Proceedings

Lúbia Vinhas and Clodoveu Davis Junior (Eds.)

Dados Internacionais de Catalogação na Publicação

SI57a Simpósio Brasileiro de Geoinformática (11. : 2011: Campos do Jordão,SP)

Anais do 12º Simpósio Brasileiro de Geoinformática, Campos do Jordão, SP, 27 a 29 de novembro de 2011. / editado por Lúbia Vinhas (INPE), Clodoveu A. Davis Junior (UFMG). – São José dos Campos, SP: MCT/INPE, 2011.
CD + On-line
ISSN 2179-4820

1. Geoinformação. 2.Bancos de dados espaciais. 3.Análise Espacial. 4. Sistemas de Informação Geográfica (SIG). 5.Dados espaço-temporais. I. Vinhas, L. II.Davis Junior, C.A. III. Título.

CDU: 681.3.06

Preface

This volume of proceedings contains papers presented at the XII Brazilian Symposium on Geoinformatics, GeoInfo 2011, held in Campos do Jordão, Brazil, November 28-29, 2011. The GeoInfo conference series, inaugurated in 1999, reached its twelfth edition in 2011. GeoInfo continues to consolidate itself as the most important reference of quality research on geoinformatics and related fields in Brazil.

GeoInfo 2011 brought together researchers and participants from several Brazilian states, and from abroad. Among the authors of the 17 accepted papers (out of 39 high-quality submissions), 13 distinct Brazilian academic institutions and research centers are represented, from all over the country. Most contributions have been presented as full papers, but both full and short papers are assigned the same time for oral presentation at the event. Short papers, which usually reflect ongoing work, receive a larger time share for questions and discussions.

The conference included special keynote presentations by Christian Freksa and Gilberto Câmara, who followed GeoInfo's tradition of attracting some of the most prominent researchers in the world to productively interact with our community, thus generating all sorts of interesting exchanges and discussions. Keynote speakers in past GeoInfo editions include 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 and Fosca Giannotti.

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

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

Lubia Vinhas
Program Committee Chair

Clodoveu A. Davis Jr.
General Chair

Conference Commitee

General Chair

Clodoveu Davis Junior
Federal University of Minas Gerais, UFMG

Program Chair

Lúbia Vinhas
National Institute for Space Research, INPE

Local Organization

Daniela Seki
INPE

Luciana Moreira
INPE

Janete da Cunha
INPE

Marcia Alvarenga
INPE

Support

SBC - Sociedade Brasileira de Computação

SELPER - Sociedade Latino Americana de Especialistas em Sensoriamento Remoto

Program committee

Ana Paula Afonso, Univ. Nova de Lisboa
Luis Otavio Alvares, UFSC
Pedro R. Andrade, INPE
Marcus V. A. Andrade, UFV
Silvana Amaral, INPE
Marcelo Azevedo, UFMG
Claudio S. Baptista, UFCG
Vania Bogorny, UFSC
Karla Borges, PRODABEL
Gilberto Câmara, INPE
Jorge Campos, UNIFACS
Tiago G. S. Carneiro, UFOP
Marcelo Tilio M. de Carvalho, PUC-Rio
Marco Antonio Casanova, PUC-Rio
Ricardo R. Ciferri, UFSCar
Julio C. L. D'Alge, INPE
Clodoveu Davis Jr., UFMG
Maria Isabel S. Escada, INPE
Guaraci Erthal, INPE
Sergio D. Faria, UFMG
Flávia F. Feitosa, INPE
Renato Fileto, UFSC
Frederico Fonseca, Pennsylvania State Univ.
Christopher B. Jones, Cardiff Univ.
Werner Kuhn, ifgi, Univ. of Muenster
Jugurta Lisboa Filho, UFV
Antônio Miguel V. Monteiro, INPE
Laercio Namikawa, INPE
Jussara Ortiz, INPE
Edzer Pebesma, ifgi, Univ. of Muenster
Raul Queiroz, PUC-Rio
Camilo D. Rennó, INPE
Armanda Rodrigues, Univ. Nova de Lisboa
Sergio Rosim, INPE
Mario J. G. da Silva, Inst. Superior Técnico de Lisboa
Marcelino P. S. Silva, UERN
Valeria Gonçalves Soares, UFPB
Valeria C. Times, UFPE
Ricardo S. Torres, UNICAMP
Stephan Winter, Univ. of Melbourne
Lúbia Vinhas, DPI/INPE

Contents

Building Geospatial Ontologies from Geographic Database Schemas in Peer Data Management Systems

Danúbia Lima[1], Antonio Mendonça[1], Ana Carolina Salgado[2], Damires Souza[1]
[1] Federal Institute of Education, Science and Technology of Paraiba, Brazil
[2] Federal University of Pernambuco, Brazil
{danubialima@gmail.com, tony2415@gmail.com, acs@cin.ufpe.br, damires@ifpb.edu.br}

Abstract. *One key issue in Peer Data Management Systems (PDMSs) is the heterogeneity of the peer schemas. To help matters, ontologies may be used as uniform conceptual representation of these schemas. In this work, we are working with geographic databases to be used in a PDMS. When dealing with geospatial data, specific problems with representation and usage occur. In this sense, we have developed an approach and a tool, named GeoMap, which builds a peer ontology from a geographic database schema. In order to provide geospatial semantics when mapping, we have defined and used a reference geospatial ontology. We present the principles underlying our approach and examples illustrating how they work by means of the tool.*

1. Introduction

Peer Data Management Systems (PDMSs) came into the focus of research as a natural extension to distributed databases in the peer-to-peer (P2P) setting [Lodi *et al*. 2008, Sung *et al*. 2005]. PDMSs are considered the result of blending the benefits of P2P networks, such as lack of a centralized control, with the richer semantics of a database [Zhao 2006]. They can be used for data exchanging, query answering and information sharing. For instance, in the areas of scientific research, the idea of setting up a PDMS to share research data among peers has already been widely discussed [Lodi *et al*. 2008, Zhao 2006].

A PDMS consists of a set of inter-related peers (data sources). Each peer has an associated schema within a domain of interest. However, PDMSs do not consider a single global schema. Instead, each peer represents an autonomous data source and exports either its entire data schema or a portion of it. Such schema, named exported schema, represents the data to be shared with the other peers of the system.

Data management in PDMSs is a challenging problem given the heterogeneity of their schemas. Due to the fact that ontologies provide good support for understanding the meaning of data, they have been used as an uniform metadata representation, i.e., each data source schema is represented by a local ontology (named *peer ontology*) [Souza *et al*. 2011, Xiao 2006]. In addition, due to semantic heterogeneity, research on PDMSs has also considered the use of ontologies as a way of providing a domain reference [Souza *et al*. 2011, Xiao 2006]. Considering a given knowledge domain, an agreement on its terminology can occur through the definition of a domain ontology which can be used as a semantic reference or background knowledge to enhance processes such as ontology matching and query answering.

One of the most representative realms of diversity of data representation is the geospatial domain. Geospatial data, besides hierarchical and descriptive components (relationships and attributes), are featured by other ones such as geometry, geospatial location and capability of holding spatial relationships (e.g., topological) [Hess 2008, Fonseca *et al.* 2003]. Furthermore, geospatial data are often described according to multiple perceptions, different terms and with different levels of detail. Syntactical aspects have been addressed by interoperability standards, such as Geography Markup Language [GML 2007]. However, the most hard-facing problem is still concerned with semantic heterogeneity.

In this work, we are working with geographic databases to be used in a PDMS called SPEED (Semantic PEEr-to-Peer Data Management System) [Pires 2009]. In order to uniformly deal with geospatial data without worrying about their specific heterogeneity restrictions (syntactic or semantic), we use ontologies as uniform conceptual representation of peer schemas. When a peer asks to enter the system, its schema (e.g., represented according to the relational or object-relational database model) must be automatically exported to a *peer ontology*. Due to the special semantics of geospatial data, this automatic extraction becomes more complex. Thus, in this work, we present an approach and an implemented tool, named *GeoMap*, for automatically building a geospatial peer ontology as a semantic view of data stored in a geographic database. During the ontology building process, a set of correspondences (relationships) between the generated ontology components and the original database schema is also automatically generated. The produced peer ontology will be later used for matching and querying processes in the PDMS. The set of correspondences will be used to translate ontological queries into the database query language (e.g., SQL) and retrieve corresponding instances from the geographic databases.

This paper is organized as follows: Section 2 introduces the SPEED system; Section 3 presents the *GeoMap* approach; Section 4 describes the developed *GeoMap tool* with some peer ontology generation examples. Related work is discussed in Section 5. Finally, Section 6 draws our conclusions and points out some future work.

2. The SPEED System

The SPEED system is a PDMS which works with three distinct types of peers, namely: data peers, integration peers, and semantic peers. A data peer represents a data source sharing structured or semi-structured data with other data peers in the system. Data peers are grouped within semantic clusters according to their semantic interest. A semantic interest includes the peer's interest theme and a local peer ontology. The interest theme is an abstract description of the peer's semantic domain, whereas the local peer ontology (LO) describes the peer's exported schema. Each semantic cluster has a special type of peer named integration peer. Actually, integration peers are data peers with higher availability, network bandwidth, processing power, and storage capacity. Such peers are responsible for tasks like managing data peers' metadata, query answering, and data integration. An integration peer maintains a cluster ontology (CLO), which is obtained through the merging of the local ontologies representing data peers' and integration peer's exported schemas. Integration peers communicate with a semantic peer, which is responsible for storing and offering a community ontology (CMO) containing elements of a particular knowledge domain (i.e., a domain ontology). Semantic peers are responsible for managing integration peers' metadata. A set of

clusters sharing semantically similar interests composes a semantic community. An overview of the SPEED architecture, with its kinds of peers and used ontologies is presented in Figure 1.

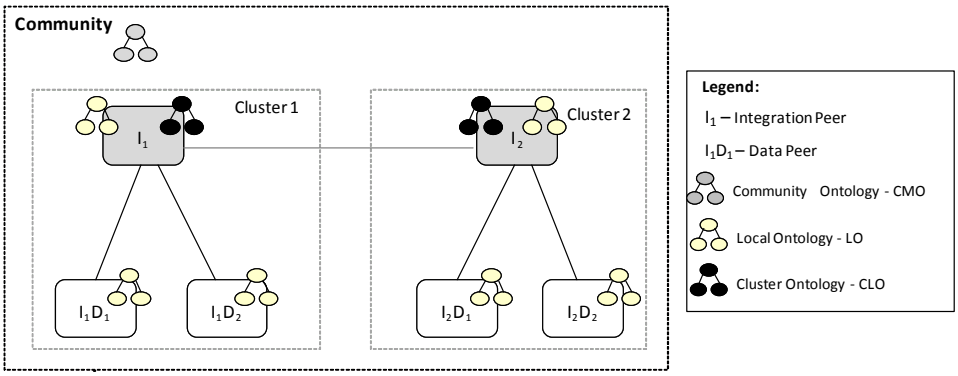


Figure 1: An Overview of SPEED Architecture

3. The *GeoMap* Approach

The database-to-ontology mapping approaches are usually classified into two main categories [Ghawi and Cullot 2009]: (i) approaches which create an ontology from a database and, (ii) approaches which map a database to an existing ontology. In the former, the objective is the creation of an ontology from a database and may include both the metadata and the data. The mappings, in this case, are the correspondences between each created ontology component (e.g., concept, property) and its original database schema concept (e.g., table, column). In the latter, the goal is to create a set of mappings between the existing ontology and the database. In our approach, we focus on the former, i.e., we build an ontology from a geographic database. Particularly, we build an ontology from the geospatial metadata. Nevertheless, the generated ontology does not contain data, i.e., the data remains in the database.

In our work, geospatial data are represented by means of the vector model. As a result, they are expressed as objects and are stored as points, lines or polygons, depending on the scale of their capture. Thus, the heterogeneity of data sources (databases) is even greater than in conventional databases: data may have multiple representations (the same data can be represented as a point in a given data source or as a polygon in another one), data may have different resolutions and different coordinate systems as well as temporal properties associated. In addition, since existing spatial database systems do not follow the same spatial data model (for instance, PostGIS [PostGis 2011] is based on the OGC specification, although Oracle is not [Oracle 2010]), there are differences when dealing with metadata from most of them. In this sense, the syntactic, semantic and spatial data format heterogeneity should be considered when creating an ontology from a geographic database.

On the other hand, an ontology is composed by concepts, properties (defined by means of domain and range information), axioms and, optionally, instances. Since an ontology is a knowledge representation technique based on Description Logics (DL) [Baader *et al*. 2003], it is usually coded using OWL (Web Ontology Language) model [Horrocks 2005]. As a result, there is a gap in terms of concept and relationship definitions between the ontology model and the database schema model. Regarding

geospatial data, there has been a lot of research looking for spatial ontology definitions [Hess *et al.* 2007, Arpinar *et al.* 2004] as well as for extensions to SPARQL language in the geospatial realm [Zhai *et al.* 2010]. Nevertheless, recently, there has been published an OGC candidate document which aims to specify a geographic query language for RDF data named GeoSPARQL [GeoSPARQL 2011]. The OGC GeoSPARQL standard will define a vocabulary for representing geospatial data in RDF as well as an extension to the SPARQL query language for processing geospatial data. We have defined specific constructs in OWL to deal with geospatial concepts and relationships, considering OWL as an extensible XML-format.

In the following, we introduce our approach by means of its main architecture. Then, we present a reference ontology which has been used to guide the mapping and the steps underlying our generation ontology process.

3.1 Architecture

Our approach, named *GeoMap*, is based on the architecture depicted in Figure 2. From a geographic database, a *peer ontology* (i.e., an application ontology) is built by means of the *GeoMap* components: at first, the database schema is extracted, then its elements are classified into spatial and non spatial ones, then its respective geospatial OWL construct is identified and, finally, the peer ontology is generated. This ontology represents, through ontological concepts and properties, the structure of the database. In order to provide semantics when accomplishing the OWL construct identification, we use a geospatial domain ontology (a reference ontology). During the generation process, an OWL document is also automatically produced to record the set of correspondences (relationships) between the generated ontology components and the original database metadata. This document will later be used to translate ontological queries into the database query language and retrieve corresponding instances.

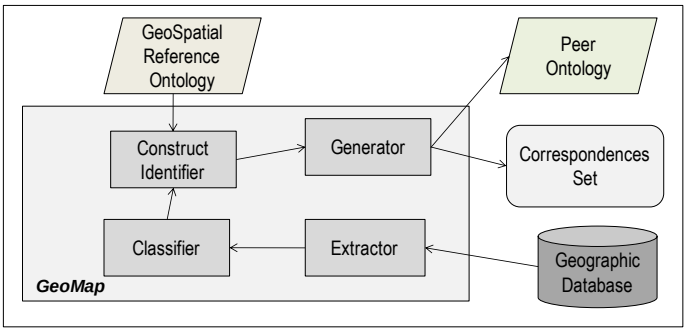


Figura 2. *GeoMap* Architecture

3.2 Reference Ontology

Using OWL as a common representation model, the SPEED system minimizes the problems of heterogeneity by transforming the schemas of data sources using ontologies. Nevertheless, regarding geospatial metadata, the set of predefined constructs in OWL does not include specifications for the description of geographical concepts and properties. Therefore, it was necessary to use some kind of background knowledge which could be a reference at the mapping process, when generating such geospatial constructs (tags). Although existing geospatial ontologies are available, we could not find out one which completely fitted our purposes. Thus, we have defined a geospatial domain ontology to be used as a reference in our process.

The reference ontology has been developed using the Protégé 3.4.4 tool [Protégé 2011]. An excerpt from this ontology is depicted in Figure 3, using the OntoViz notation [OntoViz 2011]. In this format, subtypes are associated with their supertypes through the *isa* relationship. Relationships between concepts are also presented through edges.

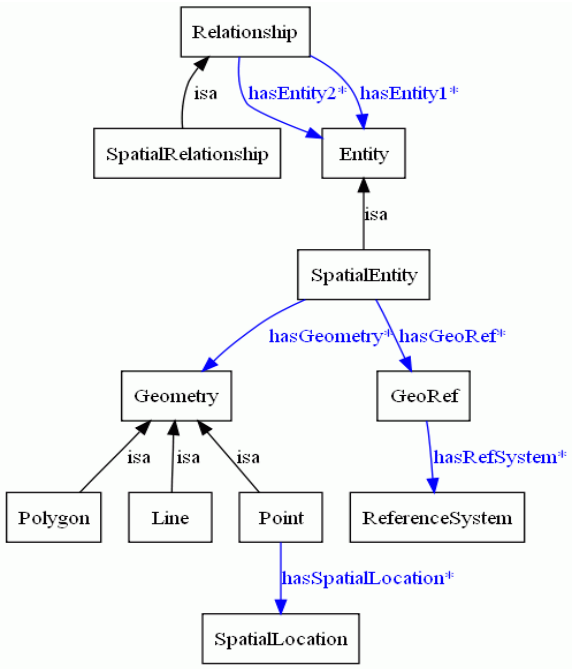


Figure 3. Excerpt from the Geospatial Reference Ontology

The reference ontology should be able to represent an abstraction of geospatial and non geospatial metadata, including, for example, entities, spatial entities, relationships, spatial relationships and geometry. In order to represent such concepts, we have defined specific high level concepts: Entity as the main concept; SpatialEntity as a specialization of Entity representing a geographic phenomenon; Relationship as a general kind of association between entities and SpatialRelationship as a specialization of Relationship regarding specific geospatial ones. A SpatialEntity has a Geometry (point, line or polygon) and is associated with a geospatial reference (a Reference System). Points are represented through spatial location, lines are defined through points and polygons are defined through lines.

3.3 Mapping Process

The mapping process used in *GeoMap* approach is based on the particular aspects described in the previous sections. An overview of its steps is presented in Figure 4. During this process, rules are applied to transform the geospatial schema elements from the database into geospatial ontology components. These rules include: (i) after connecting to the database, it extracts the database schema; (ii) it identifies, among the obtained tables, which are non-spatial, i.e., tables that have no geographic columns; (iii) meanwhile, it also identifies the spatial ones. Among the obtained tables, (iv) it identifies simple type properties (e.g., varchar and number types) and (v) it identifies

relationships which are mapped into object type properties, with the specification of domain and range, (vi) it also classifies these relationships as spatial or nonspatial ones. After processing all these steps, the tool produces an OWL file (i.e., the peer ontology) representing the geographic database schema.

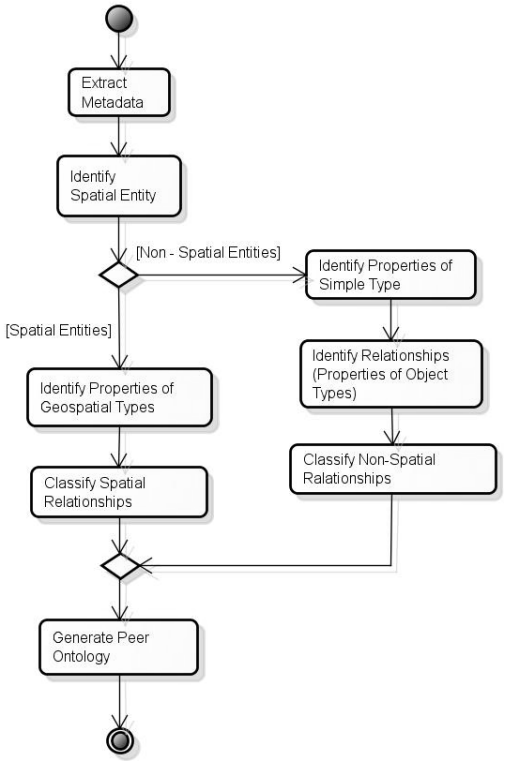


Figure 4 . Peer Ontology Generation Process

In next section, we provide some implementation issues for *GeoMap* and present the tool underlying our approach through an example.

4. The *GeoMap* Tool: Experiments and Results

The *GeoMap* tool has been implemented in JAVA as an extension of an object-relational database ontology generation tool [Franco 2009]. In this current version, *GeoMap* uses geographic databases coded in Oracle DBMS [Oracle 2010]. The Protégé-OWL API [Protégé 2011] and the Jena framework [Jena 2011] have been used for ontology manipulation.

In Figure 5, we present a use case diagram which shows the functional requirements that have been considered in the *GeoMap* tool implementation. There are two actors in the diagram. The first one is the *GeoMap* tool itself that starts the whole process of ontology mapping by connecting to the database. The database, in turn, characterizes the second actor. It is worth mentioning that, in this current version, the mapping options are initiated by a "user", i.e., manually by a *GeoMap* user. In a future version, the tool will be set in SPEED system as a service to be called whenever a peer (with geographic database) requests entering the system.

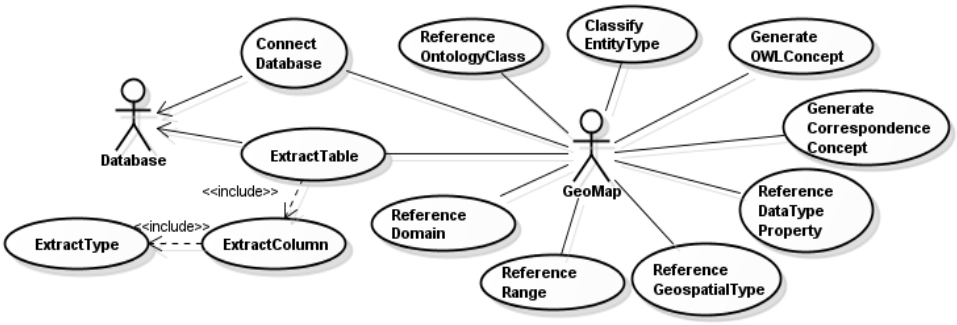


Figure 5. Use Case Diagram for the *GeoMap* Tool

After connecting to the database, *GeoMap* retrieves the existing geospatial types and tables from the database schema. Through this recovery, it is possible to extract all the columns, identifying what is a simple attribute and what represents a geometry. Then, the tool identifies the geometry type of geospatial tables - whether it is point, line or polygon. Based on the recovery of all entities and their properties (simple or objects), the tool makes use of the domain ontology as a reference of terms and creates the specific geospatial tags (i.e., constructs). Thus, when creating the spatial entities representation and referring the types of geometry, we use the reference domain ontology. When referencing the domain, we identify the domain of properties. When referencing the geographical range for a column, we specify the geometry types (line, polygon or point) present in the reference ontology.

During the mapping of these metadata, it is also possible to identify the equivalence correspondences of the generated ontology components and the existing database schema entities and properties. In order to define this set of equivalence correspondences, we build an OWL document composed by a specific construct named *IsEquivalentTo*. Such construct has been defined and used to indicate which ontology concept is equivalent to the database schema element. Also, it indicates which ontology properties are equivalent to the database schema attributes and relationships. An excerpt from a produced set of correspondences is depicted in Figure 6. In this example, we are working with a database regarding districts in São Paulo city. In the owl file, for instance, the SPATIAL_DATA column in the database schema corresponds to the DITRITOSSP_SPATIAL_DATA concept in the peer ontology, i.e., they are equivalent.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:j.0="d:\base2.owl#" >
  <rdf:Description rdf:about="http://localhost:1521#DENOMINACAO">
    <j.0:isEquivalentTo>d:\base2.owl#DISTRITOSSP_DENOMINACAO</j.0:isEquivalentTo>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:1521#BAIRRO">
    <j.0:isEquivalentTo>d:\base2.owl#BAIRROSSP_BAIRRO</j.0:isEquivalentTo>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:1521#DISTR">
    <j.0:isEquivalentTo>d:\base2.owl#BAIRROSSP_DISTR</j.0:isEquivalentTo>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:1521#CLASSE">
    <j.0:isEquivalentTo>d:\base2.owl#DRENAGEMSP_CLASSE</j.0:isEquivalentTo>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:1521#BAIRROSSP">
    <j.0:isEquivalentTo>d:\base2.owl#BAIRROSSP</j.0:isEquivalentTo>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:1521#SPATIAL_DATA">
    <j.0:isEquivalentTo>d:\base2.owl#DISTRITOSSP_SPATIAL_DATA</j.0:isEquivalentTo>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:1521#SIGLA">
    <j.0:isEquivalentTo>d:\base2.owl#DISTRITOSSP_SIGLA</j.0:isEquivalentTo>
  </rdf:Description>
```

Figure 6. Excerpt from an Obtained Set of Correspondences

At end, the *GeoMap* tool produces two outputs: (i) the peer geospatial ontology and (ii) the owl document with the set of equivalence correspondences. As a way to present *GeoMap* tool main steps execution, we provide some examples in the following.

4.1 *GeoMap* in Practice

Figure 7 shows a screenshot of the tool's main window that is split into four parts: (i) area which identifies the database name, (ii) area showing the structure obtained from the database schema, (iii) area with the *generated peer ontology* and (iv) area with the set of produced correspondences. In this example, we use a geographic database that stores attributes and geometries from the neighborhoods, districts and drainage map of São Paulo city. For instance, the database table named "DistritoSP" is represented as a polygon through the Oracle type MDSYS.SDO_GEOMETRY. Its structure is presented in an expanded view also shown in Figure 7.

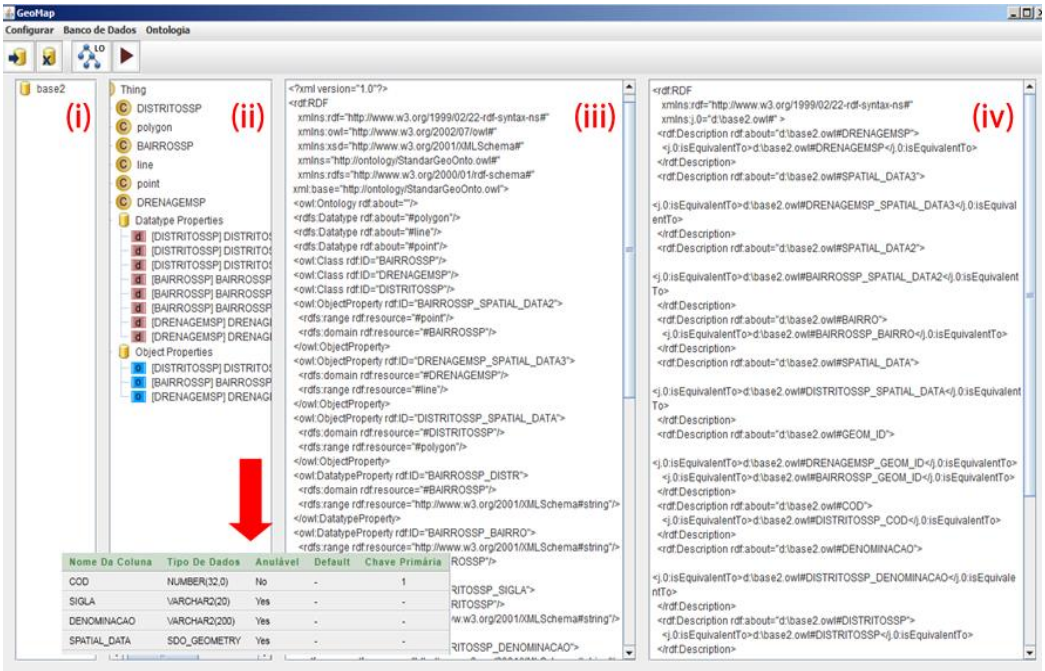


Figure 7. *Geomap* Interface and an Example of Database Schema

Particularly, another example regards using a geographic database that stores *laboratories* from IFPB Institute. According to the process explained in Section 3, the tool identifies spatial and non-spatial entities and properties from the database schema and generates the peer ontology and the set of equivalence correspondences. For the sake of visibility, we present the database schemas for the tables *Laboratórios*, *Corredor* and *TV* apart from the interface (Figure 8a). In addition, we depict the generated peer ontology from such database and the set of obtained correspondences in Figure 8b.

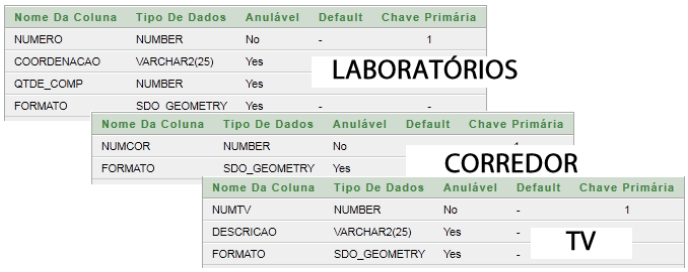
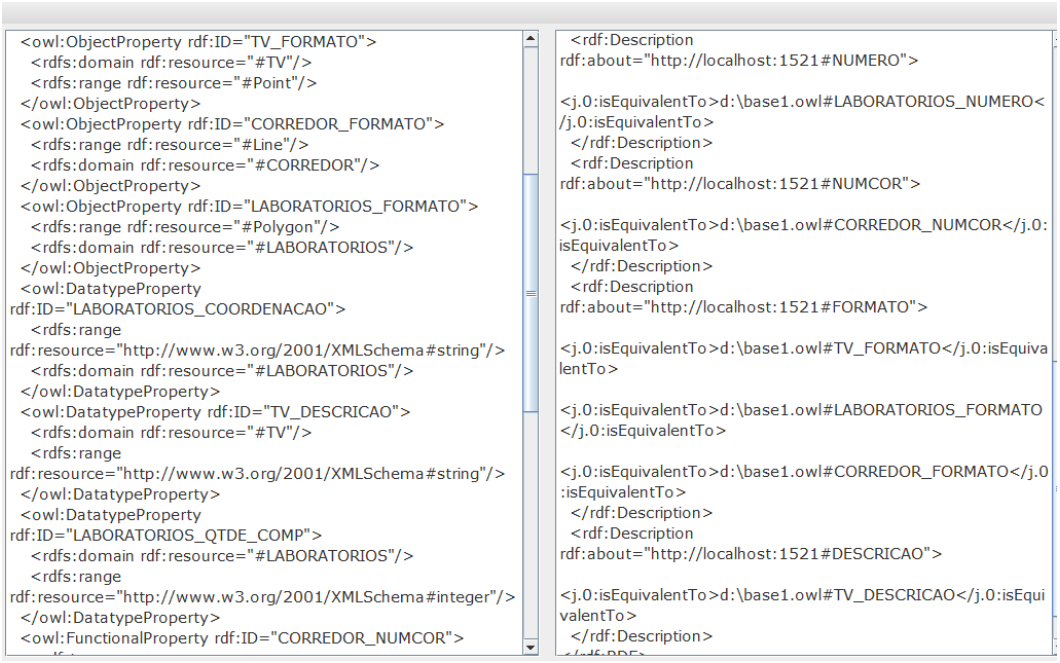


Figure 8a. Laboratories Database Schema



