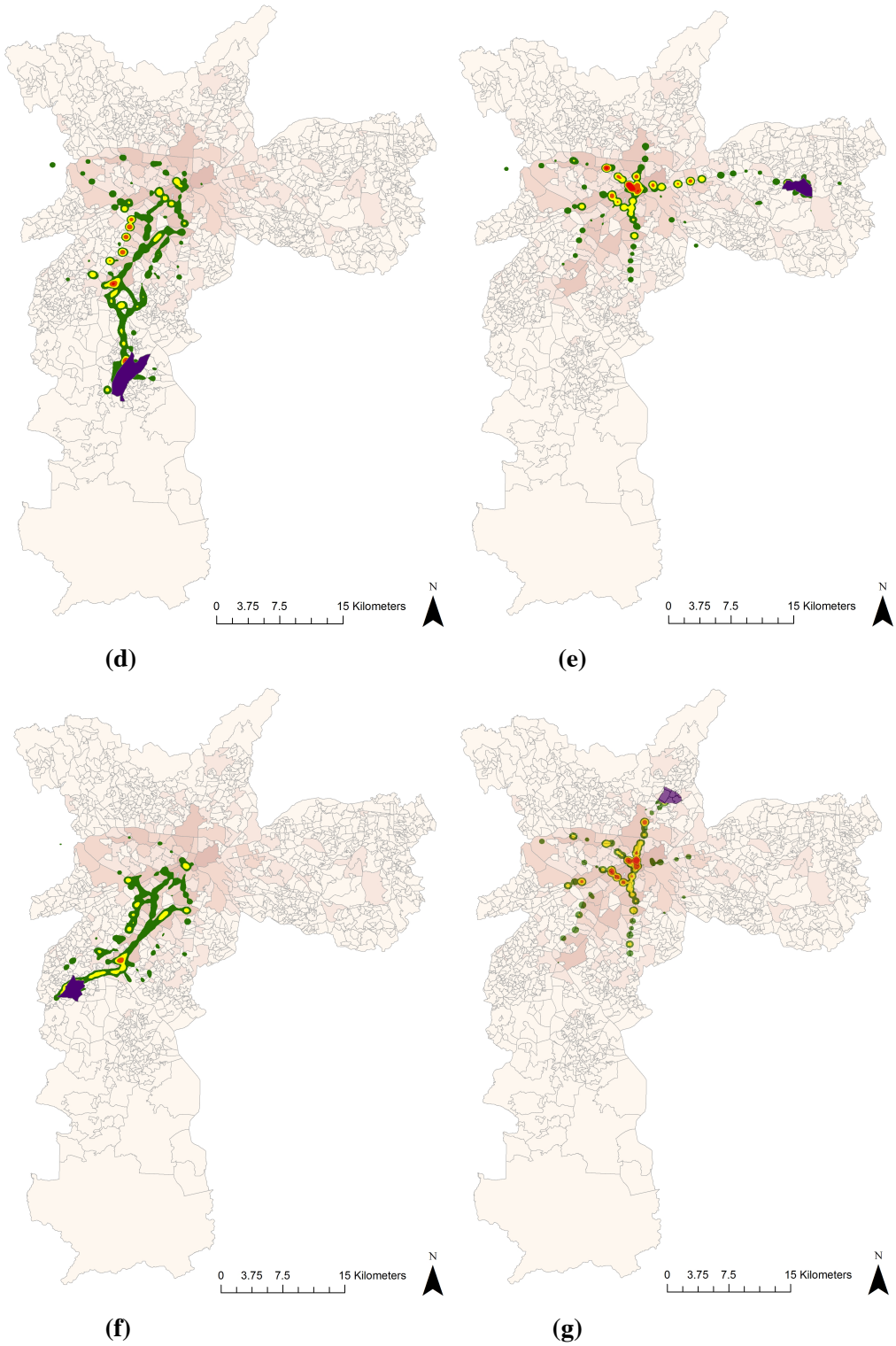


Figure 6: Smart Card Transactions Assumed as Destiny from high GJI zones

[(a) Capão Redondo / (b) Cidade Tiradentes/ (c) Parada de Taipas / (d) Grajaú / (e) Itaquera / (f) Jardim Ângela / (g) Vila Gustavo]

This patterns of lengthier displacement are in part responsable for the long duration of the trips to access the jobs. This study also showes that, in general, people spend, on average, 1h to access their jobs, and half of the public transport users spend at least 2h30min daily in commutes.

6. Conclusions

An evaluation of the spatial distribution of job accessibility in São Paulo was conducted through the correlation of home-based work trips demand and socioeconomic data of public transport users. It was proposed a gravitational job inaccessibility index (GJI) that identifies higher volumes of home-based work trips with longer durations. Origin-destination pairs of low-income population work trips were identified, and the proposed index described the job accessibility disparity between regions in the city.

A characterization of the daily home-based work displacements of this group of low-income population was obtained. The results helped to understand the gravity forces of production and attraction of work trips for the low-income population. In addition, the GJI spatialization clearly shows the consequences of social segregation, since the areas with high levels of inaccessibility are mainly peripheral areas with low-income residents and lack of job opportunities. Also, the study shows that, in general, people spend, on average, 1h to access their jobs, and half of the public transport users spend at least 2h30min daily in displacements.

The exploratory spatial analysis enabled the comprehension of the spatial distribution of the low-income population, likewise the city workplaces, representing respective potential poles of origin and destination. It was possible to conclude that São Paulo has in fact a high level of mono-centrality, characteristic that was also appointed by works of Ramos (2014). Bessa, Colli and Paula (2011) showed that not only the central area represents only 7% of the city, it also concentrates more than 2/3 of the workplaces and services.

Furthermore, in consistence with Comin (2011), this paper showed the spatial segregation of the peripheral areas, where in general reside the low-income population. In conclusion, this paper enables to visualize clearly the negative influence that the land use inequality represents on public transport demands.

7. Bibliography

Bessa, V., Colli, J., Wissenbach, T., Paula, A. (2011) "Território e desenvolvimento econômico", Atlas Geoeconômico da Cidade, Metamorfoses Paulistanas da prefeitura de São Paulo.

Bocarejo S., J. P., e Oviedo H., D. R. (2012) "Transport accessibility and social inequities: a tool for identification of mobility needs and evaluation of transport

investments". Journal of Transport Geography, 24, 142–154. doi:10.1016/j.jtrangeo.2011.12.004

Carvalho, C. (2014), "Dos trilhos para o asfalto", Caderno Mobilidade Urbana, São Paulo, Globo Comunicação e Participações S.A., p.18-25.

Church, a., Frost, M., e Sullivan, K. (2000) Transport and social exclusion in London. Transport Policy, 7(3), 195–205. doi:10.1016/S0967-070X(00)00024-X

Comin, A. (2011) "A economia e a cidade: metamorfoses paulistanas", Atlas Geoeconômico da Cidade, Metamorfoses Paulistanas - da prefeitura de São Paulo.

Demo, P (1993) "Pobreza política". Papers. São Paulo, Fundação Konrad Adenauer-Stiftung.

Ermínia, M. (2003) "Metrópole, legislação e desigualdade." Estudos avançados, http://dx.doi.org/10.1590/S0103-40142003000200013, July.

Goto, M. (2000) "Uma análise de acessibilidade sob a ótica da eqüidade-o caso da Região Metropolitana de Belém", São Carlos. 77p. Dissertação (Mestrado)-Escola de Engenharia de São Carlos, Universidade de São Paulo.

Handy, S. L., & Niemeier, D. A. (1997) "Measuring accessibility: an exploration of issues and alternatives", Environment and planning A, 29(7), 1175-1194.

Hansen, W. G. (1959) "How accessibility shapes land use", Journal of the American Institute of Planners, 25(2), 73-76.

I. B. G. E., Instituto Brasileiro de Geografia e Estatística, http://www.censo2010.ibge.gov.br/sinopse/index.php?dados=6&uf=00, August.

I. B. G. E., Instituto Brasileiro de Geografia e Estatística, "Censo" http://www. ibge. gov. br/home/estatistica/populacao/censo2010, June."

Ingram, D. R. (1971) "The concept of accessibility: a search for an operational form", Regional studies, 5(2), 101-107.

Kenyon, S., Lyons, G., e Rafferty, J. (2002) "Transport and social exclusion: Investigating the possibility of promoting inclusion through virtual mobility", Journal of Transport Geography, 10(3), 207–219. doi:10.1016/S0966-6923(02)00012-1

Mavoa, S., Witten, K., Mccreanor, T., e Sullivan, D. O. (2012) "GIS based destination accessibility via public transit and walking in Auckland" , New Zealand. Journal of Transport Geography, 20(1), 15–22. doi:10.1016/j.jtrangeo.2011.10.001

Metrô SP. (2015) "Pesquisa Origem Destino de São Paulo" http://www.metro.sp.gov.br/metro/numeros-pesquisa/pesquisa-origem-destino-2007.aspx. August.

Munizaga, M., Devillaine, F., Navarrete, C., e Silva, D. (2014) "Validating travel behavior estimated from smartcard data", Transportation Research Part C: Emerging Technologies, 44, 70–79. doi:10.1016/j.trc.2014.03.008

Ramos, F. (2014) Três Ensaios sobre a Estrutura Espacial Urbana em Cidades do Brasil Contemporâneo.

Salonen, Maria, *et al*. (2014) "Do suburban residents prefer the fastest or low-carbon travel modes? Combining public participation GIS and multimodal travel time analysis for daily mobility research." Applied Geography 53: 438-448.

Silva, J.L. (2014), "Uma nova política de mobilidade urbana", A teoria e debate, n121, february, São Paulo, Editor Perseu Abramo,

Sindzingre, A. (2005). The Multidimensionaly of Poverty: An Institutionalist Perspective. Conference The Many Dimensions of Poverty. International Poverty Center, United Nations Development Programme (UNDP), 29 a 31 de Agosto, Brasilia, DF, Brasil.

Spiekermann, K., & Wegener, M. (2006) "Accessibility and spatial development in Europe", Scienze Regionali, 5(2), p.15-46.

Spiekermann, Klaus, and Jörg Neubauer (2002), "European accessibility and peripherality: Concepts, models and indicators".

Sposati, A. (2000) "Cidade, Território, Exclusão/Inclusão Social", Congresso Internacional de Geoinformação – GEO Brasil, 1–7. http://www.dpi.inpe.br/geopro/exclusao/cidade.pdf

Sposati, Aldaíza. Exclusão social abaixo da linha do Equador. In: Véras, Maura Padini Bicudo (ed.). Por uma Sociologia da Exclusão social: o debate com Serge Paugam. São Paulo: Educ: 1999. Pp.126-138.

Villaça, F. (2011) "Urban segregation and inequality", Estudos Avançados, http://dx.doi.org/10.1590/S0103-40142011000100004, July.

World Bank (2002) "Cities on the Move", doi:10.1596/0-8213-5148-6

Yuan, Y., Raubal, M., & Liu, Y. (2012) "Correlating mobile phone usage and travel behavior–A case study of Harbin, China", Computers, Environment and Urban Systems, pages 118-130.

Desafios no Mapeamento de Esquemas Conceituais Geográficos para Esquemas Físicos Híbridos SQL/NoSQL

Danilo B. Seufitelli, Mirella M. Moro, Clodoveu A. Davis Jr.

Universidade Federal de Minas Gerais, Belo Horizonte – MG – Brazil

{daniloboecha, mirella, clodoveu}@dcc.ufmg.br

Abstract. *To the best of our knowledge, there is no generic mapping from conceptual schemas to NoSQL physical schemas. This paper tackles such problem in the context of geographic databases. We discuss the solution of mapping conceptual schemas to hybrid relational/NoSQL physical schemas.*

Resumo. *Até onde pudemos determinar, não existem ainda propostas genéricas para produzir esquemas físicos para estruturas complexas NoSQL (documentos, grafos, etc). Este artigo apresenta questionamentos quanto ao mapeamento da modelagem conceitual para esquemas físicos híbridos, de modo a conciliar modelos relacionais e não relacionais.*

1. Introdução

A modelagem conceitual de dados geográficos envolve abstrações que vão além da expressividade dos modelos de dados convencionais. Modelos de dados geográficos, como o OMT-G [1], incluem primitivas para definir alternativas de representação e relacionamentos espaciais. O mapeamento de esquemas conceituais geográficos para esquemas lógicos e físicos precisa levar em conta a semântica dessas primitivas e definir a implementação final em um sistema de gerenciamento de bancos de dados (SGBD) geográfico, como o PostGIS ou o Oracle Spatial. Embora esse mapeamento tenha sido estudado para o caso de bancos de dados objeto-relacionais espacialmente estendidos [6], a crescente disponibilidade de gerenciadores NoSQL indica que podem existir situações em que seu uso em aplicações pode ser mais vantajoso. Por exemplo, um SGDB NoSQL orientado a grafos oferece melhor desempenho em tarefas de roteamento, como demonstrado em [11].

Por outro lado, a implementação de restrições de integridade espaciais [3] é uma tarefa típica dos SGBDs tradicionais, porém não disponível nos gerenciadores NoSQL. Assim, acreditamos que existam situações em que um enfoque híbrido, ou seja, esquemas físicos que combinem SGBD relacionais e NoSQL, seja o mais indicado. Portanto, este trabalho propõe avaliar o potencial para mapeamento de esquemas conceituais geográficos para esquemas híbridos, com componentes relacionais e NoSQL.

A seguir, a Seção 2 discute brevemente trabalhos relacionados. A Seção 3 apresenta o processo de mapeamento de dados geográficos para diferentes representações NoSQL, apontando diversos desafios na hora de definir tais mapeamentos. É também apresentada uma discussão sobre os desafios encontrados. A Seção 4 conclui este artigo.

2. Trabalhos Relacionados

O OMT-G é um modelo de dados orientado a objetos que oferece primitivas para a modelagem da geometria e da topologia dos dados espaciais através de três conceitos principais: classes, relacionamentos e restrições de integridade espaciais [1]. O modelo permite

a especificação de diferentes alternativas de representação geográfica e classes de objetos com múltiplas representações.

O mapeamento de esquemas OMT-G para esquemas lógicos e de implementação foi estudado por Hora et al. [6], tendo como alvo SGBD objeto-relacionais e esquemas GML. Foram definidos algoritmos que estabelecem a equivalência entre representações conceituais mais complexas em OMT-G e estruturas de representação geográfica mais simples (e.g., pontos, linhas, polígonos) para o esquema de implementação, adicionando elementos para a implementação concomitante de restrições de integridade, de modo a respeitar a semântica das primitivas conceituais. Esses elementos incluem asserções (`CHECK`), restrições convencionais (`CONSTRAINT`s) e funções de verificação topológica implementadas como gatilhos (*triggers*).

O mapeamento para SGBDs NoSQL, por outro lado, não é ainda abordado na literatura. Isso provavelmente decorre do fato de existirem sistemas NoSQL em quatro arquiteturas distintas de armazenamento de dados: chave-valor, orientado a colunas, orientado a documentos e orientado a grafos [7, 10]. Dentre os argumentos para adoção de SGBD NoSQL estão o crescimento horizontal escalável, que visa prover uma grande quantidade de operações de leitura e escrita por segundo. Esses sistemas também notabilizam-se por serem replicáveis, potencialmente distribuídos entre vários servidores, e terem interface ou protocolo de acesso simples. Além disso, possuem um sistema de paralelismo e controle de concorrência menos estrito que o gerenciamento de transações em bancos relacionais, com distribuição eficiente de índices e uso intensivo de memória. Outra característica importante é ter a possibilidade de realizar alterações estruturais dinâmicas, em contraste com a relativa rigidez das estruturas tabulares dos SGBD relacionais.

Bugiotti et al. [2] propuseram uma metodologia de projeto de banco de dados NoSQL com o objetivo de projetar uma "boa"representação de dados NoSQL e visando obter escalabilidade, desempenho e consistência em aplicações Web da nova geração. Experimentos mostraram que o projeto de implementação deve ser conduzido com cuidado, pois o desempenho e a coerência das operações de acesso aos dados podem ser consideravelmente afetados.

Com relação a esquemas híbridos, Moro et al. [9] descrevem o ReXSA, uma ferramenta para projetar esquemas de banco de dados que combinam o armazenamento de dados relacionais e dados XML. Em outras palavras, o ReXSA avalia e recomenda um esquema de banco de dados que harmoniza modelos de dados relacionais e XML.

Assim como [9], também consideramos mapear esquemas conceituais para esquemas que combinem dados de naturezas diversas. Porém, este trabalho difere dos trabalhos citados em dois pontos: ao contário dos demais que consideram o projeto de bancos de dados NoSQL para dados convencionais, nós focamos nas especificidades dos dados geográficos; e estudamos questões relativas ao processo de mapeamento de esquemas conceituais geográficos para esquemas de implementação híbridos NoSQL/SQL.

3. Processo de Mapeamento

O mapeamento de um esquema conceitual geográfico para um esquema de implementação envolve decisões sobre diversos fatores. Com a flexibilidade dos distintos formatos utilizados por gerenciadores NoSQL, um fator a ser analisado é o tipo de armazenamento

Figura 1. Esquema OMT-G: Representação de ruas e bairros

físico a ser utilizado (e.g. chave-valor, documentos, grafos e família de colunas). Em cada um desses formatos, existem diversas possibilidades de organização dos dados e recursos de indexação. Há ainda a dificuldade na representação dos relacionamentos espaciais e restrições de integridade espaciais, visto que gerenciadores NoSQL são livres de esquema e não possuem o mesmo conceito de chave estrangeira dos SGBD relacionais.

Para dados geográficos, muitos dos bancos de dados NoSQL adotam o formato GeoJSON, que é um formato para a codificação de uma variedade de estruturas de dados geográficos. O GeoJSON adere aos padrões estabelecidos pelo Open Geospatial Consortium (OGC) e suporta os seguintes tipos de geometria: *Point, LineString, Polygon, MultiPoint, MultiLineString,* e *MultiPolygon*. Listas de geometrias são representadas por um *GeometryCollection*. Geometrias com propriedades adicionais são objetos *Feature*, e as listas de características são representados pela *FeatureCollection*.

Entretanto, dadas a complexidade e as peculiaridades das aplicações geográficas, criar uma estrutura de banco de dados por meio de esquemas GeoJSON não é uma atividade simples. Além disso, é mais fácil especificar e entender os conceitos e os relacionamentos de um sistema usando um esquema conceitual geográfico, para aproveitar a natureza visual dos diagramas de classes e outras primitivas, antes de tentar codificar diretamente as estruturas de banco de dados em esquemas GeoJSON.

Como exemplo da modelagem conceitual, a Figura 1 contém um fragmento de diagrama de classes utilizando o OMT-G. A cidade é formada por um polígono que é subdividido em bairros (subdivisão planar), de modo a estabelecer uma relação de composição espacial: toda cidade é formada por bairros, e não há bairros que se sobrepõem e nem que excedam os limites das cidades. Cada bairro possui segmentos de rua, relacionados em rede com seus cruzamentos. Nesse tipo de relacionamento, deve ser assegurado que, para cada nó exista pelo menos um arco, e a cada arco correspondam sempre dois nós.

Para exemplificar as dificuldades do mapeamento entre esquemas conceituais geográficos e esquemas de implementação NoSQL, apresentamos a seguir o mapeamento do esquema da Figura 1 para de três tipos distintos de gerenciadores NoSQL: orientados a documentos, a grafos e a família de colunas.

Mapeamento para SGBD Orientado a Documentos. Bancos de dados orientados a documentos utilizam um conjunto de coleções de atributos e valores, onde um atributo pode ser multivalorado, formando assim os documentos. Estes documentos são autodescritivos, com uma estrutura hierárquica em árvore, que pode conter mapas, coleções e valores escalares [5]. Neste artigo, é considerado o MongoDB como exemplo de gerenciador NoSQL orientado a documentos, pois suporta dados geográficos (GeoJSON) com o uso de índices espaciais. O mapeamento do diagrama da Figura 1 para MongoDB utilizando o GeoJSON como esquema de implementação é apresentado na Figura 2.

Existem diversas dificuldades quanto à representação das características geográficas no MongoDB. A primeira é a representação dos relacionamentos, pois a solução

Figura 2. Ruas e bairros na modelagem do GeoJSON

apresentada não especifica os relacionamentos espaciais entre as classes. As alternativas de solução incluem a utilização de pares chave-valor para esta representação e a utilização de vetores de subdocumentos. Com subdocumentos são três possibilidades: *(i)* segmentos de logradouro como subdocumento de cruzamento de rua (Cidade (Bairro (Cruzamento (Segmento)))); *(ii)* cruzamentos de vias como subdocumentos de segmentos de logradouro (Cidade (Bairro (Segmento (Cruzamento)))); *(iii)* todos os documentos em um mesmo nível, formando um documento único (Cidade, Bairro, Segmento, Cruzamento).

Tal diversidade provoca uma série de questionamentos, como por exemplo, determinar a melhor alternativa para atualização dos dados. A abordagem com todos os documentos aglomerados em um único documento (mesmo nível) provoca redundância dos dados. Desse modo, é necessário identificar se o tempo gasto para uma atualização de tais dados justificaria o uso de tal abordagem. Nota-se que este problema de padrões de projeto é conhecido no contexto de dados XML, no qual existem diversos formatos para os esquemas, que podem variar de acordo com o número dos seus elementos globais ou tipos (bonecas russas, fatia de salame, persianas, e o jardim do Eden) [9].

Outra dificuldade é a verificação das restrições de integridade espaciais, que em soluções SQL é realizada em *triggers*. No caso do MongoDB, verificar tais restrições requer implementar funções que utilizem as operações de log (*oplog*) para simular ações das tradicionais *triggers* de bancos de dados SQL.

Mapeamento para SGBD Orientado a Grafos. SGBDs orientados a grafos não possuem esquema, e dados são coleções de nós e arestas interligados em grafo. Cada nó representa uma entidade (ex., cidade ou bairro) e cada aresta uma ligação ou relação entre dois nós [4, 8]. Aqui, consideramos o gerenciador Neo4J para representar esta categoria, pois possui módulo para dados geográficos (Neo4J Spatial). A Figura 3 ilustra duas possibilidades para o armazenamento físico dos dados modelados em grafos: mapear diretamente cada classe para um nó (Figura 3a), e mapear as classes que representam a rede urbana com cruzamentos de rua como nós e segmentos de rua como arestas (Figura 3b).

Para verificar as restrições de integridade espaciais com o Neo4J, pode-se utilizar o *TransactionEventHandler*, similar às *triggers* dos SGBDs relacionais. Os relacionamentos espaciais podem ser representados através dos arcos que ligam os nós, como por exemplo a rede urbana formada por segmentos de rua e seus cruzamentos, conforme a Figura 1. Embora a representação em grafo apresente dificuldades no processamento de certos tipos de consultas (ex. vizinhos mais próximos), oferece melhor desempenho em tarefas de roteamento e conectividade em rede [11].

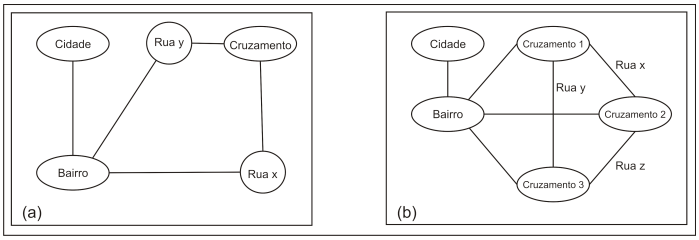


Figura 3. Ruas e bairros na modelagem em grafos

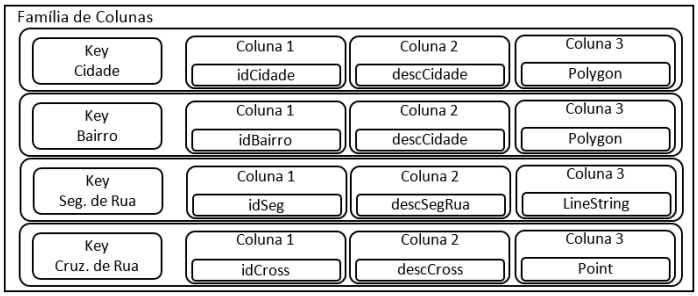


Figura 4. Ruas e bairros na modelagem em família de colunas

Mapeamento para SGBD Orientado a Colunas. Enquanto o modelo relacional define uma tabela como uma coleção de linhas, o modelo orientado a colunas (como o nome indica) organiza segundo uma coleção de colunas [12]. Aqui, consideramos o MonetDB como exemplo de gerenciador NoSQL orientado a colunas, pois este SGBD possui o *geom*, que é sua extensão para dados espaciais. A Figura 4 mapeia a Figura 1 onde cada classe corresponde a uma família de colunas, e cada uma dessas famílias possui uma chave única (*key*). Ainda é necessário usar chaves estrangeiras para representar o relacionamento entre as entidades. Outra possibilidade requer redundância dos dados, em que cada família de colunas seria uma subfamília de colunas da anterior, formando uma superfamília de colunas. O MonetDB suporta o uso de *triggers* para a verificação das restrições de integridade espaciais.

Discussão. Vários fatores da modelagem física de dados podem influenciar diretamente o desempenho dos sistemas gerenciadores NoSQL para aplicações geográficas [11]. Algumas dessas questões foram apresentadas e discutidas neste artigo, e resumidos pela Tabela 1. Considere $\surd$ para sim, $\pm$ para parcialmente e χ para não.

A variedade de soluções, estruturas e esquemas de implementação indicam que, em princípio, esses tipos de SGBD podem atuar de forma complementar. Enquanto, por exemplo, a consulta a estruturas em rede é mais eficiente se realizada em um SGBD orientado a grafos, e hierarquias territoriais são mais bem representadas em um SGBD orientado a documentos, a atualização de dados respeitando restrições de integridade espaciais

Tabela 1. Comparação entre gerenciadores NoSQL

	MongoDB	Neo4J	MonetDB
Trigger p/ RIE	χ	$\pm$	$\surd$
Relacionamentos	$\pm$	$\surd$	$\pm$
Redundância	$\surd$	χ	$\pm$
ACID	χ	$\surd$	$\surd$
SQL	χ	χ	$\surd$

(RIE) é provavelmente melhor executada em SGBD relacionais. Um estudo comparativo de desempenho foi apresentado por Santos et al. [11], confirmando o potencial para criação de esquemas de implementação híbridos, usando o melhor de cada alternativa.

4. Conclusões

Neste artigo, apresentamos as dificuldades de mapear dados geográficos para SGBDs NoSQL. É importante que as questões levantadas sejam avaliadas levando em consideração todos os tipos de modelagem física que os sistemas gerenciadores NoSQL utilizam (documentos, grafos, colunas, etc.). Novos estudos permitirão melhorar o mapeamento da modelagem conceitual para a modelagem física, unindo as características peculiares aos tipos de relacionamentos espaciais com os formatos que ofereçam melhor desempenho em consultas e atualizações de dados. Tais aspectos são os grandes gargalos de uma aplicação geográfica, bem como uma análise de parâmetros intrínsecos à aplicação a ser construída. Por exemplo, quais os tipos de dados mais frequentes, qual a carga de trabalho que será submetida ao SGBD e qual será a utilização mais frequente, entre consultas e atualizações. Desta forma, o objetivo será mapear um esquema conceitual para um modelo físico híbrido SQL/NoSQL de dados geográficos, que seja capaz de reunir as melhores características de cada paradigma para obter o máximo de desempenho em aplicações geográficas.

Agradecimentos. Trabalho parcialmente financiado por CAPES, CNPq e FAPEMIG.

Referências

[1] K. A. V. Borges, C. A. Davis Jr., and A. H. F. Laender. OMT-G: an object-oriented data model for geographic applications. *GeoInformatica*, 5(3):221–260, 2001.

[2] F. Bugiotti, L. Cabibbo, P. Atzeni, and R. Torlone. Database Design for NoSQL Systems. In *ER*, pages 223–231, 2014.

[3] S. Cockcroft. A taxonomy of spatial data integrity constraints. *GeoInformatica*, 1(4):327–343, 1997.

[4] H. Hashem and D. Ranc. An Integrative Modeling of Bigdata Processing. *International Journal of Computer Science and Applications*, 12(1):1–15, 2015.

[5] C. He. Survey on NoSQL Database Technology. *JASEI*, pages 50–54, 2015.

[6] A. C. Hora, C. A. Davis Jr, and M. M. Moro. Mapping Network Relationships from Spatial Database Schemas to GML Documents. *JIDM*, 2(1):67–74, 2011.

[7] K. Kaur and R. Rani. Modeling and querying data in NoSQL databases. In *Int'l Conference on Big Data*, pages 1–7, 2013.

[8] L. B. Marinho et al. Extracting geospatial preferences using relational neighbors. *JIDM*, pages 364–478, 2012.

[9] M. M. Moro, L. Lim, and Y.-C. Chang. Schema advisor for hybrid relational-XML DBMS. In *SIGMOD*, pages 959–970, 2007.

[10] P. J. Sadalage and M. Fowler. *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education, 2012.

[11] P. O. Santos, M. M. Moro, and C. A. Davis Jr. Comparative Performance Evaluation of Relational and NoSQL Databases for Spatial and Mobile Applications. In *DEXA*, 2015.

[12] M. Saxena, Z. Ali, and V. K. Singh. NoSQL Databases-Analysis, Techniques, and Classification. *JoADMS*, 1(2):13–24, 2014.

GeoSQL+: Um Aplicativo Online de Apoio ao Aprendizado de SQL com Extensões Espaciais

Guilherme Henrique R. Nascimento[1], **Clodoveu A. Davis Jr.**[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)
Av. Presidente Antônio Carlos, 6627 – 31270-901 – Belo Horizonte – MG – Brasil

{guilherme, clodoveu}@dcc.ufmg.br

Abstract. *Many tools focused on learning the SQL language are available online. None of them, however, focus on spatially extended databases, in which queries can involve geometric representations, topological functions, and distance calculations. This paper describes GeoSQL+, an online tool that allows posting SQL queries to a geographic database manager and visualizing the results as tables and maps. Visual results can be accumulated and interactively manipulated, composing layers with user-configurable presentation that simulate the operation of geographic information systems. GeoSQL+ intends to support teaching spatial databases to students and professionals from Computer Science and other areas, such as geosciences, urbanism, and engineering.*

Resumo. *Vários recursos para ensino da linguagem SQL estão disponíveis online. Nenhum deles, no entanto, aborda bancos de dados com extensões espaciais, nos quais as consultas podem envolver representações geométricas, funções topológicas e cálculos de distância. Este artigo descreve o GeoSQL+, um aplicativo online que permite formular consultas em SQL a um gerenciador de bancos de dados geográficos e visualizar os resultados em forma de tabelas e mapas. Resultados visuais podem ser acumulados e manipulados interativamente, formando camadas com apresentação configurável e simulando o funcionamento de sistemas de informação geográficos. O GeoSQL+ pretende apoiar o ensino de bancos de dados espaciais para estudantes e profissionais de Computação e de outras áreas, como geociências, urbanismo e engenharia.*

1. Introdução

Existem diversas diferenças entre os Sistemas de Gerenciamento de Bancos de Dados (SGBD) convencionais e os SGBD capazes de lidar com dados geográficos ou espaciais. Elementos como a representação de formas geométricas localizadas no espaço, a indexação espacial e funções topológicas e geométricas acrescentam várias características e capacidades aos SGBD espaciais, que ainda preservam integralmente a capacidade de lidar com dados alfanuméricos convencionais. Estudantes interessados em bancos de dados geográficos, e também profissionais de bancos de dados e sistemas de informação, podem se beneficiar de recursos que amenizem a curva de aprendizado das diferenças entre SGBD convencionais e espaciais, particulamente no aprendizado das extensões espaciais à linguagem SQL. Para suprir tal necessidade, foi proposto e implementado o aplicativo GeoSQL [Freitas et al., 2012].

Este artigo apresenta o GeoSQL+[1], um aplicativo online para aprendizado de SQL com extensões geográficas. O GeoSQL+ é uma reimplementação completa do GeoSQL, na qual foram incorporadas diversas novas funções e facilidades em uma arquitetura interna que foi totalmente redesenhada em bases mais atuais, utilizando bibliotecas JavaScript como OpenLayers 3. O GeoSQL foi implementado usando SVG para apresentação, o que causa limitações importantes para visualização e para a interface homem-máquina. Naturalmente, o GeoSQL+ pode ser usado como ferramenta interativa de consulta. No entanto, sua maior utilidade está na opção ao ensino do SQL com extensões espaciais que permite visualizar o resultado de aplicação de operadores topológicos, métricos e geográficos.

O presente artigo está organizado da seguinte forma: A Seção 2 apresenta trabalhos voltados ao ensino de SQL convencional e geograficamente estendido. A Seção 3 traz uma descrição do funcionamento interno do GeoSQL+ e seus módulos. A Seção 4 encerra o artigo, trazendo conclusões e listando trabalhos futuros.

2. Trabalhos Relacionados

Diversas ferramentas voltadas para o ensino da linguagem SQL foram propostas na literatura, sendo que algumas estão disponíveis online. Um exemplo dessas ferramentas é o Learn-SQL [Abelló et al., 2008], cuja arquitetura é baseada em serviços Web. O Learn-SQL possibilita que o aluno tenha um *feedback* rápido de suas consultas, bem como a avaliação de seu aprendizado, em qualquer computador com acesso à Web. Outra iniciativa é o SQLator [Sadiq et al., 2004], que implementa funções análogas às do Learn-SQL, dispondo ainda de tutoriais e de vários bancos de dados para exercitar a elaboração de consultas. A avaliação do desempenho de estudantes e aspectos específicos na formulação de consultas SQL são explorados por Prior [2003]. No entanto, nenhum dos trabalhos citados apresenta uma abordagem para o ensino de SQL com extensões geográficas.

Em um trabalho de nosso grupo, Freitas et al. [2012] propôs uma abordagem para o ensino de SQL com extensões geográficas a partir do GeoSQL, um ambiente online que oferece uma interface na qual o usuário pode submeter uma consulta SQL a um banco de dados geográfico e obter as respostas tanto em forma de tabelas quanto na forma de mapas. É possível acumular visualmente o resultado de diversas consultas, formando um mapa organizado em camadas, como em um sistema de informação geográfico (SIG).

No entanto, o aplicativo proposto por Freitas apresenta problemas decorrentes da renderização de mapas usando *Scalable Vector Graphics* (SVG), que tem baixo desempenho quando se usa muitas camadas. Outro ponto problemático é a interface do GeoSQL com o usuário, implementada com limitações em relação à necessária flexibilidade na apresentação de resultados. Assim, o GeoSQL+ foi inteiramente reprojetado e reimplementado para substituir e expandir as funções do GeoSQL original.

3. GeoSQL+

O GeoSQL+[2] foi desenvolvido usando PHP e JavaScript. Foram utilizadas as bibliotecas BootStrap para estilização, Jquery UI e Jquery para conexão ao SGBD e Openlayers 3[3]

[1]Aplicativo: `http://aqui.io/geosql`, Screencasts: `http://aqui.io/geosql/video`

[2]Código disponível em: `https://github.com/lab-csx-ufmg/geosql.git`

[3]`https://jqueryui.com/`, `https://jquery.com/`, `http://openlayers.org/`

para renderização dos mapas e implementar outras funções. O GeoSQL+ se apoia no SGBD PostgreSQL, associado à extensão espacial PostGIS[4], ambos sendo intermediados por um servidor Apache[5]. O banco de dados e o servidor PostGIS utilizados pelo GeoSQL+ são definidos em um arquivo de configuração acessível pelo instrutor.

Neste projeto foi adotada uma arquitetura em camadas, seguindo o modelo apresentado por Casanova et al. [2005] para uma arquitetura básica de SIG. Cada camada só depende dos recursos e serviços oferecidos pela camada imediatamente abaixo dela. A arquitetura em camadas apoia o desenvolvimento incremental do GeoSQL+ pois, quando uma camada é desenvolvida, alguns serviços prestados por ela, tais como o serviço de visualização, podem ser imediatamente oferecidos para os usuários. A arquitetura é também portável e mutável. Desde que sua interface fique inalterada, uma camada pode ser substituída por outra equivalente [Sommerville, 2007].

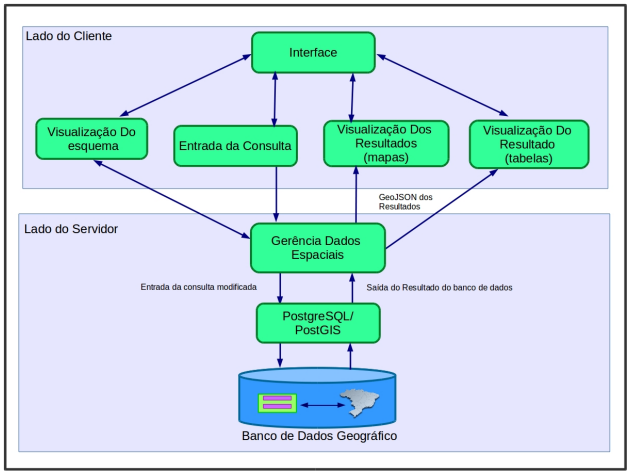


Figura 1. Diagrama de funcionamento básico do GeoSQL.

A Figura 1 apresenta a arquitetura interna do GeoSQL+. De cima para baixo, a primeira camada é responsável pela interação com o usuário. Na camada seguinte, aparecem módulos independentes, voltados para a visualização do esquema do banco de dados, entrada de consultas, visualização em formato de tabela e visualização como mapa. Cada um desses módulos possui sua conexão com o módulo de gerência de dados, que faz a conexão ao SGBD espacial.

Quando o usuário digita uma consulta SQL e solicita sua execução, o módulo de entrada de consultas faz uma requisição AJAX[6] ao módulo de gerência de dados. Em seguida, no lado do servidor, a consulta é pré-processada, acrescentando uma cláusula que permite recuperar a geometria dos objetos do resultado em GeoJSON, um formato mais favorável para apresentação. A consulta assim modificada é enviada para o banco de dados geográficos para execução. Se a consulta retornar algum resultado, um objeto GeoJSON é gerado pelo módulo de gerência de dados, e então enviado de volta para o lado do cliente, onde será consumido pelos módulos de visualização de tabelas e de

[4]PostgreSQL: `http://www.postgresql.org`; PostGIS: `http://postgis.net`
[5]`http://httpd.apache.org`
[6]*Asynchronous Javascript and XML.*

mapas e, portanto, visualizados pelo usuário através da interface. O resultado da consulta poderá ser apresentado apenas pela visualização de tabelas, caso o resultado não inclua geometrias.

A interface com o usuário do GeoSQL+ é composta por três abas: (1) *Consulta*, que apresenta o esquema do banco de dados para referência e um espaço onde o usuário formula a consulta SQL; (2) *Tabela*, que apresenta o resultado de uma consulta em formato tabular; e (3) *Mapas*, que apresenta o resultado da consulta em formato de mapa, juntamente com resultados de consultas anteriores, organizados em camadas. As subseções a seguir descrevem esses componentes da interface e seu funcionamento.

3.1. Formulação de consultas

Na aba *Consulta* (Figura 2), a parte superior apresenta o esquema das tabelas da base de dados. Nela o aluno pode verificar nomes de tabelas e outros dados do catálogo do banco de dados, como nomes de atributos. Abaixo do esquema existe um espaço para a digitação da consulta SQL, e um botão para disparar sua execução.

Figura 2. Aba "Consulta"

3.2. Visualização dos resultados

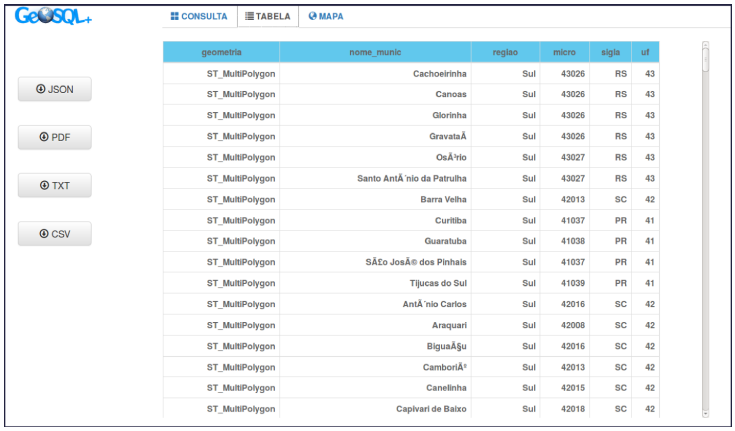


Figura 3. Aba "Tabela" com o resultado tabular de uma consulta

A aba *Tabela* contém a visualização da tabela que resulta de uma consulta. Na lateral esquerda são apresentados botões que oferecem opções para a exportação do resultado em formato JSON, PDF, TXT e CSV (Figura 3). A partir dessas opções é possível, por exemplo, colecionar arquivos de resultado de consultas para formar um relatório de atividades práticas, a encaminhar ao professor, ou ainda redirecionar os resultados para uso por outra ferramenta.

Figura 4. Aba "Mapas" com o resultado de consultas SQL

Na aba *Mapas* são visualizados os resultados das consultas que incluem atributos geográficos (Figura 4). Esse sistema de visualização funciona como uma especie de agregador de visualizações de consultas. Para cada resultado de consulta com atributos geográficos uma camada é gerada e inserida (ou empilhada) na visualização. Ao clicar na visualização e selecionar um objeto de alguma camada, um *pop-up* mostra todas as informações e atributos associados, bem como as coordenadas geográficas da posição indicada pelo mouse no momento do clique.

Outros elementos de apoio à visualização foram incluídos. No canto superior esquerdo tem-se o controle de zoom, no canto inferior esquerdo é apresentado um mapa chave, e na parte inferior da tela é apresentada uma escala gráfica. No canto inferior esquerdo foi colocado um botão que exibe informações sobre a camada-base usada (por exemplo, imagem de satélite ou mapa de fundo). O canto superior direito traz um menu/legenda, que contém controles que permitem interagir com a visualização.

Figura 5. Controles de uma camada no menu/legenda

Assim, cada camada presente na visualização corresponde a uma entrada na legenda com controles. A Figura 5 apresenta os controles disponíveis: da esquerda para direita, (1) selecionar a cor do preenchimento de polígonos na camada, (2º) selecionar a cor das bordas dos objetos da camada, (3) selecionar o atributo que será usado para rotular os objetos da camada, (4) desligar e ligar a visualização de uma camada, (5) exportar a camada no formato GeoJSON e (6) excluir a camada. A ordem das camadas pode ser mudada usando o recurso *Drag and Drop* para manipular diretamente o bloco de controles

da camada, como mostrado na Figura 4. Ao posicionar o mouse sobre uma entrada no menu, uma *tooltip* exibe a consulta SQL usada para gerar aquela camada.

Esse sistema de visualização permite ao aluno interagir com os dados resultantes das consultas, permitindo explorá-los visualmente e em comparação com elementos da camada de fundo, o que permite eventualmente que o objetivo da consulta seja conferido. Em relação ao GeoSQL original, foi obtido um ganho no desempenho aparente na renderização das camadas, viabilizando a apresentação de resultados mais complexos e volumosos.

4. Conclusões e Trabalhos Futuros

Este artigo apresenta o GeoSQL+, um aplicativo online para aprendizado de SQL e extensões espaciais. O aplicativo permite formular consultas com acesso ao esquema físico do banco, e visualizar o resultado em formatos tabular e geográfico. Os resultados das consultas podem ser exportados em diversos formatos, para uso em outras plataformas.

O uso do GeoSQL+ requer apenas um navegador, e não é necessário instalar quaisquer pacotes na máquina do cliente. O GeoSQL+ pode também ser utilizado no ensino da linguagem SQL convencional. Com isso, o aplicativo é uma opção interessante para o ensino de bancos de dados e SQL não apenas para alunos da Computação e áreas afins, mas também para alunos de áreas usuárias das tecnologias de bancos de dados geográficos e SIG, como geografia, cartografia, urbanismo e engenharia.

Em trabalhos futuros, pretendemos incorporar a seleção interativa dos bancos de dados para uso, além de recursos online de apoio à avaliação do desempenho de alunos, incluindo proposição de listas de exercícios, avaliação automática das consultas, e acompanhamento global e individual do desempenho de turmas de alunos. Pretendemos também realizar sessões de validação do aplicativo com usuários reais, dentro do escopo de uma disciplina que aborde esse conteúdo.

Agradecimentos

Os autores agradecem ao CNPq e à FAPEMIG pelo suporte a este projeto.

Referências

Abelló, A., Rodríguez, M. E., Urpí, T., Burgués, X., Casany, M. J., Martín, C., and Quer, C. (2008). LEARN-SQL: Automatic assessment of SQL based on IMS QTI specification. In *Proceedings - The 8th IEEE International Conference on Advanced Learning Technologies, ICALT 2008*, pages 592–593.

Casanova, M. A., Camara, G., Davis Jr, C. A., Vinhas, L., and de Queiroz, G. R. (2005). *Bancos de dados geográficos*. MundoGEO Curitiba.

Freitas, A. L. S., Davis Jr, C. A., and Filgueiras, T. M. (2012). GeoSQL: um ambiente online para aprendizado de SQL com extensoes espaciais. *XIII Simpósio Brasileiro de Geoinformática (GeoInfo 2012), 2012*, pages 146 – 151.

Prior, J. C. (2003). Online Assessment of SQL Query Formulation Skills. In *ACE '03*, pages 247–256.

Sadiq, S., Orlowska, M. E., Sadiq, W., and Lin, J. (2004). SQLator: An Online SQL Learning Workbench. *Proceedings of the Ninth Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, pages 223–227.

Sommerville, I. (2007). *Engenharia de software.* São Paulo: Pearson Addison Wesley.

Small Area Housing Deficit Estimation: A Spatial Microsimulation Approach

Flávia da Fonseca Feitosa[1], Roberta Guerra Rosemback[2], Thiago Correa Jacovine[1]

[1] Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas - Universidade Federal do ABC (CECS/UFABC) - São Bernardo do Campo, SP - Brazil

[2] Centro de Desenvolvimento e Planejamento Regional - Universidade Federal de Minas Gerais (CEDEPLAR/UFMG) - Belo Horizonte, MG - Brazil.

flavia.feitosa@ufabc.edu.br, rosemback@cedeplar.ufmg.br, thiago.jacovine@aluno.ufabc.edu.br

Abstract. *This paper presents our first attempts to develop a new methodology for measuring housing deficit at small areas. It combines the advantages of two types of census data: (a) individual-level sample data, which are very useful for depicting many dimensions of the housing deficit, but do not present detailed geographic information; and (b) universal data with detailed spatial resolution (census tracts), but aggregated. For that, we explore an approach based on spatial microsimulation. We simulate spatial microdata by using aggregate data as constraints to expand and allocate individual-level data to census tracts. This procedure allowed us to estimate a particular dimension of the housing déficit (housing cost) at higher spatial resolution.*

1. Introduction

The lack of decent housing for families with limited means represents a major social problem in Brazil and other developing countries. To support the development of appropriate housing policies, it is essential to characterize and quantify the multiple dimensions of housing deficit. According to the Brazilian National Housing Policy (PNH - *Política Nacional de Habitação*), federal funding towards municipal social housing projects will only be allocated to municipalities that have developed a Local Plan of Social Interest Housing (PHLIS - *Plano Local de Habitação de Interesse Social*), which includes a local diagnosis of the current housing situation.

The PNH assigns to the municipalities a role that was previously attributed to the federal and state levels. This leading role of municipalities in the development of housing diagnosis raises new methodological demands, which include a refinement of scale and a thorough discussion on the potential of available data sources (Rosemback et al. 2014).

Building a methodology for estimating housing deficit requires the definition of which data sources will be used and the identification of their limitations and potentialities. The census is the most comprehensive statistical survey carried out in Brazil. It collects data on the composition and characteristics of population, households, dwellings and their surroundings. It is available to all municipalities and is therefore a unique data source for understanding the Brazilian housing conditions.

The census survey relies on two types of questionnaires: a sample one, which is applied to a fraction of households (about 11% of the population), and a simplified one,

to the remaining households. As a result, the census provides different types of data for public use, two of which are of particular interest for this work: (a) individual-level sample data (microdata), which are very useful for depicting many dimensions of the housing deficit, but are not universal and, for confidentiality reasons, do not present detailed geographic information; and (b) universal data with detailed spatial resolution (small areas known as census tracts), but aggregate.

This paper presents our first attempts to develop a new methodology for measuring housing deficit at small areas. For that, an approach based on spatial microsimulation is explored to combine the advantages of the two types of census data (sample microdata and universal aggregate data). Spatial microsimulation, in this work, is understood as the process of generating spatial microdata by taking data at the individual level and using aggregated level constraints to allocate these individuals to small areas (Lovelace, 2014).

In the next section, we introduce conceptual dimensions of the housing deficit and the two main measurement approaches adopted for Brazilian cities – *place-based* and *household-based*. Along the description of these two approaches, we point out the potential and shortcomings of census data for capturing the multiple dimensions of the housing deficit. Afterwards, a spatial microsimulation method named "iterative proportional fitting" (IPF) is presented as a valuable resource to address the limitations identified in the current housing deficit estimation approaches. To illustrate this point, an experiment is conducted with census data from a small region of the city of São Bernardo do Campo, Brazil. In this experiment, we estimate one particular dimension of the housing deficit (housing cost) at higher spatial resolution by using universal aggregate data as constraints to expand and allocate individual-level sample data to small zones (census tracts).

2. The Housing Deficit: Dimensions and Measurement Approaches

The development of a diagnostic that is suitable for supporting social housing policies demands a multidimensional view of the housing issue. According to Rosemback et al. (2014), it is possible to point out at least seven dimensions of adequacy that must be considered in the assessment of housing needs, as presented in Table 1.

Table 1. Housing Needs: Dimensions of Housing Adequacy (Rosemback et al., 2014).

Dimension	Description
1. Housing Cost	The household spending on housing should not severely compromise the total household income.
2. Physical Suitability of the Dwelling Unit	Dwellings should be made of materials that permanently ensure weather protection, the health, privacy, and security of their residents.
3. Dwelling Unit Suitability to the Household	The household density in a building should not be excessively high. Families should not cohabit for lack of choice.
4. Environmental Safety	Dwellings should not be located in areas of environmental risks, including risks of flooding or landslides, contaminated areas, etc.
5. Legal Security	Households must have legal security of tenure.
6. Infrastructure and Public Services	Dwellings should be served by sewage, water supply, electricity network, street lighting, paving, trees, curb, sidewalk, etc.
7. Location and Accessibility	The location of dwellings should promote the integration into the city, including appropriate access to employment options, efficient public transportation, health services, school, culture and leisure;

Building a methodology that is able to measure these multiple dimensions of the housing deficit remains as an important challenge to be faced. In Brazil, it is possible to identify two different measurement approaches to address this issue. The first is a *place-based* approach that relies on the identification of so-called 'squatter settlements', i.e., inadequate human settlements occupied by low-income residents. The second approach, which is *household-based*, adopts the concept of 'housing needs' and addresses more explicitly some of the dimensions shown in Table 1.

In the *place-based approach*, the housing deficit is estimated from the count of the families living in areas demarcated as squatter settlements. It demands, therefore, data with detailed spatial information. These diagnostics often use local data provided by the municipalities and/or census data. In the latter case, the census data used is the one obtained from the simplified questionnaire and aggregated by census tracts, which is the smallest spatial unit of analysis available for public use. Due to its high spatial resolution, this data can be more easily combined with auxiliary data on dimensions that are not covered by the census survey, such as maps of risk areas ('Environmental Safety') and municipal data on legal status of land tenure ('Legal Security').

However, the variables obtained from the simplified census questionnaire can only depict the dimension 'Infrastructure and Public Services'. Another disadvantage of this data comes from the fact that it is aggregated. In this case, it is possible to know how many dwellings without connection to water supply networks *or* without garbage collection service can be found within the area of a certain census tract. Nevertheless, it is not possible to know how many dwellings are not served by both services. Thus, the identification of squatter settlements and the use of aggregated data prove to be insufficient for the characterization and quantification of the housing deficit.

The *household-based approach* demands individual-level data. The João Pinheiro Foundation (FJP) adopted this approach to develop a methodology for estimating the Brazilian housing deficit that is considered as a reference among social housing experts. This methodology relies on sample data from the 2010 Census to quantify the housing deficit of all municipalities in the country (FJP, 2013). In fact, the data obtained from the long census questionnaire represents a valuable source of information for measuring 5 of the 7 dimensions presented in Table 1. Only the dimensions 'Environmental Safety' and 'Legal Security' cannot be captured by any of the variables available in this dataset.

This data is available as microdata, which allows us to consider the household as an analytical category and also provides a richer set of information about their living conditions. It includes many variables that are not available in the aggregate data and allow multiple combinations among them. The dimension 'Housing Cost', for instance, demands a combination between information on the household spending on housing and the total household income, which can only be obtained from individual-level data.

Nevertheless, census microdata does not provide detailed spatial information. The housing deficit estimates that are calculated from this dataset are only made available for the municipality as a whole or, at best, for weighting areas, which are large geographical partitions that are used in the census sampling weighting procedures. To incorporate the two dimensions that demand auxiliary data into these estimates ('Environmental Safety' and 'Legal Security'), it is essential to improve the spatial

resolution of these results. In other words, it is important to obtain more detailed information on the location of the sampled households.

3. Spatial Microsimulation for Housing Deficit Estimation: An Experiment

Since detailed spatial information about individual-level data cannot be provided due to confidentiality reasons, we advocate that spatial microsimulation represent a valuable resource to address the problem. To demonstrate that, this work uses spatial microsimulation to combine the advantages provided by the two types of census data presented in this paper: sample microdata (more variables and individual-level, but lower spatial resolution) and universal data (higher spatial resolution and easier integration with auxiliary data sources, but less variables and aggregated). This procedure introduces new possibilities for the development of hybrid approaches (individual and place-based) for measuring housing deficit.

In general, spatial microsimulation takes microdata at the individual level (e.g., sample microdata) and uses aggregate level constraints to allocate these individuals to small areas (e.g., data aggregated by census tracts) (Lovelace, 2014). In this work, we explore a spatial microsimulation method named "iterative proportional fitting" (IPF), which is based on deterministic reweighting. Basically, the weights are calculated and adjusted for each individual observation in every census tract until the known marginal distribution of the census tract population is matched by the weighted survey microdata (Hermes and Poulsen, 2012). Information about the marginal distribution is obtained from count data aggregated by census tracts (constraint data). The output is a "spatial microdata" – in other words, a dataset that contains a single row per individual and also an additional variable that indicates the small area (census tract) where the individual is located, as the Figure 1 shows.

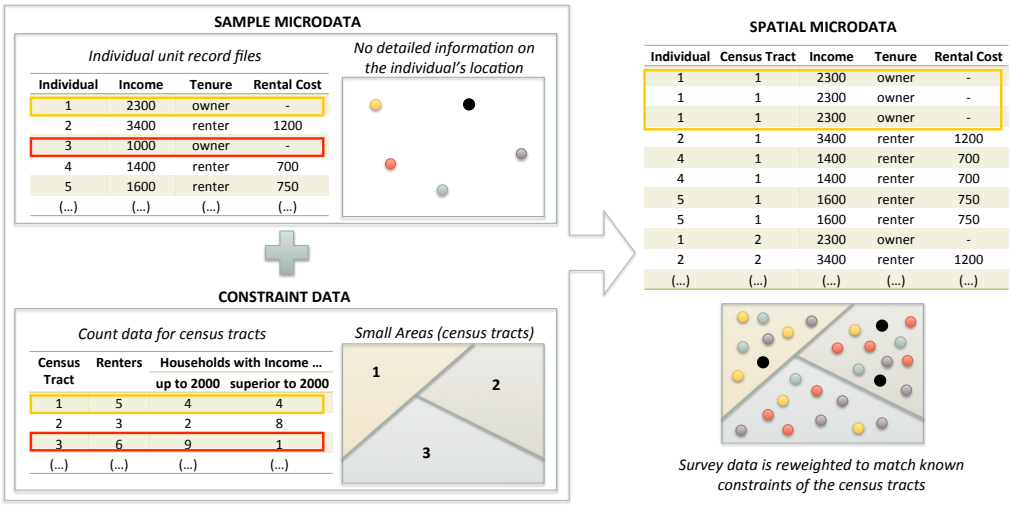


Figure 1. Creating spatial microdata using the IPF method. Adapted from Hermes and Poulsen, 2012.

In the example presented in Figure 1, the results obtained from the IPF method show us that the individual 1 from the sample microdata received a weight equal to '3' for census tract 1. This means that this individual (1) represents three individuals of the census tract 1 (that is why individual 1 appears 3 times at the Spatial Microdata table).

Meanwhile, a weight equal to zero was assigned to the individual 3 because its characteristics are untypical for this small area. Thus, the individual 3 is not allocated to this census tract.

In this work, we present an experiment that uses the IPF method to estimate the spatial variability of one of the dimensions of housing adequacy presented in Table 1, the 'housing cost'. This dimension is often represented by the 'number of households with total income of up to 3 minimum wages that spend more than 30% of their income with rental costs', which can be measured using microdata, but not using data aggregated by census tracts (higher spatial resolution). Nevertheless, we have per census tract data on the number of households with a certain tenure status (renters, owners, etc.) as well as the number of households with per capita income between certain boundaries. In the IPF method, these data can be used as aggregate constraints.

The experiment was conducted for a small portion of the city of São Bernardo do Campo, located in the Metropolitan Region of São Paulo, Brazil. Considering the sample data from the 2010 population census, it is estimated that there were 9811 low-income households with excessive rental cost in the city. Of this total, 1055 households are located in the weighting area that was chosen for the experiment (Figure 2).

The aim of the experiment is to estimate this particular dimension of the housing deficit at higher spatial resolution by using universal aggregate data as constraints to expand and allocate the individual-level sample data into the 61 census tracts contained in the selected weighting area. The procedure was conducted at the statistical and modeling software R (package 'ipfm' - Lovelace, 2014).

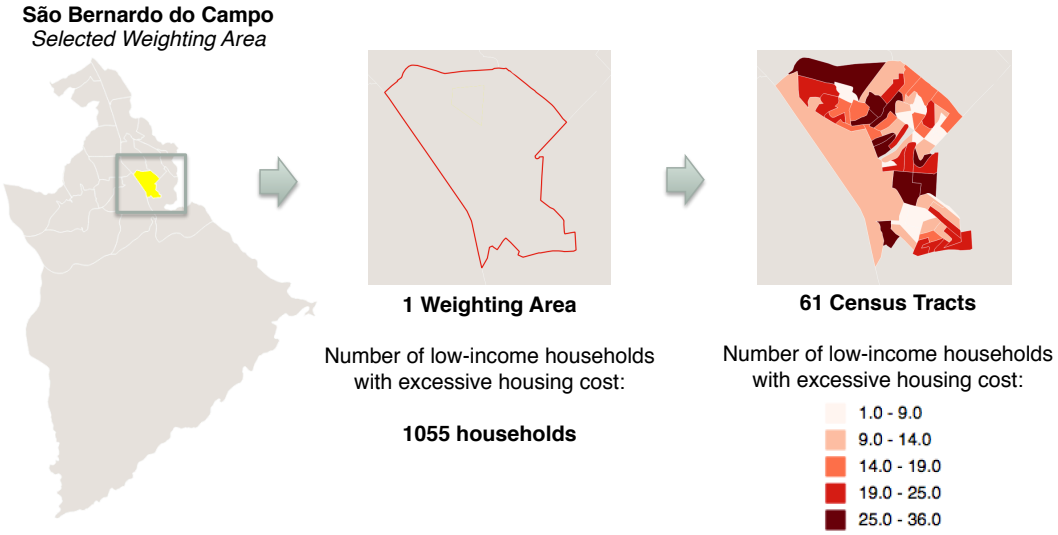


Figure 2. Experiment results: Spatial distribution of low-income households with excessive rental cost, based on simulated data.

To expand and allocate the individual-level sample data, we used the variables 'income per capita' (8 intervals of income) and 'tenure status' (6 classes) as *constraint variables*, since they are available in both datasets (individual-level and aggregate). As *target variables*, we selected variables that are only obtained from the sample microdata: "total household income" and "rental cost". With these two target variables,

it is possible identify the low-income households with excessive rental cost. The results are summarized in Figure 2.

4. Concluding Remarks

In this paper we advocate that spatial microsimulation techniques introduce new possibilities for the development of a methodology for measuring housing deficit at small areas. While the count by census tracts presents a good spatial resolution but lacks details on the households, the census sample microdata presents a richer dataset that is suitable for capturing different dimensions of the housing deficit but lacks information on the spatial location of households.

By addressing the shortcoming of both data types, and therefore both place-based and individual-based approaches for measuring housing deficit, we expect to develop a hybrid approach that is able to better depict the housing deficit in Brazilian cities. Such approach should be able to explore not only the full capability of census data, which is available for all municipalities, but also allow the integration with auxiliary data that may be available at the local level, such as natural hazard/risk assessments and municipal data on legal status of land tenure.

Additional tests must be conducted to ensure that the simulated spatial microdata is as representative as possible of the aggregate constraints. For that, we intend to explore the choice of different constraint variables and validate the resulting estimates. In addition, it is important to test and compare different methods for spatial microsimulation by exploring its main features, variability and validity comparing with external datasets.

Acknowledgements

The authors would like to thank the financial support received from CNPq (Grant 443052/2014-0) and FAPEMIG.

References

Hermes, K., Poulsen, M. (2012) "A review of current methods to generate synthetic spatial microdata using reweighting and future directions". Computers, Environment and Urban Systems, 36, 281-290.

Lovelace, R. (2014) "Introducing Spatial Microsimulation with R: A Practical". National Centre for Research Methods Working Paper 08/14. University of Leeds, Leeds, UK.

Rosemback, R., Rigotti, J., Feitosa, F. and Monteiro, A. (2014) "As dimensões da questão habitacional e o papel dos dados censitários nos diagnósticos municipais: uma sugestão de análise frente às novas exigências da Política Nacional de Habitação", XIX Encontro Nacional de Estudos Populacionais, ABEP, São Pedro, SP.

Fundação João Pinheiro – FJP. (2013). “Déficit habitacional municipal no Brasil 2010”, Fundação João Pinheiro, Centro de Estatística e Informações, Belo Horizonte, MG.

Processamento da Junção Espacial Distribuída utilizando a técnica de Semi-Junção Espacial

Sávio S. Teles de Oliveira[2], Anderson R. Cunha[2], Vagner J. do Sacramento Rodrigues[2], Wellington S. Martins[1]

[1]Instituto de Informática – Universidade Federal de Goiás (UFG)
Alameda Palmeiras, Quadra D, Câmpus Samambaia
131 - CEP 74001-970 – Goiânia – GO – Brazil

[2]GoGeo
Rua Leopoldo Bulhões, esquina com a Rua 1014
Quadra 31, Lote 07, Sala 9 Setor Pedro Ludovico
CEP 74820-270 – Goiânia – GO – Brazil

{savio.teles,anderson.cunha,vagner}@gogeo.io, wellington@inf.ufg.br

Abstract. *This paper presents a new Distributed Spatial Join algorithm with Semijoin technique in a scalable peer-to-peer platform. This technique reduces the network traffic and minimize the processing cost. Tests have demonstrated that the response time has been drastically reduced with the Distributed Spatial Join proposed in this paper.*

Resumo. *Este trabalho apresenta um novo algoritmo de Junção Espacial Distribuída utilizando a técnica de Semi-Junção em uma plataforma peer-to-peer escalável. Esta técnica reduz o tráfego de dados na rede e minimiza o custo de processamento. Testes demonstraram que o tempo de resposta foi drasticamente reduzido com a utilização do algoritmo de Junção Espacial Distribuída proposta neste trabalho.*

1. Introdução

A junção espacial é uma das operações mais importantes nos Sistemas de Gerenciamento de Bancos de Dados Espaciais [Zhou et al. 1997] e envolve o relacionamento entre duas bases de dados. Por exemplo: encontrar as rodovias que intersectam com rios (pontes).

Para processar de forma eficiente a operação de junção espacial, as pesquisas têm se concentrado em resolver o problema de forma distribuída utilizando *clusters* de computadores. Com isso, algumas questões no processamento de consultas são identificadas, tais como o tráfego de dados na rede, que impacta de forma significativa no desempenho da Junção Espacial Distribuída.

Este trabalho têm como objetivo apresentar um algoritmo de Junção Espacial Distribuída utilizando a técnica de Semi-Junção Espacial implementado em uma plataforma *peer-to-peer* escalável. Os testes demonstraram que a utilização do algoritmo reduziu drásticamente o tráfego de dados na rede e o tempo de resposta.

O trabalho está organizado da seguinte forma. A Seção 2 descreve as propostas encontradas na literatura para o processamento da junção espacial distribuída. A Seção

3 apresenta o algoritmo de junção espacial distribuído implementado neste trabalho. A Seção 4 descreve a técnica de Semi-Junção proposta neste trabalho. A Seção 5 apresenta os experimentos realizados. A Seção 6 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Correlatos

Esta Seção apresenta um grupo de trabalhos que processam a junção espacial distribuída utilizando a técnica de Semi-Junção Espacial. Alguns trabalhos encontrados na literatura, como [Mutenda and Kitsuregawa 1999, Patel and DeWitt 2000, Chung et al. 2005, Wei et al. 2008, Zhou et al. 2011, Zhong et al. 2012], processam a junção espacial em um *cluster*, mas não discutem ideias de redução de tráfego de dados na rede utilizando a técnica de Semi-Junção Espacial. Em [Tan et al. 2000] são indexadas as duas bases de dados R e S envolvidas na junção espacial e armazenadas nos servidores R_{site} e S_{site} respectivamente. O trabalho apresentado em [Ramirez and de Souza 2001] utiliza aproximações mais acuradas para processar a junção espacial. Em [Kang and Choy 2002], são propostos alguns modelos de custo para o processamento da junção espacial distribuída. Em [Karam and Petry 2005] são propostos vários modelos de custo detalhados para o processamento da junção espacial distribuída. Estes trabalhos não apresentam estratégias para processar a Junção Espacial Distribuída em mais de dois servidores como em um *cluster* de computadores.

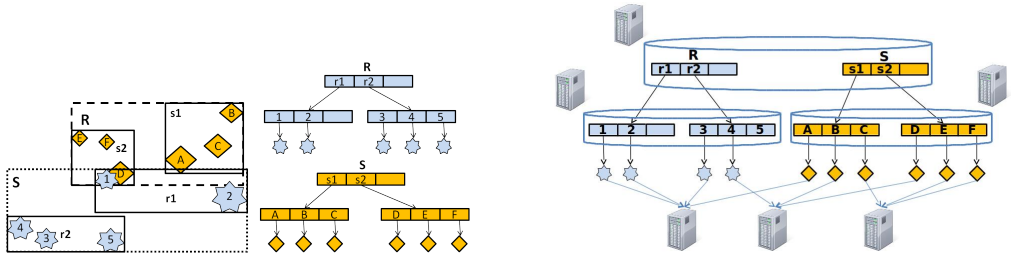
Uma plataforma de processamento da junção espacial distribuída é proposta em [Oliveira et al. 2013], com a necessidade de um serviço de nomes para descobrir qual máquina armazena cada nó da *R-Tree* distribuída. Desta forma, a junção espacial é muito penalizada pela necessidade de consulta ao serviço de nomes a cada passo do algoritmo. Além disso, o algoritmo de Semi-Junção transfere os objetos de apenas uma relação, ao invés de analisar cada tupla da Junção para decidir qual objeto transferir. Em [Farruque and Osborn 2014] é proposto um algoritmo de Semi-Junção com vários servidores particionando os índices espaciais ou através de uma representação com Bloom Filter. O algoritmo trafega na rede a base de dados da junção com menor cardinalidade. Entretanto, a base de dados de menor cardinalidade pode conter polígonos com geometrias com grande quantidade de pontos geográficos, o que aumentaria o tráfego de dados.

3. Processamento da junção espacial distribuída

A junção espacial pode ser definida a partir de duas relações R = $r_1, ..., r_n$ e S = $s_1, ..., s_m$, onde r_i e s_j são objetos espaciais, $1 \leq i \leq n$ e $1 \leq j \leq m$. A operação verifica todos os pares (r_i, s_j) que satisfazem o predicado de um operador topológico, por exemplo a interseção, isto é, $r_i \bigcap s_j \neq \emptyset$.

O processamento da junção é realizado em duas etapas: etapa de filtragem e etapa de refinamento [Patel and DeWitt 1996]. A etapa de filtragem inicia na raiz das duas relações R e S e é realizada nos nós internos da *R*-Tree*. Esta etapa gera um conjunto de possíveis respostas a consulta. A fase de refinamento é realizada nas folhas e remove deste conjunto os resultados incorretos utilizando as geometrias reais de cada objeto.

No exemplo da Figura 1(a), a etapa de filtragem analisa as raízes de R e S e a etapa de refinamento analisa os nós filhos de (r1, s1) e (r1, s2), pois estes apresentaram intersecção entre os seus respectivos MBRs na fase de filtragem. Apenas (1, D) fez parte do resultado por apresentar intersecção de suas respectivas geometrias.



(a) Visão Espacial das relações R e S (b) R*-Tree Distribuída

Figura 1. Junção Espacial

Na versão distribuída do algoritmo de junção espacial, há a necessidade de trocar mensagens na rede para acessar os objetos distribuídos. O algoritmo de junção espacial distribuída deste trabalho foi implementado utilizando a plataforma DistJoin. Esta plataforma possui uma arquitetura *peer-to-peer* escalável descentralizada com tolerância a falhas e alta disponibilidade. O protocolo *Gossip* é utilizado para disseminar informações do *cluster* e de localização dos nós da *R-Tree* para que não seja necessário um Serviço de Nomes para descobrir a localização de um nó distribuído durante a execução do algoritmo de Junção Espacial Distribuída.

A Figura 1(b) ilustra as *R*-Trees* de R e S distribuídas em um *cluster* de computadores. Quando dois objetos espaciais estão localizados em máquinas diferentes, por exemplo os objetos 1 e D, um deles deve ser trafegado na rede até o local em que está armazenado o outro objeto. O tráfego de dados na rede é reduzido neste trabalho utilizando uma técnica conhecida como Semi-Junção espacial, apresentada na Seção 4.

4. Algoritmo de Semi-Junção Espacial

Este trabalho apresenta o algoritmo de Semi-Junção Espacial executado entre cada par de objetos (r, s) das bases R e S na etapa de refinamento da Junção Espacial Distribuída. Para reduzir o tráfego de dados na rede, o objeto espacial com maior número de pontos será visto como b e o outro como a, já que a aproximação do polígono de b e o polígono de a serão transferidos pela rede. O número de pontos do polígono de cada objeto espacial e é armazenado como metadado no nó pai de e. Na Semi-Junção Espacial, o par de tuplas a e b estão localizados nos servidores A_{server} e B_{server} respectivamente.

O algoritmo segue três passos. O passo 1 do algoritmo envia a aproximação do polígono de b, denominada b', para A_{server}. Neste trabalho, o MBR é utilizado como aproximação da geometria. No passo 2, é realizada a junção entre b' e o polígono a em A_{server}. A aproximação têm menos pontos que a geometria real, o que faz com que a computação geométrica da junção entre b' e a tenha processamento minimizado, já que o custo dos algoritmos espaciais são proporcionais ao número de pontos das geometrias.

As aproximações filtram apenas os resultados que não fazem parte da resposta. Por isso, é retornado o polígono de a para B_{server}, caso b' e a apresentem intersecção e vazio, caso contrário. O algoritmo de intersecção é executado, no passo 3, entre os polígonos de a e b para verificar se apresentam intersecção. Caso apresentem, os dois objetos são retornados como um dos pares de resultados da consulta.

No exemplo da Figura 2 com o par de objetos (1, D), o polígono 1 contém mais

pontos que D e, por isso, é enviada sua aproximação, no passo 1, e retornado o polígono D como resposta no passo 2. Assim, o objeto 1 é definido como b e D como a.

O algoritmo de Semi-Junção Espacial consegue reduzir o tráfego de dados na rede, através da transmissão do objeto espacial que apresenta geometria com menos pontos. Além disso, é minimizado o processamento local, pois no passo 2 do algoritmo é realizada a intersecção entre a aproximação da geometria com maior número de pontos (b') e a geometria com menor número de pontos (a). Se estas geometrias não apresentarem intersecção, não será preciso realizar a intersecção entre a e b.

5. Experimentos

Foram utilizadas as seguintes bases de dados[1]: i) 10994 polígonos do Bioma da Caatinga com tamanho total de 275 MB, ii) 21840 pontos de Localidades do Brasil com tamanho total de 1,4 MB, iii) 5771 linhas de Hidrografia do Brasil com tamanho total de 1,4 MB e iv) 5565 polígonos de municípios do Brasil com tamanho total de 38 MB. Foram executados as seguintes junções espaciais: i) Bioma da Caatinga e Localidades que retorna 3934 itens e ii) Hidrovia e Municípios que retorna 8721 itens.

A execução dos testes de junção espacial distribuída têm como objetivo avaliar os seguintes aspectos: i) a escalabilidade do algoritmo de junção espacial distribuída; ii) o tráfego de dados na rede. Os testes foram executados com e sem a técnica de Semi-Junção Espacial em um *cluster* com 3, 6 e 12 servidores, com cada máquina contendo um processador Optiplex 780 Intel Core 2 Quad 2.83GHz com 4 Gb de memória RAM conectadas por uma rede Ethernet de 1 Gbit/segundo.

Como pode ser visto na Figura 2(a), a junção espacial entre as bases de dados de Bioma da Caatinga e Localidades, apresentou um tempo de execução drasticamente menor utilizando a técnica de Semi-Junção Espacial proposta neste trabalho, sendo 5 vezes melhor na média que os testes executados sem a técnica.

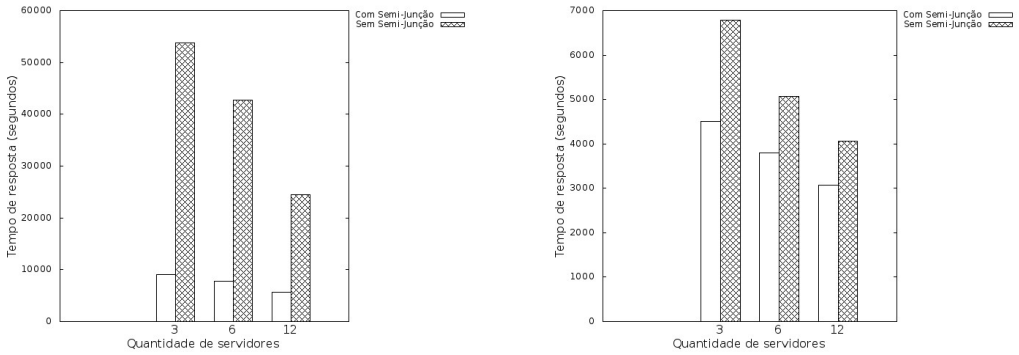
Isto ocorreu devido a redução do tráfego de dados na rede que foi aproximadamente 9 vezes menor com 3 máquinas, 11 vezes menor com 6 servidores e 19 vezes menor com 12 servidores utilizando a técnica de Semi-Junção Espacial.

A junção espacial entre as bases de dados de Municípios e Hidrovia apresentou tempo de resposta aproximadamente 1,5 vezes melhor com a técnica de Semi-Junção Espacial deste trabalho, como pode ser visto na Figura 2(b). O tráfego de dados na rede nesta junção foi reduzido em aproximadamente 1,3 com a técnica de Semi-Junção Espacial para 3, 6 e 9 servidores.

O tempo de resposta nesta junção não foi reduzido na mesma proporção que a junção apresentada na Figura 2(a), pois o tráfego de dados na rede sem a técnica de Semi-Junção é menor na junção da Figura 2(b) devido ao menor espaço em disco de Municípios em relação a Bioma da Caatinga.

O algoritmo de Junção Espacial Distribuída proposto neste trabalho foi aproximadamente quatro vezes melhor que a proposta de [Oliveira et al. 2013], já que a plataforma de [Oliveira et al. 2013] necessita consultar o Serviço de Nomes a cada acesso a um nó da

[1]Bases de dados disponibilizadas pelo Laboratório de Processamento de Imagens e Geoprocessamento - www.lapig.iesa.ufg.br



(a) Bioma da Caatinga e Localidades.

(b) Municípios e Hidrovia.

Figura 2. Tempo de resposta da Junção Espacial Distribuída.

R-Tree Distribuída. Além disso, a Semi-Junção proposta em [Oliveira et al. 2013] sempre envia os objetos da relação com menor número de pontos pela rede. Nosso trabalho analisa cada par de objetos individualmente.

O algoritmo de Junção Espacial Distribuída apresentou escalabilidade em todos os testes executados. A utilização da técnica de Semi-Junção Espacial proposta neste trabalho reduziu o tráfego de dados na rede e o tempo de resposta da Junção Espacial Distribuída.

6. Conclusões e Trabalhos Futuros

A junção espacial apresenta alto custo e pode ser processada de forma eficiente em um *cluster* de computadores para distribuir o processamento da operação. Para processar a Junção Espacial Distribuída de forma eficiente, o tráfego de dados na rede deve ser reduzido. Para tal, existe uma técnica denominada Semi-Junção Espacial. Nenhum trabalho encontrado na literatura, entretanto, processa a Junção Espacial Distribuída utilizando uma técnica de Semi-Junção Espacial eficiente em uma plataforma *peer-to-peer* escalável.

Por isso, neste trabalho foi implementado um algoritmo de processamento da Junção Espacial Distribuída utilizando uma técnica de Semi-Junção Espacial para ser processado em um *cluster* de computadores utilizando uma plataforma *peer-to-peer*. O algoritmo se apresentou escalável nos experimentos e a utilização da técnica de Semi-Junção Espacial reduziu de forma significativa o tráfego de dados na rede e o tempo de resposta.

Em trabalhos futuros, além do número de pontos das geometrias, novos metadados serão adicionados (i.e. CPU, memória, rede), para decidir quais geometrias serão transferidas e onde elas serão processadas. Também serão analisadas novas aproximações na técnica de Semi-Junção Espacial. Novos experimentos serão realizados com outras bases de dados para validar o algoritmo de Junção Espacial Distribuída em diversos cenários.

Referências

Chung, W., Park, S., and Bae, H. (2005). Efficient parallel spatial join processing method in a shared-nothing database cluster system. *Embedded Software and Systems*, pages 81–87.

Farruque, N. and Osborn, W. (2014). Efficient distributed spatial semijoins and their application in multiple-site queries. In *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on*, pages 1089–1096. IEEE.

Kang, M. and Choy, Y. (2002). Deploying parallel spatial join algorithm for network environment. In *High Speed Networks and Multimedia Communications 5th IEEE International Conference on*, pages 177–181. IEEE.

Karam, O. and Petry, F. (2005). Optimizing distributed spatial joins using r-trees. In *Proceedings of the 43rd annual Southeast regional conference-Volume 1*, pages 222–226. ACM.

Mutenda, L. and Kitsuregawa, M. (1999). Parallel r-tree spatial join for a shared-nothing architecture. In *Database Applications in Non-Traditional Environments, 1999.(DANTE'99) Proceedings. 1999 International Symposium on*, pages 423–430. IEEE.

Oliveira, S., Sacramento, V., Cunha, A., Aleixo, E., de Oliveira, T., Cardoso, M., and Junior, R. (2013). Processamento Distribuído de Operações de Junção Espacial com Bases de Dados Dinâmicas para Análise de Informações Geográficas. *XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*.

Patel, J. and DeWitt, D. (1996). Partition based spatial-merge join. In *ACM SIGMOD Record*, volume 25, pages 259–270. ACM.

Patel, J. and DeWitt, D. (2000). Clone join and shadow join: two parallel spatial join algorithms. In *Proceedings of the 8th ACM international symposium on Advances in geographic information systems*, pages 54–61. ACM.

Ramirez, M. and de Souza, J. (2001). Distributed processing of spatial join. In *Proc. of the Anais do III Workshop Brasileiro de GeoInformática GeoInfo*, volume 2001, pages 1–8.

Tan, K., Ooi, B., and Abel, D. (2000). Exploiting spatial indexes for semijoin-based join processing in distributed spatial databases. *Knowledge and Data Engineering, IEEE Transactions on*, 12(6):920–937.

Wei, H., Wei, Z., and Yin, Q. (2008). A new parallel spatial query algorithm for distributed spatial databases. In *Machine Learning and Cybernetics, 2008 International Conference on*, volume 3, pages 1570–1574. IEEE.

Zhong, Y., Han, J., Zhang, T., Li, Z., Fang, J., and Chen, G. (2012). Towards parallel spatial query processing for big spatial data. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*, pages 2085–2094. IEEE.

Zhou, X., Abel, D., and Truffet, D. (1997). Data partitioning for parallel spatial join processing. In *Advances in Spatial Databases*, pages 178–196. Springer.

Zhou, Y., Zhu, Q., and Zhang, Y. (2011). Spatial data dynamic balancing distribution method based on the minimum spatial proximity for parallel spatial database. *Journal of Software*, 6(7):1337–1344.

Mining influential terms for toponym recognition and resolution

Caio Libânio Melo Jerônimo, Cláudio E. C. Campelo, Cláudio de Souza Baptista

Systems and Computing Department
Federal University of Campina Grande (UFCG) – Campina Grande, PB – Brazil

caiolibanio@copin.ufcg.edu.br, {campelo,baptista}@dsc.ufcg.edu.br

Abstract. *The detection of toponyms present in text has appeared as a useful resource for many different applications, such as for social network analysers and for geographic search engines. The variety of ambiguities present in the geoparsing process represents one of the main challenges related to the process of detecting toponyms, bringing the need for treating this problem with careful attention. One important technique to detect toponyms is based on the presence of influential terms, which are terms that could indicate the existence of geographical references in the text. This paper presents an approach to automatically identifying relevant influential terms for a given language, as well as a set of attributes relating these terms with toponyms. The technique presented here was validated with an existing geoparser, using a training set based on online news. The results indicate the technique is effective in identifying influential terms, and has shown that the geoparser's capabilities of detecting toponyms have improved by using the generated list of influential terms.*

1. Introduction

Since the early days of the internet, many changes have occurred, especially in the way people perform searches on the network. In this scenario, new technologies related to Information Retrieval (IR) have emerged. It has been found that users often describe some kind of geographic context within their queries when performing a search in an information retrieval system [Gan et al. 2008]. A study with the logs of the Excite search engine showed that one fifth of all queries were geographical [Guillén 2007]. Thus, it is possible to realize the importance that the geographical context has in the current internet usage, which made possible the rise of new technologies related to geoparsing methods [Jones and Purves 2015] [Dhavase and Bagade 2014].

The geoparsing process consists of analysing documents, in order to find geographic references in it. The core difficulty associated with the geoparsing process consists of the different kinds of ambiguity associated with natural languages, including ambiguities related to toponyms [Leidner and Lieberman 2011]. This kind of ambiguity refers to the cases where it is not possible to determine if a specific name is related to a geographic term or to another kind of reference, such a person's name. This kind of ambiguity can lead to undesired results, especially in geographic search engines, due to the fact that these systems rely on geographic references found in documents (by the geoparser) to satisfy the user's geographical query.

For the geographic recognition in text, there are three main families of methods: Gazetteer Lookup Based, Rule-Based and Machine Learning Based [Leidner and

Lieberman 2011]. The Gazetteer Lookup Based consists of analysing texts elements (words or characters) and search for this references in a predefined set of real geographic place names, in order to verify if the searched term exists in a predefined set. The Rule Based method uses a set of rules in a DSL (domain specific language) encoding decision procedures, allowing an interpreter to decide whether a word is a geographic term. The Machine Learning Based method basically consists in analysing texts in order to find specific patterns that could indicate a presence of geographic terms in the text, based in previously learned information.

One important type of pattern, that is useful to detect different types of named entities (NE) in text (including geographic ones) is related to the detection of influential terms (ITs). These terms generally appear near the named entity. For example, the term "city of" suggests that the next term probably refers to a city name, making this term a relevant case of influential term. Ratinov and Roth (2009) included this kind of feature in a set of "context aggregation features" to develop a Named Entity Recognition (NER) method that automatically detects token's context based on existent terms in a distance window, in order to determine their context in documents. Combined with other approaches, this technique shows itself useful in determining the context of a correlated NE. In a geoparser, this method helps determine the geographic scope of a term (city names, state names, street references), and also helps in the disambiguation process, allowing to decide if a reference is a real toponym or just a person's name, by applying a contextual meaning to the NE.

The objective of this research is to propose an innovative method of automatically detecting ITs for a given language. To achieve this goal, we developed a set of heuristics with the objective of learning the relations between geographical terms (toponyms) and other terms that could indicate the presence of geographical content in a document. The heuristics presented here have been implemented within an existing geoparser (Campelo and Baptista, 2009), which is part of a search engine prototype, making possible to analyse the effectiveness of the presented method, based on the number of ITs detected and based on observed improvements of the geoparser effectiveness, in terms of corrected toponyms detected, false positives and false negatives.

To execute both the training and parsing process, we used Brazilian news from a major online newspaper (Globo G1 - www.g1.globo.com). News often have a strong geographic component, since this type of documents frequently have associations with the place where the readers live [Lieberman and Samet 2011], making this kind of documents a good scenario for both training and geoparsing processes. In order to analyse the results, we compared the detection rate between two cycles of training, collecting informations about the toponyms detected correctly and incorrectly. It was discovered a total of 1,211 ITs and, using these new terms, we observed the geoparser was able detect more toponyms in the analysed documents, with a p-value = 0.00001924.

The remainder of this paper is organised as follows. The next section presents related work. Section 3 presents our proposed approach to identifying influential terms. Then the experiments conducted to validate our approach is presented in Section 4. In Section 5 we have the discussion of the results. Finally, Section 6 concludes the paper and points to future directions.

2. Related work

Toponym recognition and resolution have been studied in different contexts, such as in information retrieval, social media geoparsing, geographic information systems and in a large variety of different applications where detecting geographic references could play a relevant role.

Many research papers related to this topic uses the idea of influential terms in toponym detection. Gelernter and Balaji (2013) presented a method of geo-parsing microtext with the objective to make these texts more readily usable for tracking news events, political unrest, or disaster response by providing a geographic overview. Their presented technique included the use of special cues (in, near, to, west, south) to identify possible location abbreviations in microtext.

Keller et al. (2008) describes an automated approach to discover geographic references taking the context into account, which relies on a window of words surrounding the word to parse, providing a generalization of the gazetteer's rule-based geoparsing. This approach has the objective of geo-parsing texts from media reports to track global disease outbreaks. Other methods also consider the context given by neighbor terms in geographic references analysis [Rauch et al. 2003] [Amitay et al. 2004].

Campelo and Baptista (2009) used a similar technique, where the occurrence of an influential term could increase the confidence that there is a geographic reference in a given document. Dominguès and Eshkol-Taravella (2015) proposed a solution to detect toponyms in custom-made maps also using predefined patterns based on verbs, locative nouns and locative prepositions (e.g., to leave, departure, arrival, beside, alongside, close to).

Although there have been proposed many approaches to detecting toponyms that rely on the presence of influential terms in the geoparsing process, a common drawback of these existing works is that they do not describe precisely how the list of influential terms are built. Moreover, in most cases, the ITs are set manually by a user, rather than automatically detected by a learning process.

3. Identifying Influential terms

There are two important factors that should be considered while designing a mechanism for automatically identifying influential terms for toponym recognition: the *type of places* and the *types of georeferences* the geoparser can deal with. The former refers to the place types such as cities and states. Existing geoparsers normally define a hierarchy of place types that it can deal with (such as city → state → country). The latter refers to types of georeferences usually found in text that can be used to infer locality, such as postcodes, phone numbers or place names. While place names can be used in a text to refer to any place type, other georeference types may be mapped to specific place types (such as phone area codes, which are usually associated with countries or states). Influential terms, in turn, are usually employed in association with one or more place types or georeference types. Thus, we say that an IT can be mapped to specific tuples <georeference type, place type>. For example, "city of" may be an influential term for cities when it is referred by its place name, but not by a postcode.

As the solution developed in this research was implemented in an existing geoparser (Campelo and Baptista, 2009), the influential terms identified were based on the types of georeferences and places it is able to recognize. The types of places the system is able to detect is based on a 5-level administrative hierarchy (i.e., city, microregion, mesoregion, state and region). On the other hand, the system can process place names, phone numbers and postcode as georeference types. In order to preserve the capabilities of the validating geoparser, the method presented here will only allow the identification of ITs related to the types of georeferences and the types of places that are currently supported by the system. However, it should be highlighted that our proposed solution is general enough to keep identifying additional ITs as the geoparser improves its capabilities.

The aim of our method is to generate a dataset consisting of a table containing all the influential terms identified and a set of related attributes, as follows:

- *Term*: an identified IT. There may be multiple rows in the table for a given IT, each of which associated with different attributes.
- *Distance*: distance in text (in number of words) from the IT to the correlated toponym.
- *Place Type*: the place type of the toponym that the IT is associated with (city = 1, microregion = 2, mesoregion = 3, state = 4 and region = 5);.
- *Georeference Type*: the type of the georeference that the IT is associated with (place name = 1, telephone reference = 2 and postcode reference = 3).
- *Relevance*: the calculated relevance for the IT. This is a number between 0 and 1 that quantifies the influence of a term to the tuple <georeference type, place type> when the distance between them in the text is equals to the value given in the field Distance.

Another important challenge in developing a learning based method of identifying influential terms is that it leads to a problem like the classic "the chicken or the egg" dilemma: the identification of influential terms must rely on known toponyms present in the text. However, for training a corpus containing thousands of news, annotating those toponyms manually would not be feasible. On the other hand, for automatically detecting toponyms, it is crucial to apply heuristics based on the presence of influential terms. We have overcome this challenge by executing the previous version of our geoparser on the training set for detecting toponyms. Afterwards, we could apply our approach to identifying ITs. This process is illustrated in Figure 1 (the training variables will be described later in this section).

The previous version of our geoparser relies on an IT table containing just 32 rows manually inserted based on human observations. Nonetheless, one could argue that such geoparser would detected a significant number of false positives, which would consequently affect the training process for identifying ITs. However, in this geoparser, each detected toponym is associated with a confidence value (from 0 to 1), and only those above a certain threshold is accepted. In a previous work (Campelo and Baptista, 2009), we found that the parser performs reasonably well (less than 30% false positives) for a threshold of 0.5. Thus, in this task, we increased this threshold to 0.6, making the results much more reliable (with approximately less than 10% false positives). The counter effect of this is that less toponyms are detected (more false negatives), which would not be acceptable for a geographic search engine, for example. However, in this

case, the decrease in the number of toponyms detected per document does not affect the efficacy of the training process for identifying ITs, as the training set of toponyms is still large enough for this task.

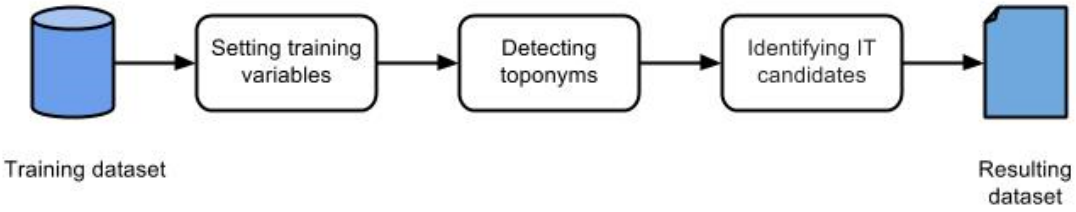


Figure 1. The flow process of the training algorithm

3.1 The IT identification algorithm

The process of identifying ITs (last box of Figure 1) is illustrated in more detail in Figure 2. As described above, our algorithm identifies toponyms using the previous version of the geoparser. After this process, the mechanism stores information about their preceding terms (Figure 1 - box 3), called IT candidates, along with information about the correlated toponym, such as: the distance between the preceding term and the toponym; the toponym classification (georeference type and place type). A rule implemented in this stage asserts that each IT candidate will be associated with the nearest toponym only. In other words, the rule ensures that there will be no other toponym between an IT and its related toponym. By implementing this rule, we observed a significant decrease in the number of false positives for IT identification, that is, the cases where identified ITs were not syntactically related to the correct toponym. In this process, if a repeated IT candidate is found, a counter associated with this IT is incremented, which represents the number of times that the term is found in correlation to a toponym, for the same classification and distance.

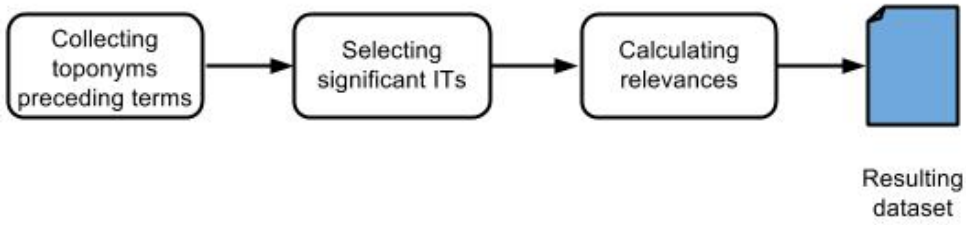


Figure 2. The process of Identifying ITs

The collection of preceding terms depends on a training variable called MAX_DISTANCE. This variable represents the maximum distance that the algorithm should consider when looking for IT candidates, always in backwards direction from the toponym. For example, in the text fragment “The next olympics will be held in the city of Rio de Janeiro”, if the toponym was “Rio de Janeiro”, and the value for this variable was 3, the terms selected for composing IT candidates will be “the”, “city” and “of”.