```
<owl:ObjectProperty rdf:ID="TV_FORMATO">
  <rdfs:domain rdf:resource="#TV"/>
  <rdfs:range rdf:resource="#Point"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="CORREDOR_FORMATO">
  <rdfs:range rdf:resource="#Line"/>
  <rdfs:domain rdf:resource="#CORREDOR"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="LABORATORIOS_FORMATO">
  <rdfs:range rdf:resource="#Polygon"/>
  <rdfs:domain rdf:resource="#LABORATORIOS"/>
</owl:ObjectProperty>
<owl:DatatypeProperty
rdf:ID="LABORATORIOS_COORDENACAO">
  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#LABORATORIOS"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="TV_DESCRICAO">
  <rdfs:domain rdf:resource="#TV"/>
  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty
rdf:ID="LABORATORIOS_QTDE_COMP">
  <rdfs:domain rdf:resource="#LABORATORIOS"/>
  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
</owl:DatatypeProperty>
<owl:FunctionalProperty rdf:ID="CORREDOR_NUMCOR">
```

```
<rdf:Description
rdf:about="http://localhost:1521#NUMERO">

<j.0:isEquivalentTo>d:\base1.owl#LABORATORIOS_NUMERO<
/j.0:isEquivalentTo>
</rdf:Description>
<rdf:Description
rdf:about="http://localhost:1521#NUMCOR">

<j.0:isEquivalentTo>d:\base1.owl#CORREDOR_NUMCOR</j.0:
isEquivalentTo>
</rdf:Description>
<rdf:Description
rdf:about="http://localhost:1521#FORMATO">

<j.0:isEquivalentTo>d:\base1.owl#TV_FORMATO</j.0:isEquiva
lentTo>

<j.0:isEquivalentTo>d:\base1.owl#LABORATORIOS_FORMATO
</j.0:isEquivalentTo>

<j.0:isEquivalentTo>d:\base1.owl#CORREDOR_FORMATO</j.0
:isEquivalentTo>
</rdf:Description>
<rdf:Description
rdf:about="http://localhost:1521#DESCRICAO">

<j.0:isEquivalentTo>d:\base1.owl#TV_DESCRICAO</j.0:isEqui
valentTo>
</rdf:Description>
```

Figure 8b. Produced Peer Ontology and Correspondences

We have accomplished some initial experiments with the *GeoMap* tool. The goal of our experiments is to check if we can assess *completeness* in terms of the produced peer ontology. According to some quality information criteria [Batista and Salgado 2007, Wang and Strong 1996], regarding PDMS systems, we have defined *completeness* as the degree to which entities and properties of the peer data source (i.e. the database schema) are not missing in the generated peer ontology. In order to measure such criterion, we have invited some users (knowledgeable about the Geospatial domain and OWL/RDF constructs) to produce a manual peer ontology from the geographic database schemas. These "gold ontologies" were compared with our produced peer ontologies. As result, we could verify that our produced peer ontologies are quite complete (ninety percent on average) in terms of the existing database elements, i.e., they include most of all the schema elements from the database. The different components obtained from the "gold" ontologies and ours regarded semantic interpretations when defining the geometry types: the expert users defined point, line and polygon as owl:class while our tool is still generating them as rdf:datatype. In fact, it indicates a probable mistake that will be corrected. Furthermore, we intend to

accomplish additional experiments with other experts and other databases in order to obtain a more concrete result.

5. Related Work

Currently, there are many approaches and tools which build ontologies from databases [Franco 2009, Cerbah 2008, Cullot *et al.* 2007, Baglioni *et al.* 2007]. However, most of them are concerned with relational databases. As an example, the DB2OWL tool map relational databases to OWL ontologies, considering particular table cases during the mapping process [Cullot *et al.* 2007]. Another example regarding relational databases is the RDBToOnto tool [Cerbah 2008]. This tool produces an ontology and allows refining its generated version. To this end, RDBToOnto provides a visual interface for accomplishing manual changes. Lubyte and Tessaris define a framework for extracting from a relational database an ontology that is to be used as a conceptual view over the data [Lubyte and Tessaris 2007]. In this work, the semantic mapping between the database schema and the ontology is captured by associating a view over the source data to each element of the ontology (i.e., by means of a GAV approach) [Halevy 2001]. Regarding object-relational databases, the work of Franco [2009] implements a tool which provides the ontology built from such kind of database.

Particularly, in the geospatial realm, Cruz *et al.* [10] developed a semi-automatic method to generate mappings between ontologies of local databases and a global one. The generated mappings are then used for query rewriting. In a closer scope, Baglioni *et al.* [2007] defined a method to access spatial database through an ontology layer. To this end, they developed a semi-automatic tool which builds an application ontology from a geographical database. They also enrich the generated ontology with semantics from a domain ontology by finding correspondences between the classes and properties of the two ontologies. This work is the most similar to ours.

Comparing these works with ours, we are able to produce the peer ontology in an automatic way, by using the semantics provided by the reference ontology. In this light, we can use any background knowledge that may support the geospatial semantics we need. Thus, for instance, we will be able to use the OGC GeoSparql standard (when it is ready for use) as our domain reference. Furthermore, in our approach, we do not need the user intervention, since this current tool will be stated as a service in SPEED system which will be dynamically executed at peer arriving time.

6. Conclusions and Future Work

This work has presented the *GeoMap* approach and tool. *GeoMap* accomplishes the extraction of metadata from a geographic database representing them in terms of a peer ontology. Also, it identifies the equivalence correspondences between the generated ontology components and the existing database schema entities and properties. To this end, it takes into account the identification and classification of geospatial and non spatial entities and properties, figuring out how they can be represented in terms of an OWL ontology. A geospatial reference ontology is used as a way to provide the semantics of geospatial relationships and types, absent from the set of existing concepts in the OWL model.

Currently, this version generates ontologies from Oracle databases. As future work, it will be extended to also extract metadata from other DBMSs, such as PostGIS

and other data sources such as GML. Another important future work concerns identifying correspondences between geospatial peer ontologies. Thereby, we will be able to reformulate and execute geospatial queries among the existing peers in the PDMS.

References

Arpinar I. B., Sheth A., Ramakrishnan C., Usery E. L., Azami M. and Kwan M. "Geospatial Ontology Development and Semantic Analytics". In: Handbook of Geographic Information Science. Blackwell Publishing (2004).

Baader F., Horrocks I., and Sattler U. (2007) "Description Logics". In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, Handbook of Knowledge Representation. Elsevier, 2007.

Baglioni M., Masserotti M., Renso C., Spinsanti L. (2007) "Building Geospatial Ontologies from Geographical Databases". GeoS 2007: 195-209.

Batista M. C., Salgado A.C. (2007) "Data Integration Schema Analysis: An Approach with Information Quality". In Proceedings of the 12th International Conference on Information Quality (ICIQ), MIT, Massachusetts, USA, October 2007.

Cerbah, F. (2008) "Learning Highly Structured Semantic Repositories from Relational Databases – The RDBtoOnto Tool". In: 5th European Semantic Web Conference (ESWC'08), pages 777-781, Tenerife, Spain (2008).

Cruz, I. F., Sunna, W., Chaudhry, A. (2004) "Semi-automatic ontology alignment for geospatial data integration". In: Egenhofer, M.J., Freksa, C., Miller, H.J. (eds.) GIScience 2004. LNCS, vol. 3234, pp. 51–66. Springer, Heidelberg (2004).

Cullot N., Ghawi R., and Yétongnon, K. (2007) "DB2OWL: A Tool for Automatic Database-to-Ontology Mapping". In: 15th Italian Symposium on Advanced Database Systems (SEBD'07), pages 491-494, Torre Canne di Fasano (BR), Italy (2007).

Fonseca, F. T., Davis Jr. C. A., Camara G. (2003) "Bridging Ontologies and Conceptual Schemas in Geographic Applications Development". *Geoinformatica*, v. 7, p. 355-378, 2003.

Franco, F. (2009) "Uma Ferramenta de Transformação de Esquemas Objeto-Relacionais para Ontologias". Trabalho de Graduação, Centro de Informática, Universidade Federal de Pernambuco (UFPE), Recife, PE, Brasil.

Geography Markup Language (2007). Available at: http://www.opengeospatial.org/standards/gml. Last access on august 15th, 2011.

GeoSPARQL documentation (2011). Available at: http://www.opengeospatial.org/projects/groups/geosparqlswg. Last access on august 15th, 2011.

Ghawi, R., and Cullot, N. "Building Ontologies from Multiple Information Sources". In 15th Conference on Information and Software Technologies (IT2009) (Kaunas, Lithuania, April 2009).

Halevy A. (2001) "Answering queries using views: A survey". VLDB Journal, 10(4):270{294, 2001.

Hess G. (2008) “Towards effective geographic ontology semantic similarity assessment”. PhD Thesis. UFRGS.

Hess G., Iochpe C., Castano S. “Towards a Geographic Ontology Reference Model for Matching Purposes”. GeoInfo 2007: 35-47.

Horrocks I. (2005) “Applications of description logics: State of the art and research challenges”. Proc. of the 13th Int. Conf. on Conceptual Structures (ICCS'05) April 2005.

Jena (2011). Jena - A Semantic Web Framework for Java. http://jena.sourceforge.net/

Lodi, S., Penzo, W., Mandreoli, F., Martoglia, R., and Sassatelli, S. “Semantic Peer, Here are the Neighbors You Want!” In: 11th Extending Database Technology (EDBT’08), Nantes, France, pp. 26-37 (2008).

Lubyte L., Tessaris S. (2007) “Extracting ontologies from relational databases”. In Proc. of the 20th Int. Workshop on Description Logics (DL’07).

OntoViz (2011). Available at: http://protegewiki.stanford.edu/wiki/OntoViz. Last access on august 18th, 2011.

OpenGIS (2011). Available at: http://www.opengeospatial.org/standards. Last access on august 15th, 2011.

Oracle 10g documentation (2010). Available at: http://www.oracle.com/technetwork/indexes/documentation/index.html. Last access on august 15th, 2011.

OWL API (2011) – available at: http://protege.stanford.edu/plugins/owl/api/. Last access on august 15th, 2011.

Pires C.E.S. (2009) “Ontology-Based Clustering in a Peer Data Management System”. PhD thesis, Center for Informatics, UFPE, 2009.

PostGIS documentation (2011). Available at: http://postgis.refractions.net/documentation/. Last access on august 15th, 2011.

Souza D., Pires C. E., Kedad Z., Tedesco P. C., Salgado A. C. (2011) “A Semantic-based Approach for Data Management in a P2P System”. To be published in LNCS Transactions on Large-Scale Data- and Knowledge-Centered Systems, 2011.

Sung L. G. A., Ahmed N., Blanco R., Li H., Soliman M. A., Hadaller D. (2005) “A Survey of Data Management in Peer-to-Peer Systems”. School of Computer Science, University of Waterloo.

Wang R. Y., Strong D. M. (1996) “Beyond Accuracy: What Data Quality Means to Data Consumers”. Journal of Management Information Systems, Vol. 12, N. 4, pages 5-33, 1996.

Xiao H. (2006) “Query processing for heterogeneous data integration using ontologies”. PhD Thesis in Computer Science. University of Illinois at Chicago, 2006.

Zhai X., Huang L., and Xiao, Z. (2010) “Geo-spatial query based on extended SPARQL”. In Proceedings of Geoinformatics. 2010, 1-4.

Zhao J. (2006) “Schema Mediation and Query Processing in Peer Data Management Systems”. Master Thesis, University Of British Columbia, October, 2006.

Proceedings XII GEOINFO, November 27-29, 2011, Campos do Jordão, Brazil. p 13-26.

Core Concepts of Spatial Information: A First Selection

Werner Kuhn

Institute for Geoinformatics (ifgi)
University of Muenster (Germany)
kuhn@uni-muenster.de

Abstract

The work reported here explores the idea of identifying a small set of core concepts of spatial information. These concepts are chosen such that they are communicable to, and applicable by, scientists who are not specialists of spatial information. They help pose and answer questions about spatio-temporal patterns in domains that are not primarily spatial, such as biology, economics, or linguistics. This paper proposes a first selection of such concepts, with the purpose of initiating a discussion of their choice and characterization, rather than presenting a definitive catalog or novel insights on the concepts.

1 Introduction

Research and development concerned with spatial information have been trying for at least two decades to integrate spatial information into mainstream information technology as well as science and society at large. For example, the vision of OGC (the Open Geospatial Consortium) went from, originally, "The complete integration of geospatial data and geoprocessing resources into mainstream computing" to today's "Realization of the full societal, economic and scientific benefits of integrating electronic location resources into commercial and institutional processes worldwide." The expectation behind the many large-scale efforts to create Spatial Data Infrastructures, nationally and internationally, is that more of the purported 80% of decisions in society with a spatial component will eventually become informed by spatial information, thereby improving business, governance, and science.

Progress toward this goal has been modest and often happened outside or even despite the efforts of spatial information experts, rather than through them. Today's most popular sources of spatial information are imagery and map data from Google, Microsoft and other companies, who acquired and developed their

technology largely outside the communities of spatial information experts. The problem has been recognized (and in part dealt with) in terms of a need for simpler models and standards than those typically produced by experts. As a result, simply structured spatial data can now be accessed, and elementary analyses performed on them, using non-specialist data formats and services built around them.

Yet, for spatial information to become a cross-cutting enabler of knowledge and analysis, such bottom-up technological solutions alone are not sufficient. Their confusing letter soup of acronyms and their growing plethora of "standards du jour" do not encourage a broader understanding and communication. An effort at the conceptual level is needed, in order to present a coherent and intelligible view of spatial information to those who may not want to dive into the intricacies of standards and data structures. Spatial information is too valuable for society to leave it up to the specialists; but as specialists, we can do a better job explaining it and demonstrating its benefits.

Spatial information answers questions about *themes* in *space* and *time*. All its varieties result from treating these three components as fixed, controlled, or measured **(author?)** [15]. For example, information about objects is produced by fixing time, controlling theme (i.e., choosing the objects of interest), and measuring space. A unified treatment of time and space turns Sinton's structure into the geo-atom <x, Z, z(x)>, which links a point x in space-time to a property-value pair <Z, z(x)>, where z(x) is the value of property Z at x [10]. The geo-atom answers the questions *what is there?* (looking for z, given x) and *where is this?* (looking for x, given z), two questions whose dualism is characteristic for spatial information.

Spatial data as such are not spatial information, but generate it once humans interpret them. For example, *1-5-3 Yaesu* is spatial data and can be interpreted in the right context as the address of a post office in a ward of Tokyo; the same goes for *6p21.3,* the locus of a gene in a chromosome. Concepts are the mental mechanisms needed to interpret data. For example, the concept of location is needed to interpret addresses or gene loci, and the concept of value is needed to interpret copyright regulations for spatial data.

The overall goal of this work is to explain spatial information and its potential for science and society through a set of core concepts. These are concepts of spatial *information,* defined here as concepts used to answer questions about themes in space and time *as well as* being represented by data or services. They include *spatial concepts*, which serve to reason about space [1] and *information concepts*, in the sense of concepts *about* spatial information, which may be spatial or not. An example of the former is *location,* referring to space and being represented digitally. An example of the latter is *value*, which refers to spatial information, but is not spatial. An example of both is *resolution*, which is a spatial measure, but also describes spatial information. The co-existence of such content and meta concepts, and the consequent need to understand both, are characteristic for information sciences in general.

[1] A comprehensive overview with an emphasis on learning is http://www.teachspatial.org.

2 Spatial information in science and society

Why bother with spatial information at all? First of all, because some of the biggest societal and scientific challenges require a better understanding of, and better decisions about, the location and interaction of things in space and time. Consider climate change, biodiversity, financial systems, poverty, security, health, energy or water supply – spatial information is essential in addressing each of these global as well as many regional and local challenges. In fact, key notions in today's scientific and social debates on these challenges are essentially spatio-temporal - consider risk, sustainability, vulnerability, or resilience. Secondly, studying spaces at smaller scales (atoms and subatomic particles, molecules, crystals, biological cells) as well as larger ones (planets and galaxies) remains among the most fascinating - as well as most costly - scientific adventures. Thirdly, solutions to *non-spatial* problems often use spatial analyses in real or metaphorical spaces, the latter ranging from the data cubes used in data mining to the spatializations used for information retrieval or mnemonic techniques.

Dealing with these and other challenges requires approaches transcending those of single disciplines, even if these disciplines have themselves broad scopes, as geography or computer science do. Transdisciplinary research addresses challenges that span multiple disciplines and have direct social relevance. Its goal is to make progress in solving these problems, not just to gain knowledge. Its approaches often benefit from exploiting spatial information, either because the problems are inherently spatial or because space and time act as unifiers and organizers for all phenomena and our knowledge about them. For example, combining satellite imagery of the Amazon rain forest with ground sensor measurements and socio-economic models reveals deforestation patterns, whose presentation to farmers, decision makers, and the general public helps reduce the depletion of this planet's lungs [1]. Recent developments in the social sciences and humanities, referred to as a *spatial turn* [17], together with numerous technological advances (navigation systems, mobile computing, high-resolution satellite imagery, virtual globes, sensors, and crowd-sourced information) further amplify and exemplify the opportunities for spatial information in science and society.

To be able to bring spatial information to transdisciplinary work, scientists of any disciplines need to be supported in understanding and exploiting *spatiality* in their theories and models. This requires an explanation of spatial information in a theoretically sound and technically informed way, maximizing the scope of applications and minimizing technological jargon. The spaces to be considered are those where transdisciplinary challenges arise, i.e., primarily those of human experience in one to three dimensions plus time. They include geographic spaces (such as a neighborhood in a city or a river catchment), indoor spaces (such as a room or a hallway), body spaces (such as a human body or organ), tabletop spaces (such as a desktop or workbench), and images. Smaller and larger spaces (such as cells, atoms or galaxies) as well as higher-dimensional ones (such as those used in statistics or data mining) are typically understood

through mappings to these experiential spaces. Among them, geographic spaces are the ones with which we have the richest set of experiences, giving geographic information science a privileged status in dealing with spatial information.

3 Concepts of spatial information

Surprisingly, a comprehensive treatment of concepts of spatial information with a transdisciplinary scope does not yet exist. All explanations of spatial information have a technological bias toward Geographic Information Systems (GIS) or a disciplinary one toward geography, surveying, or computer science. Where they discuss core concepts at all, these are often limited to spatial concepts or even geometric ones. While there is no shortage of calls to improve this situation (for an early GIS-oriented example with a broad view, see [2]), comprehensive results are missing or only starting to emerge[1].

This lack of a conceptual consensus on spatial information across disciplines, spaces, and technologies may be both a reason for, and a result of, the fact that the "science behind the systems" [7] remains largely there - *behind* the systems. Two decades after it started to call itself a science, and despite significant accomplishments [9], efforts to present geographic information science *to outsiders* as an intellectual (rather than just technological) venture lack a coherent conceptual basis[2].

As a consequence, biologists, economists, or linguists interested in testing a hypothesis using spatial information have to dig into GIS text books and standards documents, even just to understand what spatial information and reasoning could possibly do for them. Those interested in smaller or larger spaces are even worse off, since no textbooks or standards address these across disciplines. Geographic information science, thus, has to ask itself where economics would be today if its core concepts were limited to either micro- or macro-economics and hidden in specialized textbooks and manuals for spreadsheets and accounting software.

Taking up this challenge, the work presented here is neither about technologies, nor about particular disciplines or domains. It attempts to cut across their boundaries, targeting a bigger role for spatial information in science and society. It strives for an explanation of *what is special about spatial* for those who are *not* specialists of spatial information.

4 A first selection of concepts

The discussion of the proposed nine core concepts of spatial information in this section begins with location and the dual pair of field and object information. It continues with the spatial concepts of network and process, and ends with the information concepts of resolution, accuracy, semantics, and value. Each

[2]Some examples (with a variety of goals) are http://spatial.ucsb.edu/, http://uspatial.umn.edu/, http://spatial.uni-muenster.de, http://lodum.de/

concept gets only briefly characterized, in order to initiate and guide discussions. Detailed explanations exceed the available space and will be provided in a revised and extended article [13].

4.1 Location

The starting point for a journey through concepts of spatial information has to be location: spatial information is always linked to location in some way - but what exactly is location and how does it play this central role? Location information answers *where* questions: where are you? where is the appendix? where are this morning's traffic jams? Perhaps counter-intuitively, location is a relation, not a property. This is so, because nothing has an intrinsic location, even if it remains always in the same place. The house you live in can be located, for example, by a place name, an address, directions, or various types of coordinates. All of these location descriptions express relations between the *figure* to be located (your house) and a chosen *ground* (a named region, a street network, coordinate axes). How one locates things, i.e., what ground and what relation one chooses, depends on the context in which the location information is produced and used. Spatial reference systems, for example the World Geodetic System 1984 (WGS84), standardize location relations and turn them into easier to handle attributes within such a system. Yet, when data use multiple reference systems (for example, latitude and longitude as well as projected coordinates), locations need to be understood as relations and interpreted with respect to their ground (for example, the Greenwich meridian or a projected meridian).

Relating different phenomena through location is fundamental to spatial analysis. An early and famous application is Dr. Snow's 1854 finding that Cholera is spread by drinking water, after he had observed that many Cholera deaths had taken place around a certain water pump. The great power of such locational analyses results from the fact that nearby things are more related than distant things, which has been dubbed Tobler's First Law, based on its first explicit statement in [16].

4.2 Field

Fields describe phenomena that have a value everywhere in a space of interest, such as temperature. Generalizing the field notion from physics, field-based spatial information can also represent values that are statistically constructed, such as probabilities or population densities. Field information answers the question *what is there?,* where "there" can be anywhere in the space of interest. Fields are one of two fundamental ways of structuring spatial information, the other being objects 4.3. Both fix time, with fields resulting from controlled space and measured theme, and objects resulting from controlled theme and measured space. Time can also be controlled, rather than fixed. Controlling it together with space leads to space-time fields; controlling it together with theme produces object animations. Fields have been shown to be more fundamental

than objects, capable of integrating field and object views in the form of General Field models [14].

Since it is not possible to represent the infinitely many values of a field, they need to be discretized for explicit digital storage. There are two ways to achieve this, either through a finite set of samples with interpolation between them or through a finite number of cells with homogeneous values, jointly covering the space of interest. The cells can all have the same shape (forming a regular grid of square, triangular, hexagonal, or cubic cells) as in the so-called *raster model* for spatial data, which is best known from digital images. Or they can have irregular shapes, adaptable to the variation of the field, as in the finite element models used in engineering or the triangulated irregular networks (TIN) used to represent terrains.

An important kind of fields captures values on two-dimensional surfaces such as those of the earth or of the human body. These fields are typically organized into thematic *layers*. The idea of a layer is rooted in traditional paper- or film-based representations of spatial information, such as maps, and the production of models from stacked transparent layers of data about a theme. The main computational use of layers is to *overlay* them in order to relate information about multiple themes or from multiple sources.

4.3 Object

After fields, objects provide the second fundamental way of structuring spatial information. They describe individual things with spatial, temporal, and thematic properties and relations. Object information answers questions about properties and relations of objects, such as *where is this?, how big is it?, what are its parts?*, *which are its neighbors?, how many are there?*.

For many applications, one is interested in things that are *features* in the way that a nose is a feature of a face, i.e. parts of a surface. Features are important siblings of objects, but can be understood as a special case. The simplest way to carve out features from a surface is to name regions on it. Geographic places are the prototypical examples, carved out of the earth's surface by naming regions; but the same idea applies, for example, to models of airplane wings, sails, or teeth. Object and feature models can co-exist, and the general tendency today is to complement two-dimensional feature models with three-dimensional object models and provide more or less seamless transitions between them. For example, your house may be represented as a feature of the earth's surface in a digital map, as a feature of street view images, and as a three-dimensional object. The resulting blended feature-object notion pervades geography, but also exists in biology and medicine (features of cells, organs, or bodies), and generally in imaging (features extracted from images of anything).

Many questions about objects and features can be answered based on simple representations as points with thematic attributes. For example, doing a blood count or determining the density of hospitals in an area require only point representations. On the other hand, some questions do require explicit *boundaries* enclosing or separating objects. For example, determining the neighbors of a

land parcel, the extent of a geological formation, or the health of blood cells may require boundary data. The frequent occurrence of boundaries in object information, however, has mainly historical reasons, since analog representations like images or maps were digitized by drawing or following lines on them.

The so-called *vector models* for spatial data capture objects with boundaries at various levels of sophistication. Like surface fields in raster data, collections of features in vector data can be organized into thematic *layers*. Processing vector data exploits the geometry of boundaries to compute sizes, shapes, buffers around the objects, and overlays. Yet, many objects, particularly natural ones, do not have crisp boundaries [3]. Examples are geographic regions or body parts such as the head. It can be harmful to impose boundaries on such objects only for the sake of storing them in vector models. Differences between spatial information from multiple sources are indeed often caused by such more or less arbitrary delimitations. For example, boundaries of climate zones are vague by nature, and the variation in boundaries between different definitions matters much less than the overall extent and location of the zones. Thus, whether modeling objects with explicit boundaries is necessary or even desirable has to be carefully assessed for each application. It is certainly not something that the concept of an object implies.

4.4 Network

Connectivity is central to space and spatial information. The concept of a network captures binary connections among arbitrary numbers of objects, which are called nodes or vertices of the network. The nodes can be connected by any relation of interest. Network information answers questions about connectivity, such as *are nodes m and n connected?*, *what is the shortest path from m to n?*, *how central is m in the network?*, *where are the sources and sinks in the network?*, *how fast will something spread through the network?*, and many others.

The two main kinds of networks encountered in spatial information are transportation and social networks. Transportation networks (in the widest sense) model systems of paths along which matter or energy is transported, such as roads, utilities, communication lines, synapses, blood vessels, or electric circuits. Social networks capture relationships between social agents, such as friendships, business relations, or treaties.

All networks can be spatially embedded, which means that their nodes are located. This is often the case for transportation networks and increasingly for social networks. If the embedding space is a surface, networks can be organized into thematic layers, like the surface fields and feature collections encountered above.

Network applications benefit from the well studied representations of networks as graphs and the correspondingly vast choice of algorithms. Partly due to this sound theoretical and computational basis, networks are the spatial concept that is most broadly recognized and applied across disciplines.

4.5 Process

Processes are of central interest to science and society - consider processes in the environment, in a human body and its cells, and in machines or molecules. Processes that manifest themselves in field, object, or network information are considered spatial. Information about spatial processes primarily answers questions about *motion*, *change*, and *causality*.

Controlling time and measuring space generates information about *motion*; controlling time and measuring theme informs about *change*. Time is typically controlled through time stamps in spatial information. Temporal reasoning on time stamps (and on time intervals formed from them) is the basis for understanding motion and change. Migration or embolism are examples of motion. Growth, such as that of vegetation or social networks, exemplifies the change of objects or networks. Diffusion, for example in the form of climate change, collapsing house prices, or spreading innovation, is an example of a change in fields, objects or networks.

The most complex relation between spatial processes is that of *causality*. Dr. Snow's tracing of cholera to drinking water is a case of determining that one process (drinking some water from a contaminated pump) is the cause of another (contracting cholera), based on the patients being located near the pump.

Real-time spatio-temporal data from sensors and spatio-temporal simulations are the two key sources of process information. In order to makes sense of these dynamic data and models, science needs better *theories of change* [4]. One of the main benefits to be expected from a list of core concepts of spatial information is indeed to establish the conceptual foundations for such theories. If the theories can be formulated in terms of the proposed core concepts, their choice will be corroborated; if not, other concepts will have to join or replace them. For the current proposal, this means that all spatial change needs to be explained in terms of operations on locations, fields, objects, networks, and processes themselves.

4.6 Resolution

Resolution is the first and most spatial *concept of information* on this list. It characterizes the size of the units about which information is reported and applies to all three components of space, time, and theme. For example, satellite images have the spatial resolution of the ground area corresponding to a pixel, the temporal resolution of the frequency at which they are taken, and the thematic resolution of the spectral bands pictured. Vote counts have the spatial resolution of voting districts, the temporal resolution of voting cycles, and the thematic resolution of parties or candidates. Resolution information answers questions about *how precise* spatial information is, for example, when taking decisions based on the information.

Resolution characterizes information about all concepts introduced so far: location is recorded at certain granularities, fields are recorded at certain sample spacings or cell sizes, and the choice of the types of objects (say, buildings

vs. cities) and nodes (say, transistors vs. people) determines the spatial resolution of object and network information. The choice of the spatial, temporal, and thematic resolution at which spatial information gets recorded is primarily determined by the processes studied, because these involve phenomena of certain sizes, frequencies, and levels of detail. For example, migration, social networking, and the diffusion of technological innovations all involve people over months; embolism involves blood clots and vessels over hours; cancer involves cells and organs over years; climate change involves large air and water masses over decades; changing house prices involve land parcels and people over days or weeks.

Many processes need to be studied at multiple resolutions (for example, erosion) or they connect to processes at other resolutions. For example, one can think of all processes as involving some sort of motion at some resolution. All five core spatial concepts on our list can be represented at multiple resolutions: location descriptions are often hierarchical (for example, addresses); fields are often represented by nested rasters (called pyramids in the case of images); object hierarchies are expressed as part-whole relations between objects (for example, administrative subdivisions of countries); hierarchical network representations allow for more efficient reasoning (for example, in navigation), process models (for example, in medicine) are connected across levels of detail.

4.7 Accuracy

Accuracy, like precision, is a key property of information, capturing how information relates to the world. Information about accuracy answers questions about the *correctness* of spatial information. The location of a building, given in the form of an address, coordinates, or driving instructions, can in each case be more or less accurate. The spatial, temporal, and thematic components of spatial information are all subject to (in)accuracy.

Assessing the accuracy of information requires two assumptions: that there is in principle a well-defined correct value and that repeated measurement or calculation distributes in regularly around it. The first assumption requires an unambiguous specification of the reported phenomenon and of the procedure to assign values. For example, if temperatures are reported for different places, one may need to specify the level above ground to which they refer. The second assumption requires an understanding of measurement as a random process. Choosing a particular form of distribution (called a probability density function) allows for estimating the probability that a measured or computed value falls within a given interval around the correct value. Mean errors and any other accuracy data are based on these two assumptions.

Accuracy connects to resolution through the established practice of reporting all data at a resolution corresponding to the level of expected accuracy. If information is collected at multiple levels of resolution, one level can sometimes be considered as accurate when assessing the others. For example, positions determined from high-precision measurements serve as “fix points” for lower precision measurements, and objects get extracted from remotely sensed images

by determining "ground truth" for parts of an image.

4.8 Semantics

Understanding the semantics of spatial information is crucial to its adequate use. When it comes to analyzing spatial information, determining whether the same things are called the same (and different things differently) is essential to producing meaningful results and making sense of them. The challenge is to capture what the producer means with some data or services and to guide the user on how to interpret them. For example, when navigation systems use road data, they make assumptions on what the data producer meant by "road width" (paved or drivable?, number of lanes or meters or feet?). When using a spatial information service, operational terms such as distance also have to be interpreted adequately.

Semantic information answers the question *how to interpret the terms* used in spatial information. It concerns the spatial, temporal, and thematic components. While the semantics of spatial and temporal data have long been standardized through spatial and temporal reference systems, the semantics of thematic data and operations remain hard to capture and communicate. What is meant with data about land use or body tissue, for example, depends on a complex interaction between defining the intended use of some terms (say, forest or muscle) and delineating the spatio-temporal extents of their application to land or tissue.

Data and services do not have a meaning by themselves, but are used to mean something by somebody in some context. Therefore, it is impossible to fix the meaning of terms in information. However, one can make at least some of the conditions for using and interpreting a term explicit. This is what *ontologies* do: they state constraints on the use of terms. But language use is flexible and does not always follow rules, even for technical terms. An empirical account of how some terms are actually used can therefore provide additional insights on intended meaning or actual interpretation. This is what *folksonomies* deliver: they list and group terms with which information resources have been tagged.

Semantic information consists of necessarily incomplete collections of constraints from ontologies and folksonomies on the use and interpretation of terms. The constraints can use binary logic (for example, stating that a term refers to a subset of the things that another term refers to) or fuzzy logic (where such a statement is neither true nor false, but possibly true). The latter is an attempt to account for the inherent vagueness of many terms.

Yet, all constraints depend on *context*. Terms are used by somebody to mean something in a given context. Ontologies and folksonomies capture some aspects of context, but spatial information is often used in other contexts than the ones it was produced in. For example, road width data produced by traffic engineers may be quite different from those needed for navigation. In order to map between different contexts, ontologies need to be *grounded*. This means that their constraints need to refer to something outside their context, to which the constraints of other contexts can then refer as well. Spatial information

has successfully relied on grounding for centuries, through spatial reference systems. These systems refer coordinates to something outside their conceptual framework, such as physical monuments or stars. Generalizing this idea from location information to any terms used in spatial information leads to the idea of semantic reference systems [12]. These systems, once established in practice, are expected to support translations of terms used in spatial data and services from one context to another. Grounding is the basis for analytical translations of terms from one context to another. Since all constraints on meaning are non-deterministic, stochastic approaches to translation are a valid alternative to explicit grounding. For example, translations of terms across natural languages are now routinely and successfully achieved through stochastic methods.

4.9 Value

The final core concept proposed is that of value. Information about the value of spatial information answers questions about the many *roles* spatial information plays in society. The main aspect of value is *economic*, but the valuation of spatial information as a good in society goes far beyond monetary considerations. It includes its relation to other important social goods, such as privacy, infrastructure maintenance, or cultural heritage.

Setting policies on public *access* to spatial information, for example, is a pressing societal need requiring a better understanding of the many valuations involved. It is further complicated by the fact that information about indoor and geographic spaces can now be and is being collected and shared by almost everybody. This phenomenon of crowd-sourced or Volunteered Geographic Information (VGI, [8]) is profoundly altering the value of spatial information, from economic as well as institutional, ethical, and legal perspectives. For example, a key new challenge created by VGI is to understand and model *trust* in spatial information.

Given these wide ranging aspects of spatial information value, no coherent theoretical framework for it can be expected any time soon. Partial theories of value, for instance about the economic value of spatial information, are still sketchy and difficult to apply, because they involve parameters that are hard to control or measure. The cost of spatial information is no good guide to its value either, because it often reflects the high expenses for collecting the information, rather than the value of the result.

Value of information tends to accrue holistically and unpredictably, by new questions that can be asked and answered, new services that are provided. Partly for this reason, spatial information holdings have become significant assets, not only for scientists and governments, but also for enterprises in all sectors. Such assets need to be evaluated, for example in enterprise valuation, reinforcing the need for theories of spatial information value.

Even at the level of personal information management, the value of accessing and analyzing information through its spatial and temporal properties has barely been understood and tapped into yet [11]. For example, searching information by where or when it was collected or stored is highly effective, but still

only weakly supported by the web, personal computers, and smart phones.

4.10 Also-ran

It may be useful to consider some arguments against core status for some other concepts. Obviously, these may have to be reconsidered, so that this list of also-rans is part of the material for discussion.

My earlier lists of concept candidates contained nearness, spatial relations, feature, map, layer, motion, path, uncertainty, and scale. Typical reasons to exclude them from the list were that they were too broad or too narrow. In particular:

- *nearness* got generalized to *spatial relations*, but these serve to specify location and are covered there;
- *features* are now treated together with objects;
- *maps* are visualizations of mostly geographic information that exists in other forms;
- *layers* structure the representations of several concepts (fields, objects, networks) and are dealt with there;
- *motion* is only one process in space, although the most important one;
- *paths* are covered as parts of networks;
- *uncertainty* covers several concepts, of which resolution, accuracy, and semantics are covered;
- *scale* is also a catch-all for several concepts, of which resolution is on the list, extent (of a study area) is rather trivial, and support is more specialized (belonging to measurement ontology).

5 Conclusions

Achieving a stronger role for spatial information in science and society requires explaining its uses and benefits at a higher level than that of technologies and acronyms. The small set of concepts of spatial information proposed in this paper indicates a possible basis for such explanations. While it may miss or misrepresent some concepts, it provides a starting point to reach a conceptual view of our field that is accessible and intelligible to outsiders. The main goal at the moment is, therefore, to receive critical feedback and suggestions of what to add, drop, or change.

The concepts chosen and revised based on the expected feedback will then be described in more detail over the coming year[3]. These descriptions will ask and answer four questions about each concept:

[3]To participate in the discussion, please visit http://ifgi.uni-muenster.de/services/ojs/index.php/ccsi/index

1. *what* is the concept, i.e., what phenomena does it capture?
2. *where* does information about the concept come from, i.e., what are typical sources of information about it?
3. *how* is the concept *represented*, i.e., what data structures and algorithms implement it?
4. *how* is information about the concept *used*, i.e., what reasoning and analyses does the concept support?

The expected result is a catalogue of core concepts that are meaningful and useful across disciplines - a vocabulary to talk about spatial information to non-specialists. Such vocabularies, when formalized, are referred to as ontologies. While formalization is not a primary goal here, treating the concepts as nodes in an ontology and relating them to an upper level ontology or embedding them in ontology patterns will certainly help to clarify them further. Starting this work as an ontology design exercise, however, would most likely not lead to a useful set of concepts, because their relation to actual data and computations would be too weak. A subsequent ontological analysis will produce an ontology of spatial information that allows for interfacing with other domains, while relating explicitly to information technology. As such, it will complement and benefit from existing ontologies of spatial information [6, 5].

Acknowledgments

Countless discussions over the years with many colleagues and friends have encouraged and influenced these thoughts. The members of http://musil.uni-muenster.de and the students of my *Introduction to Geographic Information Science* have been very helpful critics and supporters of this work. Some anonymous reviewers of GeoInfo2011 provided very useful comments.

References

[1] Ana Paula Dutra Aguiar, Gilberto Câmara, and Maria Isabel Sobral Escada. Spatial statistical analysis of land-use determinants in the Brazilian Amazonia: Exploring intra-regional heterogeneity. *Ecological Modelling*, 209(2-4):169–188, December 2007.

[2] Peter A Burrough and Andrew U Frank. Concepts and paradigms in spatial information: Are current geographic information systems truly generic? *International Journal of Geographical Information Science*, 9(2):101–116, 1994.

[3] Peter A Burrough and Andrew U Frank, editors. *Geographic Objects with Indeterminate Boundaries*. Taylor&Francis, December 1996.

[4] Gilberto Câmara, Lúbia Vinhas, Clodoveu Davis, Fred Fonseca, and Tiago Carneiro. Geographical Information Engineering in the 21st Century. In Gerhard Navratil, editor, *Research trends in geographic information science*, pages 203–218. Springer Verlag, 2009.

[5] Helen Couclelis. Ontologies of geographic information. *International Journal of Geographical Information Science*, 24(12):1785–1809, November 2010.

[6] Andrew U Frank. Ontology for spatio-temporal databases. Spatiotemporal Databases. In Timos Sellis, editor, *The Chorochronos Approach*, pages 9–77. 2003.

[7] Michael F Goodchild. Geographical information science. *International Journal of Geographical Information Science*, 6(1):31–45, 1992.

[8] Michael F Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, November 2007.

[9] Michael F Goodchild. Twenty years of progress: GIScience in 2010. *Journal of Spatial Information Science*, 1(1):3–20, July 2010.

[10] Michael F Goodchild, May Yuan, and Thomas Cova. Towards a general theory of geographic representation in GIS. *International Journal of Geographical Information Science*, 21(3):239–260, 2007.

[11] Krzysztof Janowicz. The Role of Space and Time For Knowledge Organization on the Semantic Web. *Semantic Web - Interoperability, Usability, Applicability*, 1(1-2):25–32, 2010.

[12] Werner Kuhn. Semantic Reference Systems. *International Journal of Geographic Information Science (Guest Editorial)*, 17(5):405–409, June 2003.

[13] Werner Kuhn. The sciences before the systems: towards a transdisciplinary role for spatial information. *International Journal of Geographical Information Science*, (under review):1–15, 2012.

[14] Y. Liu, Michael F Goodchild, Q. Guo, Y. Tian, and L. Wu. Towards a General Field model and its order in GIS. *International Journal of Geographical Information Science*, 22(6):623–643, 2008.

[15] David Sinton. The inherent structure of information as a constraint to analysis: Mapped thematic data as a case study. *Harvard papers on geographic information systems*, 6:1–17, 1978.

[16] Waldo Tobler. A computer movie simulating urban growth in the Detroit region. *Economic Geography*, 46:234–240, 1970.

[17] Barney Warf and Santa Arias, editors. *The spatial turn: Interdisciplinary perspectives*, volume 26. Taylor & Francis, November 2009.

Computing Polygon Similarity from Raster Signatures

Leo Antunes[1], Leonardo Guerreiro Azevedo[1,2]

[1] Graduate Program of Infomation Systems (PPGI)
Applied Informatics Department (DIA)
Federal University of State of Rio de Janeiro (UNIRIO)
Av. Pasteur, 458, Urca, 22290-240, Rio de Janeiro, Brazil

[2] Research and Practice Group in Information Technology (NP2Tec)
{leo.antunes,azevedo}@uniriotec.br

Abstract. *Computing similarity of spatial objects is not a trivial task. It considers complex algorithms, which have high cost to compute. This work proposes a simple algorithm to compute similarity between polygons through their Four-Color Raster Signatures (4CRS) based on Jaccard index. The algorithm was implemented in SECONDO, an extensible DBMS platform. Experimental tests were conduct in order to evaluate algorithm precision and execution time compared to computing polygon similarity through real representations of polygons. Results demonstrated that raster similarity computation is three times faster than exact computation, and raster similarity precision is higher for objects with high similarity and lower for objects that are not very similar. Therefore, we point the use of the proposal when the intention is to process objects that seem to have high similarity. On the other hand, other algorithm must be employed for objects with low similarity, e.g., compute similarity on objects real representation.*

1. Introduction

Quine (1969) and Cakmakov and Celakoska (2004) present the similarity concept as fundamental for learning, knowledge and thought. A similarity metric is a measure that allows comparison of pair of things. Examples of applications where similarity can be used are: medical image databases, human gesture/motion recognition, geologic/geographic information systems, e-commerce, trademark/copyright protection, computer-aided design (Sako and Fujimura, 2000).

Holt (2003) presents spatial similarity as a subset of similarity. It corresponds to a similarity where all the entities being compared to each other have spatial components.

Spatial data consist of points, lines, regions, rectangles, surfaces and volumes (Samet, 1990). Examples of spatial data are: cities, forests, rivers, land use, partition of a country into districts etc. Spatial data is in practice connected to "non-spatial" data (e.g. alphanumeric) (Güting, 1994). Examples of non-spatial data are: names of cities, names of streets, addresses, telephone number etc.

Spatial Database Management System (SDBMS) provides the technology for Geographic Information Systems (GIS) and other applications (Güting, 1994). An important issue in database systems is efficient query processing, and the user receive a query answer in a short time. However, there are many cases where it is not easy to

accomplish this requirement. Besides, there are situations where obtaining fast answers, albeit approximate, is more important to the user than exact ones. This work concerns data compression techniques, i.e., coding mechanisms to generate reduced (or compressed) data over which queries are executed. We are using spatial data signatures to code real data: the Four-Color Raster Signature proposed by Zimbrao and Souza (1998) used to represent polygons.

This work proposes an algorithm to compute similarity of polygons from their 4CRS signatures, named as raster similarity. It employs Jaccard index (Jaccard, 1912) based on the overlapping and common areas of polygons, approximately computed using their raster signatures. Algorithms were implemented in SECONDO, an extensible database that supports non-conventional data types, for example, spatial data (Güting *et al.*, 2005). Experimental tests were conducted on real data corresponding to polygons representing municipalities from north Region of Brazil. The tests were conducted to evaluate the precision of the algorithm and execution time.

This remainder of this work is divided as follows. Section 2 presents the main concepts used in this work. Section 3 presents the proposal, and related algorithms. Section 4 is dedicated to the implementation details and experimental tests, as well as corresponding analysis. Finally, Section 5 presents our conclusions.

2. Theoretical grounding

Approximate Query Processing arises as an alternative to query processing in envirnments for which providing an exact answer results in undesirable response times. The goal is to provide an estimated response in orders of magnitude less time than the time to compute an exact answer by avoiding or minimizing the number of accesses to the base data (Gibbons *et al.*, 1997). Some examples of approximate query processing are: (i) Decision Support Systems, to present aggregate data for decision makers in reasonable time (Hellerstein *et al.*, 1997) (ii) Ad-hoc data mining, during a drill-down query sequence, the earlier queries in the sequence can be used solely to determine what the interesting queries are (Hellerstein *et al.*, 1997). (iii) Spatial OLAP (Online Analytical Processing) (Papadias *et al.*, 2001) to provide fast access to precomputed and summarized data for queries over aggregated data. (iv) Query processing: to provide feedback on how well posed a query is, and even as a tentative answer to a query when the base data is unavailable (Gibbons *et al.*, 1997).

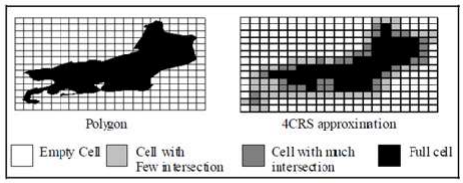
Four-Colour Raster Signature (4CRS) was proposed by Zimbrao and Souza (1998). It is a signature that stores polygon main features in an approximate and compressed representation. The signature can be accessed and processed faster than real data. It corresponds to a grid of cells (Figure 1.b) where each cell store relevant information of object using few bits (Figure 1.a). Grid scale can be adjusted in order to obtain a more compressed representation (lower scale) or a more precise representation (higher scale).

Scale change is used to ensure that signature cells of two 4CRS have same size and that the intersecting cells have the same corner coordinates. One approach to meet this requirement is cells' edge size be a power of two (2^n), and that the beginning of each cell be multiple of the same power of two ($a2^n$), as proposed by Zimbrao and Souza (1998). If this requirement is not accomplished, signatures cells may not overlap as

presented in Figure 2.a, and it is not possible to compare directly polygons signatures. Hence a better approach is perfect overlap of signature cells, as illustrated in Figure 2.b. It is important to emphasize that different signatures can have different cell size. Scale change is accomplished by grouping cells of the signature with smaller cell size, since it is not possible to subdivide a bigger cell to produce smaller ones.

Bit Value	Cell Type	Description
00	Empty	The cell does not intersect the polygon.
01	Weak	The cell contains as intersection of 50% or less with the polygon.
10	Strong	The cell contains an intersection of more than 50% with the polygon.
11	Full	The cell is fully occupied by the polygon.

(a)



(b)

Figure 1. (a) Types of Cell in the 4CRS (Zimbrao and Souza, 1998) and (b) an example of 4CRS (Azevedo *et al*., 2004)

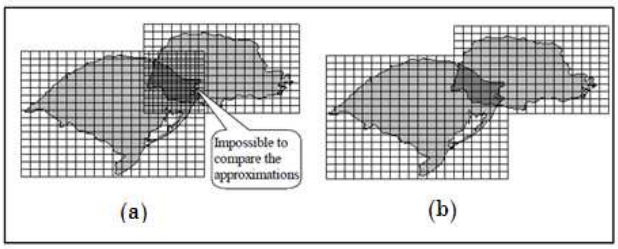


Figure 2. (a) Signatures whose cells do not overlap; (b) Perfect overlap

Zimbrao and Souza (1998) presented good results when 4CRS was used to approximate polygons in exact query processing using the Multi-step Processing of Spatial Joins architecture (Brinkhoff *et al.*, 1994). This motivates the use of 4CRS for approximate query processing, and a set of algorithms was proposed by Azevedo *et al*. (2004, 2005, 2006). These algorithms were evaluated against exact query processing and demonstrated also good results.

Approximate query processing using 4CRS corresponds to, instead of using as input the real object, use object's 4CRS signature, and return an approximate response, along with a confidence interval. As an example, the algorithm that computes the approximate area of a polygon *p* (Azevedo *et al.*, 2004) returns an area value *v*, and an interval *i* with confidence *c*. The response is that real area is between *v-i* and *v+i*, with confidence *c*.

3. Algorithms to Compute Raster Similarity

A similarity function, in an intuitive sense, returns the similarity of objects considering size, shape, and position in space. For instance, for spatial objects that have area, similarity can be computed as the ratio of intersection and union areas, as presented in Equation 1. This equation is an intuitive metric, and it is named as Jaccard index (Jaccard, 1912), as presented by Hemert and Baldock (2007).

$$S(o1,o2) = (A_{\cap}(o1,o2)) / (A_{\cup}(o1,o2)) \quad (1)$$

Where:

- o1 and o2: spatial objects that have area
- $A_{\cap}$: approximate overlapping area of polygons
- $A_{\cup}$: approximate union area of polygons

This work proposes to replace, in Equation 1, *o1* and *o2* by their 4CRS. The algorithm is presented in Figure 3. It has as input the 4CRS of two polygon, and returns a value between the interval [0, 1] that indicates polygons similarity. The algorithm employs other three algorithms: compute approximate area of polygon (Figure 4), compute intersection area of polygons (Figure 5), and compute 4CRS union (Figure 6).