Our approach allows the identification of ITs composed by more than one term, such as "south of" and "close to". For that, another training variable should be considered: MAX_COMBINATION. This variable expresses the maximum number of words that should be combined to form IT candidates (i.e., the maximum length of an IT). For the example of the text fragment given above, if the value of MAX_COMBINATION was 2, the IT candidates generated would be "the", "city", "of", "the city" and "city of". We found by empirical analysis that MAX_DISTANCE = -6 and MAX_COMBINATION=3 provide best results. Hence, these are the values adopted to perform our experiments. In this paper, we refer to ITs composed by just one term as "atomic", whilst those made by two or more terms are called "composed".

After acquiring IT candidates, the algorithm performs a selection process over these terms (Figure 2 - box 2) in order to reject terms that are not in conformance to the predefined rules. This rules consists in rejecting terms with less than two characters in length, terms formed by numbers, terms matching a specified set of stop words. These rules have shown to be effective in removing candidate terms that could decrease the quality of IT identification. Moreover, in this stage, the selected ITs must be in conformance with another training variable: ACCEPTANCE_THRESHOLD: this variable represents the minimum number of occurrences of an IT candidate in the training dataset. That is, if the number of times an IT appears in the training set is less than the value for this variable, it will be rejected and its relevance value will not be calculated.

After the selection process, the remaining IT have their relevance calculated based on the frequency that each one appeared in the training data (Figure 2 - box 3). The IT relevance is an important parameter when performing the geoparsing process using the discovered influential terms. This value can inform the geoparser how significant the IT is when it appears preceding a specific toponym, in a predefined distance. In order to calculate the IT's relevance, the solution takes into account the frequency of each candidate, based on the number of occurrences collected during the training process.

For atomic candidates, the algorithm also considers the number of occurrences that the IT appears in correlation to non-geographic references, in order to decrease its relevance value. This approach aims to reduce the false positive cases for toponym detection due to a significant reduction of the ITs relevance that are frequently not only vinculated to geographic references. This step is crucial for calculating ITs relevantes, as there are many terms that are frequently used to refer to a toponym, but are also frequently used in other contexts, such as the term "in". The approach to calculating the relevance value for atomic ITs is shown in Equation 1.

$$R_a = \frac{PCount}{PCount + NCount} \times PCount$$

Equation 1. Calculus of atomic IT's relevance

where:

- R_a represents the calculated relevance for an atomic IT.
- $PCount$ represents the occurrence counter for each IT found, in the cases where this term was associated with a toponym candidate.

- $NCount$ represents the occurrence counter for each IT found, in the cases where this term was not associated with a toponym candidate.

This is important to highlight that the system does not count occurrences of ITs (atomic or composed) that are also part of other longer composed ITs, such as the terms "in" and "north" which are both part of the term "in the north of". This changes significantly the way ITs and their attributes are identified, and we observed that the geoparser efficacy improves considerably for detecting toponyms. For example, terms like "north" and "south" are quite frequent, and could be stored in association with distance -2, such as in "north of London". However, as the term "north" rarely appears without the preposition "of", it could lead the system to detect false toponyms where any term is between it and the IT, such as in "...they went to the north. In London, people...". Thus, by considering this rule, the atomic term "north" is discarded and the term "north of" is stored in association with distance "-1". The approach to calculating the relevance value of composed IT is shown in Equation 2.

$$R_c = \frac{CCount}{CSize}$$

Equation 2. Calculus of composed IT's relevance

where:

- R_c represents the calculated relevance for a composed IT;
- $CCount$ represents the occurrence counter for each composed IT found in correlation with a toponym;
- $CSize$ is the number of composed ITs found during the training stage;

In both cases (R_a and R_c) the values are normalised by the algorithm before completing the process and storing the relevance value in the IT table.

In Algorithm 1, we present the general idea of the proposed algorithm for mining influential terms in the form of pseudo-code.

```
1. miningInfluentialTerms(trainingNewsList)
2.    n = trainingNewsList.size()  // size of the training set
3.    listIT = []  //list containing the influential terms founded
4.    for (i in n)
5.       listPT = [] //list containing the toponym' precedent terms
6.       //collecting the toponyms from an individual news article
7.       listToponyms = trainingNewsList.getNews(i).getToponyms();
8.       for (tp in listToponyms) //collecting precedent terms from toponym
9.          listPT.addAll(tp.getPrecedentTerms());
10.      for (pt in listPT) //adding each precedent term to the list of ITs
11.         if(! listIT.contains(pt)) listIT.add(pt)
12.         else listIT.get(pt).incrementCouter();
13.   for (itCandid in listIT) //removing precedent terms according to rules
14.      if(! accordingtoRules(itCandid))itCandid.markToRemove();
15.   listIT.removeMarkedTerms();
16.   for (it in listIT)  //calculating relevances
17.      if (it.isAtomic()) it.setRelevance(calculateAtomicRel(it))
18.      else it.setRelevance(calculateComposedRel(it))
19.   return listIT;
```

Algorithm 1. General idea of the proposed technique

4. Experimental Evaluation

This section presents the experiments conducted to validate our proposed approach.

4.1 Experimental Units

The performed experiments were based on web news written in Portuguese, from a Brazilian communication vehicle (http://g1.globo.com/). We decided to use this type of document because news usually have strong relations to geographic areas, since these documents often refer to the place where the readers live (e.g., city, state, country), making these documents an excellent dataset of geographic references, both to the training and the parsing processes. The news were collected in the year of 2015 by an automated tool, developed to read an RSS feed and extracting their related news. These news were collected from the "last news" category, due to this kind of subject could bring us a large number of toponyms references, as this documents are frequently associated with a specific place or region.

4.2 The prototype used

To validate the proposed solution, the methods were implemented in an existing geoparser, that is part of a search engine prototype, called GeoSEn. This is a geographic search engine, with the objective of retrieving web documents based on their geographic scope, usually specified as a parameter in the user's queries. As Campelo and Baptista (2009) describe, GeoSEn's geoparser implements a set of heuristics to detect geographic references in web documents, and the presence of influential terms is one of these heuristics. Each heuristic has a specific weight for the final value of confidence rate. This confidence value is assigned to the toponyms found in the text (by querying a local Gazetteer). In that geoparser, the ITs and their associated attributes must be informed manually by the system administrator.

Structural adaptations were performed in the system's geoparser, with the objective of implementing the entire training phase showed in Figure 1. As part of our implementation strategy, we used the original geoparser with all of its capabilities. Then, by implementing the method of discovering ITs proposed in this research, we obtained a new geoparser with the ability of detecting a large range of toponyms. The original geoparser was used to detect toponyms (Figure 1 - box 2), making possible to execute the first learning cycle based on this initial set of detected toponyms, generating the first set of discovered influential terms (Figure 1). Then, this set of detected influential terms can be used to detect toponyms more reliably, which can be the basis for further executions of the training cycle (i.e., discovering ITs based on more reliable toponyms), allowing an incremental process of learning influential terms.

4.3. Design of Experiments

The experiments performed in this research have the objective of answering the research question Q, as follows:

- *Research question (Q):* Have the discovered influential terms improved the toponyms detection rates?
- *Null hypothesis (H-0):* The discovered influential terms does not improve the toponyms detection rates.

To answer this question, our proposed experiment consists in training the parser initially with 1,000 aleatory news (set D1) and, after the training process, executing the trained geoparser with a test set of 5,000 news documents. After this first experiment step, the process was repeated with 9,000 news as a training set, totaling 10,000 news documents (set D2).

For sets D1 and D2, the system returned a copy of the analysed documents containing the detected and rejected toponyms coloured in green and red respectively, as HTML files. An example of this coloured output file is shown in Figure 3:


http://g1.globo.com/sp/campinas-regiao/noticia/2015/04/dois-jovens-morrem-apos-carro-bater-em-poste-entre-campinas-e-valinhos.html Dois jovens morrem após carro bater em poste entre Campinas e Valinhos Duas pessoas morreram e uma ficou ferida após o carro em que estavam bater contra um poste na madrugada deste sábado 25 na Avenida Francisco de Paula Souza no limite entre Campinas SP e Valinhos SP O acidente aconteceu no sentido de Valinhos próximo ao Hipermercado Carrefour Segundo a Polícia Militar o acidente aconteceu por volta das 4h O motorista Artur Sosai Cardoso e Danilo Lalier com idade entre 20 e 25 anos morreram no local Um outro jovem que estava no carro foi encaminhado ao Hospital Mário Gatti em estado grave De acordo com o Corpo de Bombeiros não havia sinal de bebida alcóolica no carro Com o impacto da batida o poste de iluminação pública caiu na avenida e uma equipe da CPFL foi acionada para fazer o conserto no início da manhã

Figure 3. Example of a coloured output file

We selected an aleatory subset of 100 of such coloured documents that showed differences between the number of accepted and rejected toponyms, totalizing 50 documents for each D1 and D2 (paired analysis), and submitted them for human examination by a group of volunteers, in order to judge the quality of the toponyms detection, in terms of correct detections (true positives), correct rejections (true negatives), false negatives and false positives. These categories of detection were considered separately for a better statistical analysis. The main steps of this experiment are summarised as follows:

1. Training the geoparser with 1,000/10,000 news;
2. Executing the geoparser process with a test set of 5,000 aleatory news for D1 and D2 cases;
3. Analysing (manually) the output consisting of coloured HTML files, to collect statistics about correct toponym detection, correct toponym rejection, false negatives and false positives;
4. Comparing the results between the two training cases (1,000 and 10,000).

5. Results and Discussion

For our proposed research question Q, the experiments were executed in a paired design, and the collected data did not show a normal distribution. Due to these characteristics of data, we chose to use the Wilcoxon test in order to analyse a possible improvement in toponyms detections for the training dataset of 1,000 (D1) and 10,000 (D2), respectively. After this first experiment, we reached the following results (Table 1):

Table 1. Results of Wilcoxon Test

Wilcoxon (α=0.05)	H != 0 (p-value)	H > 0 (p-value)	H < 0 (p-value)

Correct detections	0.00003848	1	0.00001924
Correct rejections	-	-	-
False negatives	0.00002601	0.000013	1
False positives	-	-	-

This result shows a significant difference in toponyms detection between the two dataset sizes. The difference can be shown firstly by the case where H != 0. Here, we rejected the hypothesis that D1 and D2 would have the same rate of correct detections (p-value = 0.00003848). Still with regard to correct detections, we have the H < 0 case, where p-value = 0.00001924, indicating that D2 detected a higher number of correct toponyms in relation to D1 dataset size.

Regarding false negatives, it can be noticed a difference between D1 and D2 in case H != 0 (p-value = 0.00002601), allowing us to reject the hypothesis that D1 and D2 would have the same correct rejection rate. There was still a significant change related to H > 0 case (p-value = 0.000013), denoting that the system presented a higher false negative rate when trained with D1 (in comparison to the D2 training dataset).

With reference to correct rejections and false positives, the results obtained for both D1 and D2 were almost the same for most cases, what could result in a low precision reported from the Wilcoxon test, due to the fact that this particular test depends on pairwise differences between the samples. This characteristic led us to do not execute the test for this two analysed values.

An additional experiment case was conducted, aiming to execute two incremental training cycles with the same training dataset. This experiment consisted in training the geoparser initially with 20,000 different news and collecting the number of discovered ITs. After that, we executed the training system again with the same training dataset, and collected the final number of detected ITs. Our objective was to determine whether the system could detect more ITs even with the same training documents. The results are shown in Table 2:

Table 2. Results for the second experiment

Parameter	geoparser (20000)	geoparser (40000)
Nº of discovered IT	1806	3014

This results shows a considerable increase in the number of detected IT as the training dataset increases, denoting that the system learned ITs that could not be learned in the first training case (with 20000 news). After the proposed experiments, we can answer the question Q by rejecting the null hypothesis H-0, with p-value = 0.00001924, as this value indicates that the learned ITs increased the number of correct toponyms detected. The experiment also indicates a lower rate of false negatives associated with the new IT detected (p-value = 0.000013).

6. Conclusion and Future Work

This paper presented an approach to automatic discovery of influential terms from text. The solution was implemented in a geographic search engine prototype called GeoSEn, with the objective of validating the proposed methodology in a geographically oriented system. The system's geoparser has been adapted to learn the influential terms from a training dataset and to report statistics of the execution of the parsing process, making possible to perform further analysis of the obtained results. Our results indicate that the proposed algorithm performed considerably well for automatically detecting ITs, as well as indicate that the geoparsing efficacy improves significantly when these new influential terms are used for detecting toponyms.

The methodology presented was validated with Brazilian news, written in portuguese. However, there are evidences that it can be extended for many other languages without further modifications of the parser's code, which is intended to be verified in future work. We believe that other relevant ITs could be identified if the parser is executed for other types of texts, such as more informal texts from social networks. This is also planned to be checked in future work.

References

Amitay E., Har'El N., Silvan R. and Soffer A. (2004) "Web-a-where: Geotagging web content". In Proceedings of SIGIR, Workshop on Geographical Information Retrieval, pages 273–280.

Campelo C. E. C. and Baptista C. S. (2009) "A Model for Geographic Knowledge Extraction on Web Documents." In: Advances in Conceptual Modeling - Challenging Perspectives, LNCS 5833, Edited by Carlos Alberto Heuser and Günther Pernul, Springer Berlin Heidelberg, p. 317-326.

Dominguès C., Eshkol-Taravella I. (2015) "Toponym recognition in custom-made map titles." In: International Journal of Cartography, Taylor & Francis, pp.DOI : 10.1080/23729333.2015.1055935.

Dhavase N. and Bagade A. M.. (2014) "Location Identification for Crime & Disaster Events by Geoparsing Twitter.", In: Convergence of Technology (I2CT), 2014 International Conference, pages 1 - 3.

Gan Q., Attenberg J., Markowetz A., and Suel T. (2008). "Analysis of geographic queries in a search engine log",In: *Proceedings of the first international workshop on Location and the web* ACM. (pp. 49-56).

Gelernter J., Balaji S. (2013) "An algorithm for local geoparsing of microtext." In: Geoinformatica, Volume 17, Issue 4, pp 635-667.

Guillén R. (2007) "GeoParsing Web Queries". In: Advances in Multilingual and Multimodal Information Retrieval Lecture Notes in Computer Science Volume 5152, 2008, pp 781-785.

Jones, Christopher B., and Ross S. Purves. (2015) "GIR 2014 workshop report: the 8th ACM SIGSPATIAL International Workshop on Geographic Information Retrieval.", In: *SIGSPATIAL Special* 6.3 (2015): 52-52

Keller M., Freifeld C.C., Brownstein, J.S. (2008) "Expanding a Gazetteer-based

Approach for Geo-Parsing Text from Media Reports on Global Disease Outbreaks". In: Advances in Disease Surveillance, Vol. 5, No. 3.

Leidner J. L. and Lieberman M. D. (2011) "Detecting geographical references in the form of place names and associated spatial natural language". In: SIGSPATIAL Special, 3(2):5–11.

Lieberman M. D. and Samet H. (2011) "Multifaceted toponym recognition for streaming news". In: SIGIR'11: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 843–852, Beijing, China.

Rauch E., Bukatin M. and Baker K. (2003) "A confidence-based framework for disambiguating geographic terms". In: HLT-NAACL Workshop on Analysis of Geographic References, pages 50–54.

Spectral Attributes Selection based on Data Mining for Remote Sensing Image Classification

Raian V. Maretto[1,2], Thales S. Körting[2], Emiliano F. Castejon[2], Leila M. G. Fonseca[2], Rafael Santos[3]

[1]Fundação de Ciências, Aplicações e Tecnologias Espaciais (FUNCATE)
São José dos Campos – SP – Brazil.

[2]Image Processing Division – National Institute for Space Research (INPE)
São José dos Campos – SP – Brazil.

[3]Applied Computing and Mathematics Associated Laboratory – National Institute for Space Research (INPE)
São José dos Campos – SP – Brazil.

{raian, thales, castejon, leila}@dpi.inpe.br, rafael.santos@inpe.br

Abstract. *Remote sensing images are a rich source of information for studying large-scale geographic areas. The new satellite generations have producing huge amounts of data. Data mining techniques have been emerged last years as powerful tools to help in the analysis of these data. In the area of remote sensing image analysis, software like GeoDMA, eCognition, InterIMAGE, and others are available for end users. These software provides tools to extract several attributes of the images. These attributes are then used in image classification and analysis. When dealing with high resolution multispectral satellites, we have a large quantity of attributes. In many cases, the attributes are highly correlated, and consequently may not help to separate the classes of interest. Thus, this work shows the results of an approach to analyze the correlation of the attributes between several classes of interest, selecting those that will better distinguish them. In this way, it is possible to reduce the amount of data to be used during classification and analysis, consequently reducing the computational time for classification.*

1. Introduction

The increased accessibility of the new generation high-spatial resolution multispectral sensors has improved the level of complexity required in the analysis techniques. In particular, many traditional per-pixel analysis may not be suitable to high-spatial resolution imagery, due to its high-frequency components and the horizontal layover caused by off-nadir look angles [Im et al. 2008]. Aiming to overcome this problem, in the last decades, several approaches and platforms have been developed with algorithms that consider contextual information and pixel region properties [Körting et al. 2013; Syed et al. 2005; Walter 2004].

Current software can extract several statistical, spatial, color, texture or topological attributes. However, most of them often do not help to distinguish between the classes of interest, due to its high correlation. Thus, the attributes selection phase often relies on *ad hoc* decisions about what of them can better describe the classes. The huge number of attributes available makes a detailed exploratory time-consuming and dependent on expertise [Körting et al. 2013]. Many works have proved that data mining techniques can be useful to this purpose [Dash and Liu 1997; Kohavi and Kohavi 1997; Laliberte et al. 2012].

In this context, the main objective of this work is to analyze the correlation of the spectral attributes between a set of classes of interest, in order to verify what of them best distinguish these classes. A case study is presented over a small region of the city of São José dos Campos, using a WorldView-2 image. It is important to emphasize that although this study is in a preliminary stage, the results are promising and reached improvements in the accuracy of the classification, even as a good reduction in the computational time.

2. Spectral attributes selection

Most of attributes selection approaches focuses on a global selection, analyzing the correlation for the whole set of attributes and classes, even as its capacity to distinguish between all the classes. In this work, we propose an approach based on the analysis of the best attributes to distinguish pairs of classes. For this, we applied the C4.5 decision tree algorithm [Quinlan 1993], which constructs the classification model based on the divide and conquer strategy. It applies thresholds to the object attributes, and then, observations that are smaller than these thresholds are assigned to the left branch, otherwise to the right branch [Hastie et al. 2008; Körting et al. 2013; Ruggieri 2002].

One important feature of decision tree algorithms is that they indicate the best attributes to distinguish between the classes of interest, according to the entropy measure. However, it analyses all the classes together, choosing the best attributes to distinguish between all of them. To compare the pairs of classes, we isolate only the corresponding samples to the pair being compared and then constructed a decision tree for them. Figure 1 shows three examples of the decision trees. These trees were used in the experiment presented in Section 3, and show for example, that the attribute Band Ratio of the band 2 is the best to distinguish the classes Ceramic Roof and Bare Soil.

With the attributes indicated by this analysis, we construct a matrix as shown in Table 1, which indicates what attributes are the best to separate each pair of classes, and then, these attributes will be used to the classification process. With the selected set of attributes, is expected an increase in the accuracy of the classification, even as a decrease of the computational cost.

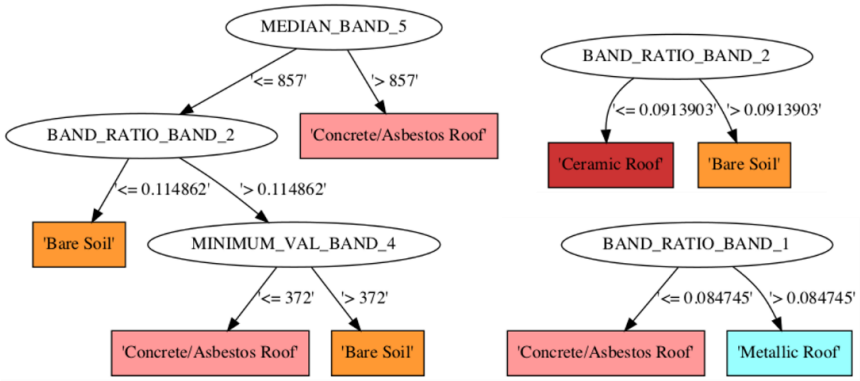


Figure 1. Examples of decision trees comparing pairs of classes.

3. Experimental Results

To evaluate the effectiveness of the approach proposed, we tested a WorldView-2 image of a small area in São José dos Campos, Brazil. The image has 8 multispectral bands with 0.5 meter of spatial resolution. It is important to keep in mind that, in this phase, the objective is not to provide the optimal classification result. The aim of this experiment is to verify the improvement in the distinction between a set of classes when using only the previously selected attributes for the classification, in comparison with the results obtained in the classification using all the spectral attributes computed for the image.

The image was segmented using the Region Growing algorithm [Bins et al. 1996], and then 19 spectral attributes were computed for each band, being 14 statistical measures (like mean, variance, standard deviation, etc) and 5 texture measures based on [Haralick et al. 1973]. Thus, we have 152 spectral attributes for each segment region. The image used and the segmentation result are presented in Figure 2.

Figure 2. Image used in the test (a) and the result of its segmentation (b).

The class typology includes roofs (ceramic, metallic, concrete and asbestos), vegetation, shadow, asphalt pavement and bare soil. Firstly, around 130 samples were collected, distributed between all the classes. Using these samples, we made two classification experiments, using 66% of them to train the decision tree algorithm, and 34% to validate the classification model.

In the first experiment, we used all the 152 attributes to build the decision tree using the C4.5 algorithm. In the second experiment, we applied the proposed approach to select the best attributes and then, we built the decision tree using only the subset of the selected attributes, comparing the validation results with the previous. Figure 1 shows some examples of the decision trees used to select the attributes, and the matrix with all the selected attributes is shown in Table 1.

Table 1. Matrix with the selected attributes for the classification.

Classes	Concrete/Asbestos Roof	Ceramic Roof	Asphalt Pavement	Vegetation	Metallic Roof	Bare Soil	Shadow
Concrete/Asbestos Roof							
Ceramic Roof	BR_B2						
Asphalt Pavement	C_B0	BR_B2					
Vegetation	MD_B4	MD_B4	MD_B1				
Metallic Roof	BR_B1	BR_B2	BR_B2	MD_B1			
Bare Soil	BR_B2 MD_B5 MIN_B4	BR_B2	BR_B1 AM_B4	ME_B4	BR_B1		
Shadow	ME_B5	ME_B5	SM_B0	BR_B4	MD_B2	MD_B5	

Where:

- AM_B4 → Amplitude on Band 4
- BR_B1 → Band Ratio of Band 1
- BR_B2 → Band Ratio of Band 2
- BR_B4 → Band Ratio of Band 4
- C_B0 → Number of valid values on Band 0
- MD_B1 → Median on Band 1
- MD_B2 → Median on Band 2
- MD_B4 → Median on Band 4
- MD_B5 → Median on Band 5
- ME_B4 → Mean on Band 4
- ME_B5 → Mean on Band 5
- MIN_B4 → Minimum Value on Band 4
- SM_B0 → Sum on Band 0

In both experiments, the results were evaluated with the 34% remaining samples (the 66% used to build the tree were not used in the validation). The decision tree built for the first experiment is shown in Figure 3. In this experiment, the classification obtained an accuracy of 63.64% in the validation, with an error of 36.36%, and the kappa value obtained was 0.57.

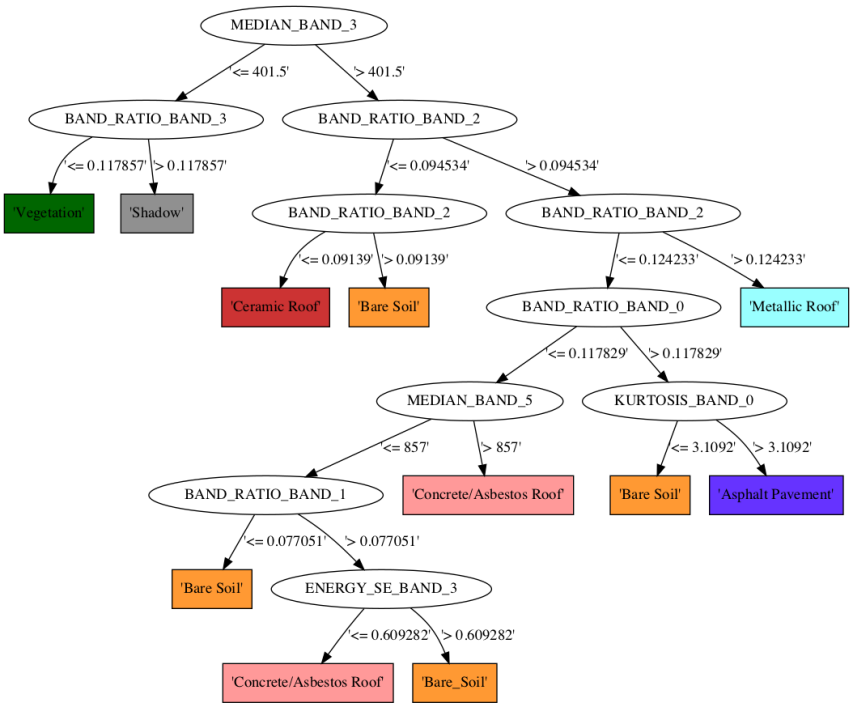


Figure 3 – Decision tree built using the whole set of attributes.

The decision tree built in the second experiment, considering only the subset of the selected attributes, is shown in Figure 4. We can see, in this second tree, several differences in the attributes used, and in the importance attached for some of them. In the decision trees algorithm, the attribute that provides the greater distinction between the classes is assigned to the root, and nodes in lower levels, receive smaller importance. In this way, the lower levels provides a finer adjustment for the classification. In this experiment, the classification obtained an accuracy of 70.45%, with an error of 29.54%, and the kappa value obtained was 0.65.

4. Concluding Remarks and Future advances

This work has shown that the attributes selection approach proposed can help for the improvement of the accuracy in the classification through Data Mining techniques. In our experiments, the results increased around 7% the accuracy when compared to the original classification. We believe that, with more adjustments in the methodology and in the models for classification, we can obtain more relevant improvements. Moreover, with the reduction on the amount of attributes used in the classification, we can also reduce the computational cost of this process.

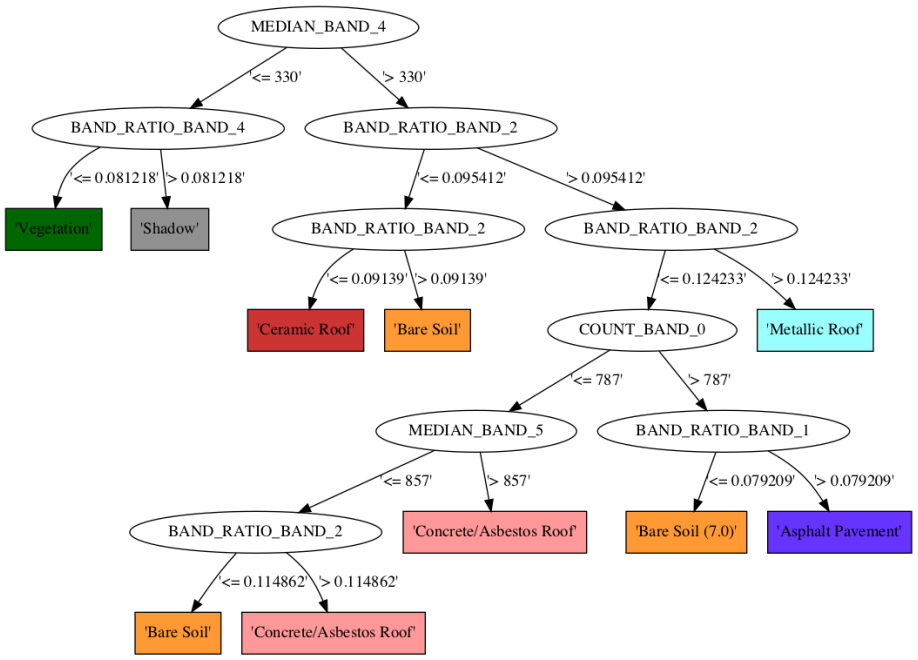


Figure 4. Decision tree built using only the subset of selected attributes.

As future steps, we must validate the results for an entire image, and then automatize the process of the comparison between the classes of interest. We aim to implement the results in GeoDMA platform and then study the improvements in the computational performance of the process, comparing with other classification processes. This work also gives way to thought about approaches using evolutionary computing or other optimization methods, aiming to improve the selection process to try to find the optimal set of attributes, in order to help the analysts to both, improve the classification results, and understand more about the data being classified.

5. References

Bins, L. S., Fonseca, L. M. G., Erthal, G. J. and Ii, F. M. (1996). Satellite Imagery Segmentation : a region growing approach. Anais VIII Simposia Brasilerio de Sensoriamenta Remoto, Salvador, Brasil, 14-19 abril 1996, INPE, p. 677–680.

Dash, M. and Liu, H. (1997). Feature Selection for Classification. Intelligent Data Analysis, v. 1, n. 97, p. 131–156.

Haralick, R., Shanmugan, K. and Dinstein, I. (1973). Textural features for image classification. IEEE Transactions on Systems, Man and Cybernetics. http://dceanalysis.bigr.nl/Haralick73-Textural features for image classification.pdf.

Hastie, T., Tibshirani, R., Friedman, J. and Franklin, J. (2008). The elements of statistical learning: data mining, inference and prediction. The Mathematical, v. 27, n. 2, p. 83–85.

Im, J., Jensen, J. R. and Tullis, J. a. (2008). Object-based change detection using correlation image analysis and image segmentation. International Journal of Remote Sensing, v. 29, n. 2, p. 399–423.

Kohavi, R. and Kohavi, R. (1997). Wrappers for feature subset selection. Artificial Intelligence, v. 97, n. 1-2, p. 273–324.

Körting, T. S., Garcia Fonseca, L. M. and Câmara, G. (2013). GeoDMA-Geographic Data Mining Analyst. Computers and Geosciences, v. 57, p. 133–145.

Laliberte, a. S., Browning, D. M. and Rango, a. (2012). A comparison of three feature selection methods for object-based classification of sub-decimeter resolution UltraCam-L imagery. International Journal of Applied Earth Observation and Geoinformation, v. 15, n. 1, p. 70–78.

Quinlan, J. R. (1993). C4.5: Programs for machine learning. San Mateo, California: Morgan Kaufman Publishers.

Ruggieri, S. (2002). Efficient C4 . 5. IEEE Transactions on Knowledge and Data Engineering, v. 14, n. 2, p. 438–444.

Syed, S., Dare, P. and Jones, S. (2005). Automatic classification of land cover features with high resolution imagery and lidar data: an object-oriented approach. … : the national biennial Conference of the …, p. 512–522.

Walter, V. (2004). Object-based classification of remote sensing data for change detection. ISPRS Journal of Photogrammetry and Remote Sensing, v. 58, n. 3-4, p. 225–238.

Using Rational Numbers and Parallel Computing to Efficiently Avoid Round-off Errors on Map Simplification

Maurício G. Gruppi [1], Salles V. G. de Magalhães [1,2], Marcus V. A. Andrade [1], W. Randolph Franklin[2], Wenli Li[2]

[1]Departamento de Informática – Universidade Federal de Viçosa (UFV)
Viçosa – MG – Brazil

[2]Rensselaer Polytechnic Institute
Troy – NY – USA

{mauricio.gruppi,salles,marcus}@ufv.br, mail@wrfranklin.org, liw9@rpi.edu

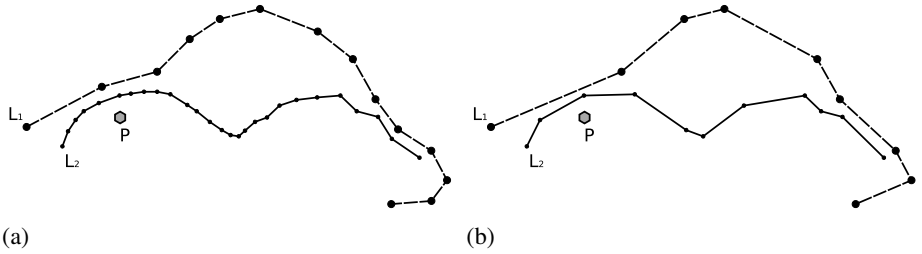
Abstract. *This paper presents* EPLSimp*, an algorithm for map generalization that avoids the creation of topological inconsistencies.* EPLSimp *is based on Visvalingam-Whyatt's (VW) algorithm on which least "important" points are removed first. Unlike VW's algorithm, when a point is deleted a verification is performed in order to check if this deletion would create topological inconsistencies. This was done by using arbitrary precision rational numbers to completely avoid errors caused by floating-point arithmetic.* EPLSimp *was carefully implemented to be efficient, although using rational numbers adds an overhead to the computation. This efficiency was achieved by using a uniform grid for indexing the geometric data and parallel computing to speedup the process.*

1. Introduction

The map simplification process, also known as map generalization, allows the production of maps with different levels of details [Jiang et al. 2013]. It consists of removing information that is not relevant to the viewer, while preserving essential features on the map. Generalization is inherent to every geographical data since every map consists of generalized representations of reality, and the more generalized a map is, the more distant it becomes from the real world [João 1998]. The output of this process is a map with more desirable properties than those from the input map. An example of generalization is scaling a map of a single town which contains detailed information about streets and buildings. When scaling this map to show nearby towns it may be necessary to simplify it so that it is not overburden by unimportant data.

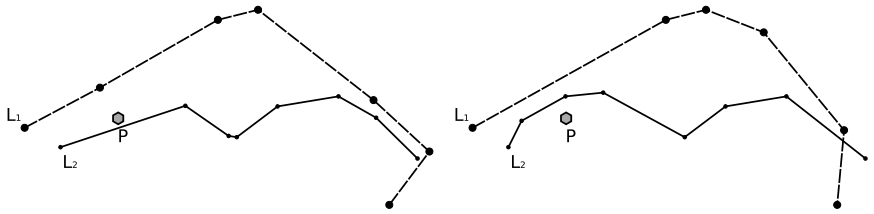
A challenge in generalization is to find a balance between simplification and reality. Map simplification can produce inappropriate results as it may affect topological relationships. These results are said to be *topologically inconsistent* and they may present relationships that are conflicting with reality. For example, the simplification can create self-intersecting lines, improper intersections between lines and polygons, etc.

Another kind of topological inconsistency is the sidedness change, that is, after performing simplification, a feature can be on a different side regarding other feature on the map. For example, after the simplification of a line, a point which was originally on the right side of this line now can be on the left side. Thus when designing simplification algorithms it is important to guarantee topologically consistent results.



(a) (b)

Figure 1. (a) Example of polylines L_1 and L_2 and a control point P. (b) Simplification of L_1 and L_2. Notice that topology consistency is preserved: no intersections were created and sidedness is maintained.



(a) (b)

Figure 2. Inconsistent simplification output: (a) P is on the wrong side of line L_2; (b) nonexisting ntersection between lines L_1 and L_2 is created.

2. Polyline Simplification

An approach for performing map simplification is to reduce the complexity of its lines. That means making simpler representation of curves or polygon edges. Usually, lines are represented by polygonal chains or polylines. A polyline is a serie of segments defined by a sequence of n vertices (v_1, v_2, ..., v_n), where each segment consists of two endpoints and adjacent segments share a common endpoint. Figure 1(a) shows an example of two polygonal chains L_1 and L_2, and also a control point P (gray hexagon) that does not belong to a polyline but is considered relevant or meaningful.

The basic idea of line simplification consists of removing points and representing the original curve using approximation with fewer vertices. Figure 1(b) presents an example of the simplification of the lines shown in Figure 1(a). Two famous and frequently used line simplification algorithms are the Ramer-Douglas-Peucker's algorithm (RDP) [Douglas and Peucker 1973, Ramer 1972] and Visvalingam-Whyatt (VW) [Visvalingam and Whyatt 1993] algorithm.

The line simplification process can bring inconsistencies to the output if some care is not taken. Figure 2 shows two examples where removing certain points from the polylines in Figure 1(a) would cause topological inconsistency: (a) after simplification, point P is on the other side of the simplified line L_2; (b) a "nonexistent" intersection between lines L_1 and L_2 is created.

Topological inconsistency may be created by some simplification algorithms such as the ones based on the RDP method. But there is another source of error that affects even algorithms that attempt to avoid inconsistencies: round-off errors resulting from floating point arithmetic. These errors occur because real numbers cannot be exactly

represented in computational systems, instead, an approximation of the real number is used [Goldberg 1991]. In order to overcome such problems, the best strategy is to make use of Exact Geometric Computation [Li et al. 2005].

In this paper is presented a method that uses rational numbers and parallel computing to solve the following variation of the generalization problem: given a set of polylines and control points, the goal is to simplify these polylines by removing some of their vertices (except endpoints) such that topological relationships between pairs of polylines and between polylines and control points are maintained. In practice, polylines may represent boundaries of counties or states, and control points may represent cities within these states. The introduction of rational numbers was used to prevent errors introduced by rounding in floating point arithmetic. The use of arbitrary precision numbers is expected to increase the overall execution time of the algorithm since its operations are more complex. In order to compensate this performance drop, parallel computing is used.

3. Related Works

In this section we describe algorithms for line simplification as well as problems that arise from floating-point arithmetic.

3.1. Algorithms for Line Simplification

Many algorithms for line simplification have been developed so far. One of the most famous is the Ramer-Douglas-Peucker's algorithm (RDP) [Douglas and Peucker 1973, Ramer 1972]. Its basic idea is to start with a very rough approximation of the original line (i.e. a straight line connecting the end vertices) and iteractively refine the approximation including, in each step, the vertex that is farthest from the current line. The method stops when the distance between the farthest vertex and the line is greater than a given threshold (the smaller the threshold the less simplified the line is).

The RDP algorithm does not take topological consistency into consideration and may generate inconsistent results. An approach proposed by Saalfeld [Saalfeld 1999] attempts to avoid such inconsistencies. It uses Douglas-Peucker's algorithm to simplify lines and then starts a refining process by adding points to the output line so that the curve no longer presents any inconsistency. Noteworthy to mention that adding points to a curve may eliminate previous inconsistencies but may create new ones.

Another approach based on Douglas-Peucker was proposed by Li et al. [Li et al. 2013]. It intends to avoid topological inconsistencies as well as cracks on polygon shapes using a strategy based on detection-point identification, which are points lying within a minimum boundary rectangle (MBR) of the bounded face formed by a subpolyline and its corresponding simplifying segment. These detection-points are used for consistency verification of the simplification process.

Visvalingam and Whyatt [Visvalingam and Whyatt 1993] proposed a method (called the *VW* algorithm) for line generalization that uses the concept of *effective area* of a point to define the priority of its removal. The *effective area* of a point v_i, for $1 < i < n$, in a polygonal chain $v_1, \cdots, v_n$, is defined as the area of the triangle formed by v_i and its two adjacent vertices, namely, v_{i-1}, v_i, v_{i+1}. The *VW* algorithm considers that the "importance" of the points are proportional to their *effective area* and, therefore, it ranks the points and simplifies the polylines by removing first the points with smaller areas.

Even though the *VW* algorithm performs simplification with good quality, it does not avoid topological problems in the map. To solve this problem, Gruppi et al.[Gruppi et al. 2015] developed *TopoVW*, a variation of the *VW* algorithm that avoids the creation of topological inconsistencies. Similarly to *VW*, *TopoVW* processes the points in an order based on their *effective area* but only removes a point v_i if its removal does not create inconsistencies in topology. When a point is removed the effective areas of its two neighbor points in the line are updated since the triangle associated with them change. *TopoVW* may be configured to stop when the number of points removed reaches a limit or when the smallest effective area of the points is greater than a given threshold.

Although some of the methods previously mentioned have mechanisms to detect and prevent topological inconsistencies created by the simplification process itself, these problems may still happen because of round-off errors related to the use of inexact arithmetic to process the points' coordinates.

3.2. Round-off Errors in Floating Point Arithmetic

The computational representation of a non-integer number is made by adjusting this number to a finite sequence of bits, this possibly causes the number to be an approximation most of the time. Furthermore, even if some numbers can be exactly represented, arithmetic operations applied to these numbers may generate a result that is not exactly correct. In geometric algorithms this is a great issue since they may result in inconsistent outputs.

Kettner et al. [Kettner et al. 2008] presented a study of how rounding in floating point arithmetic affects the planar orientation predicate and as consequence the planar convex hull problems. The planar orientation predicate is the problem of finding whether three points p, q, r are collinear, make a left turn, or make a right turn. This predicate is computed by verifying the sign of a determinant involving the points.

This determinant will be positive, negative or zero which means that points (p, q, r) form a left turn, right turn or are collinear, respectively. Due to round-off errors in floating point arithmetic the results can be classified incorrectly due to *rounding to zero*, *perturbed zero*, or *sign inversion*. Respectively, it means a non-zero result may be rounded to zero, a zero result may be mis-classified as positive or negative, and a positive result may be mis-classified as negative or vice-versa.

To observe the occurrence of issues caused by floating-point arithmetic, Kettner et al. [Kettner et al. 2008] developed a program to apply planar orientation predicate ($orientation(p, q, r)$) on a point $p = (p_x + xu, p_y + yu)$ where u is the step between adjacent floating point numbers in the range of p and $0 \leq x, y \leq 255$. This results in a 256×256 matrix containing either 1, -1 or 0 if the point corresponding to the matrix position is to the right, to the left or on the line that passes through q and r. Figure 3 shows the geometry of this experiment for $p = (0.5, 0.5)$, $u = 2^{-53}$ and $q = (12, 12)$ and $r = (24, 24)$. White cells represent correct output. The black diagonal line is an approximation of line (q, r). Black cells represent incorrect output, that is, black points above the diagonal were considered to form a right turn with the line (q, r), which is not true, it also applies to the points below the diagonal which were said to form a left turn with line (q, r). Gray cells contain points considered collinear to (q, r). According to Kettner et al., even using extended double arithmetic was not enough to overcome this issue.

As shown by [Kettner et al. 2008], these inconsistent results in

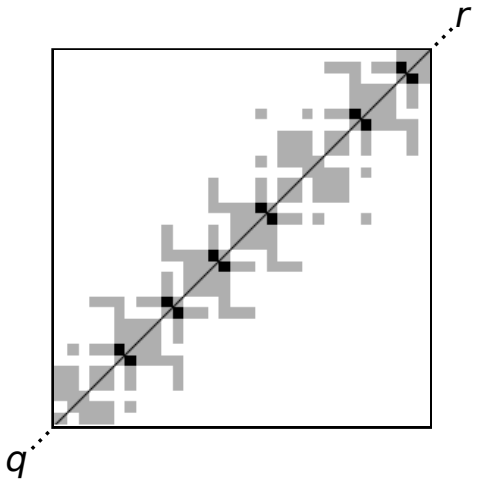


Figure 3. Geometry of the planar orientation predicate for double precision floating point arithmetic. White points represent correct outputs, gray were considered collinear and black cells points considered to be in the wrong side of the line. The diagonal line is an approximation to line (q, r). This Figure was created based on the experiments performed by Kettner et al.[Kettner et al. 2008].

$orientation(p, q, r)$ predicate could make algorithms that use this predicate (such as the Incremental Convex Hull algorithm) to fail.

A well-known technique to get around round-off errors in floating point arithmetic is the Epsilon-tweaking, that consists in comparing numbers using a relatively small tolerance value epsilon (ϵ). In practice, epsilon-tweaking fails in several situations [Kettner et al. 2008]. Snap rounding is another method to approximate arbitrary precision segments into fixed-precision numbers [Hobby 1999]. However, Snap rounding can generate inconsistencies and deform the original topology if applied consecutively on a data set. Some variations of this technique attempt to get around these issues [de Berg et al. 2007, Hershberger 2013].

One of the most robust ways for eliminating rounding errors in geometry is by using Exact Geometric Computation (ECG). According to Li [Li et al. 2005], any problem handled by other approaches can also be solved by ECG. Additionally, ECG can do even more and the solutions may be of higher quality. This can be achieved by using arbitrary precision rational numbers [Li et al. 2005], which eliminates rounding errors but considerably decreases performances as most operations are more computationally intensive.

4. Evaluation of Round-off Errors on Map Simplification

Similarly to other geometric problems, map simplification is also affected by round-off errors. As mentioned in section 3, *TopoVW* processes points in an order defined by their effective areas and only removes a point if its removal does not cause topological inconsistencies on the map. Given a polyline point v from a map, the removal of v causes a topological inconsistency if and only if there is another point (that may be a polyline or a control point) inside the triangle formed by v and its two adjacent vertices in its polyline.

If the *point-in-triangle* test fails returning a false positive a point that could have

been removed from the polyline will not be removed. If this test returns a false negative, on the other hand, topological inconsistencies may be created on the map.

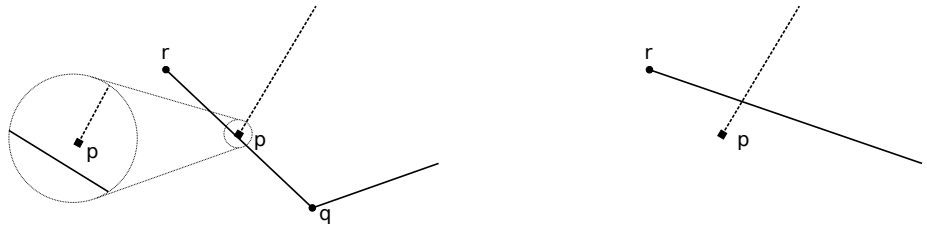
In *TopoVW*, the test to determine if a point p lies inside the triangle T formed by points r, s and t is performed by computing the barycentric coordinates of p in T, i.e., p is expressed in terms of three scalars a, b and c such that $p_x = ar_x + bs_x + ct_x$, $p_y = ar_y + bs_y + ct_y$, and $a + b + c = 1$. Point p lies in T if and only if $0 \leq a \leq 1$ and $0 \leq b \leq 1$ and $0 \leq c \leq 1$. A function *is_inside*(r, s, t, p) to perform the *point in triangle* test using the barycentric coordinates was implemented in C++. This approach is similar to the one used by Kettner et al. shown in Section 3.2.

In a similar manner to the orientation test presented in the previous section, the function *is_inside*(r, s, t, p) may also return incorrect results in two situations:

- *false inside*: erroneously determine an outer point as inside;
- *false outside*: erroneously determine an inner point as outside;

Since *is_inside* is *TopoVW*'s key operation, the method may avoid simplifying lines due to *false inside* appearance. Even more alarming, it may remove points on the presence of *false outsides*, what would change the topological relationships. Figure 4 shows an example of *false outside* simplification. In this example there are two non-intersecting lines (solid and dashed) as shown in Figure 4(a), the zoomed area shows explicitly that both lines do not intersect. Point p is inside the triangle formed by points (r, q, w) with w not shown in the figure to preserve simplicity. However, due to a *false outside* failure point q is removed resulting in an intersection as seen in Figure 4(b).

Another instance of this problem is shown by Figure 5, where a single line is simplified. Similarly to the previous example, vertex p is inside the triangle formed by (r, q, w) but it is seen as a *false outside*. Vertex q is removed by the simplification process causing the line to self-intersect as seen in Figure 5(b).