```
REAL similar(signat4CRS1, signat4CRS2)
  intersArea = approxIntersectionArea(signat4CRS1, signat4CRS2);
  IF(intersArea = 0) /* Does not exist intersection area */
    RETURN 0;
  ELSE /* Exists intersection area */
    unionSignat = unionSignat4CRS(signat4CRS1, signat4CRS2);
    unionArea = approximateArea(unionSignat);
  RETURN intersArea / unionArea;
```

Figure 3. Algorithm to compute raster similarity of polygons

The algorithm to compute polygon approximate area (Azevedo *et al.*, 2004) (Figure 4) returns polygon area summing the expected area of the polygon inside each type of signature's cells. The expected areas for *Empty*, *Weak*, *Strong* and *Full* cells are 0%, 25%, 75% and 100%, respectively.

```
REAL approximateArea(signat4CRS)
  nWeakCells = nStrongCells = nFullCells = 0;
  cellArea = signat4CRS.edgeSize * signat4CRS.edgeSize;
  FOR EACH cell IN signat4CRS.cells DO
    IF (cell.type = WEAK)
      nWeakCells= nWeakCells+1;
    ELSE IF (cell.type = STRONG)
      nStrongCells = nStrongCells + 1;
    ELSE IF (cell.type == FULL)
      nFullCells = nFullCells + 1;
  RETURN (nWeakCells * weakWeight + nStrongCells * strongWeight
         + nFullCells * fullWeight) * cellArea;
```

Figure 4. Algorithm to compute approximate area of polygon

The algorithm to compute the approximate overlapping area of two polygons (Azevedo *et al.*, 2005) (Figure 5) sums the expected area of cell types that overlap, and multiplies this value by the cell area. There are four types of cell; hence there are sixteen possibilities of types of cells that overlap, as proposed by Azevedo *et al*. (2005).

```
REAL approxIntersectionArea(signat4CRS1, signat4CRS2)
  interMBR = intersectionMBR(signat4CRS1, signat4CRS2);
  IF (signat4CRS1.edgeSize = signat4CRS2.edgeSize) then
      s4CRS = signat4CRS1;
      b4CRS = signat4CRS2;
  ELSE
      s4CRS = smallerCellSide(signat4CRS1, signat4CRS2);
      b4CRS = biggerCellSide (signat4CRS1, signat4CRS2);
  appArea = 0;
  FOR EACH b4CRS cell b THAT IS inside interMBR DO
      FOR EACH s4CRS cell s THAT IS inside cell b DO
        appArea = appArea + expectedArea[s.type,b.type];
        cellArea = s4CRS.edgeSize * s4CRS.edgeSize;
   RETURN appArea * cellArea;
```

Figure 5. Algorithm to compute overlapping (intersection) area of polygons

The algorithm to compute the signature resulting from the union of two raster signatures is used to compute raster similarity, and it is also a contribution of this work. It computes the signature as follows: if there is intersection MBR (Minimum Bound Rectangle) of the signatures, then a new signature is created and returned. On the other hand, when there is not intersection MBR, then NULL is returned. This simplification was done because if there is no intersection between the signatures, than raster similarity is zero. Some comments help to understand the algorithm (Figure 6). More details about algorithm implementations are presented by Antunes and Azevedo (2011).

```
SIGNAT4CRS unionSignat4CRS(signat4CRS1, signat4CRS2)
  IF existsIntersection(signat4CRS1, signat4CRS2)
    IF (signat4CRS1.edgeSize > signat4CRS2.edgeSize)
      b4CRS = signat4CRS1;
      s4CRS = changeScale(signat4CRS2, signat4CRS1.edgeSize);
    ELSE
      b4CRS = signat4CRS2;
      s4CRS = changeScale(signat4CRS1, signat4CRS2.edgeSize);
      /*unionMBR: MBR that encloses  MBRs of s4CRS and b4CRS */
      unionMBR = computeUnionMBR(s4CRS.MBR, b4CRS.MBR)
      /*Creates 4CRS with Empty Cells */
      n4CRS = createSignature(unionMBR, b4CRS.edgeSize, VAZIO);
      /*Mark each n4crs cell by the union of s4CRS and b4CRS cells*/
      FOR EACH b4CRS cell b that intersects n4CRS cell n DO
        n.type = b.type;
        FOR EACH s4CRS cell s that intersects n4CRS cell n DO
          IF n.type = EMPTY OR s. type = FULL
            n. type = s. type;
          ELSE IF n. type = WEAK AND s. type = STRONG
            n. type = s. type;
      RETURN n4CRS;
  ELSE
    RETURN NULL;
```

Figure 6. Algorithm to compute union of two 4CRS

In approximate query processing, along with the response, it is also important to return a confidence interval. The user can use this interval to decide if the precision of the answer is enough. Equation 2, employed by Azevedo *et al*. (2004, 2005), presents the function to compute the confidence interval for the approximate area and approximate overlapping area algorithms. To execute the calculus, it is required to compute the average and variance of expected area and overlapping expected area.

$$\text{Confidence interval (CI)} = \sum n\ n_c \times [\mu_c \pm p \times \sqrt{(\sigma_c^2/n_c)}] \quad (2)$$

Where:

- c: type of cell or combination of type of cells, according to the algorithm
- μ_c: average (expected area or overlapping expected area)
- σ_c^2: variance
- p: confidence interval, e.g., 1.96 for a confidence interval of 95%
- n_c: number of type of cells.

In this work, we propose a confidence interval for raster similarity, presented in Equation 3, based on the proposals of Azevedo *et al*. (2004, 2005).

$$CI_{\text{Raster similarity}} = [(A_\cap - \triangle CI_\cap)/(A_\cup + \triangle CI_\cup);\ (A_\cap + \triangle CI_\cap)/(A_\cup - \triangle CI_\cup)] \quad (3)$$

Where:

- $A_\cap$: approximate overlapping area of raster signatures
- $A_\cup$: approximate area of union of raster signatures
- $\triangle IC_\cap$: confidence interval variance of approximate overlapping area
- $\triangle CI_\cup$: confidence interval variance of approximate area of union of signatures

An example of confidence interval calculus is presented in Figure 7. Consider that the algorithm execution returned the following data: $A_{\cap}$: 4.95×10^6; $A_{\cup}$: 1.27×10^7; $\Delta CI_{\cap}$: 2.37×10^5; and, $\Delta CI_{\cup}$: 2.45×10^5. Using Equation 3 the confidence interval is:

$$CI_{Raster\ Similarity} = [(4.95 \times 10^6 - 2.37 \times 10^5)/(1.27 \times 10^7 + 2.45 \times 10^5), (4.95 \times 10^6 + 2.37 \times 10^5)/(1.27 \times 10^7 - 2.45 \times 10^5)]$$
$$CI_{Raster\ Similarity} = [(4.72 \times 10^6)/(1.30 \times 10^7), (5.19 \times 10^6)/(1.25 \times 10^7)]$$
$$CI_{Raster\ Similarity} = [0.364, 0.416]$$

Figure 7. Example of confidence interval calculus

4. Experimental Evaluation

4.1. Algorithm implementation

The algorithms were implemented in SECONDO - a generic environment that supports database systems implementation for a large number of data models and query languages (Güting *et al.*, 2005). It is developed as a research prototype at Fernuniversität in Hagen. Implementations in SECONDO are done in algebras. Algebras are based on the concept of second-order signature (Güting, 1993): the first signature describes type constructors and second signature describes operations over these types. As an example, raster similarity operator has the specification presented in Figure 8.