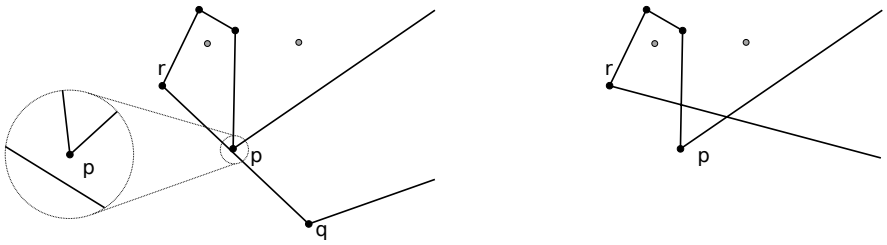


(a) (b)

Figure 4. (a) Example input on which false outside failure occur, two lines (solid and dashed) do not intersect. (b) Result of simplification with false outside, the removal of point q causes the lines intersect after simplification.

5. The *EPLSimp* Method

To avoid adding topological errors to the map in the situations described in section 4, we developed *EPLSimp*, a simplification algorithm based on *TopoVW* that uses exact arithmetic to completely avoid the round-off errors that may happen during the point in triangle tests. In *EPLSimp*, all non-integers variables are represented using arbitrary-precision



(a) (b)

Figure 5. (a) Example input of a single line and the occurrence of a false outside. (b) Simplification with a false outside point. The removal of point q produces a self-intersecting line.

rational numbers. Since exact arithmetic is usually much slower than arithmetic with floating point numbers (that usually can be performed natively on the CPU), some optimizations were implemented in order to reduce the performance penalty that it introduces.

First, similarly to *TopoVW*, we used a uniform grid to index the polyline and control points from the map. The idea is to create a regular grid, superimpose it with the map and insert in each cell c the control points and polyline points that are inside c. Then, given a triangle T, only points in the uniform grid cells intersecting T need to be tested in order to verify if there is a point inside T.

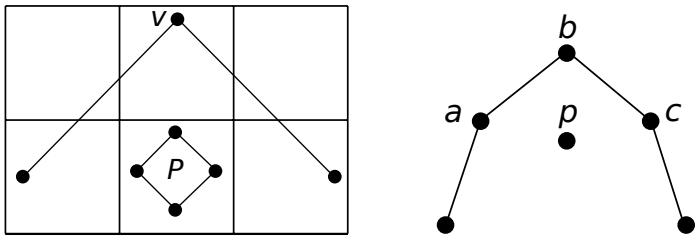
One advantage of the uniform grid over more complex data structures such as quadtrees is that it is easier to be constructed and maintained. Given a set S of points, we compute the uniform grid by performing only one pass through the dataset: for each point p in S, the cell c from the grid where p should be is computed (by dividing p's coordinates by the dimensions of the grid cells) and p is inserted in c.

Since the slowest step during the construction of the grid is the computation of the cell in which each point p is (due to the division operations with arbitrary-precision rationals), we used parallel programming to accelerate this step. The idea is to pre-compute in parallel the cell in which each point is and, after that, insert the points in the cells (this insertion step is not done in parallel in order to avoid the cost of synchronizations).

After indexing the points, the next step consists in simplifying polylines. As mentioned in section 3, *TopoVW* sorts the points based on their effective areas and processes them by removing the ones whose removal would not create topological problems in the map. To accelerate the simplification process used in *TopoVW*, we subdivided the polylines into sets such that polylines from different sets may be simplified independently in parallel not requiring the synchronization of data structures accesses.

Algorithm 1 presents the simplification algorithm and the strategy used for subdividing the polylines into sets that can be simplified in parallel. This subdivision is also performed using a uniform grid (this grid may have a resolution different from the uniform grid used for indexing the points). We create this new uniform grid and, then, insert in each grid cell the polylines that are completely inside this cell. The polylines in different grid cells could be processed independently since the triangle formed by any polyline point never contains a point from another cell. On the other hand, polylines intersecting

more than one cell cannot be processed in parallel without synchronization. For example, even though the polyline containing the vertex v in Figure 6 (a) does not intersect the cell containing the polygon P, before deleting v it is necessary to access the cell containing polygon P in order to verify if the deletion of v causes a topological inconsistency. Therefore, if the two polylines in this figure are simplified in parallel the algorithm would need to perform synchronizations.



(a) (b)

Figure 6. (a) Example where a polyline intersecting multiple cells needs to access data in a cell it does not intersect. (b) Example where the deletion of a point makes the deletion of other points infeasible .

After processing all the polylines lying completely in single cells, we repeat the simplification process for the polylines intersecting more than one cell. In order to be able to do that in parallel, we reduce the uniform grid resolution, reclassify the remaining polylines and, then, simplify the ones that lie in single cells in this new uniform grid. This process is repeated until there is no more polyline to be simplified (eventually all the polylnes will be processed since when the uniform grid is reduced to one cell all polylines that were not processed yet will lie in this unique cell).

To avoid the necessity of synchronizations between threads processing different sets of polylines, the simplification stopping criteria used in *EPLSimp* is the effective area of the points. That is, the thread simplifying a set of polylines stops the process whenever the point with smallest effective area in the set has an area greater than a given threshold. If the stopping criteria was the number of points removed, synchronizations would be necessary to ensure that all threads stop simplifying lines when the global number of removed points reaches the target number.

It is important to mention that we have considered other two parallelization strategies. First, we could pre-process the map verifying for each point if there is another point inside the triangle defined by it and its two neighbors. This pre-processing could be performed in parallel. After labeling the points that can safely be removed (that is, the ones without other points in their triangles), we could just remove the ones with smaller effective areas. This strategy would not work very well because when a point is removed the triangle of its two neighbors change. For example, in Figure 6 (b), any of the points a or b or c may be removed without changing the topological relationship between the polyline and the control point p. However, if a or c is removed the triangle associated with b will contain p and, therefore, b will not be a candidate to be removed anymore.

Another parallel strategy would be to perform the point inside triangle test in parallel. That is, given a triangle T, after using the uniform grid to select the points that are candidate to be in T we could perform the test to verify if each point is really inside T

Algorithm 1 Parallel map simplification algorithm.

1: M: input map
2: $MaxArea$: maximum effective area of a point to be removed
3: $GridSize$: initial resolution of the uniform grid used to separate the polylines.
4: **while** $GridSize > 0$ **do**
5: $ug \leftarrow GridSize \times GridSize$ uniform grid
6: **for** each polyline p in M not simplified yet **do**
7: **if** p is completely inside a cell c from ug **then**
8: Insert p into c
9: **end if**
10: **end for**
11: **for** each cell c in ug **do** //Parallel for loop
12: //Iterate in an order based on the points' effective areas
13: **for** each point v_i in polylines from $c \mid effectiveArea(v_i) < MaxArea$ **do**
14: **if** $\nexists$ point $p \mid$ *is_inside*$(v_{i-1}, v_i, v_{i+1}, p)$ **then**
15: Remove the point v_i from its polyline
16: **end if**
17: **end for**
18: **end for**
19: $GridSize \leftarrow GridSize/2$
20: **end while**

in parallel. However, preliminary experiments showed that, because of the uniform grid, the average number of points that need to be effectively tested in this step is usually small and, therefore, the performance gain obtained by processing them in parallel would not be good if compared with the overheads associated with the parallelism.

6. Experimental Evaluation

We evaluated *EPLSimp* by implementing it in C++ (the library GMPXX [Granlund and the GMP development team 2014] was used to provide arbitrary precision arithmetic) and performing experiments in some small datasets artificially generated to contain polylines and control points that would introduce topological errors in the simplification performed by *TopoVW*. Furthermore, experiments were performed in 3 real-world maps in order to evaluate the performance of *EPLSimp*. The computer used has a dual E5-2687 8-core/16-thread Intel Xeon CPU and 128 GB of RAM.

In the first set of experiments, we used the maps artificially generated to contain points in positions where the point-in-triangle tests would give a false negative answer (similar to the examples presented in section 4) and, therefore, methods such as *TopoVW* would create topological errors during the map simplification. As expected, because of the use of exact arithmetic, *EPLSimp* was able to simplify these maps without creating any topological inconsistency.

Next, we performed experiments in three datasets to verify the overhead added by the use of arbitrary precision rational numbers in *EPLSimp*. Dataset 1 was the largest dataset used in the ACM GISCUP competition 2014. It contains 30000 polyline points and 1607 control points. Dataset 2 represents the Brazilian county subdivision map avail-

able in the IBGE (the Brazilian geography agency) website and it contains 300000 polyline points and 10000 control points (the control points were positioned randomly in the map). Dataset 3 represents the United States county subdivision map available in the United States Census website and it has 4 million polyline points and 10 million control points (that were also positioned randomly in the map).

The choice of the dimensions of the uniform grid used by *TopoVW* and *EPLSimp* to index the points affects the performance of both methods and it can be performed using several strategies. For example, *TopoVW* automatically defines the grid size by computing the total number of polylines/control points in the map and chooses the grid dimension estimating the average number of points per cell close to a constant (this constant was determined experimentally). Since the best grid size for *TopoVW* may not be the best grid size for *EPLSimp* and since we want to compare the performance of these two methods, we chose experimentally, for each method and dataset, a configuration that presents the best performance (for example, in dataset 2, *TopoVW* and *EPLSimp*used grids with, respectively, 512^2 and 2048^2 cells).

The uniform grid that *EPLSimp* uses to classify the polylines that are processed in parallel was configured to have initially 256^2 cells and to iteratively reduce the resolution to half after completely processing each set of polylines that can be processed in parallel. As mentioned in section 5, this process is repeated until all polylines have been simplified, what happens, in the worst case, when the grid has only one cell.

Table 1 presents the wallclock-time (in milliseconds) of the two methods in two situations: in the first one they were configured to remove the maximum amount of points that they can remove without creating topological errors. In the second one, they were configured to remove 50% of the points. Row *initialize* contains the time for initializing the algorithm and includes the time for creating the data structures (such as the uniform grids). Row *simplify* contains the time spent in the simplification process. In all tests *EPLSimp* was tested using 16 *threads*.

EPLSimp was, on average, less than twice slower than *TopoVW*, even though we store and process all points coordinates using arbitrary precision rational numbers, that are much more computationally expensive to process than floating point numbers. This happens because *EPLSimp* was carefully implemented using techniques such as parallel computing and the uniform grid to accelerate the simplification process. It is worth mentioning that one of the advantages of the uniform grid over other indexing techniques (such as Quadtrees) is that it is easily parallelizable and can be created by performing a single pass over the data (this is particularly important for efficiency since the indexing is performed using coordinates represented by rational numbers).

Table 2 evaluates the scalability of *EPLSimp* considering 5 different number of threads. In these datasets, *EPLSimp* had a speedup of $2\times$ when two threads were used and this speedup increased slowly for larger amounts of threads. For example, the running-time using 16 threads was not much different from the time using 8 threads. Some reasons for this behavior are: first, due to Amdahl's law, sequential parts of the algorithm limits its scalability; furthermore, some polylines sets may take more time to be simplified than others, what causes load imbalance in the threads; finally, when several threads run in parallel the memory accesses may saturate the memory bus. Anyway, it is worth mentioning

Table 1. Times (in ms) for the main steps of the map simplification algorithms. Rows *Max.* represents the time for removing the maximum amount of points from the map while rows *Half* represents the time to remove half of the points.

	Dataset	1		2		3	
	Method	TopoVW	*EPLSimp*	TopoVW	*EPLSimp*	TopoVW	*EPLSimp*
Max.	Initialize	4	22	28	190	1828	5353
	Simplify	39	60	626	445	46069	57095
	Total	43	82	654	635	47897	62448
Half	Initialize	4	22	28	186	1847	5447
	Simplify	25	41	357	331	23021	48090
	Total	29	63	384	517	24868	53537

Table 2. Times (in ms) for initializing and simplifying maps from the 3 datasets considering different number of threads. The simplification was configured to remove the maximum amount of points from the maps.

		Initialization			Simplification		
	Dataset	1	2	3	1	2	3
Threads	1	71	655	26833	176	1574	250237
	2	91	568	15483	152	1150	131310
	4	54	422	9853	99	689	82641
	8	34	240	6552	61	483	62089
	16	22	190	5353	60	445	57095

that typical computers nowadays have 2 or 4 cores and, therefore, *EPLSimp* is able to present a good scalability in those computers.

7. Conclusion and Future Works

This paper presented *EPLSimp*, an algorithm for map simplification that does not produce topological inconsistencies. It uses arbitrary precision numbers to avoid round-off errors caused by floating-point arithmetic, which could lead to topological inconsistencies even in methods designed to avoid these problems, such as *TopoVW*.

EPLSimp was implemented to be efficient even though it uses arbitrary precision numbers, which are much slower to be processed than floating-point numbers. This efficiency improvement was achieved by using a uniform grid to index the geometric objects and, also, high performance computing. As a result, using 16 threads *EPLSimp* was, on average, less than twice slower than *TopoVW*, even though the latter performs all computation using inexact floating-point numbers (that are natively supported by the CPU) and then can generate "wrong" (or inconsistent) results.

For future work, we intend to develop other GIS algorithms using arbitrary precision arithmetic. Furthermore, adapting *EPLSimp* to simplify vector drawings and 3D objects is also an interesting future research topic.

8. Acknowledgement

This research was partially supported by CNPq, CAPES (Ciência sem Fronteiras), FAPEMIG and NSF grant IIS-1117277.

References

de Berg, M., Halperin, D., and Overmars, M. (2007). An intersection-sensitive algorithm for snap rounding. *Computational Geometry*, 36(3):159–165.

Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122.

Goldberg, D. (1991). What every computer scientist should know about floating-point aritmetic. *ACM Computing Surveys*, 23(1):5–48.

Granlund, T. and the GMP development team (2014). The gnu multiple precision arithmetic library.

Gruppi, M. G., Magalhães, S. V. G., Andrade, M. V. A., Franklin, W. R., and Li, W. (2015). An efficient and topologically correct map generalization heuristic. In *Proceedings of the 17th International Conference on Enterprise Information Systems (ICEIS)*.

Hershberger, J. (2013). Stable snap rounding. *Computational Geometry*, 46(4):403–416.

Hobby, J. D. (1999). Practical segment intersection with finite precision output. *Computational Geometry*, 13(4):199–214.

Jiang, B., Liu, X., and Jia, T. (2013). Scaling of geographic space as a universal rule for map generalization. *Annals of the Association of American Geographers*, 103(4):844–855.

João, E. (1998). *Causes and Consequences of map generalization*. CRC Press.

Kettner, L., Mehlhorn, K., Pion, S., Schirra, S., and Yap, C. (2008). Classroom examples of robustness problems in geometric computations. *Computational Geometry*, 40(1):61–78.

Li, C., Pion, S., and Yap, C.-K. (2005). Recent progress in exact geometric computation. *The Journal of Logic and Algebraic Programming*, 64(1):85–111.

Li, L., Wang, Q., Zhang, X., and Wang, H. (2013). An algorithm for fast topological consistent simplification of face features. *Journal of Computational Information Systems*, 9(2):791–803.

Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256.

Saalfeld, A. (1999). Topologically consistent line simplification with the douglas-peucker algorithm. *Cartography and Geographic Information Science*, 26(1):7–18.

Visvalingam, M. and Whyatt, J. (1993). Line generalisation by repeated elimination of points. *The Cartographic Journal*, 30(1):46–51.

Combining Time Series Features and Data Mining to Detect Land Cover patterns: a Case Study in Northern Mato Grosso State, Brazil

Alana K. Neves[1], Hugo do N. Bendini[1], Thales S. Körting[1], Leila M. G. Fonseca[1]

[1]Instituto Nacional de Pesquisas Espaciais – INPE

Caixa Postal 515 - 12227-010 - São José dos Campos - SP, Brasil

alana.neves@inpe.br,{hnbendini,tkorting,leila}@dpi.inpe.br

Abstract. *One product of the MODIS sensor (Moderate Resolution Imaging Spectroradiometer) is the EVI2 (Enhanced Vegetation Index). It generates images of around 23 observations each year, that combined can be interpreted as time series. This work presents the results of using two types of features obtained from EVI2 time series: basic and polar features. Such features were employed in automatic classification for land cover mapping, and we compared the influence of using single pixel versus object-based observations. The features were used to generate classification models using the Random Forest algorithm. Classes of interest included Agricultural Area, Pasture and Forest. Results achieved accuracies up to 91,70% for the northern region of Mato Grosso state, Brazil.*

1. Introduction

Since the 50s, Amazon's occupation was characterized by expanding the agriculture frontier, which resulted in extensive and accelerated transformations. This period was marked by high and continuous deforestation rates, especially in the areas located in the so-called "arc of deforestation" (BECKER, 1990, 2009). Currently, in the Amazon, large areas of pasture, agriculture, reforestation and secondary vegetation can be found and much of the primary forest is limited to protected areas (BECKER, 2009).

Due to its complexity, there is still no complete understanding of the dynamic of landscape evolution in the Amazon region. This is because of the great heterogeneity of land use and occupation since the implementation of old governmental colonization projects and new federal infrastructure projects. To help in understanding the Amazon landscape, INPE (Brazil's National Institute for Space Research), in partnership with EMBRAPA (Brazilian Agricultural Research Corporation), produces land cover data about Legal Amazon, in a project known as TerraClass – mapping of land use and land cover change in legal Amazon deforested areas (COUTINHO et al., 2013). TerraClass presents to the society information related to which are the current main activities (spatially and numerically) in deforested areas in a specific year. TerraClass information is currently available for years 2008, 2010 and 2012.

To achieve the proposed goal, most of the TerraClass interpretation and classification is done visually and manually, which is a very time consuming task. The annual agriculture mapping of TerraClass is based on an automatic method, which used minimum and maximum values of NDVI (Normalized Difference Vegetation Index)

time series. In agricultural areas, vegetation indices present low values in the beginning of agricultural cycle and high values in vegetation peaks. The difference between these two moments above a certain limit corresponds to agriculture pattern (ADAMI et al., 2015). Although there are some efforts to automate its methodology, there is still space to study more adequate data and methods to improve automatic classification results.

Since the 70s, acquisition data through remote sensing is a practice of increasingly importance and, more and more, is becoming fundamental in the knowledge of Earth's phenomena. Interpreting these phenomena only by *in situ* observations would require such an amount of resources (human, time and money).

Remote sensors, like MODIS (Moderate Resolution Imaging Spectroradiometer), have been responsible for systematically collect images of Earth, which can be converted into image time series (VUOLO, 2012). MODIS products include vegetation indices, capable of providing spatial and temporal comparisons of global vegetation conditions. The well-known vegetation indices available are the NDVI and the EVI2 (Enhanced Vegetation Index). NDVI is more sensitive to the presence of pigments such as chlorophyll, while EVI2 is related to changes in canopy structure, such as Leaf Area Index (LAI), vegetation type and vegetation physiognomy (HUETE et al., 2002). For this reason, the study of EVI2 time series also allows to obtain information about soil cover.

One of the techniques used to manipulate large amount of observations present in a time series is data mining. Data mining consists of a supporting tool, through the discovering of correlations, patterns and trends in data, combining technologies of pattern recognition, mathematics and statistics (LAROSE, 2014). Such techniques have already been employed in remote sensing, combining data mining techniques and vegetation indices time series.

Costa et al. (2015) used EVI data to differentiate pasture and native grassland in the Brazilian biome named Cerrado, comparing Support Vector Machine, Multilayer Perceptron and Autoencoder algorithms. Others efforts to classify land cover in Amazon include the use of Naïve Bayes, Nearest Neighbor and Optimum Path Forest algorithms (NOMA et al., 2013; BARBOSA ET AL., 2015). Random Forest algorithm is not so usual in remote sensing applications, but it is a powerful machine learning and it as expanding its applicability in land studies by remote sensing (RODRIGUEZ-GALIANO et al., 2012), even using vegetation indices data (NITZE et al., 2015) for image acquisition optimization for land cover classification.

Time series can be related to land patterns using feature extraction (GALFORD et al., 2008). There are several types of features, such as basic and polar features (KÖRTING, 2012), that can be combined to assist in classification models. Thus, this work aims to generate classification models to detect land cover testing different time series features, in a test area in northern Mato Grosso state, Brazil, which belongs to the arc of deforestation.

2. Methodology

In Figure 1 we present a flowchart of the employed methodology, which will be better explained as follows.

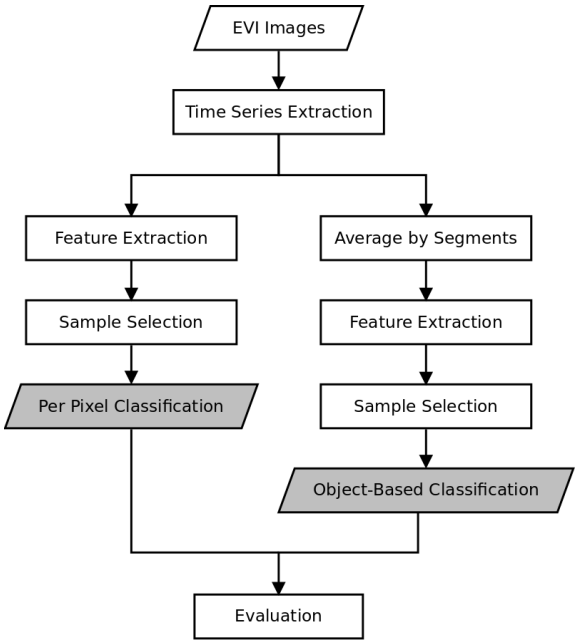


Figure 1. Methodology flowchart

2.1. Study Area

The study area (Figure 2) chosen for this work is the path-row 227-068 of TM sensor Landsat 5 satellite. The scene is located in the northern Mato Grosso (MT), Brazil and covers part of eight municipalities: Juara, Nova Canaã do Norte, Itaúba, Tabaporã, Porto dos Gauchos, Itanhangá, Ipiranga do Norte and Sinop. The scene belongs to the agriculture frontier in the arc of deforestation. MT is one of the three Brazilian states with the largest deforested area in the Amazon (MARGULIS, 2003).

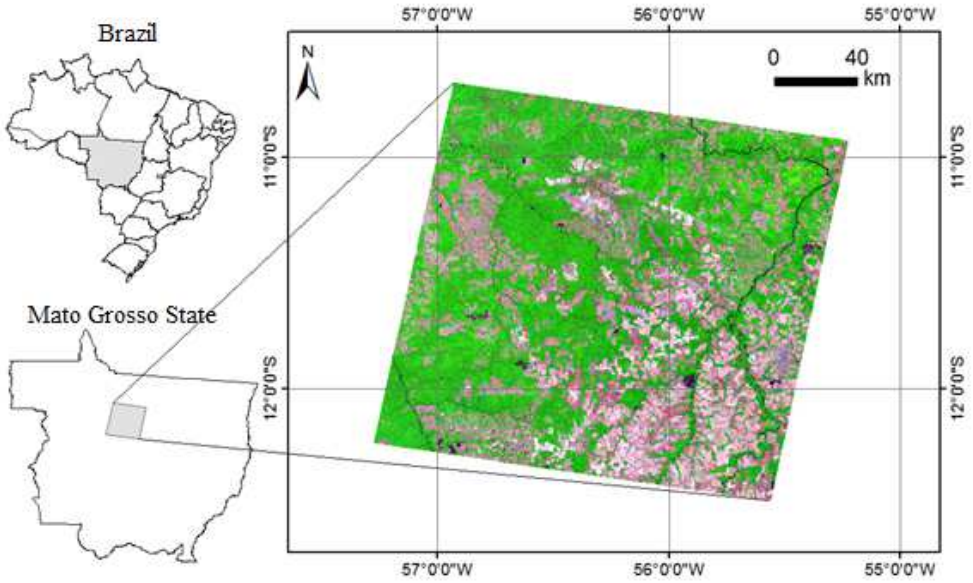


Figure *2*. Study area: TM sensor (Landsat 5) path-row 227-068.

2.2 Time Series Extraction

The data used in our experiments was obtained from the modified 2-band EVI2 (Equation 1) from the product MOD13Q1 of MODIS sensor, with spatial resolution of 250 meters and temporal resolution of 16 days (SOLANO et al., 2010).

$$EVI2 = 2.5\frac{\rho_{NIR}-\rho_{\mathrm{Red}}}{1+\rho_{NIR}+\rho_{\mathrm{Red}}} \quad (1)$$

where ρ_{NIR} is the Near Infra-red reflectance and ρ_{Red} is the reflectance of the red band.

Because of its temporal resolution, EVI2 generates cycles of around 23 observations each year, that combined can be interpreted as time series. The 46 (23 for 2008 and 23 for 2010) EVI2 images were downloaded from http://earthexplorer.usgs.gov/. For each year, images were ordered by time (Figure 3) and time series were composed for each pixel.

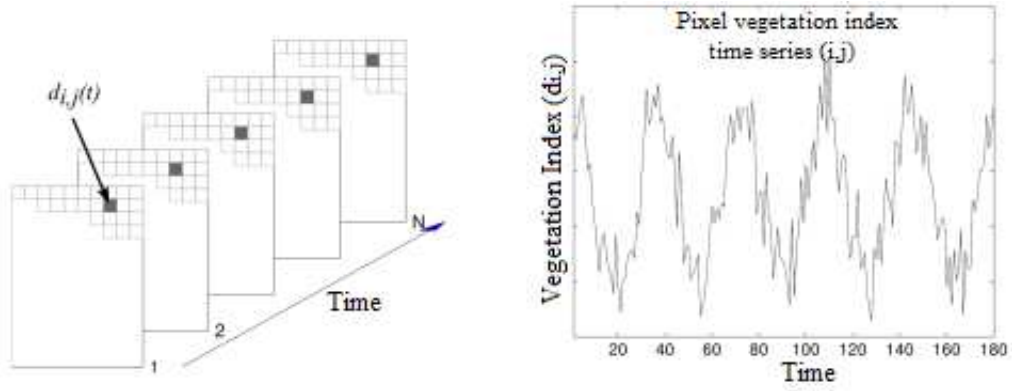


Figure 3. Per-pixel feature extraction and time series composition (Adapted from Eklundh & Jönsson, 2012).

2.3. Feature Extraction

Two approaches were used: per pixel, where each pixel has its respective time series, and object-based. Using objects means that the imagery was partitioned into homogeneous regions, so spatial, spectral and temporal characteristics can be included in the analysis (HAY & CASTILLA, 2006). In this work, objects from TerraClass were used to group pixels with similar behavior and their time series were represented by the average of all-time series of pixels present in each object. Since TerraClass information is available for years 2008 and 2010 (also 2012), we used time series from the years 2008 and 2010 in our analysis.

With temporal resolution of 16 days and spatial resolution of 250 m, EVI2 data from MODIS generates cycles of around 23 observations each year, that combined can be interpreted as time series. Several features can be extracted from each time series. In this work, two groups of features were extracted, according to the methodology proposed by Körting (2012): the so called basic and polar features. Basic features includes statistical measures such as mean, standard deviation, minimum and maximum values of the curve (Figure 4).

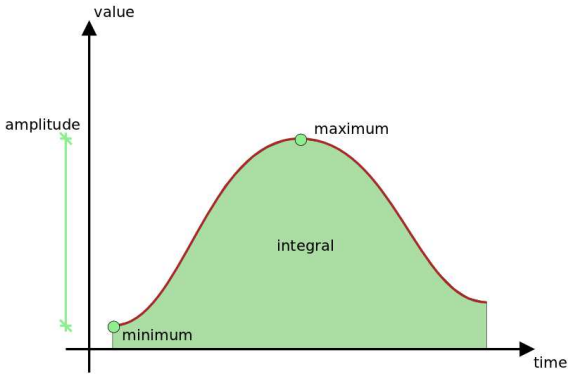


Figure 4. Example of basic features representation.

Many natural phenomena can be represented by cyclical patterns, such as agriculture. Cycles can be characterized by rise and fall oscillations in series. To support the cycles visualization, a way of plotting was proposed, adapted from Edsall et al. (1997), where each cycle value is projected in angles in the interval [0,2π] (Figure 3) (KÖRTING, 2012). This projection generates an object with a closed contour, whose properties can represent some specific behavior of the original time series.

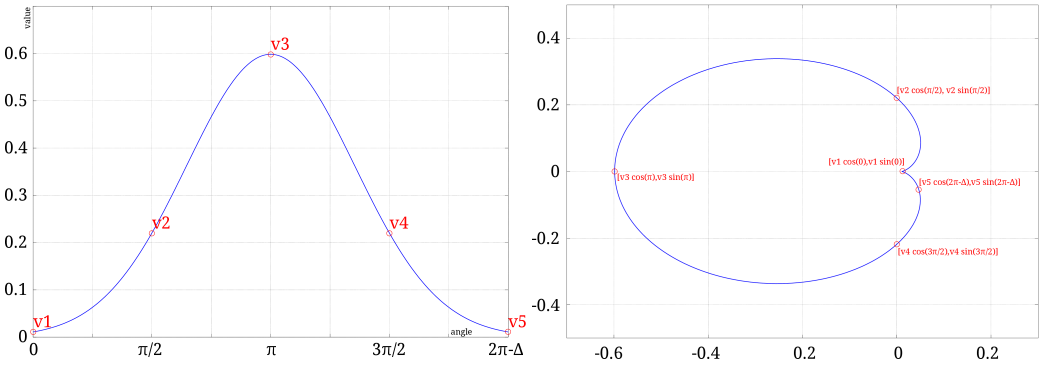


Figure 5. Example of a cycle in a time series. On left, one time series example. On the right, its polar representation according to Körting (2012).

From the polar visualization of these objects, several features can be generated such as eccentricity, angle of orientation, area per season and others. Both basic and polar features are described in the Table below.

Table 1. Description of basic and polar features from time series.

Name	Description	Type	Range
Amplitude	The difference between the cycle's maximum and minimum values. A small amplitude means a stable cycle.	Basic	[0, 1]
Area	Area of the closed shape. A higher value indicates a cycle with high EVI values.	Polar	≥0
Area per Season	Partial area of the closed shape, proportional to a specific quadrant of the polar representation. High value in the summer season can be related to the	Polar	≥0

	phenological development of a cropland.		
Circle	Returns values close to 1 when the shape is more similar to a circle. In the polar visualization, a circle means a constant feature.	Polar	[0, 1]
Cycle's maximum	Relates the overall productivity and biomass, but it is sensitive to false highs and noise.	Basic	[0, 1]
Cycle's mean	Average value of the curve along one cycle.	Basic	[0, 1]
Cycle's minimum	Minimum value of the curve along one cycle.	Basic	[0, 1]
Cycle's std	Standard deviation of the cycle's values.	Basic	≥0
Cycle's sum	When using vegetation indices, the sum of values over a cycle means the annual production of vegetation.	Basic	≥0
Eccentricity	Return values close to 0 if the shape is a circle and 1 if the shape is similar to a line.	Basic	[0,1]
First slope maximum	It indicates when the cycle presents some abrupt change in the curve. The slope between two values relates the fastness of the greening up or the senescence phases.	Basic	[-1, 1]
Gyration radius	Equals the average distance between each point inside the shape and the shape's centroid. Smaller values stand for shapes similar to a circle.	Polar	≥0
Polar balance	The standard deviation of the areas per season, considering the 4 seasons. Small value point to constant cycles, e.g. the EVI of water (with a small Area), or forest (with a medium Area).	Polar	≥0

2.4. Samples Selection, Classification and Evaluation

After the feature extraction, the automatic classification was made on software WEKA 3.6 (HALL et al., 2009). We used the Random Forest algorithm, which creates a set of decision trees used to classify the full data set. The use of this algorithm in remote sensing applications is relatively new, but it has proven to be powerful in land-cover classification (RODRIGUEZ-GALIANO et al., 2012). The number of decision trees to be used is defined by the domain's expert. In our experiments we defined this parameter empirically, based on the accuracy of results and the time needed (computational cost) to classify all data. Models were built using training samples from the year 2008 and reevaluated in 2010. To evaluate the classification accuracy in 2008, we divided the samples in two subsets. 66% of the data was used for training and 34% was used for testing. Three interest classes were discriminated: Forest, Pasture and Agriculture.

In the Random Forest algorithm, data are partitioned randomly in many subsets by the Bootstrap technique (resampling with replacement), in which some records may appear several times in the same subset while others do not appear even once. Each subset generates a decision tree and all the decision trees have a vote with a certain

weight to contribute in the decision about the class that will be assigned to the object (HAN et al., 2011).

We also tested different combination of features:

- Time Series, Basic Features and Polar Features;
- Basic and Polar Features;
- Time Series and Basic Features;
- Time Series and Polar Features;
- Only Time Series;
- Only Basic Features;
- Only Polar Features.

The classification generated by TerraClass is based on the interpretation of Landsat TM scenes, therefore objects from TerraClass were produced at the scale of 30m, differently from our input data from MODIS, whose spatial resolution is 250m. Land cover patterns from TerraClass include Annual Agriculture, Clean Pasture, Dirty Pasture, Forest, Urban Area, Mining, Occupation Mosaic, Regeneration with Pasture, Reforestation, Non Forest, Hydrography and Secondary Vegetation. Since there are different types of pasture, and also other classes which are unable to be recognized in MODIS spatial resolution, it was necessary to made some masking and a reclassification, where "Clean Pasture" and "Dirty Pasture" became a single class named Pasture. At Table 2, the reclassification made in TerraClass data to facilitate the comparison with the automatic classification. Those classes included in "Others" were not analyzed in the automatic classification, therefore their pixels were masked. Then it was considered that the image is composed only by the three targets of interest.

Table 2. Reclassification of TerraClass data for validation

TerraClass	Reclassification
Annual Agriculture	Agriculture
Clean Pasture Dirty Pasture	Pasture
Forest	Forest
Urban Area Mining Occupation Mosaic Regeneration with Pasture Reforestation Non Forest Hydrography Secondary Vegetation	Others

To test the model accuracy, we used evaluation and performance measures. As an evaluation measure of classification, one Error Matrix per year was generated for each approach.

3. Results and Discussion

The behavior of EVI2 time series in the study area can be seen in Figure 6, by pixel and object-based approach. These curves were generated with the mean of all the pixels or object time series for each class. High EVI2 values are observed on the periods between January and April, as well between October and December. This behavior reflects what is expected for the vegetation on this region, according to the annual seasonality, decreasing in greenness through the dry season and increasing during the rainy season, with an annual mean of 0.46 EVI2 and a maximum around 0.6 EVI2. The agriculture system had a more complex behavior, showing peaks next to 0.7 EVI2 between December – January, and March – April, and higher standard deviation (0,2). Similar values were found by Galford et al. (2008) in a study for detect croplands in Mato Grosso using time series wavelet analysis. In the Figure 6a we can see a more constant behavior in forest, around 0.5 EVI2, while in Figure 6b we observe that the forest mean temporal behavior was similar to pasture.

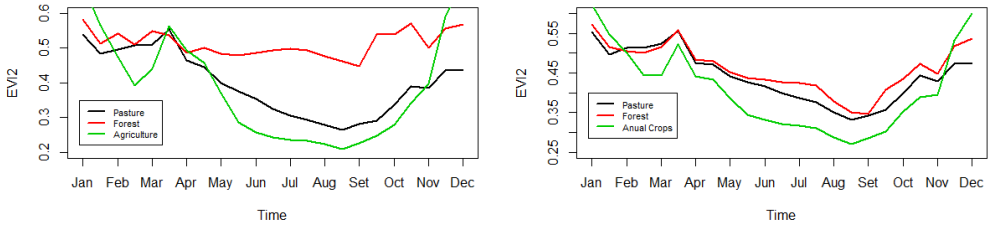


Figure 6. (a) EVI2 time series of both target by pixel based, with 65897, 219069 and 100335 for pasture, forest and agriculture respectively. (b) EVI2 time series of both target by object based approach, using 1685 segments for pasture, 1402 for forest and 492 for agriculture.

In our first experiment, resumed in Table 3, we tuned the Random Forest algorithm, by finding the best number of decision trees to be used. These performances were observed utilizing the full data (time series, polar and basic features) for the year 2008. The accuracy had little increase (less than 1%) while increasing the Number of Trees, and the Time to Build the Model almost doubled when compared to the previous. Thus, such little improvement in classification associated with the higher computational cost do not justify the use of more trees in the model. Therefore, it was chosen the number of 20 decision trees in all models used in the next results.

Table 3. Number of Trees and Performance Comparison

Number of Trees	10	20	50	100
Correctly Classified Instances (%)	90,97	91,3526	91,6056	91,6321
Time to Build Model (seconds)	257,73	482,14	1182,94	1941.07

Similarly to the obtained accuracies from Sato et al. (2013), the algorithm of Random Forest was satisfactory to distinguish patterns of land cover, although these authors have only used one Landsat image with four remote sensing products: MLME (Linear Spectral Mixture Model), NDVI (Normalized Difference Vegetation Index), NDWI (Normalized Water Index) and SAVI (Soil-Adjusted Vegetation Index).

The results of the second experiment are resumed in Table 4. The percentage of correctly classified instances for each approach shows that, in the situation studied in this work, both basic and polar features were efficient in distinguish Agriculture, Pasture and Forest, although using only the time series produced a better result (91,70%). All seven approaches obtained accuracies near 90%. Usually, the hit rate is higher in 2008, because the model was built in this year.

Table 4. Correctly Classified Instances (%) for each approach

	Per Pixel		Object-Based	
	2008	**2010**	**2008**	**2010**
Time Series, Basic Features and Polar Features	91,35	88,39	72,62	56,82
Basic and Polar Features	89,52	87,52	69,72	58,21
Time Series and Basic Features	91,43	88,34	72,62	57,37
Time Series and Polar Features	91,39	88,06	72,56	57,82
Only Time Series	91,70	88,09	72,31	54,22
Only Basic Features	89,38	87,00	69,65	57,14
Only Polar Features	84,84	83,33	64,96	53,77

Another important aspect is the fact that classification per pixel resulted in better accuracies (around 90%) than object based classification (around 60%). In Figure 7, the reference data from TerraClass and our result using automatic classification (only time series) can be compared.

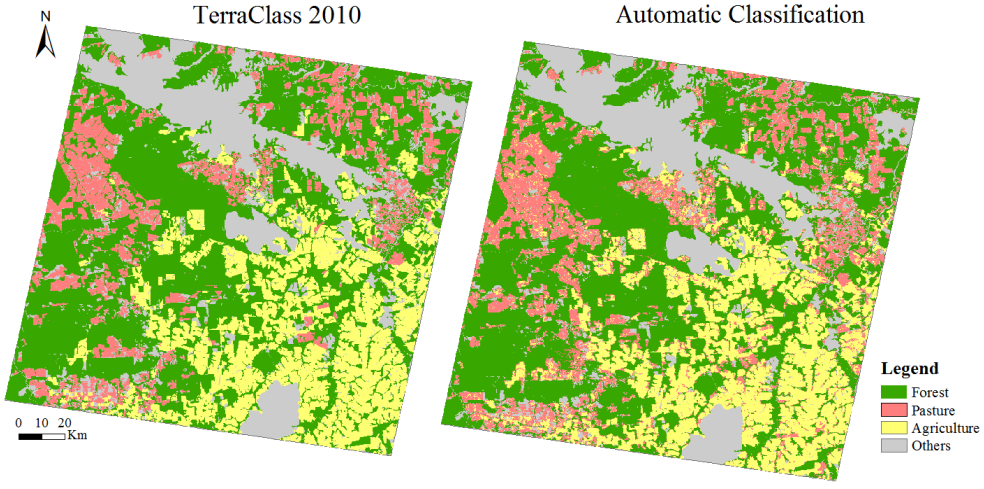


Figure 7. Comparation between Automatic Classification 2010 and its reference, TerraClass 2010. Elements of class 'Others' were not classified.

To have a more specific analysis for each class of interest, at Table 5 each class is represented and values are referring to the average of the seven approaches. As noticed before, in general the classification per pixel had a better performance, however when we observe the class Pasture, it shows that object based approaches increased correctly classified instances, which means that including spatial parameter in analyzes should improve its identification.

Table 5. Correctly classified instances (%) for each interest class.

	Per pixel		Object-based	
	2008	**2010**	**2008**	**2010**
Forest	95.82	92.81	37.65	20.75
Pasture	75.43	74.4	88.14	92.09
Agriculture	85.99	83.52	46.79	36.38

Despite the well-known good performance of object based classifications, in this case Forest and Agriculture had a better identification behavior when pixels time series were analyzed separately. According to Seyler, (2002), Pasture is a difficult class to be identified by only satellite sensor data. Because of the great quantity of mixed elements in its composition, like grass, trees, bush and others, it was harder to characterize it only by its behavior in time series.

4. Conclusions

Both basic and polar features from time series were satisfactory for the identification of the three interest classes. Forest and Agriculture classification had a great performance when using per pixel strategy, while Pasture was better differentiated when the object based approach were used. Random Forest algorithm showed to be robust enough to make a good separation between EVI2 patterns.

Although the automatic classification produced similar results to TerraClass data, it was inappropriate to make comparisons between mapped area for each approach because of the different spatial resolutions.

In future works, it is intended to analyze new interest classes and test results of segmentations that can include temporality of time series.

5. Bibliography

ADAMI, M; GOMES, A. R.; COUTINHO, A. C.; ESQUERDO, J. C. D. M.; VENTURIERI, A. Dinâmica de uso e cobertura da terra no estado do Pará entre os anos de 2008 a 2012. XVII Simpósio Brasileiro de Sensoriamento Remoto, 2015. **Anais...** João Pessoa, PB, 2015.

BARBOSA, D. P.; NOMA, A.; KÖRTING, T. S.; FONSECA, L. M. G. Um estudo experimental com classificadores baseados em regiões e perfis EVI. XVII Simpósio Brasileiro de Sensoriamento Remoto, 2015. **Anais...** João Pessoa, PB, 2015.

BECKER, B. K. Amazônia. **Série Princípios**. São Paulo: Ática, 1990. 92p.

BECKER, B. K. Amazônia: Geopolítica na virada do III milênio. Rio de Janeiro: Garamond, 2009. 172p.

COUTINHO, A. C.; ALMEIDA, C.; VENTURIERI, A.; ESQUERDO, J. C. D. M.; SILVA, M. **Projeto TerraClass: Uso e cobertura da terra nas áreas desflorestadas na Amazônia Legal**. Brasília, DF: Embrapa; Belém: INPE, 2013.

COSTA, W.; FONSECA, L.; KÖRTING, T. Classifying grasslands and cultivated pastures in the Brazilian cerrado using Support Vector Machines, Multilayer Perceptrons and Autoencoders. **Lecture Notes in Computer Science**. 1ed.: Springer International Publishing, 2015, v. 9166, p. 187-198.

EDSALL, R.; KRAAK, M.; MACEACHREN, A.; PEUQUET, D. Assessing the efectiveness of temporal legends in environmental visualization. **Proceedings of GIS/LIS**, Citeseer, p. 677{85, 1997. Available in: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.61.8955&rep=rep1&type=pdf>.

Eklundh, L., and Jönsson, P., 2012, TIMESAT 3.2 with parallel processing - Software Manual. Lund University, 88 pp.

GALFORD, G. L.; MUSTARD, J. F.; MELILLO, J.; GENDRIN, A.; CERRI, C. C.; CERRI, C. E. P. Wavelet analysis of MODIS time series to detect expansion and intensification of row-crop agriculture in Brazil. **Remote Sensing of Environment**. v. 112, p. 576-587. 2008.

HALL M.; FRANK, E., HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. **The WEKA Data Mining Software: An Update**. SIGKDD Explorations, Volume 11, Issue 1. 2009.

HAN, J.; KAMBER, M.; PEI, J. **Data mining**: concepts and techniques. 3ed. San Francisco: Morgan Kaufmann Publishers, 2011.

HAY, G. J.; CASTILLA, G. Object-based image analysis: strengths, weaknesses, opportunities and threats (swot). **The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**. OBIA, 2006.

HUETE, A.; DIDAN, K.; MIURA, T.; RODRIGUEZ, E. P.; GAO, X.; FERREIRA, L. G. Overview of the radiometric and biophysical performance of the MODIS vegetation indices. **Remote Sensing of Environment**, v. 83, n. 1, p. 195-213, 2002.

KORTING, T. S.; FONSECA, L. M.; ESCADA, M. I. S.; SILVA, F. C.; SILVA, M. P. S. GeoDMA: a novel system for spatial data mining. **IEEE International Conference on Data Mining Workshops, Pisa, Italia**, 2008. **Anais…** Pisa, Italia, 2008.

LAROSE, D. T. **Discovering Knowledge in Data: An Introduction to Data Mining**. John Wiley and Sons, Inc, 2 ed. 2014.

MARGULIS, S. **Causas do Desmatamento da Amazônia Brasileira**. 1ed. Brasília: Banco Mundial, 2003.

Nitze, Ingmar; Barrett, Brian; Cawkwell, Fiona (2015). Temporal optimisation of image acquisition for land cover classification with Random Forest and MODIS time-series. **International Journal of Applied Earth Observation and Geoinformation**, 34, 136-146.

NOMA, A.; KORTING, T. S.; FONSECA, L. M. G. Uma comparação entre classificadores usando regiões e perfis evi para agricultura. XVI Simpósio Brasileiro de Sensoriamento Remoto. **Anais...** São José dos Campos: Instituto Nacional de Pesquisas Espaciais, 2013. p. 2250–2257.

Rodriguez-Galiano, V. F.; Ghimire, B.; Rogan, J.; Chica-Olmo, M.; Rigol-Sanchez, J.P. An assessment of the effectiveness of a Random Forest classifier for land-cover classification. **ISPRS Journal of Photogrammetry and Remote Sensing**. v.67, p.93-104, 2012.

SATO, L. Y.; SHIMABUKURO, Y. E.; KUPLICH, T. M.; GOMES, V. C. F. Análise comparativa de algoritmos de árvore de decisão do sistema WEKA para classificação de uso e cobertura da terra. XVI Simpósio Brasileiro de Sensoriamento Remoto, 2013. **Anais...** Foz do Iguaçu, PR, 2013.

SOLANO, R.; DIDAN, K.; JACOBSON, A.; HUETE, A. MODIS Vegetation Index User's Guide (MOD13 Series). 2010.

SEYLER, F.; CHAPLOT, V.; MULLER, F.; CERRI, C. E. P.; BERNOUX, M.; BALLESTER, V.; FELLER, C.; CERRI, C. C. C. Pasture mapping by classification of Landsat TM images. Analysis of the spectral behavior of the pasture class in a real medium-scale environment: the case of the Piracicaba Catchment (12 400 km², Brazil). **International Journal of Remote Sensing.** v. 23. n. 23. p. 4985-5004. 2002.

VUOLO, F.; MATTIUZZI, M.; KLISCH, A.; ATZBERGER, C. Data service platform for MODIS Vegetation Indices time series processing at BOKU Vienna: current status and future perspectives . Proc. SPIE 8538, **Earth Resources and Environmental Remote Sensing/GIS Applications III**, 85380A (October 25, 2012); doi:10.1117/12.974857.