```
Name: rSimilar
Signature: (Raster4CRS, Raster4CRS) -> approxresult
Sintax: _ rSimilar _
Meaning: Returns percent. of similarity between two 4CRS with its confidence interval.
Example: query raster4CRS1 rSimilar raster4CRS2
```

Figure 8. Specification of raster similarity algorithm

All implementations employed in this work are available from the following googlecode project: http://code.google.com/p/raster4crs-project/. The project corresponds to all implementations of Raster Algebra, and the algorithm proposed in this work. In the root directory, there is a readme file that explains how to install this algebra after SECONDO installation. SECONDO is available from http://dna.fernuni-hagen.de/Secondo.html).

4.2. Experimental tests

Experimental tests were performed in order to evaluate the precision and execution time of raster similarity algorithm against polygon similarity computed through real representations of polygons. In the experimental tests, there were used a sample of 382 polygons that represents municipalities from north of Brazil (BRNorth). In order to evaluate raster similarity operator, we generated another data set that overlaps with BRNorth. Original polygons were randomly shifted in the x and y axes, as proposed by Brinkhoff *et al.* (1994), and the data set BRNorthT were generated. Figure 9 presents the data sets. Afterwards, 4CRS signatures were generated for each object of these data sets. All commands used to execute the tests are available in the googlecode project file "Experimental tests of similarity operation".

The next step was to compute the similarity. We collect time of hot execution time. The hot execution time was calculated as the average execution time from a total of 10 executions of the same query, discarding the first, the highest and the slowest times so as to avoid outliers. The time to compute similarity from real polygons was

41.187 seconds, while the time to compute similarity from polygons' 4CRS signatures was 14.406 seconds. Considering this dataset, computing raster similarity is three times faster than computing similarity from real representation of polygons.

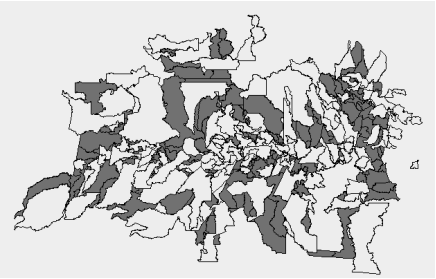


Figure 9. Overlapping of BRNorth and BRNorthT

Afterwards we studied the algorithm precision. We noticed that there were some outliers among the similarity of pair of objects. Objects with very low similarity have big error. Then, to compute the error average the results between percentiles 20 and 80 were considered, excluding extreme values that could bias the average. The error average and error standard deviation of results between percentile 20 and 80 were 13% and 10%. The error median was 10%, and it was used to divide the samples to be studied in two groups: "results above error median (more than 10% of error)" (Table 1) and "results below error median (less than 10% of error)" (Table 2). Due to lack of space, only the most interesting samples for discussion are presented.

The column labels are: (a) ID, IDT: BRNorth and BRNorthT object identifiers; (b) SB, SBT: BRNorth and BRNorthT signatures' length size (size of block); (c) NC: number of cells of signatures that overlap, discarding intersections with *Empty* cells; (d) EIA, AIA: exact intersection area and approximate intersection (overlapping) area; (e) %EAIA: percentage error of approximate intersection area (f) EUA, AUA: exact and approximate union area; (g) %EAUA: percentage error of approximate union area; (h) %RasterS: raster similarity in percentage; (i) %RealS: similarity in percentage computed from real objects; (j) %ES: percentage error of raster similarity (calculated according to Equation 3); (k) %AD: percentage of absolute difference between real similarity and raster similarity (|%RasterS - %RealS|).

$$\%ES = |RasterS - RealS| \,/\, RealS \qquad (3)$$

Table 1. Results above error median

ID	IDT	SB	SBT	NC	EIA	AIA	%EAIA	EUA	AUA	%EAUA	RasterS	RealS	%ES	%AD
234	**397**	**256**	**1024**	**2**	**124**	**312.895**	**252,212.29**	**55,347,900**	**54,263,800**	**1.96%**	**0.58%**	**0.00%**	**257,252.94%**	**0.58%**
10	**25**	**1024**	**256**	**4**	**486**	**1,045,220**	**214,822.12**	**42,117,800**	**41,680,900**	**1.04%**	**2.51%**	**0.00%**	**217,074.82%**	**2.51%**
274	164	512	256	2	1,316	21.286	1,518.03	28,184,300	28,180,500	0.01%	0.08%	0.00%	1,518.24%	0.07%
139	38	256	512	2	9,513	21.286	123.75	18,683,700	17,956,900	3.89%	0.12%	0.05%	132.80%	0.07%
2	**5**	**1024**	**512**	**26**	**5,137,260**	**10,733,300**	**108.93**	**68,657,100**	**68,681,700**	**0.04%**	**15.63%**	**7.48%**	**108.86%**	**8.15%**
140	**146**	**1024**	**1024**	**9**	**613,326**	**1,258,080**	**105.12**	**133,363,000**	**131,596,000**	**1.32%**	**0.96%**	**0.46%**	**107.88%**	**0.50%**

Table 2. Results below error median

ID	IDT	SB	SBT	NC	EIA	AIA	%EAIA	EUA	AUA	%EAUA	RasterS	RealS	%ES	%AD
68	158	512	512	29	2,820,430.00	3,092,300.00	9.64%	34,457,000.00	34,406,400.00	0.15%	8.99%	8.19%	9.80%	0.80%
188	**379**	**1024**	**256**	**22**	**11,475,800.00**	**12,765,200.00**	**11.24%**	**171,566,000.00**	**174,064,000.00**	**1.46%**	**7.33%**	**6.69%**	**9.64%**	**0.64%**
163	38	256	512	12	1,074,720.00	921,305.00	14.27%	17,361,200.00	16,252,900.00	6.38%	5.67%	6.19%	8.43%	0.52%
79	40	128	256	16	389,058.00	424,844.00	9.20%	8,625,600.00	8,716,290.00	1.05%	4.87%	4.51%	8.06%	0.36%
153	**97**	**512**	**512**	**35**	**4,785,020.00**	**4,490,630.00**	**6.15%**	**33,484,000.00**	**34,144,300.00**	**1.97%**	**13.15%**	**14.29%**	**7.97%**	**1.14%**
7	2	512	1024	9	1,770,680.00	1,837,840.00	3.79%	69,856,200.00	67,371,000.00	3.56%	2.73%	2.53%	7.62%	0.19%
79	115	128	256	32	1,036,980.00	1,050,980.00	1.35%	4,741,990.00	4,882,430.00	2.96%	21.53%	21.87%	1.56%	0.34%
280	209	128	256	31	936,733.00	940,258.00	0.38%	7,487,370.00	7,634,940.00	1.97%	12.32%	12.51%	1.56%	0.20%
24	**14**	**1024**	**1024**	**80**	**53,299,100.00**	**52,254,500.00**	**1.96%**	**141,941,000.00**	**141,296,000.00**	**0.45%**	**36.98%**	**37.55%**	**1.51%**	**0.57%**
14	14	1024	1024	61	37,309,900.00	36,984,400.00	0.87%	220,830,000.00	221,512,000.00	0.31%	16.70%	16.90%	1.18%	0.20%
75	32	256	512	13	1,146,480.00	1,181,850.00	3.09%	27,202,300.00	27,721,700.00	1.91%	4.26%	4.21%	1.15%	0.05%
9	9	1024	1024	60	39,526,900.00	39,669,600.00	0.36%	187,091,000.00	188,744,000.00	0.88%	21.02%	21.13%	0.52%	0.11%
434	419	1024	512	28	14,568,800.00	14,089,500.00	3.29%	90,799,300.00	88,080,400.00	2.99%	16.00%	16.05%	0.30%	0.05%
188	**188**	**1024**	**1024**	**163**	**123,756,000.00**	**124,048,000.00**	**0.24%**	**216,988,000.00**	**218,104,000.00**	**0.51%**	**56.88%**	**57.03%**	**0.28%**	**0.16%**
129	228	256	128	52	2,080,950.00	2,136,570.00	2.67%	6,630,980.00	6,799,360.00	2.54%	31.42%	31.38%	0.13%	0.04%
205	146	256	1024	16	5,543,130.00	5,543,720.00	0.01%	74,188,700.00	74,186,800.00	0.00%	7.47%	7.47%	0.01%	0.00%
207	263	128	256	14	306,469.00	307,855.00	0.45%	8,383,900.00	8,421,380.00	0.45%	3.66%	3.66%	0.01%	0.00%

4.2. Result analysis

Raster similarity is computed by the ratio of approximate overlapping area divided by approximate union area. So, it is important to analyze how the overlapping (intersection) area and union area values impact the precision of results. In Figure 10, Y-axis presents percentage of error, while X-axis presents the objects sorted from highest percentage error to lowest percentage error. The percentage error of the area of union of two 4CRS signature is relatively low, while percentage error of raster similarity grows along with percentage error of approximate intersection area. Therefore, if the error of approximate intersection area is small, then the error of raster similarity is small as well.

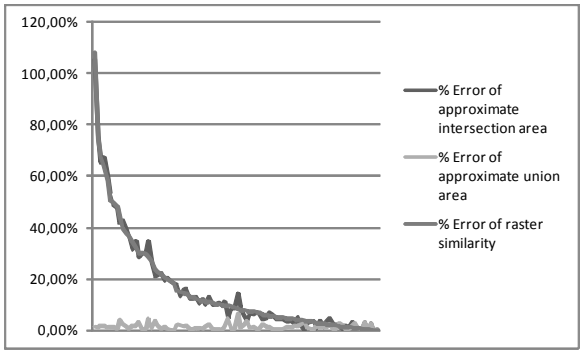


Figure 10. Percentage error of approximate intersection area, approximate union area and raster similarity

The worst error of raster similarity algorithm is presented in the first line of Table 1, corresponding to similarity of objects 234 × 397. The objects are presented in Figure 11.a. In this case, there are only two cells that intersect (column NC). Raster similarity is 0.58% (column RasterS) and real similarity is almost 0% (column RealS). The percentage error is very high (257,252.94%) (column %ES). Similar results occur with objects 10 × 25 (Table 1 and Figure 11.b). It is important to notice that in both examples it is required to execute a scale change of a signature with 256 unities of cell size to a 1024 unities of cell size (columns SB and SBT). The scale change has the goal

to ensure the execution of the algorithm on signatures of same cell size. The scale change is executed grouping a set of cells of the signature with small cell size to represent one cell of the signature with bigger cell size, as presented in Section 2.

Figure 11. Overlapping objects highlighted in Table 1 and Table 2

In the raster similarity calculus of objects 2 × 5 (Table 1 and Figure 11.c), the percentage error of raster similarity is equals to 108.86%. Raster similarity is equals to 15.63% while real similarity is equals to 7.48%. There are more cells that intersect related to the previous example (e.g., NC is equals to 26). However, this number is still small, and it was also required to execute scale change from 512 to 1024. Looking at the type of cells that intersects (*Weak* × *Weak*, *Weak* × *Strong*, *Weak* × *Full*, *Strong* × *Full*, *Strong* × *Strong*, and *Full* × *Full*) (Table 3 - line 1), we can notice that there is no intersection of *Full* × *Full* cells, which is the best case were the precision is 100%.

Table 3. Number of type of cells that overlap

Objects	W × W	W × S	W × F	S × S	S × F	F × F	Total
Obj. 2 × Obj. 5	2	7	6	6	5	0	26
Obj. 188 × Obj. 379	2	1	6	3	6	4	22
Obj. 153 × Obj. 97	3	6	8	2	5	2	26
Obj. 24 × Obj. 14	5	7	18	5	16	29	90
Obj. 188 × Obj. 188	10	9	23	7	33	81	163

On the other hand, in case of raster similarity of objects 140 × 146 (Table 1 and Figure 11.d), there is no scale change, but the number of cells is very small (9 cells). Besides raster similarity and real similarity are very small (0.96% and 0.46%, respectively). Hence the error is 107.88%.

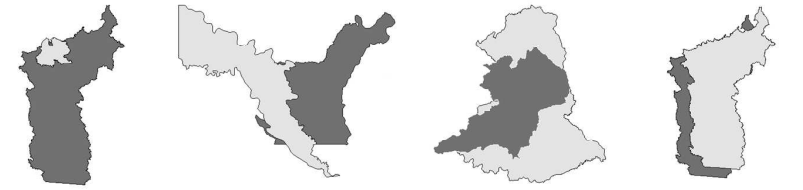
It is important to emphasize that in all cases of Table 1 where the similarity is small, the percentage of absolute difference between real similarity and raster similarity is quite small (column %AD).

We conclude that three main situations contribute to the error: (i) small number of overlapping cells; (ii) majority of overlaps involves cell types whose approximation of overlapping area consider the average (*Weak* × *Weak*, *Weak* × *Strong*, *Weak* × *Full*, *Strong* × *Full*, *Strong* × *Strong*); and, (iii) scale change.

On the other hand, there are other cases where the precision of raster similarity were quite good. For example, in the case of the similarity of objects 188 × 379 (Table 2 and Figure 12.a), the error of raster similarity is 9.64%. Raster similarity is equal to 7.33% and real similarity is 6.69%. The number of cells is small (22 cells), but now there are 4 overlaps of *Full* × *Full* cells (Table 3 - line 2), where the precision is 100%.

In another example, corresponding to similarity of objects 153 × 97 (Table 2 and Figure 12.b), the error of raster similarity is 7.97%. Raster similarity is equals to

13.15%, and real similarity is equal to 14.29%. There are two overlaps of *Full* × *Full* cells (Table 3 – line 3) and the number of cells that intersect is big (35 cells).



(a) Obj. 188 x Obj. 379 (b) Obj. 153 x Obj. 97 (c) Obj. 24 x Obj. 14 (d) Obj. 188 x Obj. 188

Figure 12. Overlapping objects highlighted in Table 3 and Table 4

In the case of similarity of objects 24 × 14 (Table 2 and Figure 12.c), the error of raster similarity is 1.51%. Raster similarity is equals to 36.98% and real similarity is 37.55%. In this case, the number of overlapping cells is 80 and, there are 29 overlaps of *Full* × *Full* cells (Table 3 – line 4).

One of the best results is the similarity of objects 188 × 188 (Table 2 and Figure 12.d). The error of raster similarity is only 0.28%, raster similarity is 56.88% and real similarity is 57.03%. The number of cells that overlap is big (163 overlapping cells) and the majority of overlaps are *Full* × *Full* cells (81 *Full* × *Full* - Table 3 – line 5).

Based on all results presented, we can conclude that the error of the algorithm (highlighted in Figure 10) can be explained because of the approximate intersection area corresponding to the overlap of cell types *Weak* × *Weak*, *Weak* × *Strong*, *Weak* × *Full*, *Strong* × *Full*, *Strong* × *Strong*. When the number of overlaps of these types of cells is small, we cannot assume Normal Distribution, as proposed by Azevedo *et al.* (2004, 2005) to estimate expected area of polygon and expected overlapping area of polygons. Hence two cases can result: (i) When the exact intersection area is close to the average, the approximate result is close to the real result; (ii) When the exact intersection area is not close to the average, the approximate result is also not close to the exact value. This is not the case when Normal Distribution can be applied (Azevedo *et al.*, 2004, 2005). Besides, we observe in our tests that the error above 10% happens when overlapping of objects are on their borders; while in the results with less than 10% of error, objects have more *Full* × *Full* cell overlaps. Therefore, it confirms that the intersection area contributes the most to the error, and it is required to improve the algorithm to compute the approximate intersection area.

Regarding confidence interval, presented in Equation 3 (Section 3), for the results with error above 10%, in 70% of cases, the real similarity was in the interval. In other words, real similarity was between minimum and maximum values computed for the confidence interval presented in Equation 2. On the other hand, for results with error below 10%, in 96% of cases, real similarity was in the confidence interval.

5. Conclusions

The main contribution of this work is a proposal of an algorithm to compute similarity of polygons from their 4CRS signatures. Other contributions were the proposal of algorithm to compute union of two 4CRS signatures and the implementations in SECONDO (Güting *et al.*, 2005) of these two algorithms and algorithms to compute

approximate area of polygons and algorithm to compute approximate overlapping area of polygons (Azevedo *et al.* , 2004, 2005).

Experimental tests were executed over real data corresponding to municipalities from North Region of Brazil. The results demonstrated the proposed raster similarity algorithm is three times faster than the algorithm that computes similarity using real representation of objects. However, raster similarity algorithm's precision varies. Because of some outliers, the percentile 20 and 80 were used to extract a reasonable and interesting sampling for analysis. Among the select objects the median value of error was identified as equals to 10%, and error values below and above 10% was analyzed.

We concluded that the errors above 10% occur when there is small overlapping of objects. The reasons for the error are: (i) small number of overlapping cells of signatures and, consequently, the value of similarity is quite small; (ii) majority of overlapping involves cell types whose approximation consider the average (*Weak* × *Weak*, *Weak* × *Strong*, *Weak* × *Full*, *Strong* × *Full*, *Strong* × *Strong*), which means that the intersection of objects are in their borders; and, (iii) the scale change required to execute the algorithm to compute union of raster signatures. In parallel, the results were quite good for cases where overlaps were bigger.

So we can state that, the bigger is the value of raster similarity, the closer it is to the real similarity. On the other hand, there is a big error in percentage when the value of raster similarity is small. So, if the use of the algorithm intends to discover objects with high similarity, our proposal is a good choice. However, in case of interest is low similarity value, it is better to execute, e.g., the algorithm to compute the real similarity.

We also evaluate our proposal to compute the confidence interval, presented in Equation 3. For the results with error above 10%, in 70% of cases, the real similarity was in the interval. On the other hand, for results with error below 10%, in 96% of cases the real similarity was in the interval. As we employed a 95% of confidence to compute the confidence interval, we can state that our proposal is adequate for high values of raster similarity, but it must be improved for low values of raster similarity.

As future work, we propose: improving the algorithm to compute approximate overlapping area, since the error of raster similarity is highly dependent from overlapping area error, as highlighted in Figure 10; execute performance evaluations considering others datasets; evaluate the use of synthetic data to identify the threshold of similarity for most useful use and recommendation of the algorithm; improve the algorithm to be used in other scenarios, e.g., to compare objects according to their shape, independent from their size and without executing scale change (e.g., compare a model of an object in small size, against a real one); implement a view for Raster objects in SECONDO, which can help to debug the algorithm, and to analyze results.

References

Antunes, L. C. R., Azevedo, L. G, 2011. **"**Polygon Similarity Calculus using Raster Signatures**"**. Technical Report DIA/UNIRIO (RelaTe-DIA), RT-0003/2011, 2011.

Azevedo, L. G., Monteiro, R. S., Zimbrão, G., Souza, J. M. (2004) "Approximate Spatial Query Processing Using Raster Signatures". In: *VI Brazilian Symposium on GeoInformatica* (GeoInfo 2004), Campos do Jordão, Brazil, p. 403-421.

Azevedo, L. G., Zimbrão, G., Souza, J. M., Güting, R. H. (2005). "Estimating the Overlapping Area of Polygon Join". In: International Symposium on Advances in Spatial and Temporal Databases, v. 1, Angra dos Reis, Brazil, p. 91-108.

Azevedo, L. G., Zimbrão, G., Souza, J. M.: (2006). Approximate Query Processing in Spatial Databases Using Raster Signatures. In: Advances in Geoinformatics. 1 ed., v. 1, Springer-Verlag, Berlin Heidelberg New York , p. 69-85.

Brinkhoff, T., Kriegel, H. P., Schneider, R., Seeger, B. (1994). "Multi-step Processing of Spatial Joins". In: ACM SIGMOD Record, v. 23 (2), p. 197-208.

Cakmakov, D., Celakoska, E. (2004). "Estimation of Curve Similarity Using Turning Function". In: Int. Journal of Applied Math, v. 15 (2), p. 403-416.

Gibbons, P. B., Matias, Y., Poosala, V., (1997). "Aqua project white paper". Technical Report, Bell Laboratories, Murray Hill, New Jersey, USA.

Güting, R.H. (1993). "Second-Order Signature: A Tool for Specifying Data Models, Query Processing, and Optimization". SIGMOD Conference, p. 277-286

Güting, R. H. (1994). "An Introduction to Spatial Database Systems". In: The Int. J. on Very Large Data Bases, vol. 3 (4), p. 357-399.

Güting, R. H., Almeida, V., Ansorge, D., Behr T., *et al.* (2005). "SECONDO: An Extensible DBMS Platform for Research Prototyping and Teaching". In: 21st Intl. Conf. on Data Engineering (ICDE), Tokyo, Japan, p. 1115-1116.

Hemert, J. V., Baldock, R. (2007). "Mining Spatial Gene Expression Data for Association Rules". BIRD 2007, LNBI 4414, Springer, p. 66-76

Hellerstein, J. M., Haas, P. J., Wang, H. J. (1997). "Online aggregation". In: Proc. of ACM SIGMOD Intl. Conf. on Manag. of Data, Tucson, Arizona, USA, p. 171-182.

Holt, A. (2003). "Spatial similarity". In: 15th Annual Colloquium of the Spatial Information Research Centre, Dunedin, New Zealand, p. 77-80.

Jaccard, P. (1912). "The distribution of flora in the alpine zone". In: The New Phytologist, vol. 11(2), p. 37–50.

Papadias, D., Kalnis, P., Zhang, J. *et al.* (2001). "Efficient OLAP Operations in Spatial Data Warehouses". In: Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases, Redondo Beach, CA, USA, p. 443-459.

Quine, W. V. (1969). "Ontological Relativity and Other Essays". In: Columbia University Press, New York.

Sako, Y., Fujimura, K. (2000). "Shape Similarity by Homotopic Deformation". In: The Visual Computer, vol. 16(1), p. 47-61.

Samet, H. (1990). "The Design and Analysis of Spatial Data Structure". Addison-Wesley Publishing Company, 1st edition.

Zimbrao, G., Souza, J. M. (1998). "A Raster Approximation for Processing of Spatial Joins". In.: Proceedings of the 24rd International Conference on Very Large Data Bases, New York City, New York, USA, p. 558-569.

Open source implementation of the Multiplicatively Weighted Voronoi Diagram as a TerraView plugin

Eng. Maurício Carvalho Mathias de Paulo[1,2]
Dr. Antônio Miguel Vieira Monteiro[2]
Dr. Eduardo Gerbi Camargo[2]

[1]Diretoria de Serviço Geográfico – DSG
Quartel General do Exército, Bloco "F", 2o Piso, Ala Norte
CEP:70630-901 – SMU – Brasília – DF, Brasil

[2]National Institute for Space Research – INPE
Caixa Postal 515 – 12245-970 – São José dos Campos - SP, Brasil

{mauricio, miguel, eduardo}@dpi.inpe.br

Abstract. *Given a point set the Voronoi diagram associates to each point all the locations in a plane that are closer to it . This diagram is often used in spatial analysis to divide an area among points. In the ordinary Voronoi diagram the points are treated as equals and the division is done in a purely geometrical way. A weighted Voronoi diagram is defined as an extension of the original diagram. The weight given usually relates to some variable property of the phenomenon represented by each point. The weighted distance is then computed as a function that depends both on the weight and on the euclidean distance. This article describes a multiplicatively weighted Voronoi diagram implementation as an open source plugin for TerraView. The algorithm used computes an approximation of the diagram using multipolygons to represent each point's area. This choice avoids the voids that might appear in most of the implementations that focus on finding the intersections and scales well in memory.*

1. Introduction

The Epidemiology and Information Coordination (Coordenação de Epidemiologia e Informação - CEInfo) of São Paulo's Municipal Health Bureau (Secretaria Municipal de Saúde de São Paulo-SP/Brazil) is researching methods to estimate supply and demand of public health care in the city. One of the parameters in the analysis is the accessibility, translated as the distance each person has to walk until reaching the nearest health center.

The health centers are represented as points and the number of treatments in each specific area is measured yearly. Because of the lack of information of where each person come from to be treated it's only possible to estimate an area from where the majority probably came. In this case a spatial partitioning of the study area among the points is a viable alternative.

The Voronoi Diagram is a method to divide the space among a set of points assigning an area to each. The main property of this division is that each area represent the space where the point is the nearest neighbor [Boots 1986]. There are a few extensions on the Voronoi Diagram that can assign areas based on some property of the points,

thus providing a way to distinguish each point by their representativeness. One of these extensions is the Multiplicatively Weighted Voronoi Diagram.

The Multiplicatively Weighted Voronoi Diagrams have been used to evaluate student allocation in educational centers [Karimi et al. 2009], logistic district attribution [Novaes et al. 2009] and computation of dominance area of health centers [Rezende et al. 2000]. The later suggests that the diagram can be used in problems similar to the estimation that CEInfo is researching.

This article first presents the theory behind Multiplicatively Weighted Voronoi Diagrams and the improvements over the Ordinary Diagram (Section 3). In Section 4 the algorithm, user interface and data structure chosen are presented. In Section 5 some preliminary results are presented suggesting its application in geographic analysis.

2. Related work

The Ordinary Voronoi Diagram has some well known algorithms that most implementations are based on. The main algorithms are the incremental insertion, divide and conquer, plane-sweep construction and embedding in a three dimensional space [Aurenhammer 1991]. Many of these implementations are available in open source libraries.

The Multiplicatively Weighted Voronoi Diagram is not necessarily convex nor continuous [Gahegan and Lee 2000]. This makes most of the popular algorithms used to build the Ordinary Voronoi to be impractical for the weighted Voronoi. Some researchers used grid approximation to build the weighted regions [Dong 2008], usually reaching slow results when compared to the Ordinary Voronoi vector-based algorithms. Some approaches uses the bisector defined by two points and finds every possible arc intersection and then reconstructs the diagram evaluating which ones are dominance area's boundaries [Lee and Gahegan 2002]. The optimal algorithm was designed using an spherical inversion in the three dimensional space [Aurenhammer and Edelsbrunner 1984].

As this paper is written, there are many open source libraries with Ordinary Voronoi Diagram implementations, but there is none for the Multiplicatively Weighted Voronoi Diagram. *CGAL* (Computational Geometry Algorithms Library) has an open source implementation of the Additively Weighted Voronoi Diagrams [Karavelas and Yvinec 2002] but as the diagram works with parabolas instead of circles, the data structures and algorithms are not reusable. Some algorithms and implementations are available to compute gridded approximations of the algorithm [Ohyama 2011] but this limits the applications when analyzing geographic vector data.

TerraView (An open source geographic information system application based on TerraLib [Câmara et al. 2008]) has an Ordinary Voronoi implementation. This article describes a vector-based Multiplicatively Weighted Voronoi implementation that is based on the incremental construction algorithm. To solve the problem of the circular arcs that are used in the diagram two approaches are suggested, one for compatibility with polygon implementations and one that extends the concept of polygons using the CurvePolygon data type [Stolze 2003].

3. Weighted Voronoi Diagrams

3.1. Ordinary Voronoi Diagram

Given a set of generator points $S = \{p_1, p_2, ..., p_n\}$, the Voronoi diagram built from the set represents the regions of the plane where each point is closer than all of the others in the set, as formalized in the Equation 1 [Boots 1986]. Therefore, the diagram assigns to each point an area where it is the nearest neighbor [Aurenhammer and Klein 2000].

The boundary of each area is defined by finding the equation for x where the distance to both generator points are equal (Equation 1). Thus the boundary of each dominance area is a line that represents the end of one generator's dominance and the beginning of the other. Figure 1 shows an Ordinary Voronoi diagram example when applied to a given set of generator points.

$$P_i = x | d(x, p_i) \leq d(x, p_j); j \in S, j \neq i \tag{1}$$

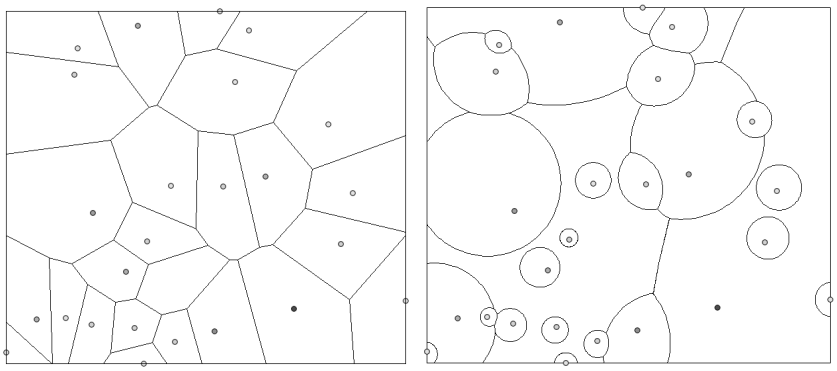


Figure 1. Ordinary Voronoi Diagram and Multiplicatively Weighted Voronoi Diagram

3.2. Multiplicatively Weighted Voronoi Diagram

As an extension, a Weighted Voronoi diagram is defined by analogy, by changing the euclidean distance $d(x, p_i)$ for the weighted distance $d_w(x, p_i)$ [Aurenhammer and Edelsbrunner 1984]. Equation 2 shows the definition of the region of dominance WP_i of a given generator p_i [Boots 1986]. In geographic applications the weight should be a property of the phenomenon mapped by the point p_i, which is considered in the spatial partitioning process [Boots 1986].

$$WP_i = x | d_w(x, p_i) \leq d_w(x, p_j); j \in S, j \neq i \tag{2}$$

Using the distance function as the ratio of the euclidean distance by the generator's weight (Equation 3) a Multiplicatively Weighted Voronoi Diagram is obtained.

$$d(x, p_i) = \frac{|x - p_i|}{w_i} \tag{3}$$

As an example, consider the time taken to reach a coordinate x in a plane. Consider the generator point p_i as a vehicle with constant speed. The time taken to reach x can be computed as $\frac{|x-p_i|}{speed}$. The higher the speed of the vehicle the lower the time distance. This function would then consider a vehicle with distance 100 and speed 10 as nearer than a vehicle with distance 50 and speed 1.

This diagram's dominance areas are bounded by pieces of Apollonius' circles [Aurenhammer and Edelsbrunner 1984]. This is a straight result since the ratio of the distances to the two generator points is constant.

$$d(x,p_1) = d(x,p_2) \rightarrow \quad \frac{|x-p_1|}{|x-p_2|} = \frac{w_1}{w_2} \tag{4}$$

A simple proof arises from setting $p_1 = (0,0)$ and $p_2 = (0,a)$ and $\frac{w_1}{w_2} = c$. Equation 5 is a circle's equation. This result was generalized for coordinates in the plane as is used to compute the boundaries of the diagram [Aurenhammer and Edelsbrunner 1984].

$$\frac{|(x_i,x_j)-(0,0)|}{|(x_i,x_j)-(0,a)|} = c \rightarrow x_i^2 + x_j^2 = c \cdot (x_i^2 + (x_j-a)^2)$$
$$(1-c)x_i^2 + (1-c)x_j^2 - 2acx_j + ca^2 = 0 \tag{5}$$

In the example of the vehicles in the plane, the vehicles are the generator points p_i. The coordinates x where the distance is equal are the places where two vehicles reach the same place at the same time. Therefore this diagram's dominance areas represent the area where each vehicle is the one that reaches every place faster than any other. This could be called the Quickest Neighbor Diagram and might have applications on search and rescue for emergency situations. This is a simple application where diagram is used as an event modeling technique.

There are also applications where the diagram is used as an area assignment method [Boots 1986]. Due to the fact that the distance's ratio is dimensionless so is the weight's ratio. Thus the weights can be any attribute in any unit. This is particularly useful in geographic applications in which the weights usually represent a non spatial parameter that is rarely in units compatible with the reference system of the point set.

The diagram can be interpreted both as a weighted area assignment, where generator points with bigger weights area assigned to bigger areas, or as a growth model, where the areas grow in different ratios starting from the generator points [Boots 1986]. Figure 1 shows a Multiplicatively Weighted Voronoi Diagram compared to an Ordinary Voronoi Diagram.

4. Implementation

4.1. Interface

The parameters necessary to compute a Multiplicatively Weighted Voronoi Diagram are the coordinates of each point and it's weights. As the project aims to be easy for future researchers, an user interface was created and integrated as a plugin on TerraView. Figure 2 shows the interface designed with *QtDesigner* (a cross-platform tool for designing

graphical user interfaces which is part of the Qt SDK) for this purpose. The code used the library *QT* 3.3.8 [Jasmin and Summerfield 2005] for compatibility with TerraView.

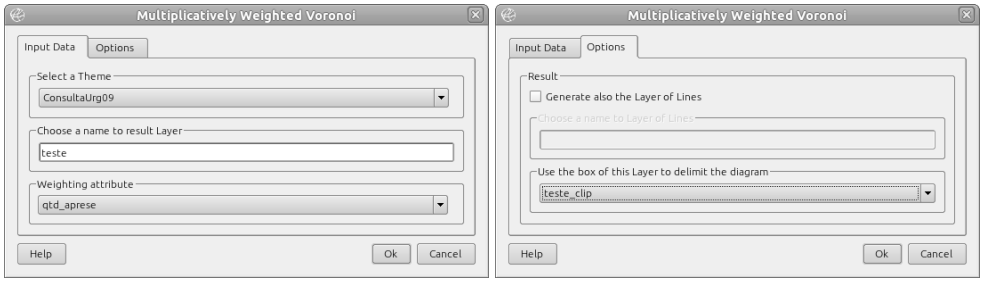


Figure 2. Interface developed for the TerraView's plugin

The first tab shows the main parameters that the user is required to enter in order to build the diagram. TerraView works tightly coupled with many database engines so the plugin lists the available options of input point layers from the currently used database connection. When the input theme is chosen the plugin lists the available attributes. An output layer name is also required for the polygon layer that is created to store the dominance areas.

4.2. Algorithm

Given two generator points p_1 and p_2 and their respective weights w_1 and w_2 the space is divided in two areas. As described in Equation 5, the weighted distance is used to divide the plane using the boundary of an Apollonius Circle. Therefore, one point receives the inner area of the circle and the other one the outer area. Figure 3 shows an example where $w_1 > w_2$.

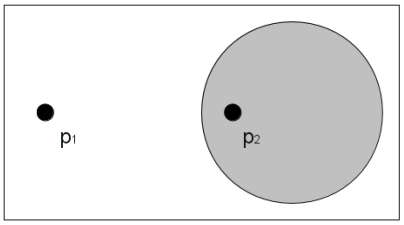


Figure 3. Dominance areas of P_1 and P_2 where $w_1 > w_2$

Without loss of generality, let $w_1 > w_2$ therefore p_1 dominates p_2 and p_1's dominance is the outer area while p_2's is the inner area of the circle. The center of the circle that represents the boundary of the dominance areas has center c_1 and the radius r_1 (Equation 6) calculated using the two points' coordinates and weights [Aurenhammer and Edelsbrunner 1984].

$$\vec{c_1} = \frac{w_2^2.\vec{p_1} - w_1^2.\vec{p_2}}{w_1^2 - w_2^2} \quad and \quad \vec{r} = \frac{w_1.w_2.|\vec{p_1} - \vec{p_2}|}{w_1^2 - w_2^2} \tag{6}$$

A Multiplicatively Weighted Voronoi Diagram represents each generator's dominance area. Therefore to compute a single point's dominance every Apollonius circle over

each other generator is computed. Figure 4 shows the dominance areas of P_1 over P_2 and P_3. In this example, P_1 dominates P_2 and is dominated by P_3. Thus the dominance area of P_1 is the intersection of the two areas, shown in gray.

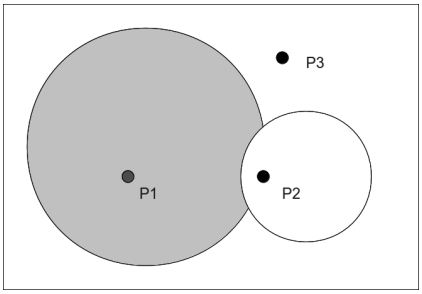


Figure 4. Intersection of the two dominance areas of the point P_1

The implemented code computes each point's dominance by intersecting every Apollonius circle created by combining the point with every other point in the set. Algorithm 1 shows the main steps to find the dominance area of a point $p[i]$ given the array p that stores every point in the set. It uses a straight forward bisector computation storing each generator point's dominance area.

Algorithm 1: Compute dominance area of the point pointList[i]