A Method for Location Recommendation via Skyline Query Tolerant to Noisy Geo-referenced Data

Welder B. de Oliveira[1], Helton Saulo[1], Sávio S. Teles de Oliveira[2], Vagner J. Sacramento Rodrigues[2], Kleber V. Cardoso[3]

[1] Instituto de Matemática e Estatística – Universidade Federal de Goiás (UFG)
Caixa Postal 15.064 – 91.501-970 – Goiânia – GO – Brazil

[2]GoGeo
Rua Leopoldo Bulhões, esquina com a Rua 1014
Quadra 31, Lote 07, Sala 9 Setor Pedro Ludovico
CEP 74820-270 – Goiânia – GO – Brazil

[3]Instituto de Informática – Universidade Federal de Goiás (UFG)
Alameda Palmeiras, Quadra D, Campus Samambaia
131 - CEP 74001-970 – Goiânia – GO – Brazil

{welder,heltonsaulo}@gmail.com, {savio.teles,vagner}@gogeo.io,
kleber@inf.ufg.br

Abstract. *This work presents a method to perform a location recommendation based on multiple criteria even when there is noise in the coordinates. More specifically, the skyline query is adapted to handle this noise by modeling the errors of geo-referenced points with an appropriate probability distribution and modifying the traditional dominance criterion used by that technique. The method is applied to a scenario in which the coordinates are set by a geocoding process in a sample of schools in a Brazilian city. It enables one to choose the level of confidence in which a point is removed from the skyline solution, i.e., the location recommendation.*

1. Introduction

Database management systems (DBMS) have been increasingly used in recommendation services or applications. Many of these applications are based on multiple, and sometimes conflicting goals, where there may be no single optimal answer. For example, a tourist may be interested in budget hotels with reasonable ratings (e.g., 3-star) that are close to the city. Traditionally, the DBMS supports the recommendation applications by returning all answers that may meet the user's requirement. However, this may not be useful if the user is overloaded by a large amount of information.

Spatial skyline queries [Borzsony et al. 2001] have gained attention due to their efficient solution for this issue. These queries retrieve the desired objects that are no worse than any other object in the database, according to all the criteria under evaluation. In other words, given a set of points, skyline comprises the points that are not dominated by other points. In our example, if there are two hotels, $h1$ and $h2$, with the same rating,

such that $h1$ is both cheaper and nearer to the city than $h2$, then $h2$ would not be presented to the user.

The spatial skyline query is directly impacted by the accuracy of the location provided by the database. Data uncertainty inherently exists in a large number of applications [Ding et al. 2014] due to factors such as limitations of the measuring devices (e.g., GPS), and inaccuracy of the geocoding algorithms, when only the street address (e.g., the hotel) is provided. Returning to example, if the hotels $h1$ and $h2$ had been incorrectly located, thus $h2$ could dominate $h1$ due the location error only.

Due to the importance of recommendation applications and the frequent problem of imprecise or noisy data, there is a relevant demand for the creation of automated solutions that are tolerant to the data inaccuracy. How to perform analysis using inaccurate locations, especially the skyline analysis, remains an important and challenging problem. In this paper, we present a novel technique to perform skyline queries over inaccurate locations.

The remainder of this paper is organized as follows. In Section 2 we briefly give an overview of the use of approaches for recommendation services. The problem of skyline queries over inaccurate locations is formally defined in Section 3. Section 4 presents our approach of recommendation service that is tolerant to inaccuracy on the spatial data. Section 5 presents results and discussion of our approach. Finally, we provide some concluding remarks in Section 6.

2. Related work

Since the introduction of the skyline operator [Borzsony et al. 2001], skyline query processing has received considerable attention in multidimensional databases. Several algorithms for skyline computation have been proposed. For example, [Tan et al. 2001] use auxiliary structures on progressive skyline computation, [Kossmann et al. 2002] show a nearest neighbour algorithm for skyline query processing, [Papadias et al. 2003] introduce the branch and bound skyline (BBS) algorithm, [Chomicki et al. 2003] present a sort-filter-skyline (SFS) algorithm leveraging pre-sorting lists, and [Godfrey et al. 2005] propose a linear elimination sort for skyline algorithm with attractive average-case asymptotic complexity. In [Groz and Milo 2015] the true skyline is returned with a high probability with less comparisons required for computing or verifying a candidate skyline.

The concept of spatial skyline query (SSQ) was introduced in [Sharifzadeh and Shahabi 2006], in which given a set of data points P and a set of query points Q, each data point has a number of derived spatial attributes, and each attribute is the point's distance to a query point.

[You et al. 2013] propose the threshold farthest spatial skyline (TFSS) and branch and bound farthest spatial skyline (BBFS) algorithms. The TFSS algorithm uses a standard set of accesses such as sorted access from distributed sources, which uses R-tree for accessing node and retrieves data objects in decreasing order of the attribute value. The BBFS algorithm uses minimum Bounding rectangle (MBR) of an R-tree for batch pruning. Full space skyline can be supported incrementally by using naïve on-line maintenance module (NMA), as described in [Huang et al. 2010].

For a spatial skyline query using Euclidean distance, efficient algorithms have

been proposed in [Son et al. 2009, Lee et al. 2011]. [Son et al. 2014] develope an algorithm using the Manhattan distance, which closely reflects road network distance for metro areas with well-connected road networks.

Some studies have also focused on the skyline query processing with moving object. In this sense, a novel probabilistic skyline model is proposed in [Ding et al. 2014] where an uncertain object may take a probability to be in the skyline at a certain point in time. [Huang et al. 2006] had introduced the continuous skyline over precise moving data. [Zhang et al. 2009] present techniques that enable inference of the current and future uncertain locations efficiently.

The present paper brings something new to the area, namely the possibility to perform skyline query even when the data is not precise. For precision we mean that the values provided by the data are exactly what can be found in reality, i.e., the data describe the reality with fidelity (with no errors from measurements, estimation or other sources). The hypothesis of precision is assumed by all current skyline procedures present in literature. We intend to change this scenario providing a more elastic approach to this kind of query, especially concerning spatial attributes.

The two closest related works in literature were made by [Pei et al. 2007] and [Lofi et al. 2013]. The first deal with what they call "uncertain" data, which have different meaning of the "imprecision" data used our work. By uncertain the authors mean that more than one record is available for each attribute for the several objects under evaluation. They provided an example with NBA players data. To each player is collected statistics like number of assists and the number of rebounds, both the larger the better. As players have different performances in different games, the values for the attributes for each player are said to be "uncertain". [Pei et al. 2007] provide two algorithms to approach the problem. Moreover, the concept of dominance probability is introduced in this paper. On the other hand, [Lofi et al. 2013] use a method based on crowd based data. This way they can perform skyline query for incomplete data. None of them, however, deal with "imprecise" data, i.e., data which contain values with some error.

In our paper, the values of spatial attributes are considered random variables and are modelled with a probabilility density function. This enables to compute dominance probabililities even for imprecise data and then provide a more noisy tolerant technique.

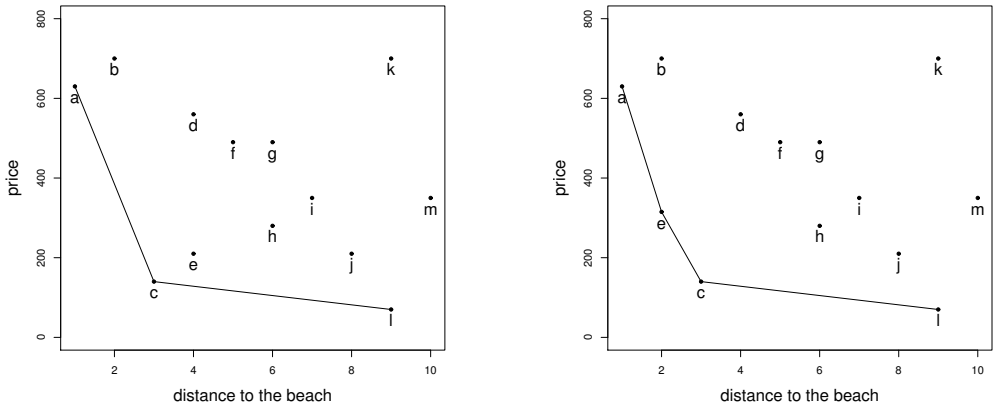
3. Formal problem definition

In order to provide a location recommendation tool able to deal with multi-criteria decision analysis, one may face the problem of noisy geo-referenced data. Furthermore, one important step that such a tool should have is discarding points that are not Pareto efficient. Pareto efficiency is an equivalent expression to skyline query and will be explained later. One problem arises in this context: "how to build a skyline query that minimizes the bad effects caused by imprecise geo-referenced data?"

Skyline query is a multi-criteria decision making technique. It aims to find a set S of all points that are not dominated by any other in the database under consideration. According to [Papadias et al. 2003], under the min condition, a point p_i dominates another point p_j if and only if the coordinate of p_i on any axis is not larger than the corresponding coordinate of p_j. In this case, the desirable points are those with the smaller values. In this

paper, only the min condition is being considered when the term dominance is mentioned. Naturally, the results are analogous for the max condition.

The Figure 1(a) shows the data for a classical example: "choose a hotel both cheap and near the beach". These are the typical conflicting criteria faced at making a decision concerning multiple goals. The nearest hotel may not be the cheapest. Note that the points a, c and l are not dominated by any other, i.e., no other is better in both dimensions then those. Therefore, $S = \{a, c, l\}$ is the skyline query for these points.



(a) Hotels near the beach: a classical skyline query example.

(b) Hotels near the beach: with a translation in point e.

Figure 1. Skyline query example

However, due to coordinates imprecision, one may not guarantee that point e is in fact dominated by the points in S. The Figure 1(b) shows how S is changed by a relatively small translation in the point e. A translation of this size in e may reflect just the imprecision in its geo-referenced position. The imprecision is measured by the error given by $error = |real - database|$, in which $real$ is the real position and $database$ is the database position. Therefore, a specific point is said to be imprecise if $error > 0$.

Two distinct types of error can occur: I) exclude a point from S when it should be in S; II) include in S a point that should not be there. In this work, we address the type I error by presenting a method to provide a skyline query solution for noisy geo-referenced data.

4. The method

In order to solve the problem proposed by this paper, three steps are required:

1. Model the error with a probability density function (PDF).
2. Generate a table of dominance probabilities via Monte Carlo method.
3. Rewrite skyline query by modifying the dominance criterion in order to take a controlled risk of a type I error.

In the following subsections, these steps will be explained in detail.

4.1. Error modeling

The error must be mathematically modeled, i.e., it is necessary to find a mathematical function that enables one to accurately predicts the behaviour of the error. For instance the probabilility of the error lies in the interval [200, 300] meters. To find this model, one must search for a PDF. A PDF is a function f that associates a value in the interval $[0, 1]$ to each set A of possible values of a random variable X.

With a PDF one can calculate the probability of a value of some random variable lies in a specific range, for example, $error$ be more than 200 and less then 300 meters. In this first step of the method, it is necessary to find a PDF that fits well the curve of the errors. A proper approach to make a guess of such a function is by the histogram or even the kernel estimation of the curve.

Since one suspects that a particular PDF fits well the curve of the errors, a formal hypothesis test may be applied in order to confirm (or not) the guess. A very well recognized hypothesis test in this sense is the Kolmogorov-Smirnov test. As states [Massey Jr 1951],

> If $F_0(x)$ is the population culmulative distribution, and $S_N(x)$ the observed cumulative step-function of a sample (i.e., $S_N(x) = k/N$, where k is the number of observations less than or equal to x), then the sampling distribution of $d = maximum|F_0(x) - S_N(x)|$ is known, and is independent of $F_0(x)$ if $F_0(x)$ is continuous.

Therefore, the distribution of *d* can be used to perform inference related to the hypothesis that $F_0(x)$ is the true populational distribution of the errors.

In our work, we employed the software `R` to perform the Kolmogorov-Smirnov test. The hypothesis may be considered **false** with at least 95% level of confidence if the computed *p-value* is at most 5%. Usually, a 95% level of confidence is considered good enough in order to discard or confirm the use of a specific probability distribution for most applications.

4.2. Table of dominance probabilities via Monte Carlo method

In this subsection, it will be evaluated the probability of a database point P dominates another, say Q in order to construct a table of dominance probabilities. This will be done by the Monte Carlo method. These two points possess the coordinates G and H in the dataset, respectively, which may be imprecise ($error > 0$). Therefore, if one computes the distances between each of those points to a third, say L, the one with the minimum distance to L in the dataset may be not the one with minimum distance in reality. Let p be the probability of P be closer than Q. In this subsection an estimative $\hat{p}$ of p is provided.

The Monte Carlo method aims to estimate a mean $M = E(X)$ by simulations with random numbers, where X is a random variable. About the history and applications of the Monte Carlo method, one can see [Metropolis 1987]. In order to estimate M, n simulated values of X should be performed, say $x_1, x_2, ..., x_n$. After generating the n values for X, its average $m = \sum_1^n x_i/n$ is computed. As n increases, the central limit theorem guarantees that m becomes arbitrarily closer to M, with the difference going to zero as n tends to infinity.

For example, to estimate the probability of getting **head** in a given coin, one can follow the mentioned steps for the variable X defined assuming the valures 0 and 1 with equal probabilility (for example 1 meaning **head**).

1. Let X be 1 if after flipping the coin the result is **head** and zero otherwise;
2. Repeat (1) n times and compute the values of X;
3. Compute the estimate $m = \sum_1^n x_i/n$;
4. Take m as an estimate for M, where M is the true probability of getting **head** by flipping the coin.

In this paper we define a random variable Y and set 1 to it if P is closer than Q in relation to L. Otherwise, the value Y is set to 0. Thus, the mean M of the Y coincides with the probability that P is the closest. As the goal is to estimate this probability, the steps (2), (3) and (4) shown above can be performed. However, it is necessary to define how the simulations will be performed.

In the case of estimating the **head** probability for a given coin, the simulation is simple: just flip the coin and compute the values. To simulate a value for Y, one can follow the paths:

1. generate a value $error_1$ and a value $error_2$ both from the chosen PDF;
2. generate an angle θ_1 and θ_2 both from an uniform distribution with parameters $(0, 2\pi)$, since it is assumed that the error is equally probable in any direction;
3. add to G a vector of length $error_1$ in the direction θ_1, resulting in the point G'. Similarly, add to H a vector of length $error_2$ in the direction of θ_2, resulting in the point H';
4. compute the distances $d_1 = |G' - L|$ and $d_2 = |H' - L|$;
5. if $d_1 < d_2$ then set 1 to Y, otherwise set 0;
6. repeat steps 1 to 5 n times with a large value of n (for instance, $n = 10,000$);
7. calculate $m = \sum_1^n y_i/n$. The value of m is a estimate for the probability that P dominates Q.

In order to provide a table of dominance probabilities as shown in Table 1, the algorithm showed above must be applied for several pairs of points P and Q placed at different distances from a reference place L. In this table, p_{ij} means the probability of P be closer to L than Q, such that in the database the distance of P to L is 100i and from Q to L is 100j. One may construct a more complete table by considering multiples of 1 meter instead of multiples of 100 meters like in this table example. Nevertheless, probabilities for intermediate or even fractional values may be estimated by interpolation.

Table 1. Structure of a dominance table

near / far	200	300	400	500
100	p_{12}	p_{13}	p_{14}	p_{15}
200	p_{22}	p_{23}	p_{24}	p_{25}
300	p_{32}	p_{33}	p_{34}	p_{35}
400	p_{42}	p_{43}	p_{44}	p_{45}
500	p_{52}	p_{52}	p_{54}	p_{55}

Based on the Table 1, the dominance criterion of skyline query can be modified. Now, it is possible to talk about probability of dominance. Instead consider the set S of

skyline query points, it is possible to construct the set W_p of points which are not dominated with level of confidence p by any other point. The new criterion is the following: "a point x dominates a point y concerning a dimension i with level of confidence p if $Prob[x_i < y_i] > p$". Therefore, there is a chance of an type I error of $(1 - p)$.

For several criteria, say d, and assuming independence between the errors of each of these dimensions, the new dominance criterion may be rewritten like: "a point x dominates y with level of confidence p if the product of the probabilities $Prob[x_i < y_i]$ is greater than p", i.e.,

$$Prob[x_1 < y_1]...Prob[x_d < y_d] > p$$

Thus, the type I error has been controlled as it was the goal of this paper. This new version of skyline query is designed to be more tolerant to geo-referenced imprecise data.

4.3. Birnbaum-Saunders distribution

In this subsection, we discuss the PDF that is used to model the geo-referenced error in the example presented in section 5, more specifically the data displayed by the Figure 2. [Birnbaum and Saunders 1969] introduced a family of Birnbaum-Saunders (BS) distributions motivated by problems of vibration in commercial aircraft that caused fatigue in materials. Although, in principle, its origin is for modeling equipment lifetimes subjected to dynamic loads, the BS distribution has been used for various other purposes, such as finance, quality control, medicine, and atmospheric contaminants. This distribution has two parameters, one of shape a and another of scale b, with b being also the median of the distribution. In addition, the BS distribution is asymmetric with positive skewness and unimodality. If a random variable T follows a BS distribution, denoted by $T \sim \text{BS}(a; b)$, then is cumulative distribution function is given by

$$F_T(t) = P(T \leq t) = \Phi\left(\frac{1}{a}\left[\left(\frac{t}{b}\right)^{1/2} - \left(\frac{b}{t}\right)^{1/2}\right]\right), \quad t > 0, \tag{1}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function. The BS model holds proportionality and reciprocal properties given by $k\,T \sim \text{BS}(a, k\,b)$, with $k > 0$, and $1/T \sim \text{BS}(a, 1/b)$, respectively. Also, when a tends to zero, the BS distribution tends to a normal model with mean b and variance τ, where $\tau \to 0$ when $a \to 0$.

5. Results and discussion

The method presented in this paper is exemplified with a data collected from a database of geocoded addresses in Goiânia-GO, Brazil. More specifically, it is a sample of 32 schools in that city. As it is common in geocoding process, the coordinates presents a significant error. To compute this error, it is necessary to have the real position of each point - the schools in this case. For these 32 schools, the right location has been collected by taking its coordinates from Google Maps application. The errors $error_i$ were calculated using the expression $error_i = |real_i - database_i|$, for $i = 1, ..., 32$. Those errors in meters are shown in y-axis of Figure 2. The cut line represents the value 350 of y-axis. Thus, one can see that most of the errors is smaller than 350 meters.

The first step is to find a PDF to model the errors. The Figure 3 shows the histogram in Figure 3(a) and a kernel density estimation with bandwith equals to 91.51 (Figure 3(b)) for the errors verified in school data. The curve suggest a highly heavy tail

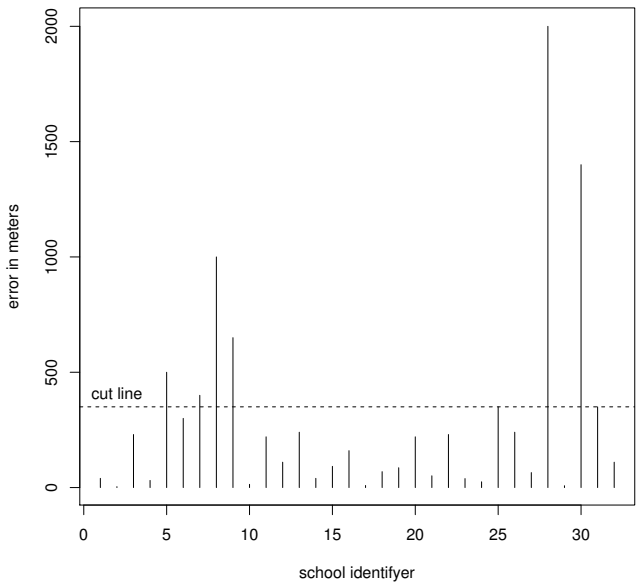


Figure 2. Location error for the 32 schools.

distribution for modeling the errors. Dispite most of the errors are relatively near to zero, there are a reasonable probabilility for some stay far beyond the mean or median of the data. Therefore, simetric options like Normal distribution are not suited for the required model. Thus, the search must rely on assymetric heavy tail PDFs. Bellow some statistics about the errors are presented.

min = 4.0

first quartile = 40

mean = 290.1

median = 135.0

third quartile = 312.5

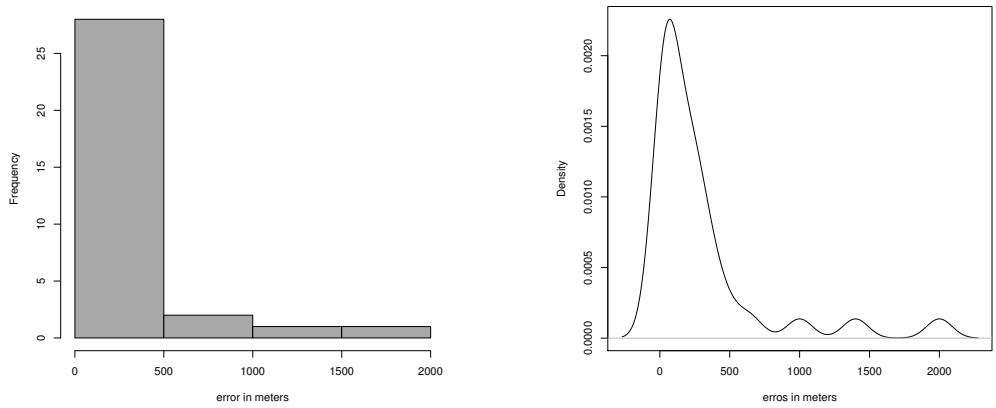
max = 2000.0

standard deviation = 433.9

pearson's second skewness coefficient = 1.07

As can been seen, the skewness is greater than 0, indicating that most values of the distribution is concentrated left to the mean and that there is a heavy tail to the right. In this context, several heavy tail and assymetric PDFs have been evaluted for modeling the error, until one in particular have been proved to achieve this purpose - the Birnbaum-Saunders (BS) distribution. BS possess the desirable requirements exposed before. In order to confirm (or not) the suspect that the error may be "well" modeled by a BS, the Kolmogorov-Smirnov test was performed. The subsection 4.3 provides more information

about this PDF.



(a) Histogram (b) Density kernel estimation for the 32 schools

Figure 3. Histogram and Kernel density estimation

The parameters a and b of the BS distribution was estimated by the Maximum Likelihood Estimation method (MLE). The parameters estimation in this model is discussed, among others, by [Lemonte et al. 2007]. Here we use a `R` implementation of MLE for this probabilility distribution. The estimated values were 1.88 for a and 100.2 for b.

To decide whether the BS(1.88, 100.2) fits well the data, an `R` implementation of the Kolmogorov-Smirnov test was used. The p-value returned by the test was 0.8745, which it is too high for refuting the hypothesis that the error does not follows a BS distribution. Therefore, the BS(1.88, 100.2) is considered a satisfactory model for the location *error* of the schools. Following the steps reported in subsection 4.2, Table 2 was constructed.

Table 2. A sample of the Dominance Table generated from the school data

near / far	200	300	400	500	600	700
100	0.62	0.71	0.76	0.81	0.84	0.87
200		0.68	0.75	0.80	0.83	0.87
300			0.69	0.77	0.82	0.86
400				0.70	0.77	0.83
500					0.70	0.78
600						0.71

Using the table, the dominance criterion was changed. For example, if P is 200 meters from a reference place L and Q is 500 meters from L, then there is a probability of 80% that P is in fact closer to L than Q. Thus, there is 1 chance in 5 to get a type I error if one considers the point P closer to L than point Q. Therefore, for a cut probability $p = 0.8$, P dominates Q. If more than one dimension was considered, then the product of the probabilities would be used like explained in subsection 4.2.

The example exposed in this section shows how the method can be used. Particularly, the first step presents some issues to be implemented efficiently, since finding a PDF for a random variable (corresponding to the error in our case) is not a trivial task. However, an alternative procedure could be employed in a future work: estimate the empirical distribution function. This approach would enable the method to be applied without the need to look for an ideal and known PDF. Nevertheless, one drawback in doing so is the lack of power that this kind of fitted has relating the a parametric approach like the ones get by finding the PDF. Also, in a future work, it is possible to implement in a programming language, a version of skyline query with the tolerance of errors proposed in this paper.

6. Conclusion

This paper presented a contribution to the area of multi-criteria decision making providing a method to perform skyline queries in the presence of noisy geocorreferenced data. Following the proposed method, it is possible to change the dominance criterion in skyline query, turning this technique more tolerant to location error. Imprecision of this type may be a common feature, for instance with coordinates obtained by a geocoding process. Therefore, the skyline query is generalized in this work from a deterministic to a probabilistic approach.

Despite the cited contribution, the proposed method only can be implemented in cases where the errors can be modeled by some known probabilility distribution. In general, this step may be hard to achieve. However, in a future work one can use the empirical distribution in order to automate this step. An advantage of this last approach would be simplicity and the declared automation. A drawback is the impossibility of choosing a parametric function like those provided by the known PDFs, which are able to provide more power in avoiding type I errors in the statistical hypothesis test (the new criterion created for dominance).

Another suggestion for a future work is to implement an end-to-end location recommendation technique, combining a first step with the noisy tolerant version of skyline query exposed in this paper with a second step related to rank the points with some optimization procedure which also handle noisy data, like, for example, that proposed by [Qin 2013]. The goal of our work has been achieved with the exposure of the method and also with its validation for a real example.

References

Birnbaum, Z. and Saunders, S. C. (1969). A new family of life distributions. *Journal of applied probability*, pages 319–327.

Borzsony, S., Kossmann, D., and Stocker, K. (2001). The skyline operator. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pages 421–430. IEEE.

Chomicki, J., Godfrey, P., Gryz, J., and Liang, D. (2003). Skyline with presorting. In *ICDE*, volume 3, pages 717–719.

Ding, X., Jin, H., Xu, H., and Song, W. (2014). Probabilistic skyline queries over uncertain moving objects. *Computing and Informatics*, 32(5):987–1012.

Godfrey, P., Shipley, R., and Gryz, J. (2005). Maximal vector computation in large data sets. In *Proceedings of the 31st international conference on Very large data bases*, pages 229–240. VLDB Endowment.

Groz, B. and Milo, T. (2015). Skyline queries with noisy comparisons. In *Proceedings of the 34th ACM Symposium on Principles of Database Systems*, pages 185–198. ACM.

Huang, Z., Lu, H., Ooi, B. C., and Tung, A. K. (2006). Continuous skyline queries for moving objects. *Knowledge and Data Engineering, IEEE Transactions on*, 18(12):1645–1658.

Huang, Z., Sun, S., and Wang, W. (2010). Efficient mining of skyline objects in subspaces over data streams. *Knowledge and information systems*, 22(2):159–183.

Kossmann, D., Ramsak, F., and Rost, S. (2002). Shooting stars in the sky: An online algorithm for skyline queries. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 275–286. VLDB Endowment.

Lee, M.-W., Son, W., Ahn, H.-K., and Hwang, S.-w. (2011). Spatial skyline queries: exact and approximation algorithms. *GeoInformatica*, 15(4):665–697.

Lemonte, A., Cribari-Neto, F., and Vasconcellos, K. (2007). Improved statistical inference for the two-parameter Birnbaum-Saunders distribution. *Computational Statistics and Data Analysis*, 51:4656–4681.

Lofi, C., El Maarry, K., and Balke, W.-T. (2013). Skyline queries in crowd-enabled databases. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 465–476. ACM.

Massey Jr, F. J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.

Metropolis, N. (1987). The beginning of the monte carlo method. *Los Alamos Science*, 15(584):125–130.

Papadias, D., Tao, Y., Fu, G., and Seeger, B. (2003). An optimal and progressive algorithm for skyline queries. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 467–478. ACM.

Pei, J., Jiang, B., Lin, X., and Yuan, Y. (2007). Probabilistic skylines on uncertain data. In *Proceedings of the 33rd international conference on Very large data bases*, pages 15–26. VLDB Endowment.

Qin, Z. (2013). Uncertain random goal programming.

Sharifzadeh, M. and Shahabi, C. (2006). The spatial skyline queries. In *Proceedings of the 32nd international conference on Very large data bases*, pages 751–762. VLDB Endowment.

Son, W., Hwang, S.-w., and Ahn, H.-K. (2014). Mssq: Manhattan spatial skyline queries. *Information Systems*, 40:67–83.

Son, W., Lee, M.-W., Ahn, H.-K., and Hwang, S.-W. (2009). Spatial skyline queries: an efficient geometric algorithm. In *Advances in Spatial and Temporal Databases*, pages 247–264. Springer.

Tan, K.-L., Eng, P.-K., Ooi, B. C., et al. (2001). Efficient progressive skyline computation. In *VLDB*, volume 1, pages 301–310.

You, G.-w., Lee, M.-W., Im, H., and Hwang, S.-w. (2013). The farthest spatial skyline queries. *Information Systems*, 38(3):286–301.

Zhang, M., Chen, S., Jensen, C. S., Ooi, B. C., and Zhang, Z. (2009). Effectively indexing uncertain moving objects for predictive queries. *Proceedings of the VLDB Endowment*, 2(1):1198–1209.

OUTCROP EXPLORER: A POINT-BASED SYSTEM FOR VISUALIZATION AND INTERPRETATION OF LIDAR DIGITAL MODELS

Gabriel Marx Bellina [a]
Francisco Manoel Wohnrath Tognoli [a,b]
Mauricio Roberto Veronez [a,b]

[a] Graduate Program in Geology (PPGEO), University of Vale do Rio dos Sinos (UNISINOS).
[b] Advanced Visualization Laboratory (VizLab), University of Vale do Rio dos Sinos (UNISINOS).
Av. UNISINOS, 950 – Zip Code 93022-000, São Leopoldo-RS-Brazil. Phone: +55 51 3591 1122
gbellina@gmail.com , ftognoli@unisinos.br, veronez@unisinos.br

ABSTRACT

The use of LIDAR-based models for natural outcrops and surfaces studies has increased in the last few years. This technique has been found to be potential to represent digitally tridimensional data, thus it increases the quality and amount of data available for interpretation by geoscientists. Researchers, in computations, face difficulties in handling the huge amount of the data acquired by LIDAR systems. It is difficult to visualize efficiently the point cloud and convert it to high-quality digital models (DMs) with specific interpretation tools. Some in-house and commercial software solutions have been developed by some research groups and industries, respectively. However, all solutions must consider the large database as the pain point of the project. Outcrop Explorer has been developed to manage large point clouds, to provide interpretation tools, and to allow integration with other applications through data exporting. In terms of software architecture, view-dependent level of detail (LOD) and a hierarchical space-partitioning structure in the form of octree are integrated in order to optimize the data access and to promote a proper visualization and navigation in the DM. This paper presents a system developed for visualization, handling and interpretation of digital models obtained from point clouds of LIDAR surveys. This system was developed considering the free graphic resources, the necessities of the geoscientists and the limitations of the commercial tools for interpretation purposes. It provides an editing tool to remove noise or unnecessary portions of the point cloud and interpretation tools to identify lines and planes, as well as their orientations, and it has different exporting formats. However, being an open source project much more collaborative development is necessary.

Key-words: Level of Detail, Octree, Terrestrial Laser Scanner, Point Clouds, OpenGL

1. INTRODUCTION

The technological evolution has brought a number of advanced solutions for applications in many different knowledge areas in the last two decades. In general, the geotechnological approach has permitted geoscientists to acquire huge amount of spacialized digital data. The more data are collected by sensors, the larger is the database size, and more processing is required to provide high-quality images or digital models for interpretation. The size of the database is directly related with the spatial resolution, *i.e.*, the number of points per unit area. This characteristic is crucial for representing outcrops and surfaces due to the multiscalar nature and any detail can be fundamental for interpretation purposes.

The Light Detection and Ranging (LIDAR) technology, especially terrestrial laser scanners (TLS), has permitted geoscientist to improve the quality of the analyses and interpretations through digital model (DM), which is a digital representation of data collected from surfaces that can be inspected, handled, and interpreted. Under a geological point of view, outcrops provide an intermediary approach between the kilometric and millimetric work scales and field data acquisition is often underexplored. As an example, the seismic data have their resolution limited to few decameters whereas lithological logs and thin sections are laterally restricted to few centimeters. Therefore, DMs have become an interesting way to integrate surface and subsurface data. This intermediary work scale allows acquiring much more data using remote sensors, *e.g.*, laser scanners with similar or superior quality obtained from the standard field equipments.

The high accuracy of the DMs allows recognition and interpretation of structures and features within a 3D-realistic scenario (Alfarhan et al. 2008; Bates et al. 2008; Olariu et al. 2008; Sima et al. 2012; Buckley et al. 2010,2013). However, the huge point clouds and the high quality of the model required to represent these type of dataset have been challenged in this knowledge area. The lack of integration between Geology, Geomatic and Computer Graphics has led researchers either to use non-specific applications to perform interpretations or to adapt interpretative routines considering the available software solutions that many times require discarding data or even applying techniques for reducing the number of points (Oryspayev et al 2012).

In-house computational systems have also been developed in the last few years by some research groups (Bellian et al. 2005; Olariu et al. 2008; Hodgetts et al. 2007; Fabuel-Perez et al. 2009), in addition to commercial applications developed by both industries and universities (*e.g.*, Polyworks®, Cyclone®, VRGS®, Coltop-3D®). They have been developed specifically to handle LIDAR data and to provide efficient visualization and navigation. However, the earlier systems are not available now, and the commercial versions are relatively expensive for academic purposes. Other options include free softwares (*e.g.*, CloudCompare, LasEdit), but none of them have specific tools for interpretation. Thus, Outcrop Explorer was developed considering the need of integrating a well-known visualization technique with efficient data management and some editing and interpretation tools. It represents a specific application for visualization, manipulation, and interpretation of georeferenced point clouds. Differently of other works already developed, Outcrop Explorer intends to be developed of a collaborative way integrating facilities and tools of different area of knowledge. For attending this premise, it was based on free graphic solutions and proposed as an open source project. Moreover, all tasks necessary among the raw data and the visualization plus interpretation can be done within the same application which is based on a new workflow.

2. COMPUTATIONAL APPROACHES ON POINT-BASED REPRESENTATIONS

The point-based concept of scene building is a recurrent topic in Computer Graphics as well as in triangulated irregular networks (TIN). Levoy and Whitted (1985) initially applied this concept for rendering surfaces, and it has been investigated more in 2000´s (Zwicker et al. 2001; Sainz and Pajarola 2004). The development of laser scanners capable to acquire million to billion points and their subsequent utilization for acquiring data from outcrops has made it necessary to develop solutions capable to handle this type of databases. The main challenge is to manage the huge amount of the data efficiently as

well as to generate high-quality DMs. The traditional visualization techniques, such as TINs, are commonly used for rendering surfaces. However, our option in using point-based rendering is based on the nature of the database and computational performance.

2.1. Data structure organization

A common approach to organize data structure is hierarchical space-partitioning data (Levoy et al. 2000; Ren et al. 2002; Botsch and Kobbelt 2003). This model can be divided in two steps. First, different levels of detail (LODs) are created, in which on the highest level the object has its full resolution. In the second step the group of LODs is linked with the structure that organizes the space (Fig 1). Using different LOD approaches, Sainz et al (2004) compared the performance and defined how much data each approach could avoid sending to the graphic hardware as well as how fast they render the necessary data.

Another concept related with data organization is octree, developed and patented in the 1980´s (Meagher, 1980,1982,1987). Afterwards, many scientific articles were published utilizing it for high-quality point-based rendering (Ren et al. 2002; Botsch and Kobbelt 2003; Pajarola 2005; Eberhardt et al. 2010). It is based on a hierarchical tree data structure in which each node can have either eight or no children. Each node can be divided more than once in eight parts, and so forth. This type of subdivision allows representing the 3D space as voxels. Partitioning, downsampling, and search operations on the point dataset are some of the resources allowed in this approach.

The standard algorithm for LOD is based on building different degrees of resolution and, at a specific moment, allowing the access only to the necessary data. The amount of data may depend on its application, like the viewer speed or distance. Luebke et al. (2003) presented a number of approaches and challenges for this technique, including a case study that used meshes as base. For the context of view-dependent LOD based on hierarchical spatial-partitioning of point clouds, the hierarchy is subjected to check the distance. Every time a node is defined as too close to the viewer, its children are brought to the same check, thus forcing more resolution to be displayed.

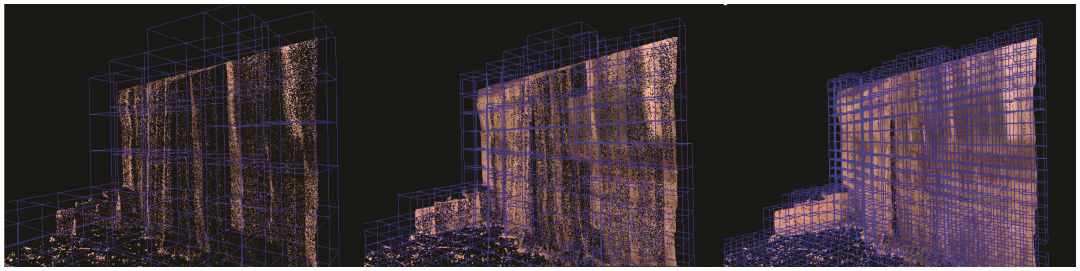


Figure 1. Example of hierarchical spatial subdivision of an adaptative point-octree.

3. DESIGN IMPLEMENTATION

The starting point for developing this project was to find a solution for handling and interpreting large amount of the data acquired with laser scanners. A view-dependent LOD together with a spatially partitioned hierarchical structure was the best choice in this case. The LOD concept along with the adopted octree structure allows sending of the only relevant data to the graphic card, and splitting of the access to the hard disk, which is a

pain point in this process. This approach served as a base for the navigation and some of the tools built subsequently.

To properly define and document the steps required to analyze and process a DM, from the point clouds to the interpreted data, few workflows have been proposed (Bellian et al. 2005; Enge et al. 2007; Buckley et al. 2008; Fabuel-Perez et al. 2010; Ferrari et al. 2012) and one more is presented herein (Fig. 2). Common steps involve data acquisition, processing, noise/obstacle editing, interpretation, and data exporting of the digital model. A common characteristic among the different workflows is the necessity of using different software products which are specific for different tasks. It increases the time of processing and interpretation, decreases the visual quality of the DMs in manipulation steps, and might face difficulties on data integration among systems.

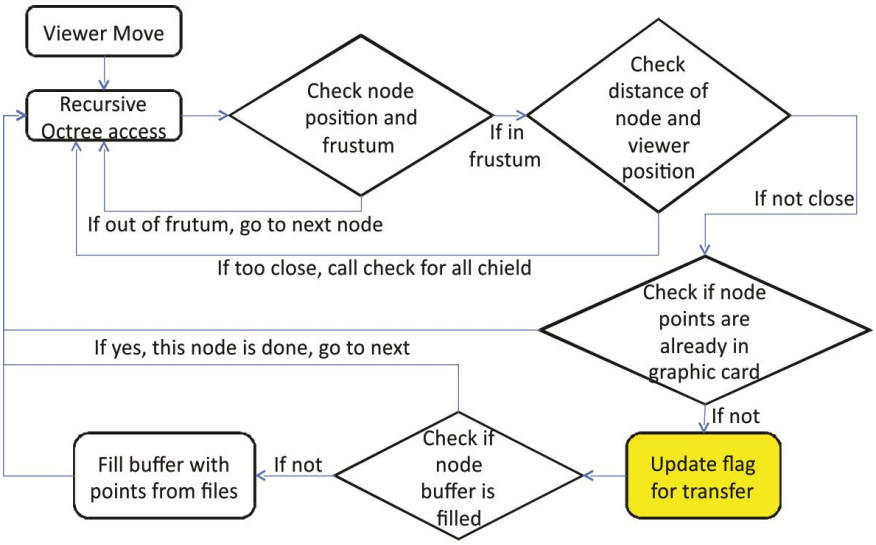


Figure 2. Logic workflow for view-dependent LOD used in Outcrop Explorer. It presents the logical sequence of tasks structured to optimize the handling and visualization of huge point clouds.

The necessity of a tool capable to integrate several steps of data acquisition and interpretation of a DM motivated this project. Outcrop Explorer is a software solution developed to process, edit, interpret, and export data and interpretations derived from LIDAR systems (Fig 3). The following requirements have been addressed: a) rendering of DM from LIDAR point clouds; b) manual noise/obstacle editing; c) manual interpretation tools such as lines and plane orientation, and d) data export. After having this structure as base, three modules were defined to split the entire process of handling the database: importing, navigation and editing tools.

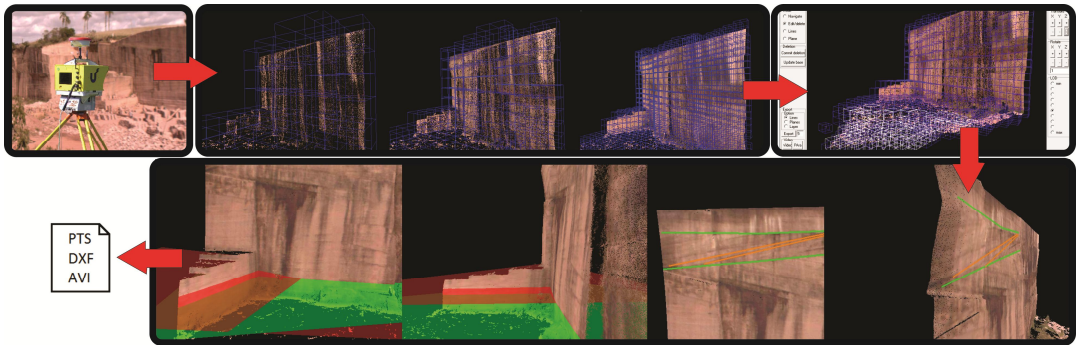


Figure 3. Field data acquisition using a terrestrial laser scanner, processing, editing, interpretation, and exportation of a digital model based on point cloud.

3.1.**Importing**

This module reads the point cloud from a file, creates the hierarchical structure, and stores it on the disk. For storing the structure, the most used solution is simple file system. This option offers great possibilities, since it allows the developers to speed up the data reading in several ways.

The hierarchical space-partitioned structure used for this project is based on a simple octree. Each tree depth level is correspondent to a level of resolution built. Two processes, one for building the structure frame and other for building the LODs, are performed. The first process defines the number of layers or the hierarchical structure depth. Two values are used, the first is the smallest spatial cube based on a constant (which can be modified by the user) and the second is calculated based on the distance between the farthest points, which have the XYZ coordinates as comparison parameters. With these data, a giant cube is calculated. This cube should contain all points; and the size of its side is defined by the following equation:

$BCS = (SCS)x\ 8^{L}$

where, BCS = biggest cube size, SCS = smallest cube size, L = number of layers.

Once the biggest cube is defined, the space inside the cube is divided into eight divisions and each one is called child of the biggest cube. Again, each cube is divided into eight subdivisions and the process repeats until the number of layers is reached. Each cube represents a node on the hierarchical structure. On a subsequent check, all the cubes that do not contain points inside their area are deleted from the structure with their children.

As a second process, the LOD is built by a bottom-up perspective. Firstly, the position of each point from the point cloud is evaluated. Based on its position, this point will be assigned to a node of the octree, which has an identifying key built with the power of 2 series (where x, y, and z use different powers). For storage, the application is built to work with the database (PostgreSQL) or system files.

3.2.**Navigation**

The displaying module operates after building and filling of the hard disk octree structure (Fig. 4). The system builds an in-memory octree, based on a structure already created on the hard disk. But the point cloud is not yet exhibited. The behavior of the system is basically divided in two steps. One manages the data to be sent to the graphic card, whereas the other manages the data to be retrieved from the database. There is an initial data load from the database, that can take either small amount of the data or the

entire database (depending on the size of the database). The user has the option of choosing which LOD will be used in the first data load, since specific tasks may not require much detail. Even though, it is recommended to use a level close to the maximum number of layers.

OpenGL was chosen as API to handle the overall communication with the graphics card interface. OpenGL offers all necessary tools to render the created outcrop model and has an extensive documentation available. A process using OpenGL updates the graphic processing unit (GPU) with the list of points currently on the main memory. The transmission of data from CPU to GPU memory is a possible bottleneck. Therefore, VBO Vertex Buffer Object (VBO) was selected to converge the efforts. The usage of buffer objects enables great amount of the data to be transferred to the graphic card memory on a single call that is always made after changing the viewer position or the LOD parameter.

At the second step, if the required nodes are not in the main memory, which means they were not taken at the first load, the system looks for this information in the database. Since this access is time consuming, a buffer environment is created. Each node accessed will have its points stored in the main memory, based on a default parameter which can be modified by the user.

Figure 4. Outcrop Explorer's visualization and navigation module. Note the high quality of the details of the rockface as evidenced by the zig-zag lines (stratifications) in the upper left corner of the point cloud, enhanced by weathering (darker areas of the wall).

3.3.**Tools**

After properly addressing the navigation, tools are offered to enable the geoscientists to interpret outcrops as presented in DMs. Most options require the ability of finding the place on OpenGL, respective to the place of mouse click. The most useful tools for this application are: to remove manually irrelevant data, to create lines, to create planes using three points as base and to export the interpretations in different formats, such as .dxf, .avi, and .pts.

3.3.1. ***Edit/delete***

During the acquisition by laser scanner, the presence of some types of obstacles, between the equipment and the target such as plants and random objects, is very common. This type of data is irrelevant but considerably increases the amount of the data to be managed in the database. To remove this unwanted information, an option was created using the octree structure. The user has the option to click at any point and the system identifies the current octree node to which the point belongs (Fig. 5). Then, the user can remove this node and its children from the current visualization or it can be deleted directly from the database. However, the second option will also require a reconstruction of the structure to remove the points from the parent nodes. This process can be done later and without affecting navigation and interpretation.

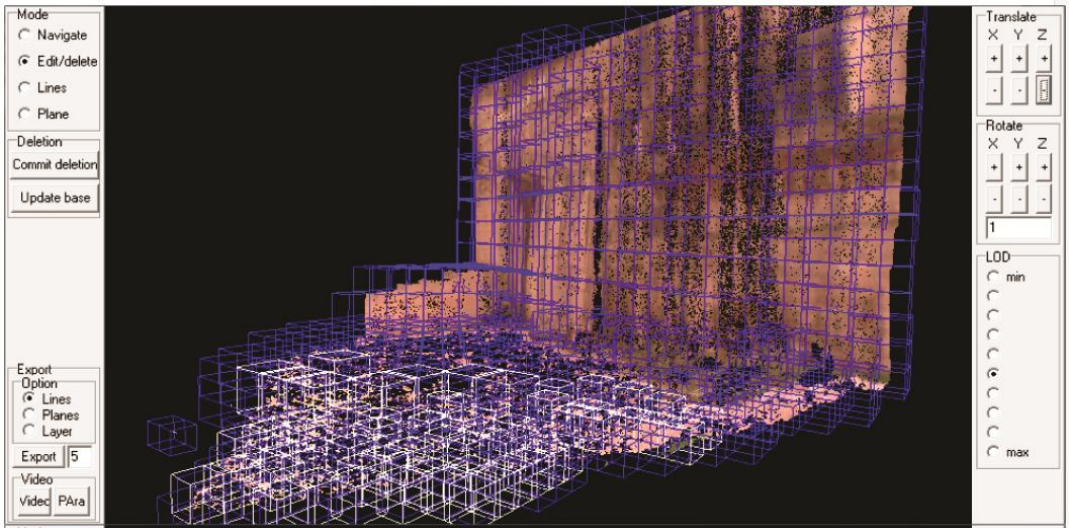


Figure 5. Digital model with the edition mode activated. Blue cubes are those that user may choose to delete. The white ones were manually selected to be deleted of the octree. Note that relevant data are in the vertical wall; the selected white cubes represent the floor of the quarry and other types of obstacles.

*3.3.2. **Lines***

With the option Lines, the interpreter can identify stratifications, beddings, fractures, faults, geometries and any kind of structures and features relevant for interpretation using mouse clicks on the screen (Fig. 6). The tool also calculates the line size according to the coordinates.

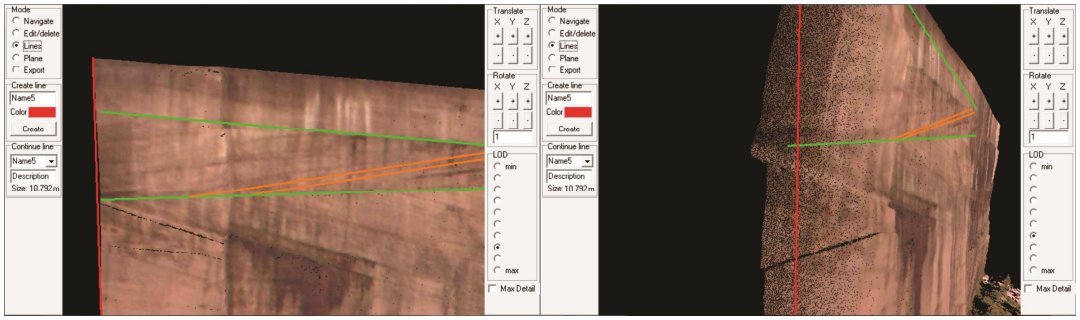


Figure 6. Interpretation lines revealing a cross-stratification directly on the digital model. Colors can be changed to enhance hierarchy or specific features. Note that lines are attached with the points as shown by the rotation of the digital model.

*3.3.3. **Planes***

Planes are almost always difficult to measure in outcrops due to the lack of good exposures in which geoscientists can take a great number of measures using a compass. In vertical outcrops, access to the upper levels is quite dangerous or even impossible. However, data from point clouds can easily provide a set of planes for the entire outcrop. Different methods can estimate the orientation of planes from point clouds such as three points, planar regression, and inertia moment (Souza et al. 2013). Outcrop Explorer is capable to create a plane using any three points on the screen (Fig. 7). The software will define and show it on the screen with an alpha factor (transparent effect). Therefore, an option to set the north direction in the tool is also created; and, consequently, the spatial orientation of the plane is automatically calculated by the system.

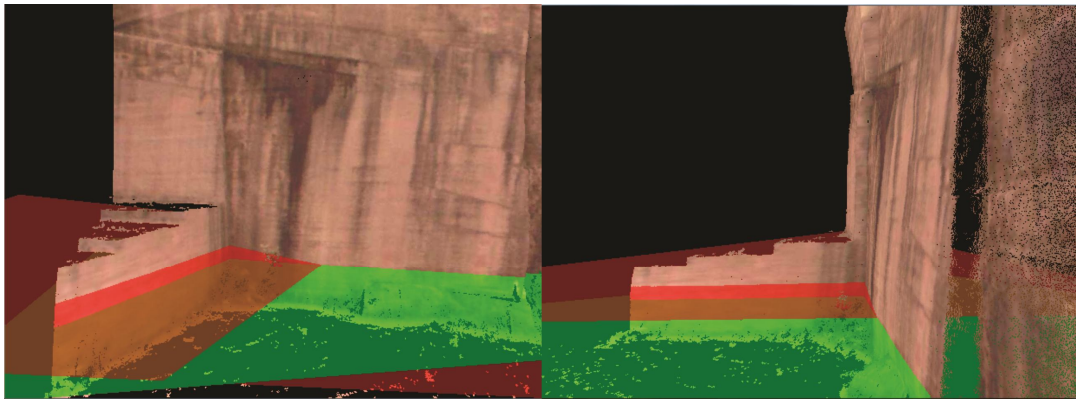


Figure 7. Planes created from three clicks directly on the digital model. Note that green and red planes represent planes with different spatial orientations, in the case, a cross-stratification.

*3.3.4. **Exporting***

The system can export final interpretation in the form of lines, planes or any specific layer of the digital outcrop model as defined by the LOD. The most of the software products used for these purposes are not capable to handle a large database, especially the CAD-based applications. Thus, it seems to be a good option to export a layer from the DOM with low amount of the data after the interpretation.

The system, for exporting, allows creating PTS files, which can be imported in CAD or GIS-based applications. Another possibility for educational purposes or even for electronic publications is to export a video from the system. Currently, only short videos with less than a minute can be exported, since the system uses AviWriter that works with BMP images resulting in a large file size.

4. DISCUSSION AND FINAL REMARKS

Performance related issues were instantly addressed during the development. The database and system files, both were tested for data access. System files solution presented

a faster data retrieval rate and was defined as the standard option, but for further developments, database can be an interesting option in case of mobile access, therefore it remains available. Optimization on file access, such as data compression or stream of data storage, can provide performance boost. In this project, the files are stored from a binary memory stream representing a buffer of points, which reduces the processing time for each file. The files are simply transferred to a memory variable. Together with the view-dependent LOD, only the necessary data is sent to the graphic card after a movement of the mouse. But even this amount can be too much in a specific condition or may not be necessary for the user at a specific moment. Therefore, an option was created in which user can easily reduce the level of detail and raise it again at any moment during the interpretation.

The navigation with low-to-moderate resolution (until layer 8 of 11) did not show any delay, but at a very high resolution, it showed some delay in moments of buffering, which could be prevented most of the times by choosing the most suitable option among followings:

(a) Have a rapid navigation with relatively low resolution or have a temporarily slow navigation for inspecting details of the DOM with high resolution (in both cases all data of the point cloud are available in the database). For example, general measurements such as height and lateral extension, delimitation of rock body geometry and interpretation of line-based structures can be done with the first option (see Fig. 6).

(b) To inspect a specific region of the point cloud or DOM with high resolution by removing temporarily lateral portions using the cube structure and working only with that specific region of the DM. This possibility can be used for detailed analyses, such as those involving inspection of high detailed features or structures and selection of specific points to define and orientate planes (see Fig. 7).

The purpose of this project was to develop a specific system for geoscientists to visualize point clouds of outcrops and surfaces acquired with laser scanners and provide interpretation tools. Considering that many solutions for this type of application are present only as in-house developments, and that interpretation tools are not available in the most of commercial software products, Outcrop Explorer intends to be the beginning of a long-term project in which users will be the main contributors for its development.

Performance in visualization will always be a big deal, since huge database is a prerequisite for digital outcrop representations based on LIDAR technique. New data management structures, high performance computing and new techniques for 3D visualization are required to improve the quality of a DOM efficiently, considering that new technologies on graphic computing arise daily. We have started a long-term project that, in this first approach, aims to provide a solution to visualize and perform basic geological interpretations in Digital Outcrop Models.

5. AVAILABILITY AND REQUIREMENTS

Outcrop Explorer is a free software developed for academic purposes. Its source code is open and researchers and users are invited to contribute with new solutions as well as to share them with the academic partners. It requires hardware configuration as follow: at least Windows XP, 2 GHz dual-core CPU, 2 GB RAM and OpenGL 1.5 GPU-enabled. All files are available at: ftp://chile.unisinos.br . If you want to download installer and related files, please contact lab-laserca@unisinos.br to obtain login and password. For technical questions, please contact gbellina@gmail.com

6. ACKNOWLEDGMENTS

The authors wish to thank both the Remote Sensing and Digital Cartography (LASERCA) and the Advanced Visualization (VizLab) Laboratories for facilities and equipments, especially associate technicians and researchers Marcelo Kehl de Souza, Leonardo Inocencio, Marcos Turani, Evandro Kirsten, Beto Reis and Rudi Cesar Comiotto Modena for their support in field and lab activities. UNISINOS' IT group is thanked for helping with file storage in the server. Kleinner Farias, Luiz Gonzaga da Silveira Jr. and Ubiratan Faccini contributed with ideas and suggestions that greatly improved a previous version of this manuscript. GMB thanks the Bureau for the Qualification of Higher Education Students (CAPES) for providing the master's PROSUP scholarship. The project was financially supported by FAPERGS (Edital 01/2010– Processo 10/0477-0) granted to FMWT; PETROBRAS− Technological Network on Sedimentology and Stratigraphy (Convênio 16 – SAP 4600242459) and FINEP (Project Moda–MCT/FINEP– Pré-Sal Cooperativos ICT – Empresas 03/2010) granted to MRV.

7. REFERENCES

Alfarhan, M., White, L., Tuck, D., Aiken, C. 2008. Laser rangefinders and ArcGIS combined with three-dimensional photorealistic modeling for mapping outcrops in the Slick Hills, Oklahoma**.** Geosphere, 4, 576-587. doi:10.1130/GES00130.1

Bates, K.T., Rarity, F., Manning, P.L., Hodgetts, D., Vila, B., Oms, O., Galobart, A., Gawthorpe, R.L. 2008. High-resolution LiDAR and photogrammetric survey of the Fumanya dinosaur tracksites (Catalonia): Implications for the conservation and interpretation of geological heritage sites. Journal of the Geological Society, 165, 115-127. doi: 10.1144/0016-76492007-033

Bellian, J.A., Kerans, C., Jennette, D.C. 2005. Digital Outcrop Models: Applications of terrestrial scanning LIDAR technology in stratigraphic modelling. Journal of Sedimentary Research, 75, 166–176. doi:10.2110/jsr.2005.013

Botsch, M., Kobbelt, L. 2003. High-quality point-based rendering on modern GPUs. In: Proceedings Pacific Graphics 2003, IEEE Computer Society Press, p. 335–343.

Buckley, S.J., Howell, J.A., Enge, H.D., Kurz, T.H. 2008. Terrestrial laser scanning in geology: data acquisition, processing and accuracy considerations. Journal of the Geological Society of London, 165, 625-638. doi:10.1144/0016-76492007-100

Buckley, S.J., Enge, H.D., Carlsson, C., Howell, J.A. 2010. Terrestrial Laser Scanning for use in Virtual Outcrop Geology. The Photogrammetric Record, 25, 225-239. doi:10.1111/j.1477-9730.2010.00585.x.

Buckley, S.J., Kurz, T., Howell, J.A., Schneider, D. 2013. Terrestrial lidar and hyperspectral data fusion products for geological outcrop analysis. Computers & Geosciences, 54, 249-258. http://dx.doi.org/10.1016/j.cageo.2013.01.018

Eberhardt, H., Klumpp, V., Hanebeck, U.D. 2010. Density Trees for Efficient Nonlinear State Estimation, Proceedings of the 13th International Conference on Information Fusion, Edinburgh, United Kingdom, July, 2010. Available at: http://isas.uka.de/Publikationen/Fusion10_EberhardtKlumpp.pdf

Enge, H.D., Buckley, S.J., Rotevatn, A., Howell, J.A. 2007. From outcrop to reservoir simulation model: workflow and procedures. Geosphere, 3, 469-490. doi:10.1130/GES00099.1

Fabuel-Perez, I.; Hodgetts, D.; Redfern, J. 2009. A new approach for outcrop characterization and geostatistical analysis of a low-sinuosity fluvial-dominated succession

using digital outcrop models: Upper Triassic Oukaimeden Sandstone Formation, central High Atlas, Morocco. AAPG Bulletin, 93, 795-827. doi:10.1306/02230908102

Fabuel-Perez, I.; Hodgetts, D.; Redfern, J. 2010. Integration of digital outcrop models (DOMs) and high resolution sedimentology - workflow and implications for geological modelling: Oukaimeden Sandstone Formation, High Atlas (Morocco). Petroleum Geoscience, 16, 133-154. http://dx.doi.org/10.1144/1354-079309-820

Ferrari, F., Veronez, M.R., Tognoli, F.M.W., Inocencio, L.C., Paim, P.S.G., Silva, R.M. 2012. Visualização e interpretação de modelos digitais de afloramentos utilizando Laser Scanner Terrestre. Geociências, 31, 79-91.

Hodgetts, D., Gawthorpe, R.L., Wilson, P., Rarity, F. 2007. Integrating digital and traditional field techniques using virtual reality geological studio (VRGS). Society of Petroleum Engineers – 69th European Association of Geoscientists and Engineers Conference and Exhibition, 83-87.

Levoy, M., Whitted, T. 1985. The use of points as display primitives. Technical Report - TR 85-022, The University of North Carolina at Chapel Hill, Department of Computer Science, 13p. Available at:
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.1180&rep=rep1&type=pdf

Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., Fulk, D. 2000. The Digital Michelangelo Project: 3-D Scanning of Large Statues. In: Proceeding SIGGRAPH'00, 14p. Available at http://graphics.stanford.edu/papers/dmich-sig00/dmich-sig00-nogamma-comp-low.pdf

Luebke, D., Reddy, M., Cohen, J., Varshney, A.,Watson, B., Huebner, R. 2003. Level of Detail for 3D Graphic. Morgan Kaufmann as part of their series in Computer Graphics and Geometric Modeling, Morgan Kaufmann, 390p.

Meagher, D. 1980. Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer. *Rensselaer Polytechnic Institute* (Technical Report IPL-TR-80-111).

Meagher, D. 1982. Geometric modeling using octree enconding. Computer Graphics and Image Processing, 19,129-147.

Meagher, D. 1987. High-speed image generation of complex solid objects using octree encoding. United States Patent Office, EP 0152741 B1. Available at:
https://docs.google.com/viewer?url=patentimages.storage.googleapis.com/pdfs/US4694404.pdf

Olariu, M. I., Ferguson J.F., Aiken C.L.V. 2008. Outcrop Fracture Characterization Using Terrestrial Laser Scanners: Deepwater Jackfork Sandstone at Big Rock Quarry, Arkansas. Geosphere, 4, 247-259. doi: 10.1130/GES00139.1

Oryspayev, D., Sugumaran, R., De Groote, J., Gray, P. 2012. LIDAR data reduction using vertex decimation and processing with GPU and multicore CPU technology. Computers & Geosciences, 43, 118-125. http://dx.doi.org/10.1016/j.cageo.2011.09.013

Pajarola, R. 2005. Stream-processing points. In: Proceedings IEEE Visualization, 10p. Available at http://www.ifi.uzh.ch/vmml/publications/older-puclications/PStream.pdf

Pajarola, R., Sainz, M., Lario, R. 2005. XSplat: External memory multiresolution point visualization. Proceedings of the International Conference on Visualization, Imaging and Image Processing, 628-633.

Ren, L., Pfister, H., Zwicker, M. 2002. Object space EWA surface splatting: a hardware accelerated approach to high quality point rendering. Computer Graphics Forum, 21, 461-470.

Sainz, M., Pajarola, R. 2004. Point-based rendering techniques. Computers & Graphics, 28, 869-879. doi:10.1016/j.cag.2004.08.014

Sainz, M., Pajarola, R., Lario, R. 2004. Points reloaded: point-based rendering revisited. In: Alexa, M., Rusinkiewicz, S. (ed.). Proceedings of the Eurographics Symposium on Point-Based Graphics, 121–128.

Sima, A.A., Buckley, S.J., Viola, I. 2012. An interactive tool for analysis and optimization of texture parameters in photorealistic virtual 3D models. International Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, I(2), 165-170. doi: 10.5194/isprsannals-I-2-165-2012

Souza, M.K., Veronez, M.R., Tognoli, F.M.W., Silveira Jr., L.G., Inocencio, L.C., Silva, R.M., Modena, R.C.C. 2013. Terrestrial Laser Scanning: Application for measuring of structures information in geological outcrops. International Journal of Advanced Remote Sensing and GIS, 2, 260-270.

Zwicker, M., Pfister, H., Van Baar, J., Gross, M. 2001. Surface splatting. In: SIGGRAPH'01 - Proceedings of the 28th annual conference on computer graphics and interactive techniques, ACM Press, New York, 371–378.

An Abstract Data Type to Handle Vague Spatial Objects Based on the Fuzzy Model

Anderson Chaves Carniel[1], **Ricardo Rodrigues Ciferri**[2], **Cristina Dutra de Aguiar Ciferri**[1]

[1]Department of Computer Science – University of São Paulo in São Carlos
13.560-970 – São Carlos – SP – Brazil

accarniel@gmail.com, cdac@icmc.usp.br

[2]Department of Computer Science – Federal University of São Carlos
15.565-905 – São Carlos – SP – Brazil

ricardo@dc.ufscar.br

Abstract. *Crisp spatial data are geometric features with exact location on the extent and well-known boundaries. On the other hand, vague spatial data are characterized by inaccurate locations or uncertain boundaries. Despite the importance of vague spatial data in spatial applications, few related work indeed implement vague spatial data and they do not define abstract data types to enable the management of vague spatial data by using database management systems. In this sense, we propose the abstract data type FuzzyGeometry to handle vague spatial data based on the fuzzy model. FuzzyGeometry was developed as a PostgreSQL extension and its implementation is open source. It offers management for fuzzy points and fuzzy lines. As a result, spatial applications are able to access the PostgreSQL to handle vague spatial objects.*

1. Introduction

Spatial applications are commonly used for spatial analysis to aid in the decision making process. They mainly analyze spatial objects that can be represented by geometries [Güting 1994, Schneider and Behr 2006]. These geometries represent spatial objects of the real world, which can be simple, such as point, line, and polygon (i.e., region), or complex, such as multipoint, multiline, and multipolygon. Furthermore, these spatial objects have exact location in the extent, i.e., their geographical coordinates clearly define their geographical positions. In addition, a region has well-defined boundaries, expressing with exactness its limit. These objects are called *crisp spatial data*.

On the other hand, real-world phenomena frequently have inexact location, uncertain boundaries, or imprecise interior [Siqueira et al. 2014]. These objects are defined as *vague spatial data*. There are several models to represent vague spatial objects, such as *exact models* [Cohn and Gotts 1995, Pauly and Schneider 2008, Bejaoui et al. 2009, Pauly and Schneider 2010], *rough models* [Beaubouef et al. 2004], *probabilistic models* [Cheng et al. 2003, Li et al. 2007, Zinn et al. 2007], and *fuzzy models* [Dilo et al. 2007, Schneider 2008, Schneider 2014, Carniel et al. 2014]. These models discuss standards to represent vague spatial data as well as their operations. However, there are no native support of vague spatial data in Spatial Database Management Systems (SDBMS), such as PostgreSQL with the PostGIS extension. This is a problem since

spatial applications increasingly require the management of vague spatial objects in situations commonly found in real world, such as the representation of soil mapping, fire and risk zones, oceans, lakes, and air polluted areas.

To fill this gap, we propose an *abstract data type* (ADT) called FuzzyGeometry. FuzzyGeometry ADT is based on the fuzzy model and it is an open-source PostgreSQL extension. The fuzzy model represents vague spatial objects by making use of the fuzzy set theory [Zadeh 1965]. We use this model since we are able to represent different vagueness levels of an object. It is possible because each point has a value in the interval $[0, 1]$, called *membership degree*, which indicates the possibility of a point to belong to a spatial object. Vague spatial data modeled by using fuzzy models are designed as *fuzzy spatial data types*, such as fuzzy point, fuzzy line, and fuzzy region. In this paper, we focus on the design and implementation of fuzzy points and fuzzy lines only, which already have several challenges for their definition and implementations. In addition, we propose operations involving them, such as fuzzy geometric set operations.

This paper is organized as follows. Section 2 summarizes related work. Section 3 describes the technical background. Section 4 presents our FuzzyGeomety ADT. Section 5 concludes the paper and presents future work.

2. Related Work

Few works in the literature implement vague spatial data types based on the fuzzy model by using a database management system or a Geographic Information System (GIS). On the other hand, exact models are also frequently adopted to represent vague spatial objects since they use well-known crisp spatial algorithms. Hence, we compare implementations based on the exact model and the fuzzy model.

Despite there are several exact models [Cohn and Gotts 1995, Pauly and Schneider 2008, Bejaoui et al. 2009, Pauly and Schneider 2010], only few implementations are based on them. Vague Spatial Algebra (VASA) [Pauly and Schneider 2008, Pauly and Schneider 2010] is an exact model that offers several spatial operations for vague points, vague lines, and vague regions. An implementation is given in [Pauly and Schneider 2008][1], which implements the VASA by adapting SQL functions to handle vague spatial objects. However, in this paper, we propose an abstract data type for vague spatial data based on the fuzzy model. While VASA has only three levels of representation for vague spatial objects (the crisp part, the vague part, and the part that certainly does not belong to the spatial object), vague spatial objects based on the fuzzy model may have several levels in the real interval $[0, 1]$. Therefore, it allows a more detailed representation of vague spatial data.

For vague spatial data based on the fuzzy model, the Spatial Plateau Algebra (SPA) [Schneider 2014] provides definitions for fuzzy spatial data that reuses crisp spatial algorithms. Hence, a fuzzy spatial object, called *spatial plateau object*, is defined as a finite sequence of pairs where each pair is formed by one crisp spatial object and a membership degree in $]0, 1]$. In addition, SPA defines spatial plateau operations to handle spatial plateau objects, such as geometric set operations. Though this implementation concept was proposed, spatial plateau data were not implemented in a SDBMS.

[1] http://www.cise.ufl.edu/research/SpaceTimeUncertainty/

Vague spatial data based on the fuzzy model are implemented in [Kraipeerapun 2004, Dilo et al. 2006]. They implement the following vague spatial data types: vague point, vague line, and vague region. A vague point is defined as a tuple (x, y, λ), where $(x, y) \in \mathbb{R}^2$ gives the location, and $\lambda \in]0, 1]$ gives the membership degree. A vague line is defined as a finite sequence of tuples $((x_1, y_1, \lambda_1), \ldots, (x_n, y_n, \lambda_n))$ for some $n \in \mathbb{N}$, where each tuple is a vague point in the vague line constructed by using linear interpolation. A vague region is composed by several vague lines and the Delaunay triangulation, which is stored together with the vague region object. A membership degree of any point inside a vague region is calculated by using linear interpolation of membership degrees of vertices of the triangle to which the point belongs. This implementation is performed in the GRASS GIS, and not in a SDBMS. They do not implement the vague geometric difference between vague lines. However, our paper proposes an abstract data type implemented a SDMS (i.e., the PostgreSQL) to manage vague spatial objects.

3. Technical Background

This section summarizes the main needed concepts to understand our proposal of an ADT to handle vague spatial data. Section 3.1 summarizes vague spatial data concepts while Section 3.2 summarizes fuzzy set theory.

3.1. Vague Spatial Data

While crisp spatial objects have exact location and well-known boundaries, vague spatial objects have inexact location, uncertain boundaries, or imprecise interior. There are distinct models to represent vague spatial data that can be classified as *exact models* [Cohn and Gotts 1995, Pauly and Schneider 2008, Bejaoui et al. 2009, Pauly and Schneider 2010], *rough models* [Beaubouef et al. 2004], *probabilistic models* [Cheng et al. 2003, Li et al. 2007, Zinn et al. 2007], and *fuzzy models* [Dilo et al. 2007, Schneider 2008, Schneider 2014, Carniel et al. 2014].

Exact models aim to reutilize existing abstract data types of crisp spatial data types (e.g. crisp points, crisp lines, and crisp regions) to represent vague spatial objects. In general, vague spatial objects are defined by using two crisp spatial objects. One object represents the vague spatial part while other object represents the well-known spatial part. The main relevant models are: *Egg-Yolk* [Cohn and Gotts 1995], *Qualitative Min-Max Model* (QMM) [Bejaoui et al. 2009], and *Vague Spatial Algebra* (VASA) [Pauly and Schneider 2008, Pauly and Schneider 2010]. Egg-Yolk model defines only vague regions, which are represented by two sub-regions: a sub-region denominates the yolk (i.e., vague spatial part) and other sub-region denominates the egg (i.e., well-known spatial part) that is contained in the yolk part. QMM model defines vague spatial objects by using two limits, a minimum limit (i.e., well-known spatial part) and a maximum limit (i.e., includes the minimum limit and extends to the part that possibly belongs to the spatial object). In addition, this model uses qualitative classifications of vagueness levels, such as *completely crisp*, *partially vague*, and *completely vague*. Finally, VASA defines a vague spatial object as a pair of disjoint or adjacent crisp spatial objects of the same type.

Rough models are based on the rough set theory [Pawlak 1982] that defines a lower and an upper approximation. Hence, a vague spatial object is represented by these

approximations. Lower spatial approximation of an object is a subset of its upper spatial approximation. Vague spatial data represented by rough models deal with vague spatial objects with inexact location as well as inexact measures [Beaubouef et al. 2004].

Probabilistic models are based on the probability density functions [Cheng et al. 2003, Li et al. 2007, Zinn et al. 2007] and the treatment of spatial vagueness is performed through objects positions and measures. In general, these models handle with expectative of a future event based on the known-characteristics. While the probability density functions are exacts, the location of an object is uncertain.

Fuzzy models, that is, models based on the fuzzy set theory [Zadeh 1965], assign membership degrees in $[0, 1]$ for each point of the location to represent spatial vagueness in different levels. There are several representations of vague spatial data by using the fuzzy set theory. Fuzzy Minimum Boundary Rectangle [Somodevilla and Petry 2004] includes the fuzzy set theory in order to define membership functions by using several Minimum Boundary Rectangles. Fuzzy spatial data types denominate fuzzy points, fuzzy lines, and fuzzy regions as well as fuzzy geometric set operations, such as, fuzzy geometric union, fuzzy geometric intersection, and fuzzy geometric difference has been defined [Dilo et al. 2007, Schneider 2008, Schneider 2014, Carniel et al. 2014]. In addition, vague partitions and their operations are defined [Dilo et al. 2007]. In this paper, we considered the fuzzy spatial data defined in [Dilo et al. 2007] as design goals to implement the FuzzyGeometry. Section 3.2 summarizes needed fuzzy set theory concepts.

3.2. Fuzzy Set Theory

Fuzzy set theory [Zadeh 1965] is an extension and generalization of the classic (crisp) set theory. In the classic theory, let X be a crisp set of objects, called the *universe*. The subset A of X can be described by a function $\chi_A : X \rightarrow \{0, 1\}$, which for all $x \in X$, $\chi_A(x)$ is 1 if and only if, $x \in A$ and 0 otherwise. On the other hand, fuzzy set theory defines a function $\mu_{\tilde{A}}$ that maps all elements of X in the real interval $[0, 1]$ by assigning *membership degrees* in a specific set. Hence, fuzzy set theory allows that an element x has different membership values in different fuzzy sets. Let X be the universe. Then, the function $\mu_{\tilde{A}} : X \rightarrow [0, 1]$ is called of *membership function of the fuzzy set* $\tilde{A}$. Therefore, each element of fuzzy set $\tilde{A}$ has a membership degree in the real interval $[0, 1]$ according to the membership function: $\tilde{A} = \{(x, \mu_{\tilde{A}}) \,|\, x \in X\}$.

Classic operations among crisp sets also are extended for the fuzzy sets. We will summarize these operations. Let $\tilde{A}$ and $\tilde{B}$ be fuzzy sets in X, then the intersection, union, and difference are defined as follows, respectively:

- $\tilde{A} \cap \tilde{B} = \{(x, \mu_{\tilde{A}\cap\tilde{B}}(x)) \,|\, x \in X \wedge \mu_{\tilde{A}\cap\tilde{B}}(x) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))\}$
- $\tilde{A} \cup \tilde{B} = \{(x, \mu_{\tilde{A}\cup\tilde{B}}(x)) \,|\, x \in X \wedge \mu_{\tilde{A}\cup\tilde{B}}(x) = \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))\}$
- $\tilde{A} - \tilde{B} = \{(x, \mu_{\tilde{A}-\tilde{B}}(x)) \,|\, \mu_{\tilde{A}-\tilde{B}}(x) = \min(\mu_{\tilde{A}}(x), 1 - \mu_{\tilde{B}}(x)\}$

An alpha-cut (α-cut) and a strict alpha-cut (strict α-cut) of the fuzzy set $\tilde{A}$ for a specific value α is a crisp set defined as follows, respectively:

- $\tilde{A}^{\geq\alpha} = \{x \in X \,|\, \mu_{\tilde{A}}(x) \geq \alpha \wedge 0 \leq \alpha \leq 1\}$
- $\tilde{A}^{>\alpha} = \{x \in X \,|\, \mu_{\tilde{A}}(x) > \alpha \wedge 0 \leq \alpha < 1\}$

When α value is 1 for the α-cut of $\tilde{A}$, the result is called of core of $\tilde{A}$.

Generalizations of fuzzy sets operations, such as intersection and union, replace the min and max operators by *triangular norms* (t-norm) and *triangular co-norms* (s-norm), respectively. A t-norm T is defined as a commutative, associative, non-decreasing binary operation on $[1, 0]$, with signature $T : [0, 1]^2 \rightarrow [0, 1]$ satisfying the following boundary conditions, $T(1, x) = x$ and $T(0, x) = 0$ for all $x \in [0, 1]$ [Klement et al. 2000]. Let $x, y \in [0, 1]$, some examples of t-norms are listed as follows:

- $T^*(x, y) = \begin{cases} x & \text{if } y = 1 \\ y & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases}$ (*drastic intersection*)
- $T_p(x, y) = ab$ (*product t-norm*)
- $T_l(x, y) = \max(0, x + y - 1)$ (*Lukasiewicz t-norm*)

For any t-norm there is an s-norm, which is obtained by De Morgan's laws. Hence, a s-norm is defined as a commutative, associative, non-decreasing binary operation on $[1, 0]$, with signature $S : [0, 1]^2 \rightarrow [0, 1]$ satisfying the following boundary conditions, $S(1, x) = 1$ and $S(0, x) = x$ for all $x \in [0, 1]$ [Klement et al. 2000]. Let $x, y \in [0, 1]$, some examples of s-norms are listed as follows:

- $S^*(x, y) = \begin{cases} x & \text{if } y = 0 \\ y & \text{if } x = 0 \\ 1 & \text{otherwise} \end{cases}$ (*drastic union*)
- $S_p(x, y) = x + y - xy$ (*probabilistic sum*)
- $S_l(x, y) = \min(1, x + y)$ (*bounded sum*)

The height of a fuzzy set $\tilde{A}$ is defined as the greatest membership value (sup) of the membership function of $\tilde{A}$ [Jamshidi et al. 1993], that is, $h(\tilde{A}) = sup_x[\mu_{\tilde{A}}(x)]$. A fuzzy set $\tilde{A}$ is called normal when $h(\tilde{A}) = 1$, and subnormal when $h(\tilde{A}) < 1$. To normalize a fuzzy set $\tilde{A}$, we apply the normalization, which is defined as $Norm_{\mu_{\tilde{A}}}(x) = [\mu_{\tilde{A}}(x)/h(\tilde{A})]$ for all $x \in X$.

The concentration (CON) of a fuzzy set $\tilde{A}$ decreases the fuzziness, while the dilation (DIL) of a fuzzy set $\tilde{A}$ increases the fuzziness [Jamshidi et al. 1993]. They are defined as follows:

- $\mu_{CON(\tilde{A})}(x) = [\mu_{\tilde{A}}(x)]^p$ for all $x \in X$ where $p > 1$
- $\mu_{DIL(\tilde{A})}(x) = [\mu_{\tilde{A}}(x)]^r$ for all $x \in X$ where $r \in]0, 1[$

Finally, there are some notations to textually represent a fuzzy set [Jamshidi et al. 1993]. The following definitions are textual representation of a fuzzy set $\tilde{A}$:

- $\tilde{A} = \sum_{x_i \in X} \mu_{\tilde{A}}(x_i)/x_i$ when X is finite and discrete
- $\tilde{A} = \int_x \mu_{\tilde{A}}(x)/x$ when X is continuous

Note that the signs of sum and integral denote the union of the membership degrees and the slash (/) denotes a separator.

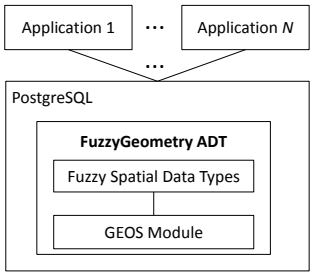


Figure 1. Architecture of the FuzzyGeometry ADT.

4. The FuzzyGeometry Abstract Data Type

We propose a novel ADT to handle vague spatial data based on the fuzzy model (Section 3.1), called FuzzyGeometry. We implemented the FuzzyGeometry ADT as a PostgreSQL extension. PostgreSQL has free license and it is an extensible database management system, which new ADTs can be implemented by using a low level program language (e.g., C language) or a high level program language (e.g. pl/pgSQL). FuzzyGeometry was implemented in the C language by using the extensibility provided by the PostgreSQL internal library.

Figure 1 shows the architecture of the FuzzyGeometry ADT in the PostgreSQL. Our ADT uses the GEOS module for crisp geometric operations that were adapted for vague spatial data based on the fuzzy model. As discussed in the Section 2, related work try to use crisp geometric set operations to handle vague spatial data. Therefore, we use the GEOS module for this purpose. GEOS module is a library in C/C++ with free source code used to handle crisp spatial data in GIS, such as the GRASS, and SDBMS, such as the PostGIS extension of PostgreSQL. As a result, external applications can use the FuzzyGeometry ADT by accessing directly the PostgreSQL.

In the next sections we will detail the FuzzyGeometry ADT. Section 4.1 presents the FuzzyGeometry data types and their textual representations. Section 4.2 details operations for each data type of the FuzzyGeometry.

4.1. Fuzzy Spatial Data Types of the FuzzyGeometry

The FuzzyGeometry ADT offers the following fuzzy spatial data types (Figure 1): fuzzy points and fuzzy lines. They can be simple or complex, and the hierarchy among them is showed in Figure 2. The highest level of hierarchy is the *FuzzyGeometry data type*. A FuzzyGeometry can be a *Simple FuzzyGeometry* or a *Complex FuzzyGeometry*. The Simple FuzzyGeometry data type cannot be instanced since it is a generalization for simple fuzzy spatial objects, which can be instanced as a fuzzy point and a fuzzy linestring (i.e., a fuzzy line). Similarly, the Complex FuzzyGeometry data type cannot be instanced since it is a generalization for complex fuzzy spatial objects that are collections of simple fuzzy spatial objects of the same type, which can be a fuzzy multipoint (i.e., a complex fuzzy point) and a fuzzy multilinestring (i.e., a complex fuzzy line). Hence, Figure 2 shows in gray, the possible data types that can be instanced. Note that the current version of the FuzzyGeometry ADT does not support for fuzzy regions.

These fuzzy spatial objects have a membership degree for each point in the space

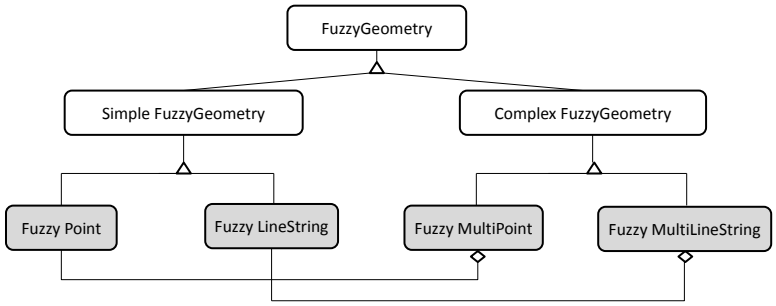
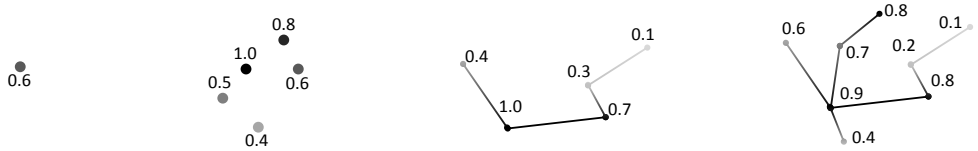


Figure 2. The hierarchy of the FuzzyGeometry data types.



(a) (b) (c) (d)

Figure 3. Examples of fuzzy spatial objects of the FuzzyGeometry: (a) a fuzzy point object, (b) a fuzzy multipoint object, (c) a fuzzy linestring object, and (d) a fuzzy multilinestring object. Darker parts have higher membership values than lighter areas.

to denote inexact location or imprecision. Such membership degree assigns the value in the real interval $[0, 1]$ for each point, which determines the spatial vagueness in a point. We will detail each possible instances of a FuzzyGeometry object according to Figure 2.

A fuzzy point is defined as (x, y, u), which (x, y) corresponds to a coordinate pair that provides the location and u is the membership degree in the interval real $]0, 1]$ for its coordinate pair. A set of disjoint fuzzy points form a fuzzy multipoint. Hence, a fuzzy multipoint is defined as a sequence of triples in the format (x, y, u). An unique fuzzy point is a special case for a fuzzy multipoint. Figure 3a shows an example of a fuzzy point object and Figure 3b shows an example of a fuzzy multipoint object.

A fuzzy linestring is defined as a sequence $((x_1, y_1, u_1), \ldots, (x_n, y_n, u_n))$ for some $n \in \mathbb{N}$, i.e., a sequence of fuzzy points. These fuzzy points are sequentially linked. For instance, (x_1, y_1, u_1) and (x_2, y_2, u_2) are two linked points and thus, they form a segment line. A membership degree of a given point in the fuzzy linestring is calculated by using linear interpolation on a segment. A fuzzy linestring object should not has self-intersection. A fuzzy multilinestring is defined as a set of fuzzy lines that can only intersect in their endpoints. An unique fuzzy linestring is a special case for a fuzzy multilinestring. Figure 3c shows an example of a fuzzy linestring object and Figure 3d shows an example of a fuzzy multilinestring object composed by 4 fuzzy linestring objects.

In the PostgreSQL, only the FuzzyGeometry data type is specified, which can be internally instanced as a fuzzy point, fuzzy linestring, fuzzy multipoint, and fuzzy multilinestring. To handle these data types by using the SQL language, we define input and output functions. An input operation transforms textual representation into an internal

representation; An output operation transforms the internal representation into the textual representation. Hence, in order to insert fuzzy spatial objects in relational tables in the PostgreSQL, we propose textual representations for simple and complex fuzzy points and lines. Therefore, by using these representations, users are able to insert fuzzy spatial objects into tables and to visualize them as results of spatial queries.

We define textual representations of fuzzy spatial data types based on the textual representations of fuzzy sets (Section 3.2). In general, firstly appears the name of fuzzy spatial data type and in parentheses, for each point, its membership degree and its coordinate pairs separated by slash. Empty fuzzy spatial objects, which contain no coordinates and membership values, can be specified by using the symbol EMPTY after the data type name. Let (x, y) be a coordinate pair, u be a membership degree in real interval $]0, 1]$, and $k, j \in \mathbb{N}$. Then, we define the textual representation for fuzzy point (i), fuzzy multipoint (ii), fuzzy linestring (iii), and fuzzy multilinestring (iv) as follows:

(i) FUZZYPOINT$(u/x\ y)$
(ii) FUZZYMULTIPOINT$(u_1/x_1\ y_1, ..., u_k/x_k\ y_k)$
(iii) FUZZYLINESTRING$(u_1/x_1\ y_1, ..., u_k/x_k\ y_k)$
(iv) FUZZYMULTILINESTRING$((u_{1_1}/x_{1_1}\ y_{1_1}, ..., u_{k_1}/x_{k_1}\ y_{k_1})_1,$
$..., (u_{1_j}/x_{1_j}\ y_{1_j}, ..., u_{k_j}/x_{k_j}\ y_{k_j})_j)$

4.2. FuzzyGeometry Operations

In this paper, we consider the grouping provided in [Güting 1994] for abstract models of spatial data to classify and define the FuzzyGeometry operations. The groups are:

(i) Operations that return spatial objects. For instance, geometric set operations.
(ii) Operations that return topological relationship between spatial objects. For instance, the topological relationships between two lines.
(iii) Operations that return numbers. For instance, metric operators.
(iv) Operations on set of objects. For instance, spatial aggregate functions, such as the geometric union on a set of objects.

FuzzyGeometry implements operations of groups (i) and (iii). In the following sections, we will show these operations according to their classifications.

4.3. Geometric Set Operations

Geometric set operations of the FuzzyGeometry ADT are: union, intersection, and difference. In general, we used the formal definition provided by fuzzy models that define vague spatial data (Section 3.1) to implement them. These geometric set operations belong to group (i), i.e., operations that return spatial data. Let $\tilde{A}, \tilde{B}, \tilde{C}$ be FuzzyGeometry objects, $s \in \{$*max s-norm, drastic s-norm, probabilistic s-norm, bounded s-norm*$\}$, $t \in \{$*min t-norm, drastic t-norm, product t-norm, Lukasiewicz t-norm*$\}$, and $d \in \{$*fuzzy difference, arithmetic difference*$\}$. Then, we define the following signatures (the prefix *FG_* is used in all operations of the FuzzyGeometry ADT):

(i) *FG_Union*($\tilde{A}$, $\tilde{A}$, s) $\rightarrow \tilde{A}$
(ii) *FG_Union*(set of $\tilde{A}$) $\rightarrow \tilde{A}$
(iii) *FG_Intersection*($\tilde{A}$, $\tilde{B}$, t) $\rightarrow \tilde{C}$
(iv) *FG_Difference*($\tilde{A}$, $\tilde{A}$, d) $\rightarrow \tilde{A}$

The union operation (i) between FuzzyGeometry objects is performed by the spatial union and the fuzzy union of the intersecting points. The fuzzy union can be executed by using a specific s-norm s. In addition, the union is only executed for FuzzyGeometry objects of same type. For instance, the union of two fuzzy linestrings yields other fuzzy linestring, which its location is the spatial union and its membership degrees for each point are calculated using the s-norm s. Other s-norms can be implemented as well. In addition, the union operation can be used as an aggregation operator, i.e., the operation (ii). This means that, given a set of FuzzyGeometry objects, this operation yields the union among all the objects contained in this set. The strategy to compute this aggregation is to perform the union operation (i) incrementally for each FuzzyGeometry object contained in the set by considering a default s-norm (i.e., the max s-norm).

The intersection operation (iii) between FuzzyGeometry objects is performed by the spatial intersection and the fuzzy intersection of the intersecting points. The fuzzy intersection can be executed by using a specific t-norm t. The intersection can be executed between FuzzyGeometry objects of different types, and the resulting FuzzyGeometry object is the lower data type by considering the hierarchy: *fuzzy linestring* > *fuzzy point*. For instance, the intersection between fuzzy linestrings and fuzzy points yields a fuzzy point or a fuzzy multipoint object composed by the commons points with membership degree calculated by using a t-norm t. Other t-norms can be implemented as well.

The difference operation (iv) between FuzzyGeometry objects is performed by the spatial difference and the membership degrees of the intersecting points are calculated by using a difference operator. The membership degrees are calculated by using fuzzy difference or arithmetic difference. The arithmetic difference is defined by $\mathit{diff}\,(a, b) = a - b$ if $a > b$; 0 otherwise. For instance, the difference between fuzzy linestrings yields a fuzzy linestring, where commons locations will have the membership degree calculated by the fuzzy difference or arithmetic difference, and the spatial difference is performed for the remaining locations.

4.4. Generic Operations

Generic operations of the FuzzyGeometry ADT are: core, boundary, set linguistic term, and crisp transform. These operations can be applied in any type of FuzzyGeometry object and belong to group (i) (operations that return spatial data). Let $\tilde{A}$ be a FuzzyGeometry object, B be a crisp spatial object, and lt be a linguistic term. Then, we define the following signatures:

(i) *FG_Core*($\tilde{A}$) $\rightarrow \tilde{A}$
(ii) *FG_Boundary*($\tilde{A}$) $\rightarrow \tilde{A}$
(iii) *FG_Set_LinguisticTerm*($\tilde{A}$, lt) $\rightarrow \tilde{A}$
(iv) *FG_CrispTransformation*($\tilde{A}$) $\rightarrow B$

The core (i) and boundary (ii) operations get the locations that have exact locations (locations with membership degree equal to 1) and vague locations (locations with membership degree less than 1 and greater than 0), respectively.

In this paper, we propose that fuzzy spatial objects (i.e., FuzzyGeometry objects) can have linguistic terms since it represents a specific vague spatial phenomenon. For instance, a fuzzy linestring object represents a determinate phenomenon. This phenomenon

can have characteristics that identify itself. An example is showed as follows. An attribute stores fuzzy linestrings that represent animal routes. Animal routes can have linguistic terms that symbolize the frequency that an animal appears in a local. For instance, *all the time*, *sometimes*, and *few times*. Hence, a fuzzy linestring object has a linguistic term associated that indicate the frequency of such animal route. For instance, an animal route R that has as linguistic term *sometimes*. This means that each point of the fuzzy linestring object $\tilde{A}$ that represents the animal route R has a membership degree to indicate the level of a frequency, which is *sometimes*. Therefore, let p be a point of $\tilde{A}$ with membership degree equal to 0.8, then it indicates 80% of chance to an animal appears *sometimes* at point p. Linguistic terms are commonly used in the fuzzy logic. To do this operation, we propose the operation (iii).

The operation (iv) transforms FuzzyGeometry objects into crisp spatial objects (i.e., in PostGIS objects). This means that, the membership degrees disappear. That is, the resulting crisp spatial object is formed by all the points with membership degree greater than 0.

4.5. Operations Based on the Fuzzy Set Theory

The operations based on the fuzzy set theory of the FuzzyGeometry ADT are: fuzzy spatial alpha-cut, fuzzy spatial strict alpha-cut, fuzzy spatial concentration, fuzzy spatial dilation, fuzzy spatial height, and fuzzy spatial normalization. These operations are adaptations of fuzzy operations (Section 3.2) to deal with fuzzy spatial objects. The fuzzy spatial alpha-cut, fuzzy spatial strict alpha-cut, fuzzy spatial concentration, fuzzy spatial dilation, and fuzzy spatial normalization operations belong to group (i) (operations that return spatial data). The fuzzy spatial height belongs to group (iii) (operations that return numbers). Let $\tilde{A}$ be a FuzzyGeometry object, $\alpha \in [0, 1]$, $p > 1$, $r \in]0, 1[$, and $h \in]0, 1]$. Then, we define the following signatures:

(i) *FG_Alphacut*($\tilde{A}$, α) $\rightarrow \tilde{A}$
(ii) *FG_Strict_Alphacut*($\tilde{A}$, α) $\rightarrow \tilde{A}$
(iii) *FG_Concentration*($\tilde{A}$, p) $\rightarrow \tilde{A}$
(iv) *FG_Dilation*($\tilde{A}$, r) $\rightarrow \tilde{A}$
(v) *FG_Height*($\tilde{A}$) $\rightarrow h$
(vi) *FG_Normalization*($\tilde{A}$) $\rightarrow B$

The fuzzy spatial alpha-cut (i) and the fuzzy spatial strict alpha-cut (ii) operations filter the locations that have membership degree equal to or greater or equal to α, respectively. These operations are useful to identify locations that contain specifics membership degrees.