```
Input: i, pointList, RegionOfInterest
Output: dominanceArea (Area of dominance of the point i)
dominanceArea=RegionOfInterest;
for j de 0 a n do
    if i!=j then
        circleIJ=ApolloniusCircle(i,j,pointList);
        dominanceArea=Intersection(circleIJ,dominanceArea);
```

The function *ApolloniusCircle(i,j,pointList)* returns a regular polygon with 360 sides centered on the circle's center and with radius calculated using Equation 6. Thus, the circle is approximated using a polygon before computing the intersections.

The function *Intersection(circleIJ,dominanceArea)* computes polygon intersections, returning a new polygon representing the intersection area. This function's computational time is $O(a \cdot b)$ where a and b, represents how many vertexes are used in each polygon. This computation is performed by TerraLib's polygon overlay functions.

4.3. Complexity

The main iteration, represented by Algorithm 1 is $O(n^2)$ where n is the number of generating points. Each dominance intersection is computed by intersecting two non-convex polygons. The polygon *circleIJ* has 360 sides, so $a = 360$. The number of edges in a region is bounded by n [Aurenhammer and Edelsbrunner 1984] so the complexity of the method *Intersection(circleIJ,dominanceArea)* is bounded by $O(n \cdot 360)$. This method is called $O(n^2)$ times so the whole computational complexity of the algorithm is bounded by $360 \cdot O(n^3)$.

For better performance the data structure chosen to represent the dominance areas should be based on circular arcs instead of MultiPolygons. The SQL multimedia standard (ISO/IEC 13249 SQL/MM) aims to standardize the CurvePolygon and MultiSurface geometry types, which represents, respectively, a closed area defined by pieces of circular arcs and a collection of CurvePolygon [Stolze 2003].

Using this kind of data structure the method *Intersection(circleIJ,dominanceArea)* drops it's complexity to $O(n)$ as it's the computational time of intersecting n pieces of circles with a circle. Therefore the algorithm is bounded by $O(n^3)$. This result is better than most approximations that achieved $O(n^4)$ [Lee and Gahegan 2002] or worse [Dong 2008] but worse than the optimal algorithm [Aurenhammer and Edelsbrunner 1984].

Both by using polygon approximation or CurvePolygon datatype the algorithm scales well in memory. This is a straight result of the iteration process that computes only one dominance area at each iteration. The implementation stores each computed area in the database, so at each step only one dominance area and one circle are in memory. Implementations that compute the intersections first then reconstruct the dominance areas need to store all the intersections in memory during processing time [Gahegan and Lee 2000].

5. Results

Figure 5 shows the result of using the implemented plugin as a tool for geographic spatial partitioning. The polygon that limits the map is São Paulo city's boundary. The points represent the health care institutes in which the number of emergency treatments done in the year of 2009 were measured. The circular arcs are the Multiplicatively Weighted Voronoi area boundaries.

Figure 5. Weighted Voronoi Diagram of São Paulo's Health Centers

In this application the diagram was used as an weighted area assignment tool.

Thus health centers that treated more people than it's neighbors were assigned bigger areas. Some areas with much lesser number of treatments received areas so small that they can be seen as with insignificant influence. This happened because there are health centers with 2 treatments per year near other centers with more than 200.000 treatments per year. This is a major improvement from the Ordinary Voronoi in this application since the later would assign areas without any regard to the relevance of each point in the space.

In the southern area of the map there is a district called Grajaú that is populated mostly by familys with low income. The health center there was assigned the biggest area in the map since it is responsible for treating more people than any other health care in the city and there is no other health center of the same size near. This is a good example of how a weighted Voronoi area can grow large if the generator point is in fact dominating it's neighbors.

Since the implementation uses an approximation of the circles as polygons instead of the proposed CurvePolygon data type a topology error was created on every intersection. This happens because every point that describes the end of a circular arc in the diagram is the result of the intersection of three circles as shown in Figure 4 [Lee and Gahegan 2002]. When the circles are approximated by polygons, the intersection of the areas is not computed using circular arcs but using lines. Figure 6 shows the spaces that are produced between the areas where there should be an intersection point.

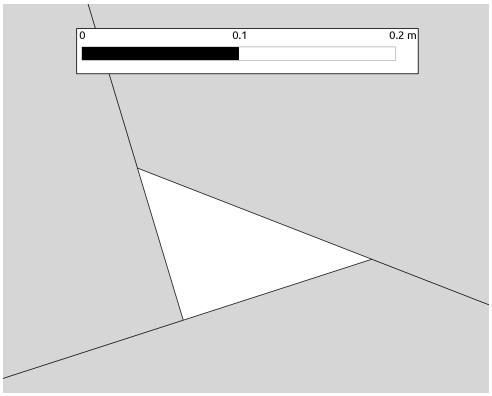


Figure 6. Approximation error in the intersections

The percent error (e)on the approximation can be found using Equation 7 as shown in Figure 7. Applying this equation to the 360 sides used by default in the implementation, the estimated percent error in the boundary of each area is 0.0001523.

$$e = \frac{R \cdot (1 - cos\theta)}{R} \rightarrow e = 1 - cos(\frac{2 \cdot \pi}{360}) \approx 0.0001523 \quad (7)$$

6. Conclusion

As the algorithm's main effort is to find each point's dominance area in each step, the void areas without dominance found in some previous researches [Aurenhammer and Edelsbrunner 1984] [Rezende et al. 2000] does not happen. From the

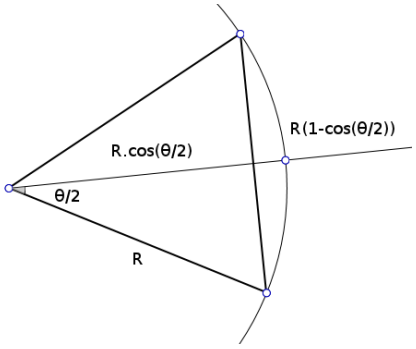


Figure 7. Circle as 360 side Polygon approximation

mathematical definition in Equation 2 there should be no area where the inequation is unsolvable. This algorithm builds by brute force the dominance areas, preserving multi part polygons, therefore areas can be discontinuous, but not empty. The application of the diagram in a geographic environment is enhanced by this improvement as empty spaces in the area partitioning are an undesired effect.

The representation of the circles as polygons before intersection brought an unnecessary speed loss. Using geometries based on circle sections would reduce the computational time to find the intersections and give better topology nodes.

The implemented code was compiled both on Windows and Linux environment which allows the tool to be used by many researchers. It's released under GPL license as a TerraView Plugin and is available through the TerraLib repository. This implementation is a proof of concept that is already usefull for many purposes. The next goal in the research is to implement the MultiCurve geometry type as defined by SQL/MM in order to improve both precision and speed.

Acknowledgments

This work is the result of the final article written for Introduction to Geoprocessing course in order to fulfill the requirements for the master's degree on Remote Sensing assigned by the Brazilian Army on National Institute for Space Research (INPE). The database of health care emergency treatments in São Paulo City were kindly provided by CEInfo for testing purposes.

References

Aurenhammer, F. (1991). Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405.

Aurenhammer, F. and Edelsbrunner, H. (1984). An optimal algorithm for constructing the weighted Voronoi diagram in the plane. *Pattern Recognition*, 17(2):251–257.

Aurenhammer, F. and Klein, R. (2000). Handbook of Computational Geometry, chapter Voronoi Diagrams. *Elsevier*, 169:170.

Boots, B. (1986). *Voronoi (Thiessen) Polygons*, volume 45. Geo Books.

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 large-scale environmental and socio-economic applications. *Open Source Approaches in Spatial Data Handling*, pages 247–270.

Dong, P. (2008). Generating and updating multiplicatively weighted Voronoi diagrams for point, line and polygon features in GIS. *Computers & Geosciences*, 34(4):411–421.

Gahegan, M. and Lee, I. (2000). Data structures and algorithms to support interactive spatial analysis using dynamic Voronoi diagrams. *Computers, environment and urban systems*, 24(6):509–537.

Jasmin, B. and Summerfield, M. (2005). C++ gui programming with qt 3.

Karavelas, M. I. and Yvinec, M. (2002). Dynamic additively weighted voronoi diagrams in 2d. In *Proceedings of the 10th Annual European Symposium on Algorithms*, ESA '02, pages 586–598, London, UK, UK. Springer-Verlag.

Karimi, F., Delavar, M. R., and Mostafavi, M. A. (2009). Space allocation of educational centers using multiplicatively weighted voronoi diagram. *WG II/2, II/3, II/4: Workshop on Quality, Scale & Analysis Aspects of City Models*.

Lee, I. and Gahegan, M. (2002). Interactive analysis using Voronoi diagrams: Algorithms to support dynamic update from a generic triangle-based data structure. *Transactions in GIS*, 6(2):89–114.

Novaes, A., Souza de Cursi, J., da Silva, A., and Souza, J. (2009). Solving continuous location-districting problems with voronoi diagrams. *Computers & Operations Research*, 36(1):40–59.

Ohyama, T. (2011). Takashi ohyama's multiplicatively weighted voronoi diagram page. `http://www.nirarebakun.com/voro/emwvoro.html`.

Rezende, F., Almeida, R., and Nobre, F. (2000). Diagramas de Voronoi para a definição de áreas de abrangência de hospitais públicos no Município do Rio de Janeiro. *Cadernos de Saúde Pública*, 16(2):467–475.

Stolze, K. (2003). Sql/mm spatial: The standard to manage spatial data in relational database systems. In *Proceedings of the BTW*.

Trust Indicator for Decisions Based on Geospatial Data

Ivanildo Barbosa[1,2] and Marco A. Casanova[2]

[1]*Seção de Ensino de Engenharia Cartográfica*
Instituto Militar de Engenharia (IME)
Praça General Tibúrcio, 80 – CEP 22290-270 – Rio de Janeiro – Brasil

[2]*Departamento de Informática*
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rua Marquês de São Vicente, 225 - CEP 22451-900 – Rio de Janeiro – Brasil
{ibarbosa,casanova}@inf.puc-rio.br

Abstract A large family of real-world applications are influenced by geospatial features and known relationships between them. Also, a very large volume of geospatial data is currently available from different sources, prepared for different applications, and with different levels of uncertainty. This paper presents a brief analysis about spatial, thematic, technical and temporal aspects of uncertainty and about how they influence the reliability of decisions based on such datasets. It also proposes an indicator to quantify the inherent reliability of such data, based on their provenance, completeness, spatial coverage and data lifetime. In particular, the indicator is applicable in the context of planning applications for which geospatial data are relevant in order to rank or discard available geospatial datasets.

Keywords: data reliability, geospatial uncertainty

1 Introduction

In the past decades, geospatial data producers considerably improved methods to acquire, process and distribute geospatial data (vectors, images, aerial photographs, thematic maps, etc) to the different kinds of users, who depend on such data to their decision making processes. Government agencies and collaborative initiatives offer online data that can be accessed through traditional user interfaces or through Web services.

However the best geospatial datasets will never show complete fidelity to the reality wherever and whenever users need to use them. Maps are designed to bring a controlled level of uncertainty, considered irrelevant for some kinds of users and applications. Despite the fact that data providers adopt the most reliable methods and use the most precise and accurate data acquisition platforms, some uncertainty will remain in geospatial data.

A wide variety of planning applications depends on geospatial data, either static or time-varying, and some of them demands low tolerance for uncertainties. Therefore, users must assess the geospatial data sources in order to select those best suited to their applications (according to the related knowledge base) [1,2].

This paper proposes an approach to evaluate the reliability of geospatial datasets in the context of planning applications that takes into account *spatial*, *thematic*, *temporal* and *technical* aspects.

The paper is organized as follows. Section 2 summarizes the standard quality indicators for geospatial data [3]. Section 3 analyzes the sources of indeterminacy and proposes alternative quality indicators for geospatial data. Section 4 applies the concepts proposed to plan routes for off-road vehicles based on geospatial data from several sources. Section 5 summarizes the approach and discusses future research lines to refine present results.

2 Quality indicators for geospatial datasets

Geospatial data are typically produced to meet the requirements of a given set of applications. The production processes are guided by well-defined specifications to provide a controlled level of accuracy and uncertainty which is adequate for the set of applications in question. Hence, different geospatial datasets covering the same area may have different characteristics.

When geospatial data were represented as printed maps, the only way to evaluate their accuracy was by comparing the represented coordinates of a number of geographic features with their real coordinates measured over the terrain. The differences, as well as their standard deviation, should be less than specified thresholds [4]. Today, geospatial data are digitally represented as vectors, matrices (images or coverages), lists of coordinates and databases, which demand the adoption of proper criteria to assess quality [3].

Quality references are relevant metadata so that the ISO 19115 standard [5] defines a package to deal this issue. In the Data Quality package, one may store the reports of measurements procedures (described in ISO 19114 [6]) and the description of process steps, and the respective data sources used, to create the dataset (also known as *lineage*).

The current specification to assess the quality of a geospatial dataset evaluates its *completeness*, *logical consistency*, *positional accuracy*, *thematic accuracy* and *temporal accuracy*.

Completeness indicates the omission or excess (*commission*) of geographic features, attributes and relationships in the dataset over its declared geographical extents. It may influence query results by improperly accepting or reject features. For raster data, pixels usually do not have null values. However, some applications consider a specific pixel value to representing the absence of value.

Logical consistency provides information about the adherence to rules related to the data structure, attributes and relationships. Such rules allow matching the attributes provided with the list of required attributes and to verify if the provided data is in conformance with the defined domains and formats.

Accuracy reflects preoccupations with spatial, thematic and temporal issues. The specified data quality reports point to absolute values and conformance to some specification, demanding further analysis.

3 Criteria for reliability

Although there are specifications to assess quality, the results obtained are not enough to assign any reliability index to data. Metadata about identification and spatial reference are also necessary to analyze the usability of a dataset. In this section, we propose a set of criteria to assess reliability for geospatial data, which better matches the requirements of planning applications, among others. The

proposed criteria are *spatial coverage*, *data completeness*, *provenance* and *lifetime*.

3.1 Spatial Coverage

Spatial coverage indicator is proposed to assess reliability for geospatial data by analyzing spatial aspects. The first aspect of spatial coverage points to evaluate *how the dataset extents cover the area of interest for planning*: fully, partially or not at all. Both the dataset and the area of interest extents are predefined by, respectively, the dataset design and the application specification. Ideally, the geospatial datasets must therefore cover an area that contains the area of interest, as otherwise the planning process may be affected by lack of available data. The polygons used do define the extents of both the dataset and the area of interest may be compared to compute the overlapping area among them and return the percentage of the area of interest covered by the dataset. The example in Figure 1 illustrates how datasets (dashed boxes) cover the area of interest (continuous line polygon). No dataset covers the whole area of interest (continuous line), so the reliability for each dataset would be lower than 100%. Even if the datasets were merged, the reliability of the whole dataset would be lower than 100%.

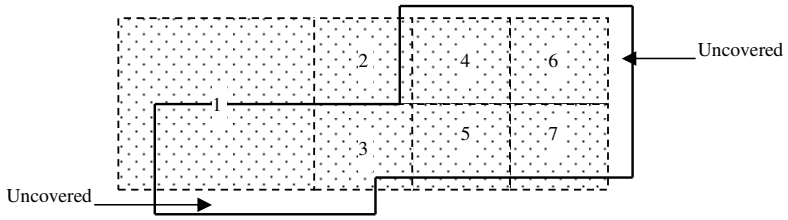


Figure 1 Example of spatial coverage: dashed boxes represent geospatial datasets that cover partially the area of interest represented by the continuous line bounded polygon

When partial coverage occurs, it is suggested to divide the planning area into *covered* from *uncovered areas*. Using the example of Figure 1, the original area would be divided in nine parts: seven areas covered by the respective datasets (100% covered) and the two remaining two not covered.

The total overlap of the areas does not provide completely reliable information. The analysis discussed before considered only the geographical extents of data. However, it is necessary to ensure that all the existing features are represented, by measuring *completeness (omission* and *commission)* [6] or using the respective metadata, when available. It aims at indicating whether all the

geometries relative to features are represented in the extents. Despite the fact that attributes are not fulfilled, these features exist and may be identified even by visual analysis.

After checking the integrity of the representations, the next approach is to analyze the individual geometries stored in the dataset. Data producers align their methodologies and materials in order to achieve a precision coherent with a predefined scale, called *equivalent scale* (usually defined by its denominator [5]). However, the exhibition scale may be controlled by the user and generalization rules must be applied to simplify the representation [7, 8]. Generalization for vector data restricts the types of visualized features, enhances relevant features or types of features in a given context, displaces or omits some features according to their both dimensions and specified precision, and simplifies some features representations. Larger equivalent scales (denominators) imply less detailed geometries. Equivalent scale also impacts the criteria to evaluate omission because some features may be not represented in some scale ranges for datasets considered complete.

Considering raster data (pictures, images and grids), a pixel represents a regularly sized portion of the terrain, either based on the signals captured by the sensors or by transforming vector to raster. Details smaller than the area covered by one pixel are ignored. Therefore, the equivalent scale must be compatible with the pixel size.

In order to assess the reliability of a dataset, it is necessary to compare its *equivalent scale* to a specified reference value, called here *proper equivalent scale for application* (PESA). In order to integrate vector and raster data and to facilitate the understanding of practical consequence of the concept, the *spatial resolution* (the size, in meters, of the smallest detail represented at the respective scale) may be used instead of the equivalent scale value.

3.2 Data Completeness

Completeness, in this context, is related to the *thematic* integrity of the representation of the existing features, that is, how comprehensive are the non-geometric attributes in a database and also in the raster data. These attributes describe the feature and an analysis of their values is relevant to select geographic features according some specific condition. Queries in incomplete databases will

accept or reject mistaken features due to match (or mismatch) attributes and conditions.

Despite the fact that the notion of completeness described in [3] and [5] partially merges the concepts of spatial coverage and data completeness, we distinguish these issues in this paper. While an *uncovered area* misses geometries (and respective attributes), an *incomplete area* misses only some attributes values, although there are geometries for every features.

Null values indicate lack of information in table registries and raster representations. Furthermore, although not explicitly indicated, default values are used to replace absent of data, thereby becoming an indicator for lack of data – field values in vector data and pixels in raster data.

Absent data may be estimated by mining available data, as reported for example by Pearson [9]. However, the estimation methods and models may also embed uncertainty, so it is recommended to distinguish the reliability of the measured (or observed) values from the estimated values.

Figure 2 presents a proposed ranking for reliability based on data completeness. It is suggested to assign values between 0 and 1, proportional to the percentage of fulfilled (or estimated) values. The assigned reliability values will be higher when the dataset satisfies the conditions at the top of the figure. On other hand, lower values will be assigned if the conditions at the bottom of the figure hold. The heights of intermediary boxes illustrate the differences between the ranges of values for reliability at each of the conditions shown and are intended to be qualitative – out of scale. The blank area represents the upper limits for reliability in some cases and is out of scale (in order to fit the text).

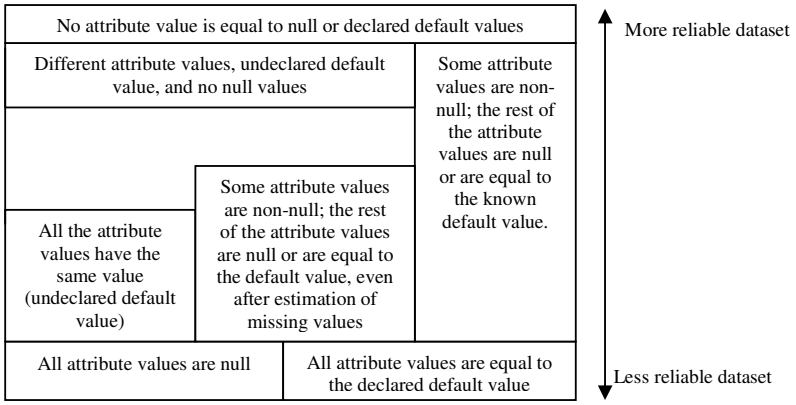


Figure 2 Relative values for reliability based on the data completeness criterion

3.3 Provenance

The usual concept of provenance is related to the source or history of some product (see [10] for a survey). In general, different entities may produce geospatial datasets aiming at achieving different levels of accuracy for different applications. It implies in more (or less) severe specifications, more (or less) accurate equipments and methods, and so on. At the context of geospatial data quality, provenance is related to the lineage.

As a simple example, consider the use of portable GPS receivers by non-corporative users to acquire geospatial data (receiving, grouping and labeling waypoints and tracks) and to publish the acquired data on the Web. A typical use is to georeference features not represented on conventional maps. On other hand, such data is defined only by geometry, with no post-processing to minimize GPS errors [11, 12]. The reliability of such datasets tends to be lower than those created with more accurate methods and equipments.

In general, geospatial data producers should abide to specifications that define methods, equipments, precisions and contents when publishing their datasets. On one extreme of the (trusted) provenance scale, we may classify government agencies that deploy Spatial Data Infrastructures (SDI) adopting standards for files and Web services. Data users rely on such standards and on the reputation of producers to assess data quality. On the other extreme, we may include companies that provide datasets for specific purposes. In this case, users will perhaps depend on some methodology to assess data quality with respect to provenance. We may include in this second category academic institutions that produce data for customized applications, according to standards published by national or international organizations. However, datasets thereby produced may be useful only for the purpose they were created, due to their particularities.

To ensure the adherence of data characteristics to the specifications, it would be recommended that a certification be issued by competent institutions to warrant the usability of that dataset at that particular application – here called *warranty of conformance to application specifications* (WCAS). However, this certification is frequently ensured by the customer himself empirically. In some cases, the producer has an independent auditing department to evaluate this adherence according to predefined legal or technical limits. Therefore, assigning trust certification to producers will warrant the usability of the datasets they

produce. Both options involve legal discussions about the certifier authorities and technical issues about criteria for certifying datasets and producers.

To summarize, the quality of the data in this context depends on the application, on the reputation of the data provider (academic, industrial and government), and on the ability of the user to assess and warrant the datasets. Therefore, the definition of absolute weight values to different data providers is not a simple task. So, we propose a ranking analogous to that proposed in section 3.2, illustrated in Figure 3. The assigned reliability values will be higher when the dataset satisfies conditions at the top of the figure. On other hand, lower values will be assigned in conditions at the bottom of the figure.

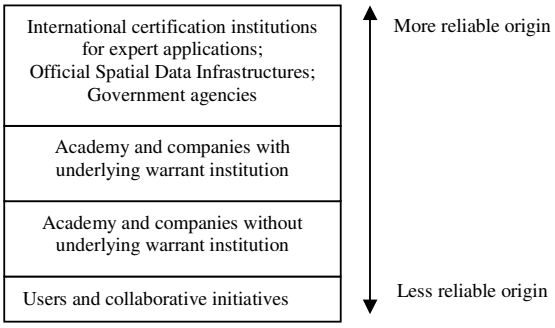


Figure 3 Relative values for reliability based on the provenance criterion

3.4 Lifetime

"*Maps are like milk: their information is perishable, and it is wise to check the date*" [8]. This statement reflects the caution of users about any kind of data and motivates the discussion about dataset reliability after some elapsed time.

The first approach to assess the lifetime for a dataset depends on its age, defined as the time elapsed between the acquisition of the data (certified by the data quality reports of temporal accuracy [6]) and its use by the application. In most planning activities, more recent datasets are preferred to older ones. However, even the most recent datasets may be outdated because geographic features types represented in the dataset change at different rates. In some cases, the representation of each geographic feature has its own distinct indicator. However, it is suggested to consider a single lifetime value for the whole geospatial dataset.

Furthermore, to come up with a reasonable estimation for the rate of change of the data may be quite difficult, although some reasonable

approximations are feasible. For example, natural feature types, such as physiographic and hydrographic ones, present slower changes, usually caused by natural phenomena, such as geological movements and long-term weather events. Man-made feature types tend to change faster (including physiographic, hydrographic and vegetation features), as a result for example of expanding populated areas or increasing infrastructure (transport systems, energy production and distribution, *etc.*).

It is therefore necessary to introduce the concept of *safe age for data application* (SADA), meaning the maximum time interval after date of creation (or last update) the dataset may be considered unchanged. It depends on the application, the equivalent scale, the feature type and the potential of human influences.