The fuzzy spatial concentration (iii) and dilation (iv) operations decrease and increase the membership degrees of the locations, respectively. These operations are useful to analyze or edit locations in order to intensify or smooth the spatial vagueness. In addition, these operations can change the meaning of linguistic term associated with the vague spatial object.

The fuzzy spatial height (v) operation returns the fuzzy height of the membership degrees of a FuzzyGeometry object. This operation is necessary for the fuzzy spatial normalization (vi) operation. If the height of a FuzzyGeometry object is 1, than it has a core. Otherwise, the normalization (vi) can be used to enforce an existence of a core, which will be the locations with the higher membership degree.

5. Conclusions and Future Work

In this paper, we proposed a novel abstract data type called FuzzyGeometry to handle vague spatial objects based on the fuzzy model in the PostgreSQL. Vague spatial data are an important representation of real-world phenomena that have vague characteristics, i.e., inexact location or uncertain boundaries. Fuzzy model can be adequately used for representation of spatial vagueness. Although this model was proposed, implementations are limited and have not been incorporated into SDBMS. Hence, FuzzyGeometry ADT advances in the state of art to handle vague spatial objects in a SDBMS.

Several operations have been proposed and implemented to handle FuzzyGeometry objects. Among them is the implementation of geometric set operations. Additionally, we proposed the use of linguistic terms to characterize a vague spatial object. Further, we propose textual representations in order to insert and retrieve FuzzyGeometry objects in a spatial database. Finally, we propose new operations based on the fuzzy set theory, such as fuzzy spatial concentration and fuzzy spatial dilation.

Future work will deal with the topological relationship between vague spatial objects based on the fuzzy model, such as the implementation of the predicate "overlap" [Carniel et al. 2014]. Further, the implementation of the fuzzy region data type. We will also propose algorithms in order to extract information by using fuzzy inference [Jamshidi et al. 1993] on vague spatial objects based on the fuzzy model.

Acknowledgments

The authors have been supported by the Brazilian research agencies FAPESP, CAPES, and CNPq.

References

Beaubouef, T., Ladner, R., and Petry, F. (2004). Rough set spatial data modeling for data mining. *International Journal of Intelligent Systems*, 19(7):567–584.

Bejaoui, L., Pinet, F., Bedard, Y., and Schneider, M. (2009). Qualified topological relations between spatial objects with possible vague shape. *International Journal of Geographical Information Science*, 23(7):877–921.

Carniel, A. C., Schneider, M., Ciferri, R. R., and Ciferri, C. D. A. (2014). Modeling fuzzy topological predicates for fuzzy regions. In *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, pages 529–532, New York, NY, USA. ACM.

Cheng, R., Kalashnikov, D. V., and Prabhakar, S. (2003). Evaluating probabilistic queries over imprecise data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 551–562, New York, NY, USA. ACM.

Cohn, A. and Gotts, N. (1995). The 'egg-yolk' representation of regions with indeterminate boundaries. In *Geographic objects with indeterminate boundaries*, pages 171–187. Francis Taylor.

Dilo, A., Bos, P., Kraipeerapun, P., and de By, R. A. (2006). Storage and manipulation of vague spatial objects using existing GIS functionality. In Bordogna, G. and Psaila, G., editors, *Flexible Databases Supporting Imprecision and Uncertainty*, volume 203, pages 293–321. Springer Berlin Heidelberg.

Dilo, A., de By, R. A., and Stein, A. (2007). A system of types and operators for handling vague spatial objects. *International Journal of Geographical Information Science*, 21(4):397–426.

Güting, R. H. (1994). An introduction to spatial database systems. *The VLDB Journal*, 3(4):357–399.

Jamshidi, M., Vadiee, N., and Ross, T. J. (1993). *Fuzzy Logic and Control*. Prentice-Hall.

Klement, E. P., Mesiar, R., and Pap, E. (2000). *Triangular Norms*. Springer.

Kraipeerapun, P. (2004). Implementation of vague spatial objects. Master's thesis, International Institute for Geo-Information Science and Earth Observation.

Li, R., Bhanu, B., Ravishankar, C., Kurth, M., and Ni, J. (2007). Uncertain spatial data handling: Modeling, indexing and query. *Computers & Geosciences*, 33(1):42–61.

Pauly, A. and Schneider, M. (2008). *Quality Aspects in Spatial Data Mining*, chapter Querying vague spatial objects in databases with VASA, pages 3–14. CRC Press, USA.

Pauly, A. and Schneider, M. (2010). VASA: An algebra for vague spatial data in databases. *Information Systems*, 35(1):111–138.

Pawlak, Z. (1982). Rough sets. *International Journal of Computer & Information Sciences*, 11(5):341–356.

Schneider, M. (2008). Fuzzy spatial data types for spatial uncertainty management in databases. In Galindo, J., editor, *Handbook of Research on Fuzzy Information Processing in Databases*, pages 490–515. IGI Global.

Schneider, M. (2014). Spatial Plateau Algebra for implementing fuzzy spatial objects in databases and gis: Spatial Plateau data types and operations. *Applied Soft Computing*, 16(3):148–170.

Schneider, M. and Behr, T. (2006). Topological relationships between complex spatial objects. *ACM Transactions on Database Systems*, 31(1):39–81.

Siqueira, T. L., Ciferri, C. D. A., Times, V. C., and Ciferri, R. R. (2014). Modeling vague spatial data warehouses using the VSCube conceptual model. *Geoinformatica*, 18(2):313–356.

Somodevilla, M. J. and Petry, F. E. (2004). Indexing mechanisms to query fmbrs. In *IEEE Annual Meeting of the Fuzzy Information*, pages 198–202 Vol.1.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8:338–353.

Zinn, D., Bosch, J., and Gertz, M. (2007). Modeling and querying vague spatial objects using shapelets. In *Proceedings of the International Conference on Very Large Data Bases*, pages 567–578, Vienna, Austria.

Improvements of the divide and segment method for parallel image segmentation

Anderson Reis Soares[1], Thales Sehn Körting [1], Leila M. Garcia Fonseca [1]

[1]Instituto Nacional de Pesquisa Espaciais (INPE)
Caixa Postal 515 – 12.245-970– São José dos Campos – SP – Brazil

{anderson.soares, thales.korting, leila.fonseca}@inpe.br

Abstract. *Remote Sensing is an important source of information about the dynamics of Earth's land and oceans, but retrieve information from this technique, is a challenge. Segmentation is a traditional method in remote sensing, which have a high computational cost. An alternative to suppress this problem is use parallel approaches, which split the image into tiles, and segment each one individually. However, the divisions among tiles are not natural, which create inconsistent objects. In this work, we extended our previous work, which used non-crisp borders computed based on graph-theory. By applying this non-crisp line cut, we avoid the post-processing of neighboring regions, and therefore speed up the segmentation.*

1. Introduction

Remote Sensing is an important source of data, spatial programs, such as Landsat, that collected more than 4 million images of the Earth's surface over 40 years, are important sources to understand the dynamic of land (Bolch et al., 2010). However, the development of methods to process and analyze this data, even with current computational power, is a challenge.

Image segmentation is a traditional method in remote sensing, which demands a lot of computational power and is widely used, especially more recently with the emergence of the Geographic Object-Based Image Analysis (GEOBIA). According to Körting et al. (2013) GEOBIA firstly identify regions in the image using segmentation, then extract neighborhood, spectral and spatial descriptive features and afterwards combine regions and features for object classification. However, these elements also turn GEOBIA a complex method because the difficulties related to image segmentation (Pinho et al., 2008) and the many different methods needed to model patterns (Hay and Castilla, 2008).

Segmentation is a fundamental problem in all image-processing applications. Soille (1999) defined segmentation as a process to split an image grouping the pixels by a similar attribute, such as the gray level, so the line which splits the areas, ideally, must be an edge. Gonzalez and Woods (2008) defined an edge, as a region where the intensity of pixels varies abruptly.

The results of segmentation must create uniform areas, which allow a simpler interpretation by the users and simpler representation for classification algorithms. Because of this, algorithms must consider the context, scale, neighborhood, meaning, and computational resources, and for that, this technique demands certain computational

power (Körting et al., 2011). Because of this, parallel approaches became an alternative to suppress this computational cost.

Parallel architectures have become quite popular for image analysis applications (Seinstra and Koelma, 2004). The parallel implementation of segmentation divides the process into different threads. For that, the image is split into tiles, usually using crisp lines. However, this may generate inconsistent objects since the divisions among tiles are not natural.

In our previous work (Körting et al., 2013), we proposed to create non-crisp borders between the image, using an algorithm based on the graph theory to find the best line cut over an edge image, obtained using the magnitude of gradient image. In this article, we extended this approach, to find optimal line cuts in both horizontal and vertical directions, using directional high-pass directional filters and low-pass filter. With this combination, blurred borders are created thus minimizing the occurrence of inconsistent objects.

2. Related Work

Usually, the images are split with crisp lines, however, according to Wassenberg et al. (2009), this is not acceptable because border objects are not correctly handled, and for that, inconsistent segments are created. So, it is needed to merge the segments after segmentation, to combine the tiles and recreate the full image. However, other problems are created; one of them is merge the neighboring blocks without prejudicing the homogeneity in bordering regions. The second problem is the reproducibility of the results (Happ et al., 2010; Körting et al., 2013).

Happ et al. (2010) employed the traditional parallel segmentation, using multithreading parallel implementation of a region-growing algorithm proposed originally by Baatz and Schape (2000). The use of crisp lines imply a post-processing step to treat the boundary segments.

Different approaches have been proposed for solving the problem of splitting an image without using crisp borders. According to Basavaprasad and Hegadi (2012), based on the graph theory the image elements are better structured, and because of this, solve the image problems became more simple and the computation more efficient. However, this approach increases the amount of data to be handled, but it has several attractive properties as highlighted by Felzenszwalb and Huttenlocher (2004). One of them is the possibility to use minimum cost algorithms to find the best path between two nodes. In this case, it is possible to speed-up the process of finding the best cutting line.

Brejl and Sonka (1999) proposed a method for image segmentation, in which the borders are detected automatically based on learning. In this graph-based optimal border detection method, the features were selected from a predefined global set using radial-basis neural networks.

Shi and Malik (2000) proposed an approach that treats the segmentation process as a graph-partitioning problem, through an adjacency matrix connecting all pixels of the image, and use it to perform the partitioning of the graph using normalized cuts.

Körting et al. (2013) also proposed an approach based on the graph theory. The authors proposed to divide the image into adaptive tiles, were the borders of tiles were built along the line of the maximum magnitude of the gradient image. The strategy to find

the adaptive tiles used the Dijkstra's algorithm, which is a graph-based approach to the design of image processing operators based on connectivity. This method considers one image as a directed graph whose nodes are the image pixels and whose arcs are the neighboring pixel pairs. This way, the division lines should follow the natural border of the segments, in most of the cases.

Lassalle et al. (2015) proposed a different approach. The graph is not used to find the best cut line, but to perform a scalable tile-based framework for region-merging algorithms. With such techniques, the authors expected to obtain identical results, with respect to processing the whole image at once.

3. Method

As an alternative for the traditional parallel method, we extended our previous method (Körting et al. 2013) which create adaptive tiles, based on the magnitude of the gradient image, hereby called *previous approach.* In this work, trough directional high-pass filters combined with a mean filter, hereby called *our approach*, our algorithm improved in finding the cutting line. Figure 1 shows the workflow.

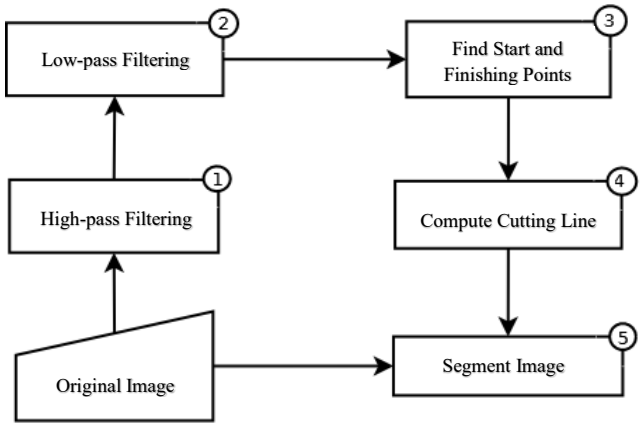


Figure 1. Scheme for segmenting images using tiles.

3.1. High-pass filtering

The first edge image was obtained using the *previous approach*. According to Gonzalez and Woods (2009), the gradient of pixels, is computed as the two-dimensional column vector, which indicates, for each pixel, the intensities of the border in horizontal and vertical directions (Equation 1). The magnitude of this vector points out the border's strength, Equation 2.

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1)$$

$$mag(\nabla f) = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial x} \right)^2 \right]^{\frac{1}{2}} \quad (2)$$

Another way to obtain the edges in an image is using directional filters. This type of filter enhance the edges in specific directions. Numerous filters has been proposed, such as Roberts, Sobel and Prewitt (Prewitt, 1970). In this work, we applied the Prewitt high-pass filters, since it produces less noise them others, as highlighted by Schowengerdt (2007). When applying a south filter (shown in Figure 2) or a north filter, the edges are

enhanced on south or north directions, respectively, so it is easier for the algorithm to find the best path on horizontal direction.

$$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 2. Prewitt South Filter applied to images using a 3x3 neighborhood.

3.2 Low-pass Filtering

As can be seen on the example image, Figure 3, the high-pass filter normally highlights local maximum in the image. To follow a path in some edge image, we should consider not only the local maximum, but also neighboring pixels to avoid holes in the path, as occurred on our *previous approach*. In *our approach*, we fix those points using a low-pass filter, such as the mean filter, in this edge image obtained from the high-pass filter, Figure 3 (c). As can be seen on Figure 3 (d), the local maximum pixels are now blurred by the low-pass filter, because of this, the algorithm have new possibilities, and the chances of fall in a path with holes are reduced**.**

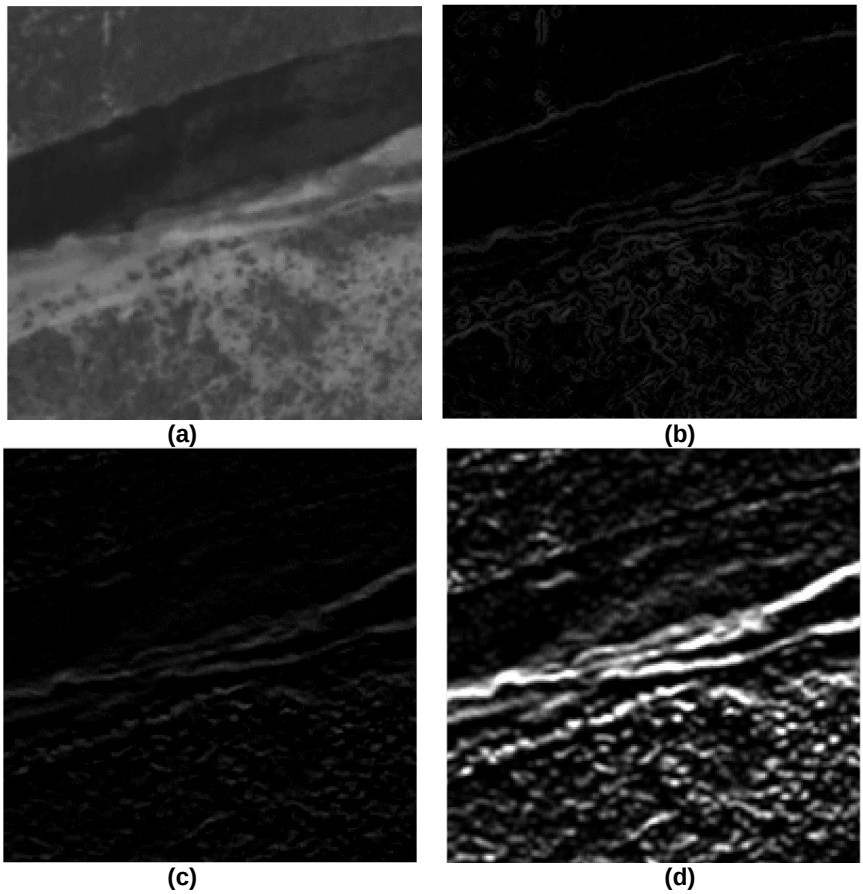


Figure 3. a) Original image, b) gradient image, c) directional high-pass image, and d) the previous result (c) with a low-pass filter.

In Figure 4 we show the histogram of the edge image obtained from both approaches. The variations on the image using *our approach* are more evident. Therefore, it is easier to find the best path. With this edge image, blurred by the low-pass filter, it is possible to create an adjacency matrix to find the optimal cutting line.

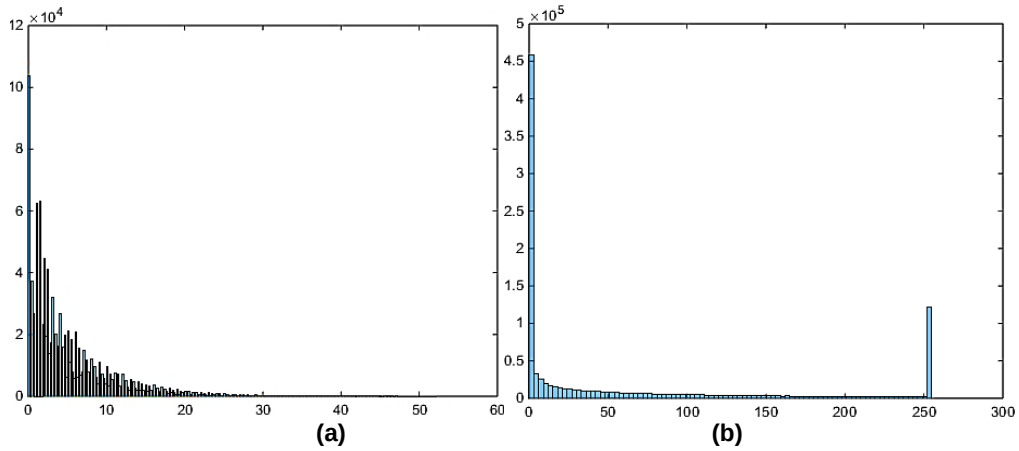


Figure 4. a) Histogram from edge image of *previous approach*, b) Histogram from edge image of *our approach* (b).

3.3. Find Starting and Finishing Points

The choice of the border is based on the pixel value. The starting point is the pixel with the highest value on the first column (or line, in case the cutting line is vertical). In case there are more than one pixel, the first on the row (or column) is assumed as the starting point. Same process is used to define the finishing point.

3.4 Compute Cutting Line

After define the starting and finishing points, the adjacency matrix is created. According to Körting et al. (2013) the adjacency matrix is a graph, whose nodes are the image pixels and whose arcs are an adjacency relation between pixels. The adjacency between the pixels is defined by five connections, including top, top-right, right, bottom-right and bottom pixels, as shown in Figure 5.

The arcs between nodes are weighted according to the pixel value. If one pixel is not a border, the distance, or the value of arc, is assigned as infinite, this way the algorithm should not go through this arc. When the adjacency pixel is a border, the distance assigned to the arc is minimum, therefore, the algorithm should go through them. With this graph is possible to apply an algorithm to compute the minimum cost between the starting and finishing points. As previously mentioned, we have employed the Dijkstra's algorithm (Dijkstras, 1959).

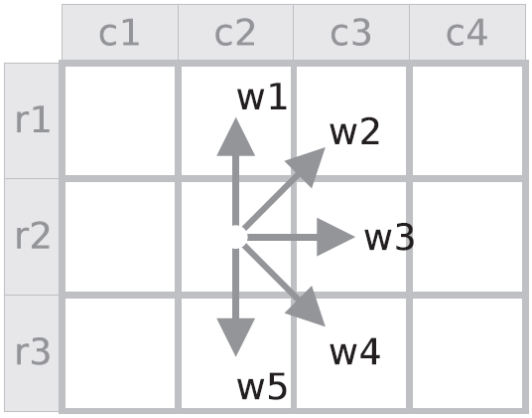


Figure 5. The five weights associated to each pixel to build the adjacency matrix. (Körting et al., 2013).

As the graph demands a lot of computational power, compute the cutting line over all image became a bottleneck, so is necessary create blocks to limit the size of the graph. Using this strategy, the candidate line will be created based on shortest path between starting and finishing edges over each blocks. To avoid tiles with very different sizes, must be defined a region of interest, a buffer zone, limited by a maximum displacement between the candidate line. This line define the non-crisp border between the tiles, and is used to split the original image.

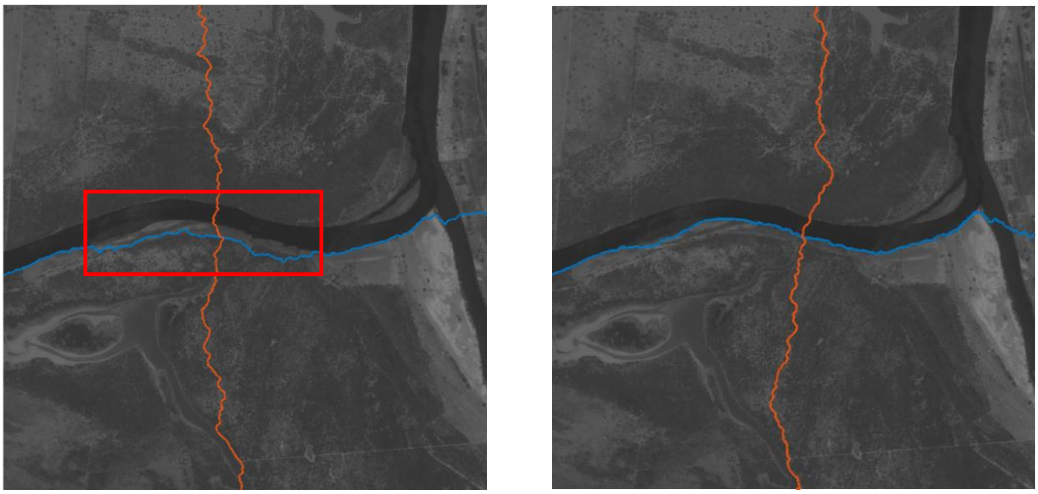
3.5. Segment Image

After computing the cutting lines, they are used to create the tiles. Each thread of a parallel segmentation scheme receive a tile, which will be segmented individually through the multiresolution segmentation algorithm (Baatz and Schäpe, 2000). Finally, after the segments are created, they are merged, creating the result, which does not need a post-processing step.

4. Results

For evaluation of the algorithm, we used three images with different contexts. The first image is a crop (1000 × 1000 pixels) of a region in the state of Bahia, Brazil, obtained by sensor HRC of satellite CBERS-2B. The second image is a crop of a Quickbird scene from São Paulo, Brazil, with 1000 × 1175 pixels. The pixels of this image have a spatial resolution of 0.6m. The third image is a crop of a WorldView-2 scene (2400 × 3200 pixels), of a region in the city of São José dos Campos, Brazil. This image has a spatial resolution of 0.5m.

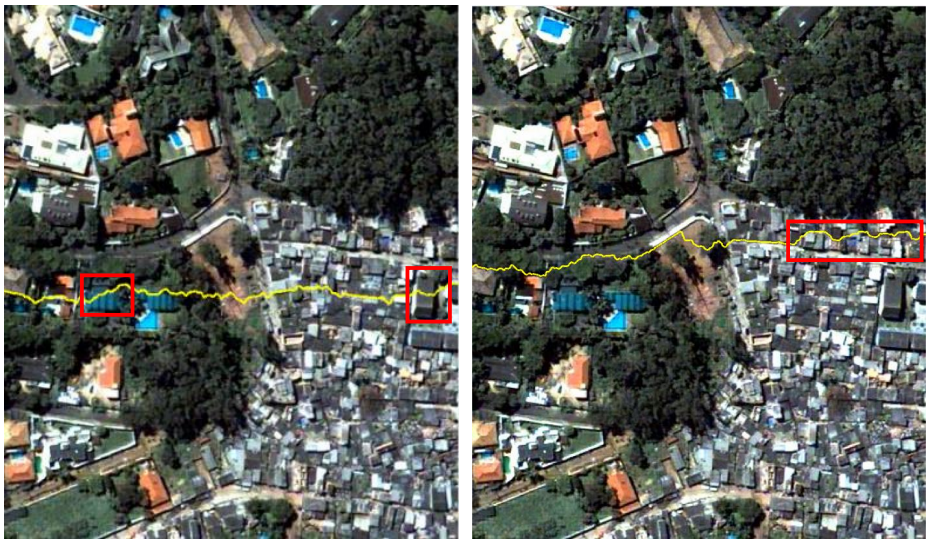
All tests were performed using equivalent parameters, on all edge images. On the first experiment, Figure 6, the cutting line created using the *previous approach*, and *our approach*, followed different paths, for both directions. As can be observed on Figure 6 (a), the cutting line using *previous approach* do not followed the edge of the river, which we consider the ideal path for this image, this mistake was caused by a maximum local pixel, which was fixed using *our approach* as can be seen of Figure 6 (b). On the vertical direction, the path followed was different, but on this direction, there are not major feature to easily split the image.



(a) (b)

Figure 6. Cutting lines created using a) *previous approach* and b) *our approach*.

On the second experiment, Figure 7, the cutting line produced by both approaches were different. Using the *previous approach*, the line cut gone through a building area, this result divided some buildings in two parts as highlighted in Figure 7 (a). Using *our approach* was obtained a better cutting line, in most parts of image. At the building area, there were not major cuts on buildings, despite in some points the line did not gone through a path we consider ideal, following the street on the right side of the image.



(a) (b)

Figure 7. Cutting lines created using a) *previous approach* and b) *our approach*.

However, problems occurred over trees, which must be caused by the shadows over the canopy, this variation created the edges. Another issue in *our approach* is caused

the intersection of cutting lines, which may create inconsistent objects, depending on the segmentation.

On the third experiment, Figure 8, the cutting line using *our approach* split the objects in image without major problems. Even passing near trees, the cutting lines gone to almost an optimal path, only one building, on horizontal direction, were split, which was caused by shadows. On the vertical direction, again, the cutting line split one building (as highlighted in the image), which was caused by the difference on illumination on the roof.

Figure 8. Cutting line created using *our approach*.

To analyze how the non-crisp lines effects on segmentation, we compared the results of a segmentation on tiles using crisp lines, which demands a post-processing step to merge neighboring regions obtained by different tiles, and *our approach*. Ideally, the segments created through segmentation should create homogeneous areas, however, with crisp lines the segments generated sometimes do not satisfy this condition, as exhibit on Figures 9 (a) and 10 (a). On Figure 9 the post-processing in those step did not merged some segments (note the roofs on center left of image and the soil area on the left side of image).

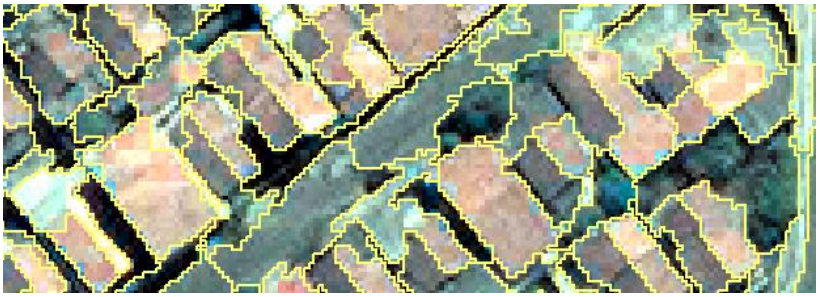
On Figure 10, the piece roof in the center of image was too small to create an individual region, therefore, it was merged in the region containing trees. Due to this problem, the bottom region with the rest of the roof was not merged, because the spectral difference between these two regions is too high.

(a)

(b)

Figure 9. The intersection of tiles. Results using a) *crisp lines* and b) *our approach*.

(a)



(b)

Figure 10. The intersection of tiles. Results using a) *crisp lines* and b) *our approach*.

The presence of those crisp lines on the segments may generate some problems to classify the images, since the metrics used to characterize the objects and perform the classification are based on characteristics of the segments.

5. Conclusion

In this article we presented improvements to the divide and segment approach (Körting et al. 2013) to parallel image segmentation. The *previous approach* used the gradient image to compute the cutting line, but local maximum points influenced the results. We solved this problem in *our approach* using Prewitt directional high-pass filter combined with a low-pass filtering, with this strategy the edges are blurred and the creating new possibilities to the algorithm to find the best path.

However, in some cases the intersection of cutting lines may create some inconsistent objects, caused by the shape of lines. Dealing with these regions remains an open problem, currently unsolved by our approach. Future works include solving the problem of inconsistent objects caused by the intersection of cutting lines, and implement this approach on TerraLib library (Câmara et al. 2008).

References

Baatz, M. and Schape, A. (2000). Multiresolution Segmentation: an optimization approach for high quality multi-scale image segmentation. In: Angewandte Geographische Informationsverarbeitung, 12, 2000, Wichmann-Verlag. Proceedings... p. 12-23, WichmannVerlag: Heidelberg, 2000.

Basavaprasad, B. and Hegadi, R. S. (2012). Graph theoretical approaches for image segmentation. Aviskar–Solapur Univ Res J, 2, pp. 7-13.

Bolch, T., Menounos, B. and Wheate, R. 2010. Landsat-based inventory of glaciers in western Canada, 1985–2005. *Remote Sensing of Environment*, Vol. 114, No. 1, pp. 127–137. doi: 10.1016/j.rse.2009.08.015.

Brejl, M. and Sonka, M. (1999). Medical Image Segmentation: Automated Design of Border Detection Criteria from Examples. *Journal of Electronic Imaging*, v. 8, pp. 54-64.

Câmara, G., Vinhas, L., Ferreira, K., Queiroz, G., Souza, R., Monteiro, A., Carvalho, M., Casanova, M., and Freitas, U. (2008). TerraLib: An open source GIS library for largescale environmental and socio-economic applications. *Open Source*, p. 247-270.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, p. 269–271.

Felzenszwalb, P. F., and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. In: *International Journal of Computer Vision*, v. 59, n. 2, p. 167-181.

Gonzalez, R. C. and Woods, R. E. (2008). Digital Image Processing. 3rd Edition, Prentice Hall. ISBN 0-13-168728-8.

Happ, P., Ferreira, R., Bentes, C., Costa, G. and Feitosa, R. (2010). "Multiresolution segmentation: a parallel approach for high resolution image segmentation in multicore

architectures", In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.

Hay, G., Castilla, G. (2008). Geographic Object-Based Image Analysis (GEOBIA): A new name for a new discipline. In: Object Based Image Analysis: Spatial Concepts for Knowledge-Driven Remote Sensing Applications, Edited by Blaschke, T., Lang, S. and Hay, G. Springer, Heidelberg.

Körting, T. S., Castejon, E. F. and Fonseca, L M. G. (2011). Divide and Segment - An alternative for parallel segmentation. In**:** *Proceedings of XII GEOINFO*, pp. 97–104. INPE, Campos do Jordão.

Körting, T. S., Fonseca, L. M. G., and Câmara, G. (2013). GeoDMA—Geographic data mining analyst. *Computers & Geosciences*, 57, 133-145.

Körting, T. S., Castejon, E. F. and Fonseca, L. M. G. (2013). The Divide and Segment Method for Parallel Image Segmentation. In: Advanced Concepts for Intelligent Vision Systems. Springer International Publishing, pp. 504-515.

Lassalle, P., Inglada, J., Michel, J., Grizonnet, M. and Malik, J. (2015). A Scalable Tile-Based Framework for Region-Merging Segmentation. In: IEEE Transactions On Geoscience And Remote Sensing, Vol. 53, n. 10. pp. 5473 - 5485.

Pinho, C., Silva, F., Fonseca, L. and Monteiro, A. (2008). Intra-urban land cover classification from high-resolution images using the C4.5 algorithm. In: ISPRS Congress Beijing 7.

Prewitt, J. MS. (1970). Object enhancement and extraction. *Picture processing and Psychopictorics*, v. 10, n. 1, p. 15-19.

Seinstra, F.J., Koelma, D. (2004). User transparency: a fully sequential programming model for efficient data parallel image processing*,* In: *Concurrency and Computation: Practice and Experience*, 16, pp. 611–644.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8), pp. 888–905.

Soille, P. (2013). Morphological image analysis: principles and applications. Springer Science & Business Media.

Schowengerdt, R. A. (2007). Remote Sensing Models and Methods for Image Processing. 3rd Edition, Elsevier.

Wassenberg, J., Middelmann, W. and Sanders, P. (2009). An Efficient Parallel Algorithm for Graph-Based Image Segmentation. In: *Computer Analysis of Images and Patterns*, pp. 1003–1010. Springer.

Embedding Vague Spatial Objects into Spatial Databases using the VagueGeometry Abstract Data Type

Anderson Chaves Carniel[1], **Ricardo Rodrigues Ciferri**[2], **Cristina Dutra de Aguiar Ciferri**[1]

[1]Department of Computer Science – University of São Paulo in São Carlos
13.560-970 – São Carlos – SP – Brazil

accarniel@gmail.com, cdac@icmc.usp.br

[2]Department of Computer Science – Federal University of São Carlos
15.565-905 – São Carlos – SP – Brazil

ricardo@dc.ufscar.br

Abstract. *Spatial vagueness has been required by geoscientists to handle vague spatial objects, i.e., spatial objects that do not have exact locations, strict boundaries, or sharp interiors. However, there is a gap in the literature in how to handle these objects in spatial database management systems since they mainly provide support to crisp spatial objects, i.e., objects that have well-defined locations, boundaries, and interiors. In this paper, we fill this gap by proposing VagueGeometry, a novel abstract data type that handles vague spatial objects, includes an expressive set of vague spatial operations, and its implementation is open source. Experimental results show that VagueGeometry improved the performance of spatial queries with vague topological predicates from 23% up to 84% if compared with functionalities available in current spatial databases.*

1. Introduction

Spatial database management systems (spatial DBMS) and Geographical Information Systems (GIS) mainly provide support to handle *crisp spatial objects* to represent real-world phenomena by using points, lines, and regions. Crisp spatial objects characterize spatial phenomena with exact locations and whose shape and boundary are precisely defined [Schneider and Behr 2006]. Examples are cities with their political boundaries. For their handling, spatial operations like geometric set operations (e.g., union), topological predicates (e.g., overlap), and numerical operations (e.g., distance) are defined and used in spatial queries.

However, geoscientists are increasingly interested in modeling spatial real-world phenomena that do not have exact locations, strict boundaries, or sharp interiors. This characterization leads to *spatial vagueness*. In general, it has a spatial extent that *certainly* belongs to the real-world phenomena (i.e., the *kernel*) and a spatial extent that *maybe* belongs to the real-world phenomena (i.e., the *conjecture*) [Siqueira et al. 2014].

There are a number of approaches proposed in the literature that define models to represent spatial vagueness, which can be classified as *probabilistic models* [Li et al. 2007, Zinn et al. 2007], *fuzzy models* [Kraipeerapun 2004, Dilo et al. 2007, Dilo et al. 2006, Carniel et al. 2014], and *exact models* [Clementini and Di Felice 1997,

Pauly and Schneider 2008, Pauly and Schneider 2010, Bejaoui et al. 2010]. These models introduce concepts and notions of *vague spatial objects* by formally defining spatial data types for *vague points*, *vague lines*, and *vague regions*. They also introduce vague spatial operations to handle them, i.e., *vague geometric set operations* (e.g., vague geometric union), *vague topological predicates* (e.g., vague overlap), and *vague numerical operations* (e.g., vague distance).

There are several advantages of incorporating vague spatial objects and their operations into spatial queries, such as to provide a more realistic representation of application environments, to allow users to manipulate vague spatial objects found in real-world phenomena, and to provide an efficient processing of operations on vague spatial objects. For instance, in an ecological application, users aim to manage habitats of species and polluted areas of rivers. The habitats and the polluted areas of rivers are represented by vague regions. This means that habitats have locations where species certainly appear and locations where species maybe appear. Further, rivers have unpolluted areas and areas where there is some kind of pollution. By using such data, a user can ask the following query: "Find all polluted areas of rivers that *maybe overlap* with habitats of species". In this query, spatial vagueness is represented by the *maybe overlap* predicate, which will return true when the overlap occurs to some extent, i.e., occurs with some degree of uncertainty.

To handle spatial objects in spatial applications, *abstract data types* (ADT) have been used in spatial DBMS and GIS. An ADT for spatial data types aids the use of spatial operations in spatial queries by hiding their complexities from the user. While ADTs for crisp spatial data are deeply implemented in the literature, this is not the case for vague spatial data. Although there are approaches that provide ADTs for vague spatial data [Dilo et al. 2006, Kraipeerapun 2004, Pauly and Schneider 2008, Pauly and Schneider 2010, Zinn et al. 2007], they face several limitations. First, they only provide support for a small subset of vague spatial operations. Second, they do not support textual and binary representations of vague spatial objects. Finally, they do not support SQL operators to manipulate results of vague spatial operations.

In this paper, we fill this gap in the literature. We propose a novel ADT named VagueGeometry to incorporate vague spatial objects into a spatial DBMS. VagueGeometry is based on the *exact model* since this model reuses existing concepts and implementations of crisp spatial data and formally defines a complete set of vague spatial operations. The main advantage to use implementation of crisp spatial data is that they are well defined and their efficiency is largely explored in the literature. Among the exact models [Bejaoui et al. 2010, Pauly and Schneider 2008, Pauly and Schneider 2010, Clementini and Di Felice 1997], VagueGeometry is based on the *Vague Spatial Algebra* (VASA) [Pauly and Schneider 2008, Pauly and Schneider 2010] since it formally defines simple and complex vague spatial data types. Further, VASA introduces a more expressive algebra than the models described in [Bejaoui et al. 2010] and [Clementini and Di Felice 1997] by including a huge set of operations, such as vague geometric set operations, vague topological predicates, and vague numerical operations.

VagueGeometry greatly overcomes the aforementioned limitations. Other major characteristics of the proposed VagueGeometry are described as follows.

- It offers textual and binary representations for vague spatial objects, which make possible their use to define, insert, and retrieve vague spatial objects by using

textual or binary representations. Further, these representations can be used as a way of communication and interoperability between different spatial applications.
- It implements an expressive set of spatial operations for vague spatial objects. To comply with this goal, VagueGeometry includes the specification of vague geometric set operations, vague topological predicates, and vague numerical operations. As a result, the use of VagueGeometry empowers the management of vague spatial objects in spatial applications by users.
- It supports SQL operators that allow users to handle results of vague topological predicates and vague numerical operations.
- It is open source and implemented in the open source PostgreSQL DBMS with the PostGIS spatial extension. This means that spatial applications are able to access directly a spatial DBMS containing vague spatial objects and handle these objects by using vague spatial operations accordingly.
- It includes an efficient improvement to process vague topological predicates in spatial queries.

This paper is organized as follows. Section 2 surveys related work. Section 3 summarizes the Vague Spatial Algebra. Section 4 introduces the proposed VagueGeometry. Section 5 details the improvement in the processing of vague topological predicates. Section 6 describes performance tests. Section 7 concludes the paper.

2. Related Work

There are few approaches that implement ADTs for handling vague spatial objects in spatial DBMS and GIS [Dilo et al. 2006, Kraipeerapun 2004, Pauly and Schneider 2008, Pauly and Schneider 2010, Zinn et al. 2007]. These approaches mainly differ from our proposed VagueGeometry in the practicable applicability of the user to handle vague spatial objects in spatial queries. We compare these approaches with VagueGeometry by considering the following functionalities related to the support provided by them: (i) textual representation, (ii) binary representation, (iii) vague geometric set operations, (iv) vague topological predicates, (v) vague numerical operations, (vi) SQL operators, and (vii) coupling with a spatial DBMS.

The majority of the approaches does not provide textual and binary representations (functionalities (i) and (ii)) to allow the user to insert and retrieve vague spatial objects. While in [Zinn et al. 2007] is provided input and output functions for vague spatial objects by using binary format, this is not the case for the textual representation. The approaches described in [Pauly and Schneider 2008] and [Pauly and Schneider 2010] define vague spatial objects by using extensive terminal command lines without any textual or binary representations. The approaches described in [Kraipeerapun 2004] and [Dilo et al. 2006] support options to represent vague spatial objects by using files in the format of the GRASS GIS. However, the binary and textual formats are not understandable for users and depend on a specific system, thus limiting interoperability between spatial applications. On the other hand, VagueGeometry defines textual and binary representations for vague spatial objects.

Regarding functionality (iii), the approaches described in [Zinn et al. 2007], [Pauly and Schneider 2008], and [Pauly and Schneider 2010] implement vague geometric union, intersection, and difference between vague points, lines, and regions.

The approaches described in [Kraipeerapun 2004] and [Dilo et al. 2006] do not implement the vague geometric difference between vague lines. Regarding functionalities (iv) and (v), some approaches described in [Pauly and Schneider 2008] and [Pauly and Schneider 2010] provide support to vague topological predicates and vague numerical operations, while other approaches in [Zinn et al. 2007], [Kraipeerapun 2004], and [Dilo et al. 2006] do not provide support for these operations, and therefore, limit the management of vague spatial objects and the type of queries that can be processed. Our proposed VagueGeometry implements an expressive set of spatial operations for vague spatial objects, which includes the specification of vague geometric set operations, vague topological predicates, and vague numerical operations.

Furthermore, there is no related work that support SQL operators to handle results of vague spatial operations (functionality (vi)). Although the approaches described in [Pauly and Schneider 2008] and [Pauly and Schneider 2010] propose some operators, they do not implement them. However, offering SQL operators is an important functionality since it allows users to intuitively handle results of spatial operations in SQL queries. Therefore, differently from related work, VagueGeometry supports SQL operators for vague spatial operations.

Finally, regarding functionality (vii), the approaches described in [Kraipeerapun 2004] and [Dilo et al. 2006] do not implement vague spatial objects in a spatial DBMS, while the approaches described in [Pauly and Schneider 2008], [Pauly and Schneider 2010], and [Zinn et al. 2007] offer this functionality. However, the implementation of [Pauly and Schneider 2008] and [Pauly and Schneider 2010][1] is based on the Oracle, which has license restrictions. On the other hand, VagueGeometry is an open source implementation based on the PostgreSQL.

3. Vague Spatial Algebra

A vague spatial object in the Vague Spatial Algebra (VASA) [Pauly and Schneider 2008, Pauly and Schneider 2010] is defined as a pair of crisp spatial objects of the same spatial data type, which must be disjoint or adjacent. The first object represents the *kernel* part, while the second object represents the *conjecture* part. In addition, VASA is characterized to separate the portions of space of a spatial object by considering minimum and maximum approximations. As a result, a spatial object can comprise or expand according to a minimum limit (i.e., the kernel part) and a maximum limit (i.e., the conjecture part). Formally, let $\alpha \in \{\textit{crisp point}, \textit{crisp line}, \textit{crisp region}\}$. Then, a vague spatial data type in VASA is defined by $v(\alpha) = \alpha \times \alpha$, such that for an object $o = (o_k, o_c) \in v(\alpha)$, the property $disjoint(o_k, o_c) \vee meet(o_k, o_c)$ holds. The kernel and conjecture of o are symbolized by o_k and o_c, respectively.

VASA defines the following operations to handle vague spatial objects: vague geometric set operations, vague topological predicates, and vague numerical operations. Vague geometric set operations are defined by reusing the crisp geometric set operations.Vague topological predicates are based on the three-valued logic, and thus, can return *true*, *false*, or *maybe*. Let A and B be two vague spatial objects. A vague topological predicate is evaluated by using the well-known crisp 9-intersection matrix [Schneider and Behr 2006] for the following combinations $A_k \times B_k$, $A_k \times (B_c \oplus B_k)$,

[1] http://www.cise.ufl.edu/research/SpaceTimeUncertainty/

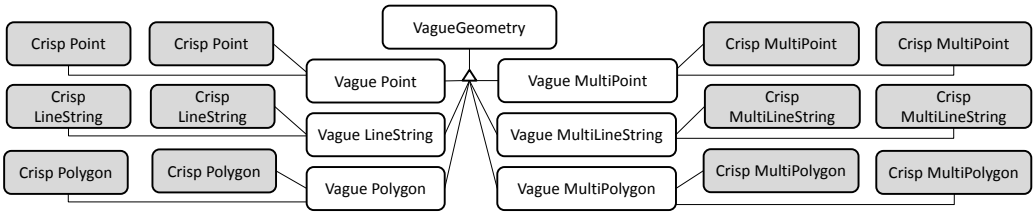


Figure 1. The vague spatial data types of VagueGeometry.

$(A_k \oplus A_c) \times B_k$, and $(A_k \oplus A_c) \times (B_c \oplus B_k)$, where $\oplus$ denotes the crisp geometric union. Finally, vague numerical operations return a pair of numeric values corresponding to a minimum and a maximum value. For instance, the area of a vague region object has a minimum value that corresponds to the area of its kernel and a maximum value that corresponds to the area of the union between its kernel and its conjecture. Details of the formal definitions for vague spatial operations of VASA are given in [Pauly and Schneider 2010].

4. The VagueGeometry Abstract Data Type

In this section, we propose VagueGeometry, an ADT to handle vague spatial objects in a spatial DBMS. VagueGeometry was implemented by using the C language and the extensibility provided by the PostgreSQL internal library. It is based on VASA, and thus we make use of the spatial operations provided by PostGIS and GEOS to implement the vague spatial data types and their operations. GEOS[2] is a C/C++ library that implements crisp spatial data types and their crisp spatial operations according to the OGC specifications[3]. A detailed documentation of VagueGeometry is available at `http://gbd.dc.ufscar.br/vaguegeometry/`.

4.1. Representation of Vague Spatial Objects

Figure 1 depicts vague spatial data types of VagueGeometry which can be simple or complex. Simple vague spatial data types named *vague point*, *vague linestring*, and *vague polygon* denote simple vague points, simple vague lines, and simple vague regions, respectively. Complex vague spatial data types named *vague multipoint*, *vague multilinestring*, and *vague multipolygon* denote complex vague points, complex vague lines, and complex vague regions, respectively. We employ this notation to follow the same notation used by the OGC specification to denote crisp spatial objects. Note that a vague spatial object of VagueGeometry is composed of a pair of disjoint or adjacent crisp spatial objects of the same spatial data type, which is showed in gray in Figure 1.

To use VagueGeometry in spatial queries, we propose textual and binary representations for vague spatial objects. We present our proposed representations by first detailing the *textual representations*. They are: (i) *Vague Well-Known Text* (VWKT), (ii) *Vague Geography Markup Language* (VGML), (iii) *Vague Keyhole Markup Language* (VKML), and (iv) *Vague Geographic JavaScript Object Notation* (vGeoJSON). These textual representations are based on the OGC specifications that use the following textual

[2]http://trac.osgeo.org/geos/
[3]http://www.opengeospatial.org/

representations for crisp spatial objects: Well-Known Text (WKT), Geographic Markup Language (GML), Keyhole Markup Language (KML), and Geographic JavaScript Object Notation (GeoJSON).

The VWKT, VGML, VKML, and vGeoJSON representations are defined as follows. Let A be a VagueGeometry object, which can assume different data types (Figure 1), formed by the kernel A_k and the conjecture A_c. Let $name$ be a function that returns a keyword representing the data type of A. For instance, $name(A)$ returns the keyword `VAGUEPOINT` if A is a simple vague point object. Finally, let ***WKT***, ***GML***, ***KML***, and ***GeoJSON*** be functions that get a crisp spatial object as input and return its respective textual representation. The textual representations for a VagueGeometry object A are:

(i) $VWKT(A) = name(A)(\ WKT(A_k);\ WKT(A_c))$
(ii) $VGML(A) =$ <vgml:$name(A)$><vgml:Kernel>$GML(A_k)$</vgml:Kernel>
<vgml:Conjecture>$GML(A_c)$</vgml:Conjecture></vgml:$name(A)$>
(iii) $VKML(A) =$ <vkml:$name(A)$><vkml:Kernel>$KML(A_k)$</vkml:Kernel>
<vkml:Conjecture>$KML(A_c)$</vkml:Conjecture></vkml:$name(A)$>
(iv) $vGeoJSON(A) =$ {"type": "$name(A)$", "kernel": $GeoJSON(A_k)$,
"conjecture": $GeoJSON(A_c)$}

We now move our discussion to the proposed *binary representation*, called *Vague Well-Known Binary* (VWKB). It is based on the *Well-Known Binary* (WKB) representation for crisp spatial objects documented in the OGC specification. Our VWKB representation is defined as follows. Let id be a function that returns an integer in the binary format symbolizing the data type of A. For instance, $id(A)$ returns 1, in the binary format, if A is a simple vague point object. Let ***WKB*** be a function that gets a crisp spatial object as input and returns its respective WKB representation. Let $endianess$ be an flag that indicates the way in which the bytes are organized in main memory (i.e., either *big-endian* or *little-endian*). The VWKB representation for a VagueGeometry object A is:

$$VWKB(A) = endianess + id(A) + WKB(A_k) + WKB(A_c),$$

where $+$ denotes the union between the serialized data.

VagueGeometry supports textual and binary representations to allow its use in different spatial applications. Hence, spatial applications based on XML or web services that use XML as communication are able to use the VGML and VKML representations. Web applications that utilize JavaScript as main language are able to use the vGeoJSON representation. Applications that manage binary files are able to use the VWKB representation. Finally, for general purpose, applications can make use of the VWKT representation. It is important to note that these representations also provide interoperability between applications since a vague spatial object has an unique representation.

4.2. Vague Spatial Operations

VagueGeometry provides support to *input and output operations*, *vague geometric set operations*, *vague topological predicates*, and *vague numerical operations*. While *input operations* transform textual or binary representations into a VagueGeometry object, *output operations* transform a VagueGeometry object into a textual or binary representation.

Vague geometric set operations get two VagueGeometry objects as input and yield another VagueGeometry object. VagueGeometry implements *vague geometric union*, *vague geometric intersection*, and *vague geometric difference*.

Regarding vague topological predicates, VagueGeometry supports *vague coveredBy*, *vague covers*, *vague crosses*, *vague disjoint*, *vague equals*, *vague inside*, *vague intersects*, *vague meets*, and *vague overlap*. These predicates are based on the three-valued logic, and can return *true*, *false*, or *maybe*. A predicate returns *true* if a relationship *certainly* occurs, *false* if a relationship *certainly not* occurs, and *maybe* if a relationship *occurs to some extent*. To deal with it, VagueGeometry also includes the VagueBool data type. As a result, a VagueBool object can assume *true*, *false*, or *maybe* as value, which correspond to the possible return values of vague topological predicates. In addition, it is possible to use crisp spatial objects as input, which is handled as a vague spatial object containing only the kernel part.

Finally, vague numerical operations supported by VagueGeometry are: *vague area of a vague region object*, *vague length of a vague line object*, and *farthest* and *nearest distance between two vague spatial objects*. These operations return two numeric values, which symbolize the minimum and the maximum values of an operation. To deal with it, VagueGeometry also includes the VagueNumeric data type. As a result, a VagueNumeric object is composed of a pair of double values, which correspond to the minimum and the maximum values returned by vague numerical operations.

As can be noted, our proposed VagueGeometry implements an expressive set of vague spatial operations, which includes the specification of vague geometric set operations, vague topological predicates, and vague numerical operations.

4.3. SQL Operators

We propose SQL operators to handle VagueBool and VagueNumeric objects, i.e., the result of vague topological predicates and vague numerical operations, respectively. For the vague topological predicates, we propose the *logical operators and* (&&), *or* (||), and *not* (!), and the *boolean operators* $\sim$, $\sim\sim$, and $\&$. Logical operators employ the three-valued logic. They get VagueBool objects as input and yield another VagueBool object. The logical operators && and || are binary operators, while ! is a unary operator. For instance, users can use these operators to specify the condition "*VG_Meets*(*vg1*, *vg2*) || *VG_Overlap*(*vg1*, *vg2*)" in a SQL query to indicate that two VagueGeometry objects $vg1$ and $vg2$ meet *or* overlap. On the other hand, *boolean operators* are unary operators that get a VagueBool object as input and have true or false as possible return values. Therefore, a boolean operator transforms a vague topological predicate into a boolean restriction. The operator $\sim$ yields true if the VagueBool object is *true* or *maybe*, and false otherwise. The operator $\sim\sim$ yields true if the VagueBool object is *maybe*, and false otherwise. The operator & yields true if the VagueBool object is *true*, and false otherwise. For instance, users can evaluate if two VagueGeometry objects *maybe* overlap by specifying the condition "$\sim\sim$*VG_Overlap*(*vg1*, *vg2*)" in a SQL query.

Regarding the vague numerical operations, we propose the binary operators $=$ and $\sim$, which get a VagueNumeric object and a numeric value as input and yield true or false. The operator $=$ yields true if the numeric value is equal to the minimum value of the VagueNumeric object, and false otherwise. The operator $\sim$ yields true if the numeric value is between the minimum and the maximum value of the VagueNumeric object, and false otherwise. For instance, users can use this operator to specify the condition "*VG_Area*(r) $\sim$ 800" in a SQL query to restrict vague region objects in the attribute r that have approximately 800 of area.

5. Efficient Processing of Vague Topological Predicates

The implementation of VagueGeometry includes an improvement for the processing of vague topological predicates. The proposed improvement, called *MBRs for Vague Topological Predicates* (*MBRVP*), makes use of *Minimum Boundary Rectangles* (MBR) of the kernel and conjecture parts of vague spatial objects to return the results of vague topological predicates when *some situations hold*. In these situations, MBRVP can avoid the use of crisp 9-intersection matrices to evaluate the vague topological predicate. As a result, the time to process spatial queries can be reduced.

We consider two situations, named *disjointness between MBRs* and *set containment between MBRs*. The disjointness between MBRs encompasses two specific cases, as depicted in Figure 2a. The first case occurs if the MBRs of the union between the kernel and the conjecture of two vague spatial objects are disjoint. The second case occurs if the MBRs of the kernel and the conjecture of two vague spatial objects are disjoint. Note that the second case can happen even when the first case holds.

Formally, let A and B be two vague spatial objects. Let also MBR_o be a MBR of a crisp spatial object o. The *disjointness between MBRs* $S_d(A, B)$ yields true if $((MBR_{A_k} \cup MBR_{A_c}) \cap (MBR_{B_k} \cup MBR_{B_c}) = \varnothing) \vee (MBR_{A_k} \cap MBR_{B_k} = \varnothing \wedge MBR_{A_k} \cap MBR_{B_c} = \varnothing \wedge MBR_{A_c} \cap MBR_{B_k} = \varnothing \wedge MBR_{A_c} \cap MBR_{B_c} = \varnothing)$, and false otherwise. By using this definition, we are able to return *true* for the vague disjoint predicate if $S_d(A, B) = true$ holds, and return *false* for the predicates of vague meets, vague intersects, vague overlap, and vague equals if $S_d(A, B) = false$ holds. Otherwise, the respective predicate is evaluated with the computation of crisp 9-intersection matrices. It includes the case of an intersection between MBRs of the conjecture parts since we cannot guarantee that the conjecture parts really intersects due to the dead space of the MBRs.

Regarding the set containment between MBRs, it also encompasses two specific cases (Figure 2b). The first case occurs if the MBR of the kernel of the first vague spatial object is not inside the MBR of the union between the kernel and the conjecture of the second vague spatial object. Hence, the interior of the kernel of the first vague spatial object intersects the exterior of the second vague spatial object. The second case occurs if the MBR of the kernel of the first vague spatial object and the MBRs of the kernel and the conjecture of the second vague spatial object are disjoint.

Formally, let A and B be two vague spatial objects. Let also MBR_o be a MBR of a crisp spatial object o. The *set containment between MBRs* $S_{sc}(A, B)$ yields true if $(MBR_{A_k} \nsubseteq (MBR_{B_k} \cup MBR_{B_c})) \vee (MBR_{A_k} \cap MBR_{B_k} = \varnothing \wedge MBR_{A_k} \cap MBR_{B_c} = \varnothing)$, and false otherwise. By using this definition, we are able to return *false* for the predicates of vague inside and vague coveredBy if $S_{sc}(A, B) = true$ holds. Otherwise, the respective predicate is evaluated. Similarly, if $S_{sc}(B, A) = true$ holds, then we can return *false* for the predicates of vague contains and vague covers, and evaluate the respective predicates otherwise.

6. Performance Evaluation

The advantages of our proposed solutions (i.e., the VagueGeometry ADT and the MBRVP improvement) were analyzed through experimental tests that process spatial queries with vague topological predicates. We analyze topological predicates since they incur high costs of processing and they are very common in spatial applications.

(a) (b)

Figure 2. Examples of the situations where the MBRVP improvement should be applied. Figure 2a: disjointness between MBRs. Figure 2b: set containment between MBRs. Black regions represent the kernel, while gray regions represent the conjecture.

6.1. Experimental Setup

We used a data set composed of 100,000 vague regions synthetically generated as follows. First, we constructed a Voronoi diagram of 200,000 crisp points randomly generated, which produced the same number of crisp regions. Second, for each crisp region, we added more points to increase its complexity. As a result, each crisp region was formed by averagely 313 points. Third, we created pairs of crisp regions that were disjoint or adjacent. Each pair was created by selecting randomly a crisp region and then by selecting the nearest crisp region that was disjoint or adjacent to the first one. We randomly assigned a crisp region as the kernel and the other crisp region as the conjecture. After creating a pair, we discarded the used regions, such that these regions were not used to create another pair. In the total, we generated 100,000 pairs of crisp regions.

We considered the following topological predicates: *vague disjoint*, *vague overlap*, *vague inside*, *vague intersects*, *vague coveredBy*, and *vague meets*. The workload was composed of 100 spatial range queries for each vague topological predicate. We also defined a query window for each spatial range query, which was composed of a vague region object that had the rectangular format for the kernel and the conjecture. Therefore, we randomly generated 100 different query windows.

We defined the following configurations: (i) *baseline* that used current functionalities provided by the PostgreSQL with the PostGIS spatial extension; (ii) *VagueGeometry* that used the proposed VagueGeometry ADT without improvements; and (iii) *VagueGeometry+* that used VagueGeometry empowered with the proposed MBRVP improvement. For the *baseline* configuration, we implemented vague topological predicates by using the Procedural Language/PostgreSQL (PL/pgSQL), which had "TRUE", "FALSE", or "MAYBE" as possible return textual values. For their use, we stored the kernel and the conjecture of each vague spatial object in separated columns in a relational table.

Note that we did not employ the approaches surveyed in Section 2 in the performance comparisons due to the following limitations. While the approaches proposed in [Zinn et al. 2007], [Kraipeerapun 2004], and [Dilo et al. 2006] *do not provide support for vague topological predicates*, the approaches described in [Pauly and Schneider 2008] and [Pauly and Schneider 2010] are *specifically implemented in Oracle*, which has license restrictions. Further, we used PostgreSQL in the performance tests to isolate the effects of the DBMS, and thus providing a fair comparison.

Table 1 depicts the SQL templates of the spatial range queries used in the three configurations. Consider the template for the *baseline* configuration. *baselineTable* is a

Table 1. Templates of the SQL spatial range queries.

Configuration	SQL Template
baseline	SELECT id FROM baselineTable WHERE R = P(kernel_geo, conjecture_geo, QW_k, QW_c)
VagueGeometry *VagueGeometry+*	SELECT id FROM vaguegeom WHERE O P(vg, QW)

table composed of three attributes: (i) *id* that is the primary key, (ii) *kernel_geo* that represents the kernel of the vague region objects, and (iii) *conjecture_geo* that represents the conjecture of a vague region object. Further, R is a textual return value that may contain "TRUE", "FALSE", or "MAYBE", P is the vague topological predicate, and QW is the query window. Regarding the *VagueGeometry* and the *VagueGeometry+* configurations, *vaguegeom* is a table that stored vague region objects in the attribute *vg* by using our proposed VagueGeometry. In addition, O corresponds to the use of the SQL operators introduced in Section 4.3. This means that the operator $\sim\sim$ was used to specify that P returned maybe, the operator $\&$ was used to specify that P returned true, and the combination of the operator $\&$ with the operator ! (i.e., $\&!$) was used to specify that P returned false. Note that the SQL templates are equivalent, i.e., they generate the same result, but using the specific functionalities provided by the corresponding configurations.

The experiments were conducted on a computer with an Intel® Core™ i7-4770 processor with frequency of 3.40GHz, 2 TB SATA hard drive with 7200 RPM, and 32 GB of main memory. The operating system was CentOS 6.5 with Kernel Version 2.6.32-431.el6.x86 64. We employed PostgreSQL 9.3.3, PostGIS 2.2.0, and GEOS 3.4.2.

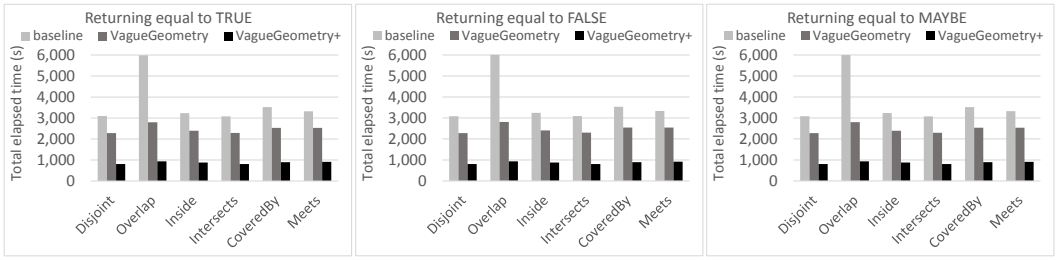
We collected the elapsed time in seconds. In detail, we executed 100 spatial range queries for each vague topological predicate and each value of return. Further, we executed each spatial range query 10 times and calculated its average elapsed time. Furthermore, we performed the tests locally to avoid network latency and we flushed the system cache after the execution of each query.

6.2. Performance Results

Figure 3 depicts the performance results. For each configuration and each return value of the three-valued logic (i.e., true, false, and maybe), we gathered similar elapsed times for processing the spatial queries. This means that the performance results showed a similar complexity for each return value.

Clearly, the performance of the *VagueGeometry* configuration overcame the *baseline* configuration. In fact, the internal structures of the VagueGeometry were more efficient than the definition of vague topological predicates by using current functionalities of the spatial DBMS. The performance gain imposed by the *VagueGeometry* configuration over the *baseline* configuration ranged from 23% up to 53%, where the performance gain is calculated as the percentage that determines how much more efficient one configuration was than another configuration.

Despite the expressive performance gains obtained by the *VagueGeometry* config-



(a) (b) (c)

Figure 3. Performance results for each vague topological predicate considering the returning values of true (a), false (b), and maybe (c).

uration, we gathered yet better results with the improvement proposed in Section 5. The *VagueGeometry+* configuration leaded to a performance gain that ranged from 72% up to 84% if compared with the *baseline* configuration. Further, compared to the *VagueGeometry* configuration, the *VagueGeometry+* configuration provided a performance gain that ranged from 63% up to 66%. This means that the use of the MBRVP improvement drastically reduced the necessity of processing crisp 9-intersection matrices in vague topological predicates.

Regarding storage space, the *baseline* configuration required 961 MB, the *VagueGeometry* configuration required 960 MB, and the *VagueGeometry+* configuration required 963 MB. We can conclude that the storage cost were almost the same. In addition, the storage of the MBRs of the kernel and the conjecture of each VagueGeometry object in the *VagueGeometry+* configuration did not introduce overhead in the execution of the spatial queries.

7. Conclusions and Future Work

In this paper, we proposed VagueGeometry, a novel abstract data type to handle vague spatial objects in the PostgreSQL with the PostGIS spatial extension. VagueGeometry empowers the management of spatial applications by offering textual and binary representations for vague spatial objects and by providing an expressive set of spatial operations, including vague geometric set operations, vague topological predicates, and vague numerical operations. As facilities, VagueGeometry introduces SQL operators for manipulating results of vague topological predicates and vague numerical operations. We also introduced MBRVP, an improvement to VagueGeometry to speed up the performance of spatial queries to process vague topological predicates.

Comparisons of VagueGeometry with current functionalities available on PostgreSQL showed that VagueGeometry provided better performance results for spatial queries with vague topological predicates. The performance gain of VagueGeometry varied from 23% up to 53%. Empowered with MBRVP, VagueGeometry provided even better results, which varied from 72% up to 84%.

Future work will deal with the extension of VagueGeometry to allow the use of index structures. Another future work refers to the development of specialized spatial join algorithms for vague spatial objects by using index structures.

Acknowledgments

The authors have been supported by the Brazilian research agencies FAPESP, CAPES, and CNPq.

References

Bejaoui, L., Pinet, F., Schneider, M., and Bédard, Y. (2010). OCL for formal modelling of topological constraints involving regions with broad boundaries. *GeoInformatica*, 14(3):353–378.

Carniel, A. C., Schneider, M., Ciferri, R. R., and Ciferri, C. D. A. (2014). Modeling fuzzy topological predicates for fuzzy regions. In *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, pages 529–532, New York, NY, USA.

Clementini, E. and Di Felice, P. (1997). Approximate topological relations. *International Journal of Approximate Reasoning*, 16(2):173–204.

Dilo, A., Bos, P., Kraipeerapun, P., and de By, R. A. (2006). Storage and manipulation of vague spatial objects using existing GIS functionality. In Bordogna, G. and Psaila, G., editors, *Flexible Databases Supporting Imprecision and Uncertainty*, volume 203, pages 293–321. Springer Berlin Heidelberg.

Dilo, A., de By, R. A., and Stein, A. (2007). A system of types and operators for handling vague spatial objects. *International Journal of Geographical Information Science*, 21(4):397–426.

Kraipeerapun, P. (2004). Implementation of vague spatial objects. Master's thesis, International Institute for Geo-Information Science and Earth Observation.

Li, R., Bhanu, B., Ravishankar, C., Kurth, M., and Ni, J. (2007). Uncertain spatial data handling: Modeling, indexing and query. *Computers & Geosciences*, 33(1):42–61.

Pauly, A. and Schneider, M. (2008). *Quality Aspects in Spatial Data Mining*, chapter Querying vague spatial objects in databases with VASA, pages 3–14. CRC Press, USA.

Pauly, A. and Schneider, M. (2010). VASA: An algebra for vague spatial data in databases. *Information Systems*, 35(1):111–138.

Schneider, M. and Behr, T. (2006). Topological relationships between complex spatial objects. *ACM Transactions on Database Systems*, 31(1):39–81.

Siqueira, T. L., Ciferri, C. D. A., Times, V. C., and Ciferri, R. R. (2014). Modeling vague spatial data warehouses using the VSCube conceptual model. *Geoinformatica*, 18(2):313–356.

Zinn, D., Bosch, J., and Gertz, M. (2007). Modeling and querying vague spatial objects using shapelets. In *Proceedings of the International Conference on Very Large Data Bases*, pages 567–578, Vienna, Austria.

3-D Reconstruction Of Digital Outcrop Model Based On Multiple View Images And Terrestrial Laser Scanning

Reginaldo Macedônio da Silva[1,2],Maurício Roberto Veronez[1,2], Luiz Gonzaga Jr[1,3], Francisco M. W. Tognoli[1,2], Marcelo Kehl de Souza[1,2], Leonardo Campos Inocencio[1]

[1]VIZLab, Advanced Visualization Laboratory, UNISINOS - São Leopoldo, RS, Brazil
[2]PPGEO, Graduate Program on Geology, UNISINOS - São Leopoldo, RS, Brazil
[3]PIPCA, Applied Computer Science Graduate Program - UNISINOS, São Leopoldo, RS, Brazil

Abstract

This paper presents a comparative study about 3D reconstruction based on active and passive sensors, mainly LiDAR – Terrestrial Laser Scanner (TLS) and raster images (photography), respectively. An accuracy analysis has been performed in the positioning of outcrop point clouds obtained by both techniques. To make the comparison feasible, datasets are composed by point clouds generated from multiple images in different poses using a consumer digital camera and directly by terrestrial laser scanner. After preprocessing stages to obtain these point clouds, both are compared, through the positional discrepancies and standard deviation. A preliminary analysis has been shown the feasible employment of digital image jointly 3D reconstruction method for digital outcrop modeling, concerning with data acquisition at low cost without significantly lost of accuracy when compared with LiDAR.

Key words: *LiDAR, 3D Reconstruction, Digital Outcrop Model, Structure From Motion, Terrestrial Laser Scanner (TLS), Multiple View Geometry, Digital Image.*

1. INTRODUCTION

The increasing advances in new technologies have emerged a couple of unexplored new opportunities in the field of technologies applied to geosciences. Thus it is required to test and evaluate the best path to use these technologies. Nowadays in geology, we have efficient tools to obtain three-dimensional (3D) data that include color and intensity, allowing accurate measurements layers thickness for inaccessible places, for example, outcrops. Three-dimensional digital models, especially those are obtained from terrestrial laser scanner and more recently from multiples digital images have been intensively employed.

One technique that has quickly evolved is georeferenced geological information by the System GNSS (Global Navigation Satellite System). This system has allowed more efficient, both in accuracy and in time gain, integration of the different products in a single geological reference system, ensuring greater reliability in the processes of generation of three-dimensional geological models (Pringle *et al*, 2004; Thurmond *et al*, 2005; White & Jones, 2008).

The use of digital mapping technologies have grown in the last ten years, in particular the use of terrestrial laser scanner and topography equipment's, integrated systems with satellite navigation and geographic information (Xu *et al*., 2001; Alfarhan *et al*., 2008), thus replacing numerous photographic mosaics routinely used in the interpretation of large outcrops.

Terrestrial laser scanners are able to capture few hundreds of millions of georeferenced points. This device, to define three-dimensional coordinates of points on a surface, emits laser pulses with the aid of a scanning mirror. When a pulse hits an object a portion of the energy returns back to the equipment. The distance between the sensor and the object is measured based on the time lag between emission and return of the pulse. The calculation of the coordinates of each point, obtained by the laser scanner, is possible from a point with known coordinates in the source pulse. Thus, the study of outcrops gets a new impulse by the ability to quantify the data estimated or ignored due to lack of access.

The use of LiDAR technology, especially terrestrial laser scanner, in studies of outcrops is expanding due to the ease of acquisition of precise, fast and automated georeferenced data.

This technology is being used for this purpose from a decade (Bellian *et al*., 2002), but only in recent years the number of scientific articles have increased significantly. However, the topics of interest are quite diverse, and include: methodological approaches (Bellian et al., 2005; Abellan et al. 2006; Enge et al., 2007; Buckley et al., 2008; Ferrari et al., 2012), reservoirs (Pringle et al., 2004; Phelps & Kerans, 2007; Kurtzman et al., 2009; Rotevatn et al., 2009; Enge & Howell, 2010; Fabuel-Perez et al, 2009,2010), fractured rocks (Bellian et al., 2007; Olariu et al., 2008; Jones et al., 2009; Zahm & Hennings, 2009), erosion rates (Wawrzyniec et al., 2007), synthetic seismic model (Janson et al., 2007), orientation of basaltic lava flows (Nelson et al., 2011) and classification of spectral patterns (Inocencio et al., 2014).

Photo-realistic 3D modeling has been a research topic, which addresses the quick generation of three-dimensional calibrated models using a hand-held device (Se & Jasiobedzki, 2006). This technique allows the creation of 3D models, both for visualization and measurements, based on multiple images. Several studies (Leung, 2006; and Aliaga *et al*., 2006) have used this photogrammetry technique for reconstruction of 3D models, and have analyzed the effects and methods for image-based modeling from multiple images (Szeliski, 2010). In geology, we aim this technique can be applied in analysis of outcrop in three dimensions in the laboratory at low cost when compared with LiDAR. Besides, it can to improve and facilitate virtual interpretations (Baltsavias *et al*, 2001; Enge *et al*, 2007).

Thus the aim of this study is to quantify, through control points, the error positional of outcrops mapped by image based modeling technique and by LiDAR, and to perform a comparison of the positional errors.

2. 3D RECONSTRUCTION MODEL FROM MULTIPLE IMAGES

Using multiples images (photographs), we can (re)construct three-dimensional models. It is the reverse process of obtaining photographs from 3D scenes. When a 3D scene is projected in 2Dcplane and depth is lost. A 3D point corresponding to a specific image point is constrained to be on the line of sight. From a single image, it is impossible to determine a point on the line of sight that corresponds to the image point. However if two images are available, then the position of a 3D point can be found at the intersection of the two projection rays. This process is called triangulation. Therefore, this process requires the multiple pass approach, that begins from camera calibration process to relate measuring range of the sensor with the real world quantity that it measures. It is necessary first to understand the mathematical model of a camera to calibrate it. For this purpose, we have adopted a *projective camera* model (pinhole camera), which has been widely adopted as camera model in computer vision, since it is simple and enough accurate enough for most applications.

The pinhole camera is illustrated in Figure 1 (a), while a slightly different model, where image plane is on front of the center of projection, is expressed in Figure 1 (b).

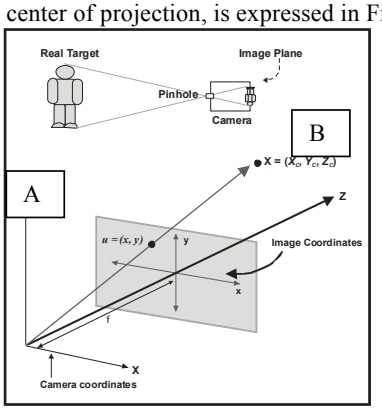


Figure 1 – Pinhole camera (A), Model (B)

To understand multi-view geometry, first of all we have considered relationship between two cameras (or sequentially moving one camera), which is actually called **epipolar geometry**. The epipolar geometry is the geometry of intersecting planes of images. Using the

common points between the images, along with the intersection of planes, it is possible to calculate the 3D position of objects in the scene.

We have show that there is a geometric relationship between corresponding points in two images of the same scene. This relationship depends only on the intrinsic parameters of the two cameras and their relative translation and rotation Figure 2.

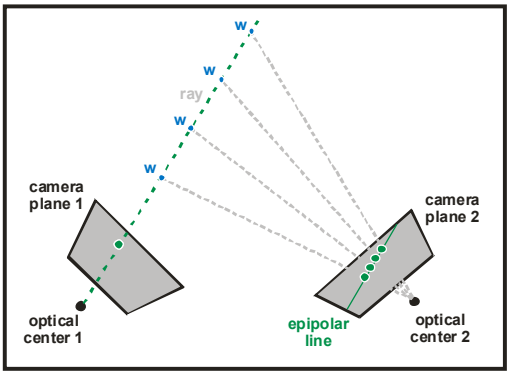


Figure 2 – Two cameras with epipolar constraints

Consider a single camera viewing a 3D point **w** in the world, passing through $\mathbf{x_1}$ and optical center $\mathbf{c_1}$. From one camera, it is impossible to identify the point in the ray. The projection of the ray in image plane 2 defines an *epipolar line*. Therefore, the point in the first image plane (Camera 1) corresponds to a constrained line in the second image plane (Camera 2). This relationship is called as *epipolar constraint*. The constraint on corresponding points is a function of the intrinsic and extrinsic parameters. If intrinsic parameters area given, then the extrinsic ones can be determined, and so the geometric relationship between the cameras. Another advantage is that, given the intrinsic and extrinsic parameters of the cameras, corresponding point of one image can be found easily through a 1D search along the *epipolar line* in the other image.

A mathematical model can capture the relationship between two cameras (two images) and can provide 3D point determination. In a general context, the mathematical constraint between the positions of corresponding points $\mathbf{x}_1$ and $\mathbf{x}_2$ in two normalized cameras can be obtained by *essential matrix* (note that either camera calibration or a different matrix – fundamental matrix – is required). Details about essential matrix can be obtained from any good computer vision literature (Harttley R. & Zisserman, 2004). This matrix can provide the above described parameters, mainly the camera matrices (*resectioning process)* and their parameters. Using a series of 3D-2D image plane correspondences it is possible to compute camera pose estimation. It utilizes camera parameters of the right camera that minimize the residual error of the 3D-point reprojections.

In another approach three or more cameras, instead of two can be considered. In three views, there are six measurements and so three degrees of freedom. However, it is for lines that there is the more significant gain. In two-views the number of measurements equals the number of degrees of freedom of the line in 3D-space, i.e., four. Consequently, there is no possibility of removing the effects of measurement errors. However, in three views there are six measurements on four degrees of freedom, therefore a scene line is over-determined and can be estimated by a suitable minimization over measurement errors.

For the computing purpose, we have implemented this sequence of concepts in an in-house computer vision library, using OpenCV (Brahmbhatt , 2013) – for computer vision and image processing support, Google Ceres-Solver library[1] – for modeling and solving large complicated nonlinear least squares problems and Eigen library[2] – a high-level C++ library of template headers for linear algebra, matrix and vector operations, and numerical solvers.

[1] https://code.google.com/p/ceres-solver/
[2] http://eigen.tuxfamily.org

3. MATERIALS AND METHODS

3.1 Materials

The study area is an outcrop of the Rio Bonito Formation, Lower Permian of the Paraná Basin, called Morro Papaléo and located at Mariana Pimentel, Rio Grande do Sul state, southern Brazil (
Figure 3), between the geodetic coordinates, latitudes 30°18'10"S and 30°18'40"S and longitudes 51°38'40"W and 51°38'30"W in the datum SIRGAS2000. The mentioned area is an abandoned quarry originally exploited for kaolin. It is a three-dimensional outcrop with a good exposure of rocks such as fossiliferous siltstone, carbonaceous siltstone, pebbly mudstone and sandstone.

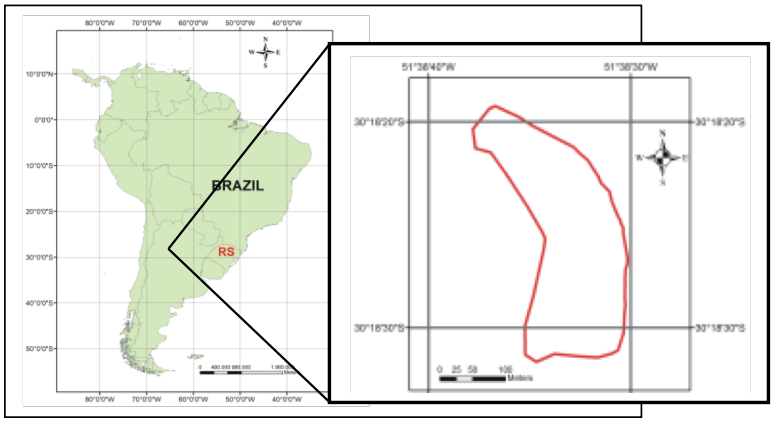


Figure 3 - Location map of the study area.

We implemented points serving as support for the georeferencing of the point clouds obtained by LiDAR as image-based modeling technique. The georeferenced points were tracked with Hyper-RTK GNSS equipment and were supported by geodetic point, implanted on top of the outcrop. These georeferenced points (P1 and P2, Table 1) were used as a support for measuring coordinates of points on the surface outcrop, which subsequently, were used to analyze the positional error. As a result tracking points (P1 and P2) were obtained in the system coordinates (Figure 3, & Table 1) UTM:

UTM COORDINATES			
POINTS	**E (m)**	**N (m)**	**Ellipsoidal height – h (m)**
P1	438125,808	6646812,115	136,775
P2	438135,602	6646873,338	137,468

Table 1 - Plane coordinates in UTM projection of the points of support for Surveying, Central Meridian at 51° SIRGAS2000 Reference System (Geocentric Reference System for the Americas).

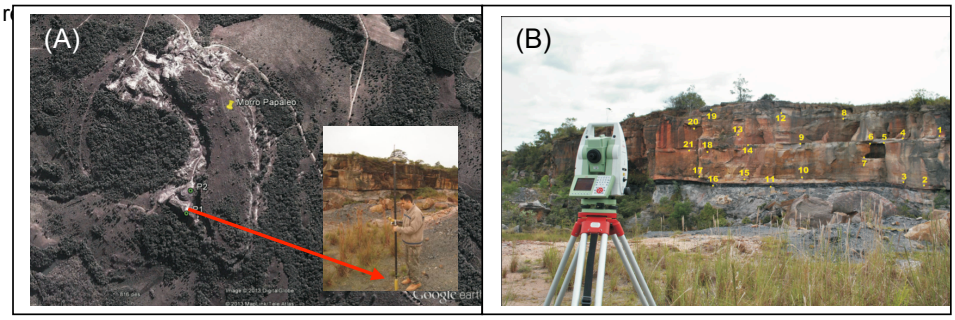


Figure 4 - Tracking of points P1 and P2 with the use of GNSS-RTK (A). Points of the surface outcrop measured with total station (B).

To obtain the coordinates on the surface of the outcrop we used a total station (Leica Viva TS15, Figure 4B), supported at points (P1 and P2) with GNSS-RTK. It was adopted as a criterion for selection of local points emphasizing on the contrast of colors and other well-defined characteristics. This facilitated the identification of the point cloud, both in terrestrial laser scanner and image-based modeling. With the total station, 21 points at the surface of the outcrop were measured, as illustrated in Figure 4B. These coordinates were used as parameters to determine the positional quality of the outcrop study.

For imaging the outcrop was used Leica Scanner Station C10, was used with resolution point cloud ranging between 2mm to 4cm.

The point cloud was processed to eliminate unnecessary information such as, vegetation and fallen rocks in front of the outcrop. In outcrop, sandstone predominates in *Morro Papaléo* and these rocks are in the point cloud in Figure 5.

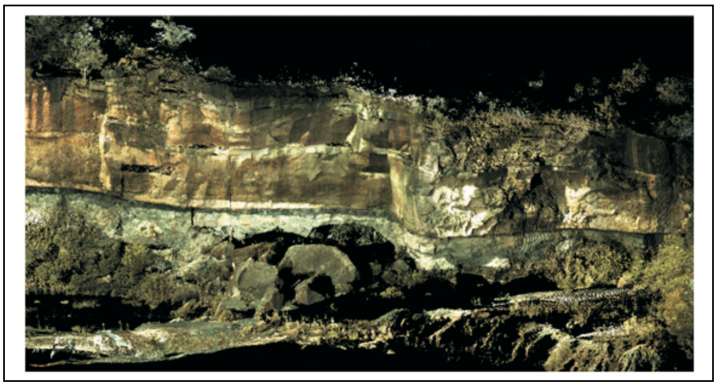


Figure 5 - Point cloud obtained with the terrestrial laser scanner

The same outcrop was photographed with a Nikon D3000 digital camera at a resolution of 7 Megapixels. The procedure for the collection of photos in the field was adopted to keep approximately the same distance between the camera and the outcrop (Figure 6). Another procedure was adopted to consider the top and bottom of the outcrop in the same photo. The photos were taken from different positions in order to obtain approximately 60% overlap between images.

Figure 6 – Pictures obtained with the camera (A). Positions of camera (B).

The processing of digital photos and reconstruction of the outcrop were based on image-base modeling technique. We have reconstructed the 3D outcrop and generated a cloud of points and georreference following the same procedures used in the generation of Digital Outcrop Model (DOM) obtained with the TLS.

4. RESULTS AND DISCUSSION

By comparing the results for the generation of DOMs, based on LiDAR technique and reconstruction of 3D objects from photos, it became evident that the image based modeling (Figure 7A) for photos allowed a visual resolution of better quality. However, the model generated by terrestrial laser scanner (Figure 7B) allowed control of spacing (resolution) of the points in the point cloud, whereas, there is no such control in image-based modeling from photos.

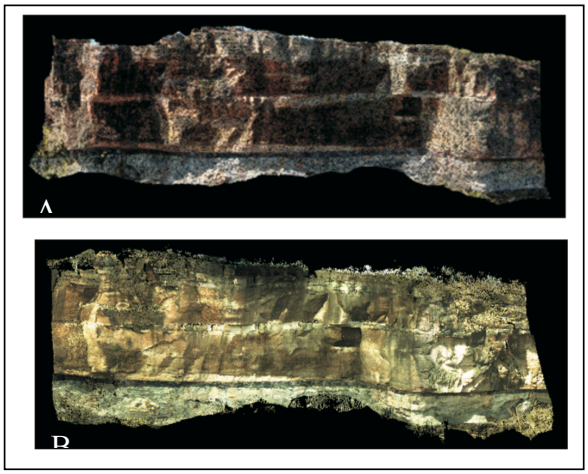


Figure 7 - 3D Reconstruction from photos (A) and terrestrial laser scanner (B).

Putting the image-based modeling and terrestrial laser scanner point cloud together, the Chi-Square test indicated 95% confidence level for georeferencing the differences were not significant between the control data and the techniques evaluated. In the comparison of the relative error models of the techniques used. It was observed that the difference became smaller than 5 cm as shown in Table 2.

Table 2 - Difference between linear measurements obtained from the models generated

Lines	Terrestrial Laser Scanner (meters)	Image-Based Modeling (meters)	Difference (meters)
1	1.6134	1.6375	-0.0241
2	2.3313	2.3451	-0.0138
3	1.8380	1.7960	0.0420
4	1.7010	1.6892	0.0118
5	2.8580	2.8669	-0.0089

5. CONCLUSION

The digital outcrop modeling technique can assist in outcrop interpretation, mainly for places that are hard to reach, due to large size and height of the outcropping or for security reasons. The paper results have shown the image-based modeling techniques can be feasible in this context of application instead of LiDAR, due the average linear error is under 40 cm. The cost of LiDAR equipment is much higher than a digital camera; hence the image-based modeling can provide good quality results at a lower cost,too.

The relative precision measurements performed from the point cloud obtained from image-based modeling had an error below 5 cm (Table 2) than that for the point cloud obtained from terrestrial laser scanner, which allows analysis of geological features for data modeling.

This study argue that image-based modeling techniques can to assist in getting the point cloud in places with occlusions by shading or obstructions around the object of the study, which is not possible using LiDAR technique.

The georeferencing of the point clouds from the image-based modeling technique allowed overlapping of point cloud from the LiDAR technique; it proves that the model generated from photos can be associated to a reference system. This, in turn, allows integration of other information obtained from other data sources.

6. REFERENCES

Alfarhan, M.; white, L.; Tuck, D.; Aiken, C. **Laser rangefinders and ArcGIS combined with three-dimensional photorealistic modeling for mapping outcrops in the Slick Hills, Oklahoma,** Geosphere, June 1, 2008; 4(3): 576 - 587.

Aliaga, D. G.; Zhang, J.; Boutin, M. Simplifying the Reconstruction of 3D Models using Parameter Elimination. Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007. 14-21 Oct. 2007. p.1-8.

Baltsavias, E. P.; Favey, E.; Bauder, A.; Bosh, H. And Pateraki, M.. Digital surface modelling by airborne *laser* scanning and digital photogrammetry for glacial monitoring. Photogrammetric Record, n. 17, p. 243-273, 2001.

Bellian, J. A.; Beck, R.; Kerans, C. **Analysis of hyperspectral and LiDAR data: Remote optical mineralogy and fracture identification,** Geosphere, December 1, 2007; 3(6): 491 - 500.

Bellian, J. A., Jennette, D. C., Kerans, C., Gibeaut, J., Andrews, J., Yssldyk, B., Larue, D., 2002, 3-Dimensional digital outcrop data collection and analysus using eye-safe laser (LiDAR) technology: American Association of Petroleum Geologists (AAPG). Search and Discovery Article 40056, (http://www.searchanddiscovery.net/documents/beg3d/index.htm).

Bellian, J. A.; Kerans, C.; Jennette, D. C. Digital outcrop models: applications of terrestrial scanning LiDAR technology in stratigraphic modeling. Journal of Sedimentary Research, n. 75, p.166–176. 2005.

Brahmbhatt S. Practical OpenCV. Apress. November 13, 2013

Buckley, S. J.; Howell, J. A.; Enge, H.D; Kurz, T. H. Terrestrial *Laser* Scanning in Geology: Data Acquisition Processing and Accuracy Considerations. Journal of the Geological Society, London; 2008, v. 165; ISSUE: 3, p. 625-638. DOI: 10.1144/0016-76492007-100.

Enge, H. D.; Buckley, S. J.; Rotevatn, A.; Howell, J. A. **From outcrop to reservoir simulation model: Workflow and procedures,** Geosphere, December 1, 2007; 3(6): 469 - 490.

Enge, H. D. & Howell, J. A. **Impact of deltaic clinothems on reservoir performance: Dynamic studies of reservoir analogs from the Ferron Sandstone Member and Panther Tongue, Utah.** AAPG Bulletin, February 1, 2010; 94(2): 139 - 161.

Fabuel-Perez, I., Hodgetts, D.; Redfern, J. **A new approach for outcrop characterization and geostatistical analysis of a low-sinuosity fluvial-dominated succession using digital outcrop models: Upper Triassic Oukaimeden Sandstone Formation, central High Atlas, Morocco,** AAPG Bulletin, June 1, 2009; 93(6): 795 - 827.

Fabuel-Perez, I.; Hodgetts, D.; Redfern, J. **Integration of digital outcrop models (DOMs) and high resolution sedimentology - workflow and implications for geological modelling: Oukaimeden Sandstone Formation, High Atlas (Morocco).** Petroleum Geoscience, May 1, 2010; 16(2): 133 - 154.

Harttley R. & Zisserman A. Multiple View Geometry in Computer Vision. Cambridge University Press, 2 edition, April 2004.

Janson, X.; Kerans, C.; Bellian, J. A.; Fitchen, W. **Three-dimensional geological and synthetic seismic model of Early Permian redeposited basinal carbonate deposits, Victorio Canyon, west Texas** AAPG Bulletin, October 1, 2007; 91(10): 1405 - 1436.

Jones, R. R.; Kokkalas, S.; Mccaffrey, K. J. W. **Quantitative analysis and visualization of nonplanar fault surfaces using terrestrial laser scanning (LiDAR)--The Arkitsa fault, central Greece, as a case study,** Geosphere, December 1, 2009; 5(6): 465 - 482.

Jones, R. R.; Mccaffrey, K. J. W.; Imber, J.; Wightman, R.; Smith, S. A. F.; Holdsworth, R. E.; Clegg, P.; Paola, N. de; Healy, D.; Wilson, R. W. **Calibration and validation of reservoir models: the importance of high resolution, quantitative outcrop analogues,** Geological Society, London, Special Publications, January 1, 2008; 309(1): 87 - 98.

Kurtzman, D.; El Azzi, J. A.; Lucia, F. J.; Bellian, J.; Zahm, C.; Janson, X. **Improving fractured carbonate-reservoir characterization with remote sensing of beds, fractures, and vugs,** Geosphere, April 1, 2009; 5(2): 126 - 139.

Leung, C. W. Y., Efficient methods for 3d reconstruction from multiple images 3D, 2006, 263p., Ph.D, Thesis, Scholl of Information Technology and Electrical Engineering, Depto of Engineering, University of Queensland, February 2006.

Nelson, C. E.; Jerram, D. A.; Hobbs, R. W.; Terrington, R.; Kessler, H. **Reconstructing flood basalt lava flows in three dimensions using terrestrial laser scanning,** Geosphere, February 1, 2011; 7(1): 87 - 96.

Olariu, M. I.; Ferguson, J. F.; Aiken, C. L. V.; Xu, X. **Outcrop fracture characterization using terrestrial laser scanners: Deep-water Jackfork sandstone at Big Rock Quarry, Arkansas,** Geosphere, February 1, 2008; 4(1): 247 - 259.

Phelps, R. M. & Kerans, C. **Architectural Characterization and Three-Dimensional Modeling of a Carbonate Channel Levee Complex: Permian San Andres Formation, Last Chance Canyon, New Mexico, U.S.A.** Journal of Sedimentary Research, November 1, 2007; 77(11): 939 - 964.

Pringle, J. K.; Westerman, A. R.; Clark, J. D.; Drinkwater, N. J.; Gardiner, A. R. 3D high-resolution digital models of outcrop analogue study sites to constrain reservoir model uncertainty: an example from Alport Castles, Derbyshire, UK. Petroleum Geoscience, 10, 343–352. 2004.

Rotevatn, A.; Buckley, S. J.; Howell, J. A.; Fossen, H. **Overlapping faults and their effect on fluid flow in different reservoir types: A LiDAR-based outcrop modeling and flow simulation study,** AAPG Bulletin, March 1, 2009; 93(3): 407 - 427.

Se, S. & Jasiobedzki P. **Photo-realistic 3D Model Reconstruction,** IEEE International Conference on Robotics and Automation, Orlando, Florida, USA., May 1, 2006; 3076 - 3082.

Szeliski R. Computer Vision: Algorithms and Applications (Texts in Computer Science). Springer, 2011 edition, October 2010.

Thurmond, J. B.; Drzewiecki, P. A.; Xu, X. Building simple multiscale visualizations of outcrop geology using virtual reality modeling language (VRML). Computers and Geosciences, 31, 913–919. 2005.

Wawrzyniec, T. F.; Mcfadden, L. D.; Ellwein, A.; Meyer, G.; Scuderi, L.; Mcauliffe, J.; Fawcett, P. **Chronotopographic analysis directly from point-cloud data: A method for detecting small, seasonal hillslope change, Black Mesa Escarpment, NE Arizona,** Geosphere, December 1, 2007; 3(6): 550 - 567.

White, P. D. & Jones, R. R. **A cost-efficient solution to true color terrestrial laser scanning,** Geosphere, June 1, 2008; 4(3): 564 - 575.

Zahm, C. K. & Hennings, P. H. **Complex fracture development related to stratigraphic architecture: Challenges for structural deformation prediction, Tensleep Sandstone at the Alcova anticline, Wyoming,** AAPG Bulletin, November 1, 2009; 93(11): 1427 - 1446.