Figure 4 illustrates the relationship between the time related aspects mentioned above. The edges of the cube (adapted to facilitate visualization) represent *human occupation* – amount and distribution of fixed population at the area of interest, *economic activities* – indication of land use and the consequent potential to change features, and data *lifetime*. The surface represents the threshold for acceptable values. It is not flat because the relationship among the concepts is not linear, demanding further research to model it. However, the relative relationships about lifetime are preserved: less human occupation and less economic activities imply higher lifetime values for the dataset, and vice-versa. Statistic data about socioeconomic aspects may be retrieved from governmental institutions responsible for census.

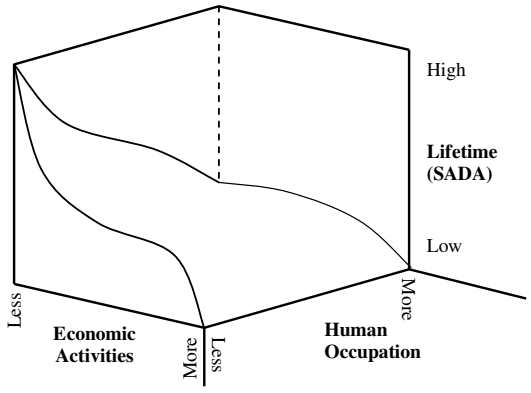


Figure 4 Relationship between lifetime, dataset age and socioeconomic aspects

In order to improve the reliability of geospatial datasets according to this criterion, it is necessary to create policies for checking and for updating feature types with low values for SADA. Even not changing the features, the age will be counted from the last verification.

4 Combining the indicators

After establishing strategies to assess each quality indicator, we must indicate how to compose them to create a unique quality indicator. There are two main approaches to deal it: the first one considers the geometric mean of normalized indicators for lifetime (*L*), spatial coverage (*S*), data completeness (*D*) and provenance (*P*), while the second one defines a qualitative classification to indicate application profiles based on fuzzy concepts. The geometric mean was chosen instead the arithmetic mean because it is indicated to handle rates. Hence, the reliability for some datasets may be assigned as 0, meaning that dataset offers no reliability. A fuzzy approach may provide some reliability for rejected datasets and will be useful when no available datasets meet the specifications and the user may choose a less imperfect dataset. This paper will deal the first approach, suggesting the second one for further discussions.

The first step is to locate both the extents of the dataset and of the area of interest, dividing the whole area to define areas fully covered by the datasets (as illustrated in Figure 1). Planning in uncovered areas must be avoided due the total lack of information.

Spatial resolution (indicator *R*) does not affect only accuracy, but also simplifies or deletes some features representations. Therefore, it is recommended to reject ($R = 0$) datasets with spatial resolution lower (coarse) than the specified one. For finer values, the spatial resolution indicator value *R* is assigned as 1.

In areas covered by datasets with proper resolution, the spatial completeness indicator will define spatial coverage indicator *S*.

The criterion to assign the indicator for provenance *P* may be simplified, by considering the existence of warrant certification (assigning value 1 to *P*). The absence of such certification assigns a partial value to P (0.5, for example).

After computing how long a dataset is valid for a specific application, the assigned value for lifetime indicator *L* will be 1, if the specified SADA is larger than the dataset age. Otherwise, the assigned value will be 0.

Hence, only provenance and data completeness provide values other than 0 and 1 for their respective indicators, *P* and *D*, belonging to the interval [0,1]. The proposal to compute an overall indicator *I* for an individual dataset is presented in (1).

$$I = L \cdot \sqrt[3]{P \cdot S \cdot D} \tag{1}$$

This definition of *I* assumes that all indicators have the same relevance. However, specific indicators may have different influences for some applications. In this case, the definition of *I* may be modified by assigning weights to balance the influence of the indicators. However, indicators will be not accurate enough to demand variations.

When the application requires data from multiple datasets, we would compute a separate indicator I_i for each dataset. The final indicator might be defined as a geometric mean of individual indicators.

$$I = \sqrt[n]{\prod_{i=1}^{n} I_i} \tag{2}$$

5 Conclusions

This paper proposed an approach to evaluate reliability indicators for geospatial data in the context of planning applications. The approach did not question thematic, temporal and positional accuracies measurement [6], but it rather relied on metadata in data quality package (*provenance* and *data completeness*), in identification package (*spatial coverage* and *spatial resolution*, or *equivalent scale*), and external data based on socioeconomic factors (*lifetime*). It means that some datasets should be replaced due to incompleteness (of attribute values), obsolescence and inadequate scale (insufficient level of spatial details). Further studies may deal with the cases where all datasets were rejected, using fuzzy criteria to compose an indicator for "best" fitting.

The use of concepts of *WCAS*, *PESA* and *SADA* aims at supporting the ranking process to select or discard geospatial datasets based on their reliability values. However, further studies are necessary to refine both criteria and threshold values to rank datasets reliability, either isolated or combined. In these cases, one

might rely on expert knowledge to obtain more meaningful indicators for the evaluated datasets face to the target application.

References

1. Russell S, Norvig P (1995) Artificial Intelligence: a modern approach, Prentice Hall, New Jersey.

2. Taştan, H, Altan, MO (1999) Spatial Data Quality, In: Proceedings of 3rd Turkish-German Joint Geodetic Days.
http://www.hgk.msb.gov.tr/haritalar_projeler/bildiriler/cbs/makale(pdf)/cbs_tek_bil3.pdf . Accessed 20 July 2011.

3. International Organization for Standardization (2002) ISO 19113: Geographic information -- Quality principles.

4. Brasil (1984) Decreto Nº 89817. Estabelece as Instruções Reguladoras das Normas Técnicas da Cartografia Nacional.

5. International Organization for Standardization (2003) ISO 19115: Geographic information -- Metadata.

6. International Organization for Standardization (2003) ISO 19114: Geographic information -- Quality evaluation procedures.

7. Mackaness WA, Ruas A, Sarjakoski LT (2007) Generalisation of Geographic Information: Cartographic Modelling and Applications. Elsevier, Amsterdam.

8. Monmonier, M (1996) How to Lie with Maps. 2nd Edition. University Of Chicago Press, Chicago.

9. Pearson RK (2005) Mining Imperfect Data: Dealing with Contamination and Incomplete Records. SIAM, Philadelphia.

10. Marins ALA (2008) Modelos Conceituais para Proveniência, Dissertation, Pontifical Catholic University of Rio de Janeiro.

11. Monico JFG (2008) Posicionamento pelo GNSS – Descrição, Fundamentos e Aplicações. UNESP, Presidente Prudente.

12. Gopi, S (2005) Global Positioning System: Principles and Applications. Tata McGraw-Hill Education, New Delhi.

Using OGC Services to Interoperate Spatial Data Stored in SQL and NoSQL Databases

Cláudio de Souza Baptista, Odilon Francisco de Lima Junior, Maxwell Guimarães de Oliveira, Fabio Gomes de Andrade, Tiago Eduardo da Silva, Carlos Eduardo Santos Pires

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

{baptista, cesp}@dsc.ufcg.edu.br,
{odilonflj, maxmcz, fabiocefetpb, tiagoes}@gmail.com

Abstract. *Spatially-enabled social networks like Twitter and Foursquare have produced huge volumes of geo-referenced information which has been in general stored in NoSQL databases. The need to bring together the entire spectrum of geo-referenced information takes us to the traditional problem of interoperability between SQL and NoSQL spatial databases. This paper proposes a solution for the integration of geographic data stored in both SQL and NoSQL databases using OGC WMS and WFS interoperability services. Experiments conducted on PostgreSQL-PostGIS and CouchDB-GeoCouch spatial databases have demonstrated that it is possible to submit queries using the same syntax for SQL and NoSQL spatial databases in a simple and transparent manner for the user's application.*

1 Introduction

Due to the large volume of data generated on the Internet today, new forms of data storage and data processing are required. We are living in an era of social networks that generate a huge amount of information. For instance, Twitter generates more than 12 Terabytes/day of information which needs to be stored for future reference. Such information can be geocoded. Moreover, the Web 2.0 technology has given rise to several location-based social network services, e.g. Foursquare, Gowalla, Whrrl, Loopt, and Brightkite.

The traditional architectures of Database Management Systems (DBMS) for storing structured data have proven inadequate to deal with this enormous volume of data, known as big data. For these applications, NoSQL databases provide distributed storage and indexing techniques using map/reduce functions [Dean and Ghemawat 2008]. However, the ubiquitous spatial dimension in data sets along with the popularity of spatial applications and also the supporting devices for geo-referenced data gathering, such as smartphones, GPS and cameras, have all contributed to an increase in this

information volume. As a result, some NoSQL databases, such as CouchDB[1] and MongoDB[2], provide support for spatial data.

There is still the pressing need to combine this volume of georeferenced information that emerges from social networks like Twitter and Foursquare with the traditional geo-referenced information, stored, for example, in SQL spatial databases or in spatial data infrastructures. For instance, to view checkin or checkout data from a particular group of users who are shopping within one kilometer buffer of a particular street in the city. This problem involves interoperability between SQL spatial DBMS such as PostgreSQL[3] - PostGIS[4] or Oracle Spatial[5]; and NoSQL spatial database, such as CouchDB - GeoCouch[6] or MongoDB. In other words, we are addressing a problem of geographical data interoperability from highly heterogeneous information sources. At least two strategies may be used to solve this problem. One of them would be to employ a mediator-wrapper architecture [Wiederhold 1992], in which a wrapper would be written to communicate with the NoSQL spatial database, and integrate all database schemas into a common, single relational data schema.

The second strategy would be to implement OGC interoperability services[7] among spatial data, such as Web Map Service (WMS) and Web Feature Service (WFS) on the NoSQL spatial database layer, so as to integrate it to the SQL spatial DBMS by means of a map server; as for instance, GeoServer. This second solution allows any client that implements the WMS and WFS services, for example, the OpenLayers, to submit a query by using the same syntax for SQL and NoSQL spatial databases.

Given these two strategies, we have opted for the second one, which is our main contribution in this paper. To the best of our knowledge, there has been no such NoSQL and SQL spatial database integration using standard OGC web services so far. Consequently, the main contributions of this paper include:

- the implementation of a service layer (OGC WMS and WFS) for the NoSQL CouchDB-GeoCouch database;
- the design and implementation of an architecture to enable interoperability between spatial data stored in SQL and NoSQL databases through OGC services standards; and
- the implementation of a Web map viewer to deploy spatially-aware applications using the proposed interoperable architecture.

[1] The Apache CouchDB Project, http://couchdb.apache.org/
[2] The MongoDB Official Website, http://www.mongodb.org/
[3] The PostgreSQL OpenSource Database, http://www.postgresql.org/
[4] The PostGIS support for geographic objects to PostgreSQL, http://postgis.refractions.net/
[5] The Oracle Spatial, http://www.oracle.com/technetwork/database/options/spatial/overview/introduction/index.html
[6] The GeoCouch – A Spatial index for CouchDB, https://github.com/couchbase/geocouch/
[7] OGC Interoperability Services, http://www.opengeospatial.org/

The remaining of the paper is organized as follows: section 2 discusses related work. Section 3 focuses on the proposed architecture. Section 4 addresses a case study to validate the proposed ideas. Finally, section 5 concludes the paper and points out further work to be undertaken.

2 Related Work

The integration of geographic data stored in different information sources constitutes an old challenge for the geospatial data community; a challenge that has been extensively approached in the literature. An important work was proposed by the project SANY [Havlik et al. 2009]. In this project, a service was designed to provide a single point of access to data spread across the various nodes of a network of sensors. However, this service only supports data provided by the standard Sensor Observation Service[8]. On the other hand, the ORCHESTRA project [Usländer 2007] describes a spatial data infrastructure for risk management applications. The architecture used for its implementation permits the addition of geographic data coming from different sources of information. However, the data must be provided in the form of feature types encapsulated in OGC web services so as to be associated with the infrastructure.

In recent years, the need to process and manage large volumes of data has called for the implementation of effective alternatives to accommodate these tasks. This need has contributed towards the popularization of cloud computing in the geospatial domain. For instance, the map/reduce functions have been used to carry out a number of tasks in the geographical domain, such as the generation of spatial indexes [Akdogan et al. 2010] [Cary et al. 2009], query processing [Jardak et al. 2010], and prediction of natural disasters [Hasenkamp et al. 2010]. However, none of these articles addresses the need for providing the interoperability of their data sets with other existing data.

Moreover, a NoSQL database application to the geospatial domain was proposed by [Miller et al. 2011]. In their work, spatial data are stored in a database implemented in CouchDB. The approach is to use a two-tier architecture for retrieving data from mobile devices. However, in that work there is no interoperability between SQL and NoSQL spatial databases.

The increasing volume of data provided by some geographic data applications has placed a great demand for new ways of storing and managing this kind of information. Lately, these tasks are being addressed via NoSQL databases. Currently, the data offered by this type of database can only be accessed through its native interfaces, which limits access by users and its interoperability with data coming from other platforms. This limitation reinforces the need for a service that allows geographic data to be retrieved using open and standardized interfaces, for which no knowledge of data storage is required.

[8] OGC Sensor Observation Service, http://www.opengeospatial.org/standards/sos/

3 Proposed Architecture

This section describes the architecture used to solve the interoperability problem addressed by this paper. The architecture of our system was developed on three layers: application, service, and persistence. Figure 1 shows the proposed architecture.

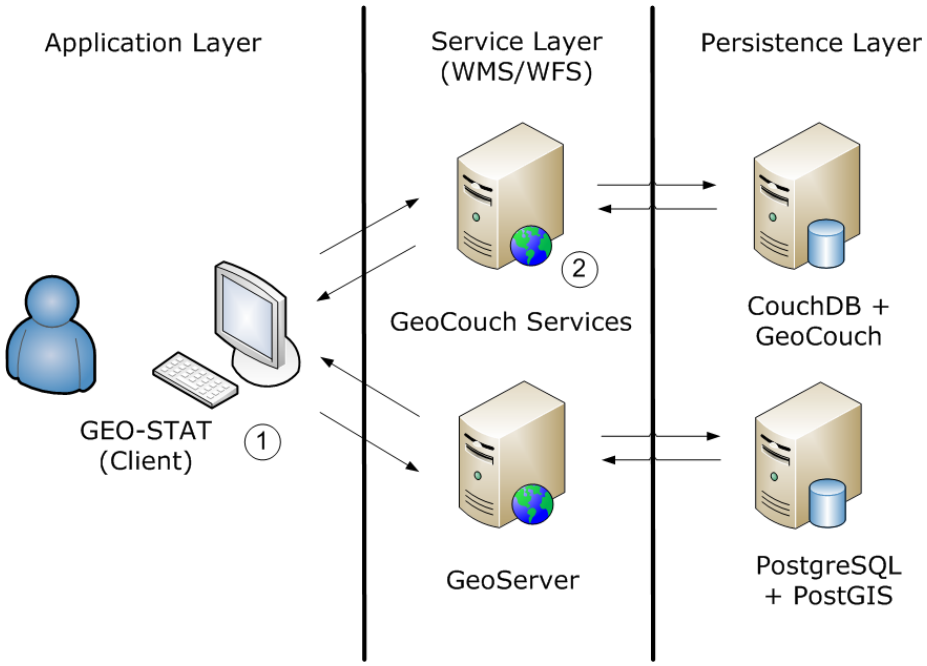


Figure 1: Three-tier architecture for interoperability of spatial data stored in SQL and NoSQL databases.

The application layer is responsible for the interaction between users and services. In our prototype, we used GEO-STAT (**Geo**graphic **S**patio-**T**emporal **A**nalysis **T**ool), a Web map viewer that we have developed (see component 1 of Figure 1). The GEO-STAT tool is based on the Google Maps API, and it works with any server that offers spatial data via WMS and WFS services. This tool provides components to visualize spatial data and spatiotemporal data. It also allows the implementation of spatial queries and the application of spatial filters. In addition, the tool offers an intuitive interface for data mining, based on spatio-temporal clustering and association rules, enabling the visualization of results through map layers. This makes possible, for instance, the implementation of comparative studies between transactional and derived data. Finally, GEO-STAT enables the immediate, practical and intuitive integration and visualization of spatial data available on any publicly accessible server that offers WMS and WFS services.

The service layer defines an interface of how certain features can be accessed by the application layer. Our main contribution lies on this layer, through the **GeoCouchServices** module, a spatial data server that implements WMS and WFS services for the NoSQL GeoCouch database. By means of GeoCouchServices one can

pose queries to a NoSQL database with the same syntax used to query a SQL database in a simple and transparent manner (see Figure 1). The syntax is defined by WMS and WFS standards. It is simple because only the service operations (e.g. GetCapabilities, GetMap, and GetFeature) must be invoked in order to formulate both spatial and non-spatial queries. Transparency to the user is obtained since these services work regardless of the data sources (e.g. GeoServer, Map Server, and GeoCouchServices). It is worthwhile mentioning that the proposed integration between SQL and NoSQL databases is also applicable to non-spatial data.

The GeoCouchServices was developed according to the Model-View-Controller (MVC) architectural pattern. Its purpose is to separate business logic from presentation logic and from application flow control. Figure 2 shows the dependence relationships and the architecture of the GeoCouchServices model (highlighted), component 2 of the architecture, described in Figure 1, for service requests.

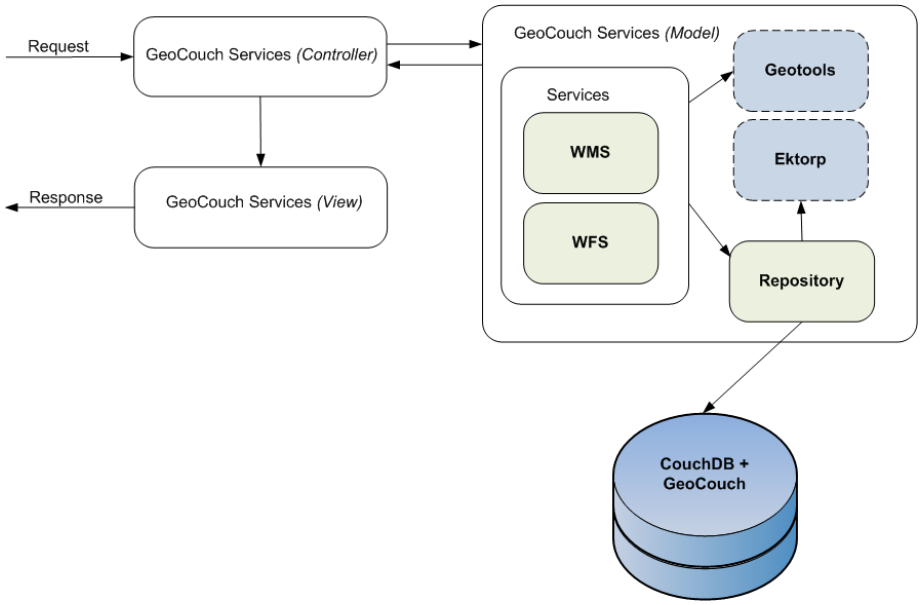


Figure 2: GeoCouchServices MVC architecture.

Upon receiving a request from the application layer, the controller of this service analyzes the request and redirects it to its model responsible for carrying out the request. The service first checks whether the attributes needed to meet the request have been provided. In case a required attribute has not been provided, a service exception is transmitted and a response is generated in the required format. Whatever the outcome, the controller receives a response from the model and forwards the response to the application layer.

For instance, when a GetMap request – including all its mandatory parameters – is submitted, the GeoCouchServices forwards it to the WMS module. The WMS module – via the Repository module – performs a Bounding Box search for the

requested layer in GeoCouch which then returns a file in the GeoJSON format (an open format for encoding a variety of geographic data structures). The Repository module performs a parsing of the GeoJSON file, and transforms it into a collection of features. This collection is then sent to the WMS module which generates the requested map.

Regarding the GeoCouchServices, versions 1.3.0 and 1.1.0 of the WMS and WFS services were implemented, respectively. The implementation of the WMS protocol included only the mandatory operations (GetCapabilities and GetMap). The optional GetFeatureInfo operation is currently being developed. We have also implemented the read-only WFS protocol, including the operations GetCapabilities, DescribeFeatureType and GetFeature.

To implement the WMS and WFS services, we used the GeoTools library [Turton 2008]. The Repository module is a component of the model responsible for forwarding spatial queries to GeoCouch. It is also responsible for querying non-spatial data in CouchDB; through the EKTORP library[9].

At the persistence layer we used CouchDB-GeoCouch for NoSQL data. Despite the possibility of manipulating spatial data in NoSQL with CouchDB and MongoDB, we preferred the former because of the existence of a more complete API that supports most types of existing spatial data.

CouchDB is a document-oriented schema free database. A database is stored as a collection of documents JSON (JavaScript Object Notation), and all interaction is performed entirely by using the HTTP protocol through a RESTful interface [Fielding 2000]. The data are indexed and searched by setting map-reduce views, similar to stored procedures. A view consists of a map function and, optionally, of a reduce function.

In the proposed architecture, we used a spatial extension for CouchDB, called GeoCouch. This architecture stores documents in the GeoJSON format. The GeoJSON[10] emerged as a simple pattern of spatial data format for the Web. This format can represent the following geometric types: point, multipoint, line, multiline, polygon, and multipolygon [Mische 2011].

4 Case Study

This section presents a case study aiming to validate the solution proposed in this paper.

Setup

We configured two servers; each providing WMS and WFS services. The first server used a SQL database; while the second one used a NoSQL database. Both servers stored spatial records about the Brazilian state of Paraíba, including all its 223 municipalities, the highways that cross the state, and all fire outbreaks detected in the state in 2010. All

[9] EKTORP Library – Java API for CouchDB, http://code.google.com/p/ektorp/

[10] GeoJSON – JSON geometry and feature description, http://geojson.org/

records are stored using the WGS84 projection. The records used are real-world data, and were obtained from the Water Management Executive Agency of the State of Paraíba (AESA) and from the National Institute for Space Research (INPE).

For the server that stores a SQL database, we have used the PostgreSQL 8.4 DBMS with a PostGIS spatial extension, version 1.5. In this server, OGC services were made accessible via GeoServer 2.1.0 map server. For the server with a NoSQL database, we have used the CouchDB database with a GeoCouch spatial extension made available by the CouchBase 1.1 package. The OGC services were available from the GeoCouchServices.

Based on this previously established design, we conducted two experiments for the present case study. The goal of the first experiment was to observe the functionality of the OGC requests made to the WMS and WFS services offered by GeoCouchServices. The second experiment evaluated the possibility of interoperability between spatial databases based on SQL and NoSQL.

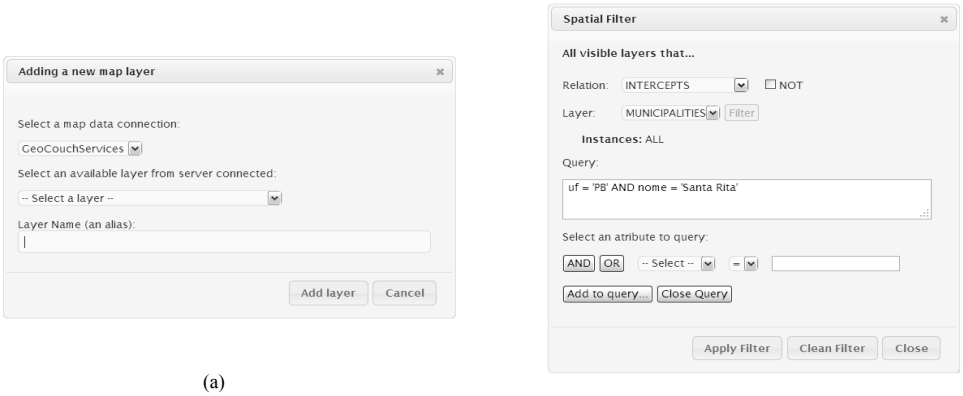
Experiment 1: Checking OGC requests placed to WMS and WFS services

Functional tests were performed on the implemented GeoCouchServices accessing the NoSQL server. The result set was compared to GeoServer in order to validate the accuracy of our implementation. These tests aimed at exploring GetCapabilities and GetMap requests from the WMS service; and GetFeature from WFS, through the use of resources from both servers by means of the GEO-STAT map viewer.

Two servers were configured using the GEO-STAT environment: the server based on GeoServer (SQL), and the other one based on GeoCouchServices (NoSQL). At this stage, for each server, it was only necessary to define an identifying name (alias) for the connection and to provide a way to access the server.

Once the connection configuration through GEO-STAT was established, it was possible to add geospatial layers, and then run the required tests. On selecting a layer to be displayed on the map, the GEO-STAT used the GetCapabilities request (WMS) to return to the list of available layers. Figure 3(a) shows how layers are added using our map viewer. The list of layers available is generated by the application using the response from the GetCapabilities request. Figure 3(b) shows how a spatial filter may be applied to the selected layers. It is possible to spatially filter all added layers in the map (regardless of the source server) by applying a selection filter in to one of them, e.g. filter by municipalities layer where cities with ‘uf’ attribute equals ‘PB’ (Paraíba), and ‘name’ attribute equals ‘Santa Rita’.

Figure 4 shows a map containing the municipalities in the state of Paraíba (223 multipolygons); and the highways that cross the state (959 multilines). These data were obtained using the GetMap request (WMS) sent to the GeoCouchServices.



(a) (b)

Figure 3: Some GEO-STAT forms to: a) add layers; b) apply a spatial filter to the added layers.

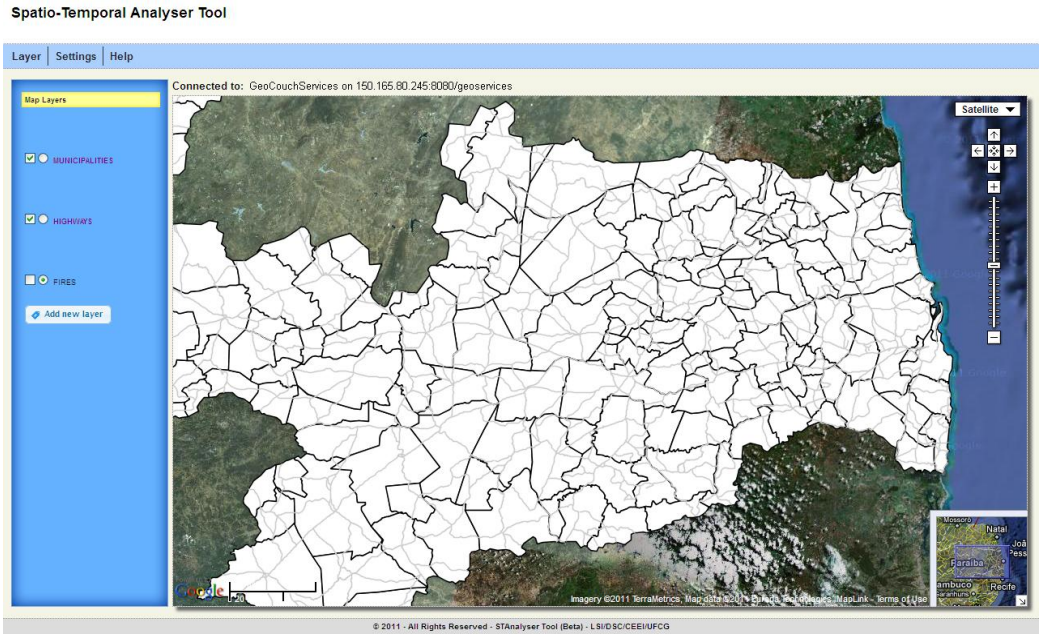


Figure 4: Municipalities and highways in the state of Paraíba provided by the GeoCouchServices using the GEO-STAT map viewer.

The exploitation of the GetFeature (WFS) request was also made possible through the GEO-STAT viewer, which allows us to specify a query using its intuitive graphical interface shown in Figure 3 (b).

Figure 5 shows the result of a query which inquires about all fires that happened in the city of Santa Rita in 2010 (141 points). The data are received by the application as a response of a GetFeature (WFS) request sent to the GeoCouchServices.

To perform this query we first added the layers MUNICIPALITIES and FIRES from GeoCouchServices. Then we applied a spatial filter based on municipalities layer

where the attribute name is equals to 'Santa Rita'. The GEO-STAT viewer, through the GetFeature request, receives all geometry from MUNICIPALITIES corresponding to applied filter and, using again the GetFeature request, uses this received response to request geometries from FIRES that spatially intersects with the geometry of the city of Santa Rita.

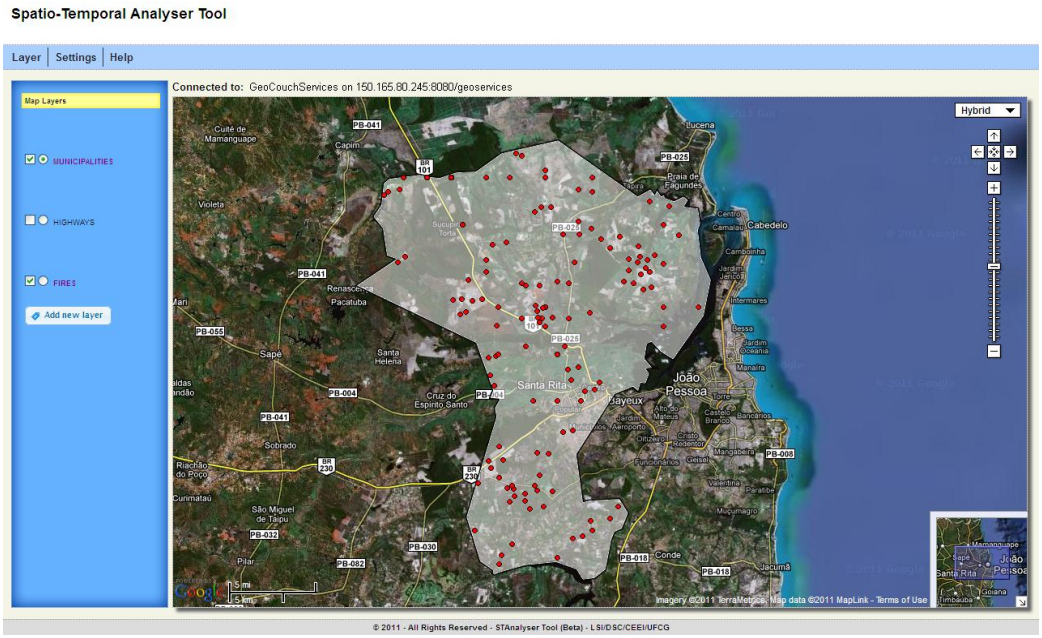


Figure 5: Fires occurred in Santa Rita in 2010 viewed in GEO-STAT, using data provided by the GeoCouchServices.

The comparative tests using the WMS and WFS requests showed that the GeoCouchServices worked satisfactorily, and returned the information similar to GeoServer, as it was expected.

Experiment 2: Evaluation of interoperability between the GeoCouchServices and the GeoServer

To evaluate the interoperability between GeoCouchServices and the GeoServer, i.e., the interoperability between spatial databases based on SQL and NoSQL, we posed queries using WMS and WFS services so that these could access spatial data from both servers.

Since our main goal is to analyze interoperability, we did some modifications on the data stored in the servers. In the first server, based on SQL, we left available only data related to municipalities and highways in the state of Paraiba. In the second server, based on NoSQL, we left available only data about fire outbreaks.

Again we used the GEO-STAT viewer to carry out this assessment. From it, we inserted into the map the MUNICIPALITIES and HIGHWAYS layers provided by the GeoServer. Then we inserted into the same map the FIRES layer made available from the GeoCouchServices. From this point onwards, we made the following spatial query:

Show all fires detected in the city of Monteiro in 2010.

This query may be formulated in the same way as shown in Figure 3 (b). The query is conducted by the GEO-STAT map viewer following two steps. In the first step, the GEO-STAT retrieves along with the GeoServer the geometry and the corresponding identifier of the city of Monteiro in the GML format. Afterwards, the GEO-STAT uses the GetMap (WMS) request to apply the filter in the MUNICIPALITIES layer providing the parameter 'featureid' in the request.

In the second step, the geometry that comes from the first step is used as a filter for a new query sent to the GeoCouchServices, where information is requested on all fires (geometries) that are inside the area represented by that geometry. Table 2 shows the GetFeature request to GeoCouchServices with the geometry of the city of Monteiro retrieved from GeoServer.

Table 2: GetFeature request to GeoCouchServices formed by data from GeoServer.

```
http://150.165.80.245:8080/geoservices/wfs?
   request=GetFeature&version=1.1.0&
   typeName=fires&outputFormat=GML3&
   FILTER=

<Filter xmlns="http://www.opengis.net/ogc"
        xmlns:gml="http://www.opengis.net/gml">
   <Intersects>
      <PropertyName>geometry</PropertyName>
      <gml:MultiSurface srsDimension="2"
                srsName="urn:x-ogc:def:crs:EPSG:4326">
         <gml:surfaceMember>
            <gml:Polygon>
               <gml:exterior>
                  <gml:LinearRing>
                     <gml:posList>
                     -7.94818296 -37.34987508
                     -7.945308   -37.34184996
                     -7.94235897 -37.3172821
                     -7.93552302 -37.31436
                     -7.93376604 -37.30863384
                     ...
                     -7.94818296 -37.34987508
                     </gml:posList>
                  </gml:LinearRing>
               </gml:exterior>
            </gml:Polygon>
         </gml:surfaceMember>
      </gml:MultiSurface>
   </Intersects>
</Filter>
```

The response to this query is received by GEO-STAT in GML format. It contains the geometries and corresponding identifiers (54 points). Then, a new GetMap request with the 'featureid' parameter is sent to GeoCouchServices. The parameter contains all the ids of geometries (fires) separated by commas. The result is shown in Figure 6.

We have successfully implemented interoperability between spatial databases based on SQL and NoSQL in a simple and transparent manner for the user application, satisfying, as a result, our case study and validating the proposed solution.

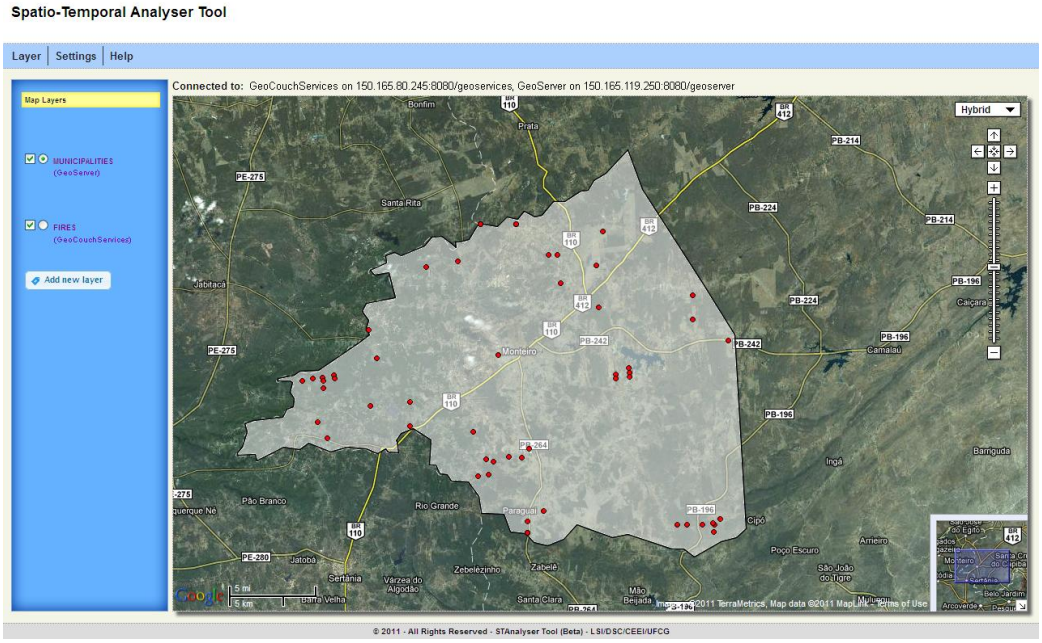


Figure 6: Fires occurred in Monteiro in 2010 using data provided by the GeoCouchServices and the GeoServer.

5 Conclusion and Future Work

This paper proposed a solution that enables interoperability between geographic data stored in SQL and NoSQL databases, using OGC WMS and WFS services.

The functional tests of requests to WMS and WFS services offered by NoSQL server have showed that they work satisfactorily, returning information in much the same way the GeoServer does. Additional tests have demonstrated that it is possible to achieve interoperability between spatial databases based on SQL and NoSQL in a simple and transparent way that will certainly help the user application.

There are at present many ongoing research issues related to the proposed interoperability solution. An objective that will certainly be the focus of our future endeavors will be to conduct experiments on the performance and scalability of services delivered. Another important issue is related to how to add other NoSQL spatial databases such as MongoDB to our architecture.

References

Akdogan, A., Demiryurek, U., Banaei-Kashani, F., and Shahabi, C. (2010). Voronoi-based geospatial query processing with mapreduce. In *Proceedings of the 2010 IEEE*

Second International Conference on Cloud Computing Technology and Science, CLOUDCOM '10, pages 9–16, Washington, DC, USA.

Cary, A., Sun, Z., Hristidis, V., and Rishe, N. (2009). Experiences on processing spatial data with mapreduce. In *Proceedings of the 21st International Conference on Scientific and Statistical Database Management*, SSDBM 2009, pages 302–319, Berlin, Heidelberg. Springer-Verlag.

Dean, J. and Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51:107–113.

Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. PhD thesis. University of California, Irvine, USA

Hasenkamp, D., Sim, A., Wehner, M., and Wu, K. (2010). Finding tropical cyclones on a cloud computing cluster: Using parallel virtualization for large-scale climate simulation analysis. In *Proceedings of the IEEE 2nd International Conference on Cloud Computing Technology and Science*, CLOUDCOM'10, pages 201–208, Washington, DC, USA.

Havlik, D., Bleier, T., and Schimak, G. (2009). Sharing sensor data with sensors and cascading sensor observation service. *Sensors*, 9(7):5493–5502.

Jardak, C., Riihijärvi, J., Oldewurtel, F., and Mähönen, P. (2010). Parallel processing of data from very large-scale wireless sensor networks. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 787–794, New York, NY, USA. ACM.

Miller, M., Medak, D., and D., O. (2011). Two-tier architecture for web mapping with nosql database couchdb. *Geoinformatics Forum*, pages 62–71.

Mische, V. (2011). CouchDB and GeoCouch. Erlang Factory Lite Munich, http://www.erlang-factory.com/upload/presentations/359/geocouch-online.pdf.

Turton, I. (2008). Open Source Approaches in Spatial Data Handling. Chapter 8: GeoTools. In *Advances in Geographic Information Science* 2, Springer Berlin Heidelberg, pages 153–169.

Usländer, T. (2007). Reference model for the orchestra architecture. Available at: http://portal.opengeospatial.org/files/?artifact_id=20300.

Wiederhold, G. (1992). Mediators in the architecture of future information systems. *Computer*, 25:38–49.

The Spatial Star Schema Benchmark

Samara Martins do Nascimento[1], Renata Miwa Tsuruda[2], Thiago Luís Lopes Siqueira[2,3], Valéria Cesário Times[1], Ricardo Rodrigues Ciferri[2], Cristina Dutra de Aguiar Ciferri[4]

[1]Informatics Center, Federal University of Pernambuco, UFPE, 50.670-901, Recife, PE, Brazil, +55 81 2126-8430

[2]Department of Computer Science, Federal University of São Carlos, UFSCar, 13.565-905, São Carlos, SP, Brazil, +55 16 3351-8573

[3]São Paulo Federal Institute of Education, Science and Technology, IFSP, 13.565-905, São Carlos, SP, Brazil, +55 16 3351-9608

[4]Department of Computer Science, University of São Paulo at São Carlos, USP, 13.560-970, São Carlos, SP, Brazil, +55 16 3373-8172

{smn, vct}@cin.ufpe.br, {renata_tsuruda, ricardo}@dc.ufscar.br, prof.thiago@ifsp.edu.br, cdac@icmc.usp.br

Abstract. *Spatial Data Warehouses (SDWs) enable the simultaneous processing of multidimensional queries and spatial analysis. In the literature, little attention has been devoted to the development of benchmarks for analyzing the performance of query processing over SDWs. In this paper, we propose a novel benchmark, called Spatial SSB, designed specifically to perform controlled experimental performance evaluation of SDWs environments. The Spatial SSB proposes a non-redundant SDW schema and controls: the generation of data, the query selectivity and the data distribution in the extent. In addition, the Spatial SSB provides the increase of the data volume, varies the complexity of spatial objects' geometries, and generates a certain number of objects that intersect an ad hoc spatial query window.*

1. Introduction

The experimental performance evaluation of databases systems is carried out mainly using benchmarks to provide a supervised generation of synthetic data and a controlled execution of queries over the synthetic datasets [Barbosa, Manolescu and Yu, 2009]. According to Gray (1993), the research on conventional data warehouse (DW) reached maturity enough to motivate the creation of benchmarks focused in its analysis, such as the *TPC-H* benchmark [Poess, M. et al., 2000] and the *Star Schema Benchmark* (*SSB*) [O'Neil, P. et al., 2009]. Conventional DW stores strictly numeric and alphanumeric data that can be represented by standard data types of SQL. On the other hand, a spatial data warehouse (SDW) consists of a DW that stores spatial data in one or more dimensions or in at least one measure of a fact table [Stefanovic, N. et al., 2000] [Malinowski, E. et al., 2008]. In this sense, the storage of spatial data in DWs allows SOLAP (Spatial On-Line Analytical Processing) query processing, which are based on

predicates that refer to data stored as vector geometries and then enable the simultaneous processing of multidimensional queries and spatial analysis [Rigaux et al., 2002].

However, existing benchmarks for DW do not consider spatial predicates, and the single benchmark in the literature for SDW, called *Spadawan* (*Spatial Data Warehouse Benchmark*) [Siqueira et al., 2010], has some drawbacks, such as: (i) it does not support one-dimensional vector objects (e.g. lines); (ii) it does not enable the adjustment of the complexity of the spatial objects (i.e. number of points that compose the geometry of each spatial object), such as an increase in the number of vertices of polygons; (iii) it does not allow a controlled distribution of spatial data in the extent; (iv) it does not enable queries to retrieve spatial objects based on a given percentage of the extent and therefore on a given selectivity; and (v) it does not define a specific scale factor to generate increasing volumes of spatial data. These are important issues that are tackled by our proposed benchmark.

In this paper, we propose the *Spatial Star Schema Benchmark* (*Spatial SSB*) to evaluate the performance of SOLAP queries over SDWs. Our benchmark extends the *SSB* to enable the storage and the processing of spatial data in dimension tables. The *Spatial SSB* manipulates only synthetic data, ensuring an accurate control over the selectivity of both conventional and spatial data. Also, it defines specific characteristics that can significantly degrade the performance, e.g. the increase of data volume and the increase of the complexity of polygons. Furthermore, aiming at generating synthetic data, we developed a data generator called *Spatial Geometry Generator*, used to produce the location and distribution of *regions*, *nations*, *cities*, *streets* and *addresses*, which are represented respectively by the following spatial data types: polygons, lines and points. Also, the *Spatial SSB* provides predefined spatial hierarchies, e.g. *region_geo* $\preceq$ *nation_geo* $\preceq$ *city_geo* $\preceq$ *street_geo* $\preceq$ c_*address_point_geo*, with the granularity level of *region* being the highest and the granularity level of *address* being the lowest. Regarding the workload, the proposed SOLAP queries of the *Spatial SSB* were obtained by modifying the existing *SSB* queries, reusing the complex operations regarding conventional data and additionally including spatial predicates in each query, to allow the evaluation of topological relationships among spatial attributes.

In order to investigate the impact of different properties for generating synthetic data, we conducted two experiments. Firstly, we investigated the effects of increasing the complexity of spatial data, i.e. the number of points that compose the geometry of each spatial object. Secondly, we tackled the increase of the number of spatial objects according to a given scale factor. The test configurations included the *Spatial SSB*'s SDW schema and a workload composed of spatial and multidimensional queries with controlled query selectivity as well.

This paper is organized as follows. Section 2 surveys related work, Section 3 presents the proposed benchmark *Spatial SSB*, Section 4 details the workload, Section 5 discusses the spatial data generation process, Section 6 describes the experiments using the *Spatial SSB* and finally, Section 7 concludes the paper.

2. Related Work

TPC-H [Poess, M. et al., 2000] and *SSB* [O'Neil, P. et al., 2009] are well-known benchmarks for conventional DWs. *TPC-H* is a decision support benchmark that consists in a suite of business oriented analytical queries and a voluminous fact constellation DW. It represents historical data from orders and sales of a company. *SSB* is based on *TPC-H*, but provides a simpler star

schema [Kimball and Ross, 2002] that was designed by applying several modifications on the original *TPC-H* schema. However, both the *TPC-H* and *SSB* benchmarks cannot be used for SDWs, since they do not allow the generation and storage of spatial data and do not provide the evaluation of spatial predicates.

The *Spadawan*, on the other hand, is a benchmark aimed at performance analysis of SDWs. However, this benchmark only supports point and polygon geometries to represent spatial data. Also, it does not allow varying the amount of points of polygons, thus making it impossible a further analysis over the complexity of spatial objects. *Spadawan* proposes the growth of the spatial data volume by the replication of the geographic objects. However, the increase of the spatial data volume is not based on the scale factor. Another limitation of *Spadawan* is that it does not include changes to all *SSB* queries and does not provide combinations of query windows that refer to different spatial granularity levels in the same query.

In this paper, we propose the *Spatial SSB*, a benchmark for SDWs based on a star schema that allows performance evaluation of queries involving spatial predicates. *Spatial SSB* advances in the state of the art overcoming all the aforementioned limitations, since it considers other spatial data types such as lines to represent *street* networks, ensures a greater control over the selectivity of both conventional and spatial data and generates multidimensional and spatial data automatically. In addition to ensuring the automatic generation of data using its own data generator, the *Spatial SSB* allows the investigation of how increasing data volumes impair query processing performance, by providing a means of varying the number of points denoting the shape of each spatial object (i.e. points of the geometry) or by selecting a database scale factor. Furthermore, the proposed set of *Spatial SSB's* queries also includes innovative aspects since they range from simple queries, with only one level of spatial granularity, to complex queries based on more than one query window that are related to different levels of granularity.

3. The *Spatial SSB*

The *Spatial SSB* schema is shown in Figure 1 and was adapted from the *SSB* schema to include spatial data. It is composed of a fact table *Lineorder*, two dimension tables to store conventional data (i.e. *Part* and *Date*) and six spatial dimension tables to store geometries (i.e. *Customer*, *Supplier*, *Region*, *Nation*, *City* and *Street*). The tables *Customer* and *Supplier* reference the spatial dimension tables *Region*, *Nation*, *City* and *Street* through foreign keys, maintain conventional attributes and the spatial attributes *c_address_point_geo* and *s_address_point_geo* that store the geometries of *Customer* and *Supplier addresses*, respectively. The spatial dimension tables were created following a predefined spatial hierarchy (e.g. *region_geo* $\preceq$ *nation_geo* $\preceq$ *city_geo* $\preceq$ *street_geo* $\preceq$ s*_address_point_geo*) according to the granularity levels. This hierarchy is defined in terms of the *containment* spatial relationship [Malinowski, E. et al., 2008].

The *Spatial SSB* schema is considered hybrid since it eliminates any redundancy in the storage of geometries. As the separate storage of spatial and conventional attributes has been recommended in SDW [Siqueira et al., 2009], we have designed a non-redundant schema based on the claim that computing additional joins is less costly than storing a large amount of redundant spatial data in the spatial dimension table and processing them to answer SOLAP queries. However, we have not created a spatial dimension table for *Customer* and *Supplier addresses* because all spatial objects that represent them are distinct, are points and have a 1:1 association with the dimension table primary key values. For this case, the joint storage of spatial and conventional data does not impair the performance of SOLAP queries [Mateus et al., 2010]. For

each spatial dimension table, a specific spatial data type was used to represent the spatial attribute. Polygons were used in *regions*, *nations* and *cities*, while lines modeled *streets* and points represented *addresses*.

The cardinality of the Spatial SSB schema depends on the *conventional* scale factor (CSF) that corresponds to the *SSB*'s scale factor and on the introduced *spatial* scale factor (SSF). The CSF and the SSF may vary independently for conventional and spatial data. However, SSF must be equal or less than CSF to guarantee a 1:1 association among addresses (i.e. addresses of suppliers and customers) considering conventional and spatial data. For greater CSF or SSF values, larger data volumes will be generated (e.g., SSF = 10 generates ten times more spatial data than SSF = 1). For instance, it is possible to increase the number of spatial objects, to assess how an increasing number of geometries impact the query processing performance.

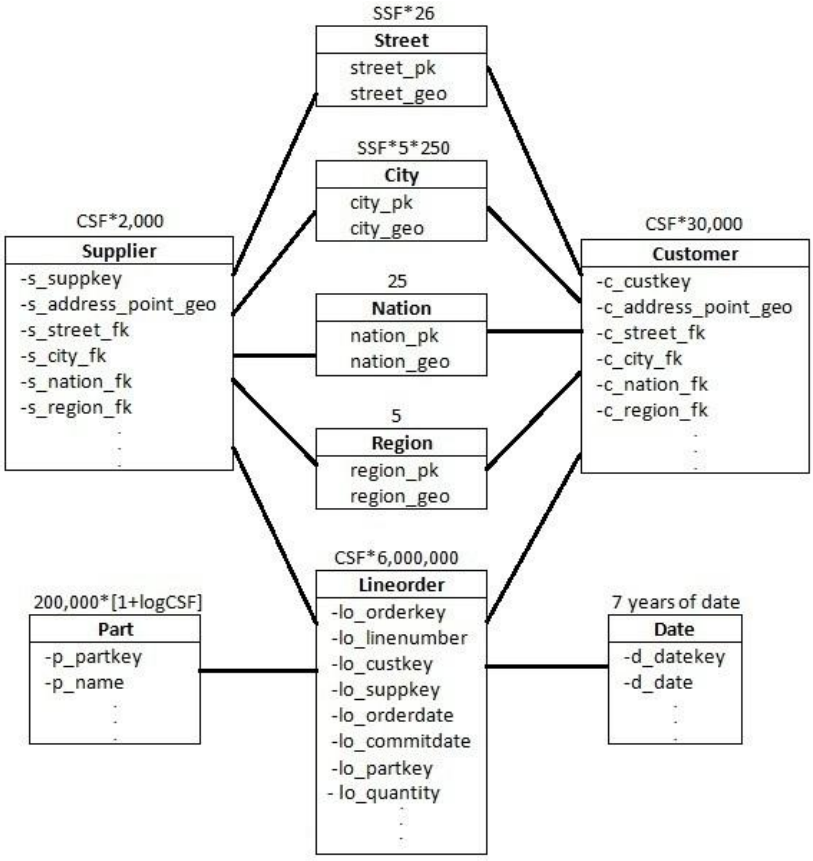


Figure 1. *Spatial SSB* schema

4. Workload

The *Spatial SSB* extends all the *SSB* queries by including spatial predicates based on different query windows (QWs), according to the granularity of the spatial dimension tables and also considering an empty area. The empty area represents oceans, aiming at identifying how an area without intersection with the spatial objects can impact the query processing cost and the query selectivity. The QWs can be (i) predefined, or (ii) can be generated and placed on the extent to comply with a given query selectivity. Each QW overlaps a specific area of the *extent*, retrieving a number of spatial objects and evaluating the spatial relationship *intersection.*

For the predefined QWs, five of them correspond to a different granularity level (*i.e. region*, *nation*, *city*, *street* and *address*), and the last QW intersects the empty area. These six QWs are quadratic, have a correlated distribution with spatial data and their sizes are proportional to the spatial granularity. Aiming at controlling the query selectivity, the *Spatial SSB* enables the retrieval of spatial objects based on a given percentage of the spatial data volume. Thus, the *Spatial Geometry Generator* computes the QW that will retrieve a given number of spatial objects. As a result, the acquisition of a number of objects through the use of an ad hoc query window is not defined a priori, but can vary according to user requests.

The *Spatial SSB* queries assess the performance of conventional and spatial predicates. Regarding the selectivity of a query, it is given by multiplying the Filter Factor (FF) and the cardinality of the table, then obtaining the number of required tuples from the fact table. FF is calculated from the conventional and spatial predicates chosen, which determine the conventional filter factor (CFF) and the spatial filter factor (SFF), respectively. As a result, FF = CFF * SFF. Note that the predefined query windows produce fixed values for the SFF, while the query windows generated to comply with a given selectivity vary the values of the SFF and they can reduce or increase the value of the FF. The *Spatial SSB* queries have the additional properties: use of one or two query windows and the definition of queries for each spatial data type that enable the query selectivity variation.

The queries are shown in Figures 2 to 6 and described in Table 1. The query selectivity values for the six predefined QWs are available at http://gbd.dc.ufscar.br/spatialssb/. Figure 7 shows an example of predefined QW that intersects 5 regions (i.e. R1 to R5), but does not intersect the empty area (i.e. EA). In this example, there are five objects distributed in the extent. The replacement of the conventional predicate of the original queries Q1.1, Q2.1, Q3.1 and Q4.1 of the SSB produced different levels of granularities for the new queries of the Spatial SSB (i.e. *region*, *nation*, *city*, *street*, *address* and *empty area*), obtaining different selectivity results shown in Tables 2, 3, 4 and 5. The results given on the variation of selectivity are based on the modified conventional predicate, and for all examples, the region granularity level was used, without considering the empty area of the extent.

```
SELECT SUM (LO_EXTENDEDPRICE*LO_DISCOUNT) AS REVENUE
FROM LINEORDER, DATE, CUSTOMER, SUPPLIER
WHERE LO_ORDERDATE = D_DATEKEY
AND LO_SUPPKEY = S_SUPPKEY
AND LO_CUSTKEY = C_CUSTKEY
AND D_YEAR = 1993
AND LO_DISCOUNT BETWEEN 1 AND 3
AND LO_QUANTITY < 25
    AND INTERSECTS (REGION, QW1)
    AND INTERSECTS (NATION, QW2)
    AND INTERSECTS (CITY, QW3)
    AND INTERSECTS (STREET, QW4)
    AND INTERSECTS (ADDRESS, QW5)
    AND INTERSECTS (EMPTY_AREA, QW6)
```

Figure 2. Query Q1.1 of *Spatial SSB*

```
SELECT SUM (LO_REVENUE), D_YEAR, P_BRAND1
FROM LINEORDER, DATE, PART, SUPPLIER
WHERE LO_ORDERDATE = D_DATEKEY
AND LO_PARTKEY = P_PARTKEY
AND LO_SUPPKEY = S_SUPPKEY
AND P_CATEGORY = 'MFGR#12'
    AND INTERSECTS (C_REGION, QW1)
    AND INTERSECTS (C_NATION, QW2)
    AND INTERSECTS (C_CITY, QW3)
    AND INTERSECTS (C_STREET, QW4)
    AND INTERSECTS (C_ADDRESS, QW5)
    AND INTERSECTS (EMPTY_AREA, QW6)
GROUP BY D_YEAR, P_BRAND1
ORDER BY D_YEAR, P_BRAND1
```

Figure 3. Query Q2.1 of *Spatial SSB*

```
SELECT C_NATION, S_NATION, D_YEAR, SUM (LO_REVENUE)
AS REVENUE
FROM CUSTOMER, LINEORDER, DATE, SUPPLIER
WHERE LO_ORDERDATE = D_DATEKEY
AND LO_CUSTKEY = C_CUSTKEY
AND LO_SUPPKEY = S_SUPPKEY
    AND INTERSECTS (C_REGION, QW1) AND INTERSECTS (S_REGION, QW1)
    AND INTERSECTS (C_NATION, QW2) AND INTERSECTS (S_NATION, QW2)
    AND INTERSECTS (C_CITY, QW3) AND INTERSECTS (S_CITY, QW3)
    AND INTERSECTS (C_STREET, QW4) AND INTERSECTS (S_STREET, QW4)
    AND INTERSECTS (C_ADDRESS, QW5) AND INTERSECTS (S_ADDRESS, QW5)
    AND INTERSECTS (EMPTY_AREA, QW6)
AND D_YEAR >= 1992 AND D_YEAR <= 1997
GROUP BY C_NATION, S_NATION, D_YEAR
ORDER BY D_YEAR ASC, REVENUE DESC
```

```
SELECT D_YEAR, S_CITY, P_BRAND1,
SUM (LO_REVENUE – LO_SUPPLYCOST) AS PROFIT
FROM DATE, CUSTOMER, LINEORDER, PART, SUPPLIER
WHERE LO_CUSTKEY = C_CUSTKEY
AND LO_SUPPKEY = S_SUPPKEY
AND LO_PARTKEY = P_PARTKEY
AND LO_ORDERDATE = D_DATEKEY
    AND INTERSECTS (C_REGION, QW1) AND INTERSECTS (S_NATION, QW1)
    AND INTERSECTS (C_NATION, QW2) AND INTERSECTS (S_CITY, QW2)
    AND INTERSECTS (C_CITY, QW3) AND INTERSECTS (S_STREET, QW3)
    AND INTERSECTS (C_STREET, QW4) AND INTERSECTS (S_ADDRESS, QW4)
    AND INTERSECTS (EMPTY_AREA, QW6)
AND (D_YEAR = 1997 OR D_YEAR = 1998)
AND P_CATEGORY = 'MFGR#14'
GROUP BY D_YEAR, P_BRAND1
ORDER BY D_YEAR, P_BRAND1
```

Figure 4. Query Q3.1 of *Spatial SSB*

```
SELECT D_YEAR, C_NATION, SUM (LO_REVENUE – LO_SUPPLYCOST) AS PROFIT
FROM DATE, CUSTOMER, LINEORDER, PART, SUPPLIER
WHERE LO_CUSTKEY = C_CUSTKEY
AND LO_SUPPKEY = S_SUPPKEY
AND LO_PARTKEY = P_PARTKEY
AND LO_ORDERDATE = D_DATEKEY
        AND INTERSECTS (C_REGION, QW1) AND INTERSECTS (S_REGION, QW1)
        AND INTERSECTS (C_NATION, QW2) AND INTERSECTS (S_NATION, QW2)
        AND INTERSECTS (C_CITY, QW3) AND INTERSECTS (S_CITY, QW3)
        AND INTERSECTS (C_STREET, QW4) AND INTERSECTS (S_STREET, QW4)
        AND INTERSECTS (C_ADDRESS, QW5) AND INTERSECTS (S_ADDRESS, QW5)
        AND INTERSECTS (EMPTY_AREA, QW6)
AND (P_MFGR = 'MFGR#1' OR P_MFGR = 'MFGR#2')
GROUP BY D_YEAR
ORDER BY D_YEAR
```

Figure 5. Query 4.1 of *Spatial SSB*

```
SELECT D_YEAR, C_NATION, SUM (LO_REVENUE – LO_SUPPLYCOST) AS PROFIT
FROM DATE, CUSTOMER, LINEORDER, PART, SUPPLIER
WHERE LO_CUSTKEY = C_CUSTKEY
AND LO_SUPPKEY = S_SUPPKEY
AND LO_PARTKEY = P_PARTKEY
AND LO_ORDERDATE = D_DATEKEY
        AND INTERSECTS (C_REGION, QW1) AND INTERSECTS (S_REGION, QW1)
        AND INTERSECTS (C_NATION, QW2) AND INTERSECTS (S_NATION, QW2)
        AND INTERSECTS (C_CITY, QW3) AND INTERSECTS (S_CITY, QW3)
        AND INTERSECTS (C_STREET, QW4) AND INTERSECTS (S_STREET, QW4)
        AND INTERSECTS (C_ADDRESS, QW5) AND INTERSECTS (S_ADDRESS, QW5)
        AND INTERSECTS (EMPTY_AREA, QW6)
AND (P_MFGR = 'MFGR#1' OR P_MFGR = 'MFGR#2')
GROUP BY D_YEAR
ORDER BY D_YEAR
```

Figure 6. Query Q4.3 of *Spatial SSB*

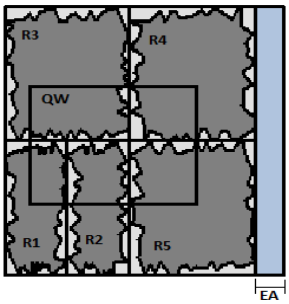


Figure 7. QW on the *extent of Regions*

Table 1. Queries of *The Spatial Star Schema Benchmark*

Query	Characteristics
Q1.1	Returns the increased revenue that is resulted from the elimination of discounts in the company, at a certain scale of percentage for products shipped in a given year, intersects for the different levels of granularity, as shown in Figure 2.
Q1.2	Changes made to the conventional predicates of the query Q1.1, as follows: (i) *d_yearmonthnum* = 199401, (ii) *lo_quantity* is between 26 and 35 and (iii) *lo_discount* is between 4 and 6.
Q1.3	Modifies the conventional predicates of *Spatial SSB* query of type Q1.1, are used: (i) *d_weeknuminyear* = 6; (ii) *d_year* = 1994; (iii) *lo_quantity* is between 36 and 40, and (iv) *lo_discount* is between 5 and 7.
Q2.1	Compares the revenues for some brands of products, grouped by years of orders, intersects for the different levels of granularity, as shown in Figure 3.
Q2.2	Changes made to the previous query type by using *p_brand1* between 'MFGR#2221' and 'MFGR#2228'.
Q2.3	This is obtained by changing the traditional predicate of query Q2.1 of *Spatial SSB*. The modification is done to use *p_brand1* = 'MFGR # 2221'.
Q3.1	Provide the revenues associated with sales and order transactions for a certain period of time. This query uses two QWs for each granularity level, with the spatial predicate intersects, as shown in Figure 4.
Q3.2	Changes were made at conventional predicate that was before analyzed in a certain range of years in order to be computed according to months per year to vary the selectivity.
Q4.1	Query Q4.1 aims to measure the profit from the subtraction of costs from revenues. It is illustrated in Figure 5.
Q4.2	It is an extension of Q4.1, changing the conventional predicate to consider the calculation of profits within a period of time, i.e. in 1997 or 1998.
Q4.3	The conventional and spatial predicates were changed of query Q4.1, to obtaining different selectivity results, as shown in Figure 6.

Table 2. Variation of Selectivity of Query Q1

Query Q1	Selectivity
Q1.1	0.39% to 1.95%
Q1.2	0013% to 0065%
Q1.3	0.0015% to 0.0075%.

Table 3. Variation of Selectivity of Query Q2

Query Q2	Selectivity
Q2.1	0.16% to 0.80%.
Q2.2	0032% to 0.16%
Q2.3	0.02% to 0.1%

Table 4. Variation of Selectivity of Query Q3

Query Q3	Selectivity
Q3.1	3.43% to 85.71%
Q3.2	0.048% to 1.19%

Table 5. Variation of Selectivity of Query Q4

Query Q4	Selectivity
Q4.1	1.6% to 40%
Q4.2	0.046% to 11.42%
Q4.3	0.05% to 1.14%.

5. Data Generation

Aiming at automating the conventional data loading process of the *Spatial SSB*, we implemented a component called *VisualTPCH+SSB*, which is responsible for creating the schemas and loading data from the *SSB* data generator. This component is described in Section 5.1. In addition, for the generation of spatial data, we implemented the *Spatial Geometry Generator* to produce the location and distribution of geometries for *regions*, *nations*, *cities*, *streets* and *addresses*, which is detailed in Section 5.2. Together, these components give rise to *VisualSpatialSSB* tool that is available at http://gbd.dc.ufscar.br/spatialssb/.

5.1 The *VisualTPCH+SSB* Component

The *VisualTPCH+SSB* component manages the storage, generation and load of data for *SSB* schema. The schema is graphically displayed and significant features are available: deleting attributes of a table, renaming tables and deleting tables. The component also offers a graphical visualization of aggregation levels of the generated schema and interactive features, as highlighting the direct ancestral or descendent of the graph of materialized views, deleting vertices and visualizing the SQL command that generates a given vertex (i.e. materialized view) are available. Finally, the data generator of *VisualTPCH+SSB* loads a dataset for each of the considered *benchmarks*.

5.2 The Spatial Geometry Generator

The *Spatial Geometry Generator* component of the *Spatial SSB* benchmark generates rectangles to guide the creation of spatial geometries. Each rectangle is a MBR (*Minimum Bounding Rectangle*) as described as follows and shown in Figure 8. The generation of spatial data contained in the MBRs is based on the *quadtree* space-partitioning model that considers quadrants that are formed from a recursive partitioning of the *extent* [Ghazel, M. et al., 2000]. The space is recursively decomposed into four sub-regions, called "quadrants", which may have different sizes, but are similar in its shape. Figure 8 also illustrates the empty area in a dark color.

The *extent* size is 0 to 1 in both horizontal and vertical axes. The *extent* is partitioned according to a given predefined amount of *regions*, *nations* and *cities*. The spatial data are generated according to a predefined spatial hierarchy, e.g. *region_geo* $\preceq$ *nation_geo* $\preceq$ *city_geo* $\preceq$ *street_geo* $\preceq$ c_*address_point_geo*. The number of *regions* is not limited, enabling to partition the *extent* area in *n* disjoint *regions*. Initially, the partition occurs in the *x* axis, dividing the *extent* in two sub-regions (Figure 8b) and after that, one partition in the *y* axis, forming three sub-regions (Figure 8c), and finally, another partition in the *y* axis, forming four sub-regions (Figure 8d). If necessary, a new partition occurs in the southwest sub-region, dividing it into two sub-regions (Figure 8e), and following this, new partitioning may occur in this southwest sub-region and so on (Figure 8f e 8g). This partitioning is always done clockwise.

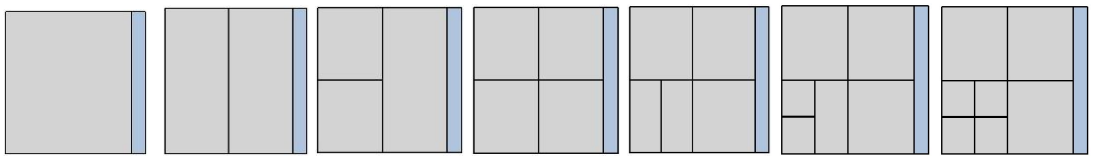


Figure 8. *Extent* partitioning

The distribution of spatial data in the MBR initially considers a margin of M = 0.1% in all sides of the MBR, and thus generates the points inside the margin limit M (Figure 9). Then, the total amount of points (i.e. the complexity of the polygon) is divided by the number of sides of the MBR (i.e. 4 sides), and the points are evenly distributed among the sides, as shown in Figure 10. This distribution of spatial data ensures that the generated geometries will be polygons. The points were generated applying a random function to one of the axes. For the *x* axis, we consider this coordinate growing and continuous and generate the *y* axis from the random function; and, when dealing with the *y* axis, we consider, now, this coordinate growing and continuous and vary the *x* axis from the random function, as can be seen in Figure 10. Figure 11 shows the algorithm to generate points.

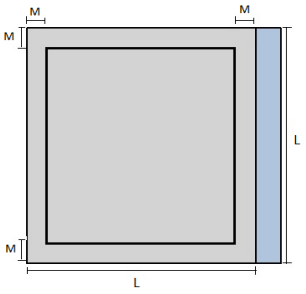


Figure 9. Margin of MBR, Where M = 0.01*L

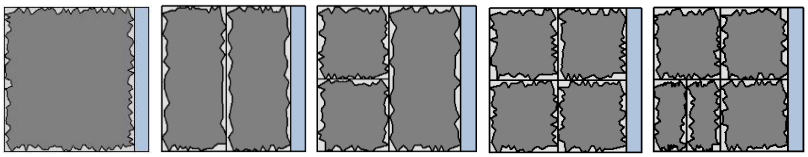


Figure 10. Generation of polygons on MBR

Algorithm 1: CreatePoint (n, x1, y1, x2, y2)

Input: n number of points (the polygon complexity); x1, y1, x2, y2 coordinates of points that form the *extent*.
Output: A file that stores the coordinates of points that compose the polygons.
1 *Read* the number of regions.
2 Margin ← 0.1%; // the margin in x and y axes are defined here.
3 Auxiliary ← n / 4
4 *Write* in the file the first point of the geometry.
5 *While* it does not reach the end of the last calculated point within the range of margin.
6 Uniformly and randomly *increment* a coordinate in an axis and *Generate* the points in the other axis.
7 *FinalResult* ← write in the file the generated points.

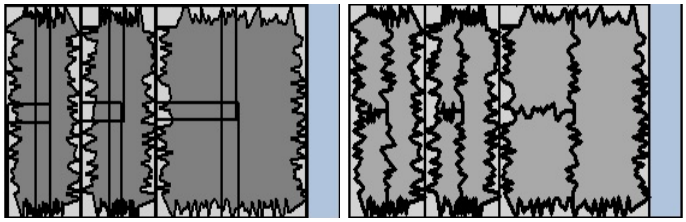
Figure 11. *Extent* partitioning algorithm

In Algorithm 1 of Figure 11, in line 1, the number of regions that are generated is read. In line 2, a margin of 0.1% in the MBR is created to accommodate the generation of points representing the spatial object's geometry. Therefore, margin sizes in the x and y axes are calculated and the ranges of the coordinates, in both *x* and *y* axes, are generated. In line 3, an auxiliary variable divides the total number of points by the number of sides (always considering a MBR) and each side will now have nearly the same number of generated points. In line 4, the first point of the geometry will be written to the file, as well as the *x* position (after the computation of the margin) and the *y* position. A loop is done to check if the generated points are still within the range referenced in this margin and if the points are being generated in a growing and continuous way in both axes.

The number of *nations* is not limited, enabling the partition of a given region in *n* disjoint *nations*. For generating the data distribution of *regions*, we slightly modified the algorithm to represent data distribution in MBRs (Figure 11), as follows. The subdivision of the MBRs is performed within the geometries for *regions*. A margin for *x* axis is obtained from the subtraction of the coordinates x_n and x_0 (last and first point in the axis x of the region), and then multiplying this result by 0.02. Similarly, a margin for *y* axis is obtained from the subtraction of the

coordinates y_n and y_0 (last and first point in the axis y of the region), and then multiplying this result by 0.02. With each new subdivision, new margins are generated and it is within this space that data distribution are generated. An example of this subdivision can be seen in Figure 12a, in which a MBR is shown with *regions* and within each *region* there are the MBRs of *nations*, in this example, three *nations* were generated by *region*. The points distributed into the margin to compose the boundaries of the polygons were generated using a random function that is similar to the random function of *regions*. The polygons of generated *nations* can be seen in Figure 12b.

The algorithm for generating *city* geometries that also were represented by polygons follows the generation of *nations*. The coordinates of *cities* use the points referring to the sides of *nation* geometries added to the generated points within the margin. The amount of points is previously known and is uniformly distributed to all sides of the polygon. Each *city* geometry contains lines to represent *streets*. We generated streets as a rectangular grid of lines that intersect each other, with the same number of lines in each axes. We consider a margin in both *x* and *y* axes to take it as starting points to the generation of lines. We also consider five thousand points to represent a *street*. The generation of lines can be seen in Figure 13. The generation of *addresses* (i.e. point geometries) considers a distribution of one address per street and no address is generated at the intersection of *streets*, as shown in Figure 14.



a. MBR **b. Points**

Figure 12. Generation of *nations*

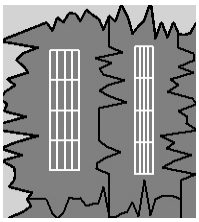


Figure 13. Street

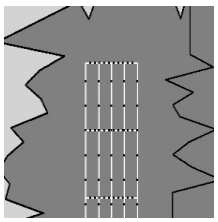


Figure 14. Address

6. Experimental Evaluation

The *Spatial Geometry Generator*, which is the spatial data generator of the *Spatial SSB*, allows that a varying number of spatial objects be distributed in the *extent* and each spatial object be represented by a varying set of points that represents the complexity of the spatial object. Thus, in addition to increasing the data volume by considering the scale factors CSF and SSF defined for the SDW schema of the *Spatial SSB* benchmark, this enlarged volume of data can also be achieved by varying the complexity of geometries of spatial objects.

The characteristics of the proposed *Spatial SSB* benchmark were investigated through two sets of experimental tests, considering three types of *Spatial SSB* queries: (i) The Query Q2.2 was chosen because it uses one QW for each level of granularity, and this QW is predefined; (ii) The Query Q3.1 was selected because two predefined QWs for a given level of granularity were used; and (iii) The Query Q4.3 was chosen because it uses two QWs related to different levels of granularity, and these ad hoc QWs were specified by an user defined percentage of intersection with the *extent*, which retrieved 5% of the spatial objects stored in the SDW. The goal of performing experiments based on the query Q4.3 is to increase the complexity of processing queries with low selectivity by using two QWs related to two different levels of spatial granularity, but that retrieve a small percentage of the spatial objects. The tests illustrate the flexibility of the proposed *Spatial SSB* benchmark for building ad hoc QWs, using predefined QWs and controling selectivity.

The first set of tests aimed at checking the impact of increasing the complexity of spatial objects on the query processsing of SDWs, i.e. increasing the number of points of each geometry of the spatial objects stored in SDWs. These test results are discussed in Section 6.1. The second set of tests investigated the impact of increasing the number of spatial objects. These test results are detailed in Section 6.2. Finally, all the experiments were conducted on a computer with 2.66 GHZ Intel Core i5 processor, 3GB of main memory, 5400 RPM SATA 320 GB hard disk, operating system Linux Ubuntu 9.10, PostgreSQL 8.2.5 and PostGIS 1.3.3.

6.1 Increasing the Complexity of Objects

In this section, we verify the impact of SOLAP query processing performance caused by an increasing number of points that represent each spatial object. We used a database with CSF and SSF equal to 1. The generation of spatial data created 5 disjoint *regions*, 5 *nations* per *region*, totaling 25 *nations*, 1,250 *cities*, 26 *streets*, 2,000 *addresses* for *Supplier* and 30,000 *addresses* for *Customer*. We investigated three configurations that varied from each other according to the number of points generated per region and per nation: 200 points, 20,000 points and 200,000 points. For each dataset, we collected the elapsed time in milliseconds.

The Query Q2.2 was issued against the nation granularity level, with a total of 25 *nations* in the *extent*. The results are shown in Table 6, indicating an increase in query processing cost of 289%, when this time is compared between the smallest number of points and the largest number of points. The greater the number of points used for representing spatial objects, the greater the data volume that impaired the query processing cost. Therefore, the *Spatial SSB* can be used to generate datasets storing spatial objects with distinct complexities, and this can introduce increasing query processing costs.

Another test was conducted, using the Query Q3.1 and two QWs for the same level of granularity. We considered the level of granularity of *regions*, obtaining a total of 5 spatial objects. Table 6 shows the performance results. An increase of 5,814% was obtained when the spatial object representation was changed from 200 to 200,000 points. Therefore we also can conclude that increasing the data volume through varying the complexity of spatial objects was directly related to processing performance losses for the Query Q3.1.

The Query Q4.3 proposes the use of two QWs for different levels of spatial granularity. For this test, it was considered the level of granularity of: (i) customer *region*, with a total of 5 spatial objects, and supplier *nation*, with a total of 25 spatial objects. Table 6 depicts the performance results, showing that there was an increase of 10,063% in the elapsed time of Q4.3 when the representation of an object changed from 200 points to 200,000 points.

Table 6. Elapsed time to process each SOLAP query (milliseconds)

Number of Points	Q2.2	Q3.1	Q4.3
200	363,060	53,299	18,362
20,000	494,942	56,812	24,621
200,000	1,414,542	3,152,581	1,866,190

6.2 Increasing the Number of Spatial Objects

In this section, we verify the impact of SOLAP query processing performance caused by an increasing number of spatial objects. We used the same workbench described in Section 6.1. However, we generated data with CSF and SSF equal to 2, which produced twice the data volume of the datasets described in Section 6.1. Therefore, the generation of spatial data created 5

distinct *regions*, 5 *nations* per *region*, totaling 25 *nations*, 2,500 *cities*, 52 *streets*, 4,000 *addresses* for *Supplier* and 60,000 *addresses* for *Customer*. Besides, we considered the level of granularity city, with 2,500 spatial objects, except for the query Q4.3 that used two levels of granularity: customers' *nation* and suppliers' *city*. For each dataset, we collected the elapsed time in milliseconds.

The performance results described in Table 7 and Figure 15 show that a significant increase in query processing costs was obtained for the CSF and SSF equal to 2, when compared to the same spatial complexity with a CSF and SSF equal to 1 (i.e. Table 6). For instance, the Query 2.1 spent 363,060 milliseconds for handling spatial objects composed of 200 points considering CSF and SSF equal to 1, while the same query spent 3,764,143 milliseconds for processing spatial objects composed of 200 points considering CSF and SSF equal to 2. Considering each query individually, Table 7 indicates an increase of 54% for the Query Q2.2 with regard to query processing costs, when comparing the smallest set of points, i.e. 200 points, with the largest, 200,000 points. For the queries Q3.1 and Q4.3, the increase was of 113% and 94%, respectively.

Table 7. Elapsed time to process each query (in milliseconds)

Number of Points	Q2.2	Q3.1	Q4.3
200	3,764,143	1,944,788	2,636,336
20,000	4,310,462	2,144,696	3,438,380
200,000	5,813,175	4,145,552	5,133,934

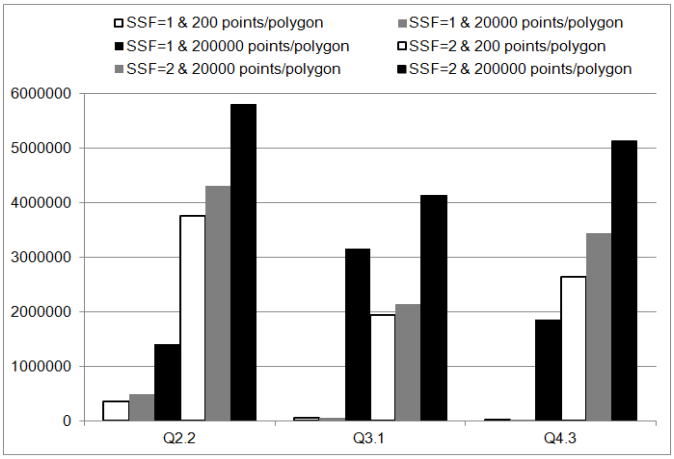


Figure 15. Performance comparison: datasets with scale factor 1 vs scale factor 2

It is noted that the increased volume of data, whether caused by the complexity of the spatial objects or caused by the number of spatial objects, highly impaired the SOLAP query processing performance. Therefore, the *Spatial SSB* can be used to generate datasets storing an increasing number of spatial objects, and this property introduces increasing query processing costs.

7. Conclusion

In this paper we proposed a new spatial data warehouse benchmark called *Spatial SSB* (Spatial Star Schema Benchmark). It is composed by a set of SOLAP queries that was derived from changes made to SSB workload to incorporate spatial predicates for different spatial granularity levels. The dataset is synthetic and is created by a specific data generator, called *Spatial Geometry*, to generate points, lines and polygons. The proposed benchmark allows controlling spa-

tial distribution, the geometric shapes, the data volume, the data complexity and the data selectivity encompassing the main spatial data types.

As future work, we aim to add other types of spatial hierarchies such as those described in [Malinowski, E. et al., 2005] [Malinowski, E. et al., 2008] [Stefanovic, N. et al., 2000]. Another indication of additional research is to incorporate different spatial data types such as complex polygons and vague spatial objects in data generation and in spatial and multidimensional query processing as well [Viswanathan and Schneider, 2011]. Also, an interesting investigation concerns the verification of how data distribution and some SOLAP query issues can affect the performance of a spatial data warehouse.

References

Barbosa, D., Manolescu, I, Yu, J. (2009) "Application Benchmark". Encyclopedia of Database Systems, Springer, p. 99-100.

Ghazel, M., Freeman, G.H. and Vrscay, E.R. (2000) "An effective hybrid fractal-wavelet image coder using quadtree partitioning and pruning". In: IEEE CCECE. p. 416-420.

Gray, J. (1993) "Database and Transaction Processing Performance Handbook". The Benchmark Handbook for Database and Transaction Systems, Morgan Kaufmann, 2nd Edition. p. 99-100.

Kimball, R. and Ross, M. (2002) "The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling". John Wiley & Sons, Inc.

Malinowski, E. and Zimányi, E. (2005) "Spatial Hierarchies and Topological Relationships in the Spatial MultiDimER Model". In: BNCDB. p. 17-28.

Malinowski, E. and Zimányi, E. (2008) "Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications (Data-Centric Systems and Applications)". Springer.

Mateus, R.C., Siqueira, T.L.L., Times, V.C., Ciferri, R.R. and Ciferri, C.D. (2010) "How does the spatial data redundancy affect query performance in geographic data warehouses?". In JIDM, v.1, n.3, p. 519-534.

O'Neil, P., O'Neil, E., Chen, X. and Revilak, S. (2009) "The Star Schema Benchmark and Augmented Fact Table Indexing". In: TPCTC. p. 237-252.

Poess, M. and Floyd, C. (2000) "New TPC benchmarks for decision support and web commerce". In SIGMOD Record, v.29, p. 64-71.

Rigaux, P., Scholl, M. and Voisard, A. (2002) "Spatial Databases with Application to GIS". Morgan Kauffman.

Siqueira, T.L.L., Ciferri, C.D., Times, V.C., Oliveira, A.G. and Ciferri, R.R. (2009) "The impact of spatial data redundancy on SOLAP query performance". In JBCS, v. 15, n. 2, p. 19-34.

Siqueira, T.L.L., Ciferri, C.D., Times, V.C. and Ciferri, R.R. (2010) "Benchmarking Spatial Data Warehouses". In: DaWaK. p. 40-51.

Stefanovic, N., Han, J. and Koperski, K. (2000) "Object-BasedSelective Materialization for Efficient Implementation of Spatial Data Cubes". In IEEE TKDE, v. 12, n. 6, p. 938-958.

Viswanathan, G., Schneider, M. (2011) "OLAP Formulations for Supporting Complex Spatial Objects in Data Warehouses". In: DaWaK. p. 39-50.

A Susceptible-Infected Model for Exploring the Effects of Neighborhood Structures on Epidemic Processes – A Segregation Analysis

Leonardo Bacelar Lima Santos[1], **Raian Vargas Maretto**[1], **Líliam César de Castro Medeiros**[1], **Flávia da Fonseca Feitosa**[1], **Antônio Miguel Vieira Monteiro**[1]

[1]Instituto Nacional de Pesquisas Espaciais – São José dos Campos, SP, Brasil

{santoslbl,raian,lccastro,flavia,miguel}@dpi.inpe.br

Abstract. *This work explores and analyzes the effects of neighborhood structures on disease spreading in a compartmental epidemic CA-model. The main goal is to investigate how different neighborhood configurations are able to affect the spatial and temporal distribution of infected and susceptible individuals and the chance of having members from these different groups interacting with each other. It uses the idea of "activity-space neighborhood", which extends traditional contiguity-based neighborhoods to capture interactions beyond those established in a residential environment. To depict the spatial distribution of infected and susceptible individuals along the simulation steps, we introduce the use of spatial segregation indices, traditionally adopted in urban studies, to an epidemiological context.*

1. Introduction and Motivation

Over the last decades, studies on epidemiology have recognized the importance of adopting a systemic view of epidemic processes. However, most of these approaches are based on ordinary differential equations or statistical models [19, 5, 26], which are unable to explore spatially-explicit patterns and interactions that are relevant to understand the dynamics of disease transmission.

To overcome these limitations, many studies have started using Cellular Automata (CA) models [35, 32] to obtain new insights on how epidemic processes evolve in time and space [22, 18, 16, 30, 33, 24]. Cellular automata are self-reproductive dynamic systems, where time and space are discretized [29]. They are composed of a lattice of cells, called cellular space, each one with a pattern of local connections to other cells, and subjected to given boundary conditions [33, 27]. Each cell can assume a state, among a enumerable set of states, which can change at every time-step according to local transition rules (deterministic or stochastic) based on the states of the cell and possibly of its neighbors. CA-based models have a long tradition in modeling and simulation of complex spatial phenomena, and its potential has been widely recognized in several fields of study [28, 1, 6, 34, 7].

In CA models, the concept of *neighborhoods* is a key component, since it determines how the elements that build up the model interact [14]. To say that two cells are neighbors means that one is exerting some sort of influence over the state (spatial, temporal and/or behavioral) of the other. Thus, it is an essential aspect to represent interactions between individuals in a society.

This paper investigates the effects of neighborhood structures on disease spreading by using a susceptible-infected (SI) epidemics CA-model. Despite its simplicity, the SI model can be used for early detection of infections diseases outbreaks [25] and has the advantage of being easily extended to models that accommodate additional categories [20], such as the susceptible-infected-susceptible (SIS) model – commonly used to represent bacterial infections –, the susceptible-infected-recovered (SIR) model – usually applied to viral infections –, or others epidemiological compartmental models [15]. The SI model can be also applied to problems of other nature, called general epidemic process, as forest fire propagation [2, 3] or the spread of information on a society [31].

This work dedicates particular attention to the spatial distribution of infected and susceptible individuals. Its main goal is to investigate how different neighborhood configurations are able to affect the spatial and temporal distribution of infected and susceptible people and the chance of having members from these different groups interacting with each other. It introduces the idea of "activity-space neighborhood", which extends traditional contiguity-based neighborhoods (e.g., Moore or Von Neumann) to capture not only interactions established in a residential environment, but also those resulting from other daily activities, such as work or school.

In addition, this study explores how the initial distribution of infected individuals influences the spatiotemporal patterns of disease spread. To depict the spatial distribution of infected and susceptible individuals along the simulation steps, we introduce the use of spatial segregation indices, traditionally adopted in urban studies, to an epidemiological context. Global and local spatial segregation indices are computed: while global indices express the segregation state for the city as a whole, local indices are able to depict the degree of segregation in different points of the simulated environment and can be visualized as maps [12].

The development of the model presented in this work was carried out using the TerraME (Terra Modeling Environment) platform [10]. TerraME is an extended CA-based computational framework for spatial dynamic modeling that implements the concepts of Nested Cellular Automata (Nested-CA). It is an extension of Lua programming language [17], and uses the geoprocessing library TerraLib [9] for handling geospatial data.

2. Methodology

The dynamics of disease dissemination is modeled by a two-dimensional cellular automata grid in which each cell represents an individual. For each time step, each individual can take one of two states: susceptible or infected. To keep the model as simple as possible, births and natural deaths were ignored, as well as incubation periods due to pathogen replication and possible mortality induced by disease.

To investigate the impact of different neighborhood structures on the dynamics of epidemic processes, we performed simulation experiments using two types of neighborhoods. The first one is the classical Moore neighborhood [35], which comprises the eight cells surrounding a central cell (Figure 1(a)). This type of vicinity is used on dissemination processes that depend on contiguous proximity relations, like forest fire propagation [2, 3], cell to cell infection [36] or tumor growth [28]. The second type of neighborhood relies on the concept of activity space, that is, the space in which people live from

day to day, where individuals interact on a daily basis [21]. This is particular important for simulating epidemic processes in which contact relations between individuals are a relevant issue to consider (e.g., respiratory diseases). Therefore, the activity-space neighborhood includes not only relations established in a residential environment, which can be represented by the Moore neighborhood, but also additional ones, such as the relations resulting from work, school and leisure activities. To represent the latter, we added a random component to the neighborhood, as illustrated in Figure 1(b).

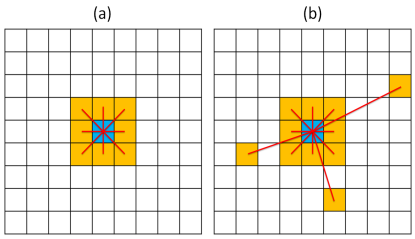


Figure 1. Types of neighborhood: (a) Moore and (b) Activity–space.

For the neighborhoods based on contiguity, we considered the null boundary condition [11], meaning that the cells located on the borders have as neighbors only those cells immediately adjacent to them into the grid. For the random component of the activity-space neighborhood, an average of 2 neighbors for each cell was considered. For that, we considered that each cell of the lattice can be neighbor of a particular cell with probability $2/N^2$, where N^2 is the number of cells in the cellular space.

Initially, the model assumes that a percentage ρ of the population is infected. To test the importance of the spatial distribution of initial disease focus, these early infected individuals were distributed in two ways: concentrated in a central cluster or randomly scattered in the lattice.

Mixing the two types of neighborhood with the two initial spatial distributions of infected people, we performed four experiments based on the following combinations:

Experiment A. Moore neighborhood with a centralized initial disease focus;
Experiment B. Moore neighborhood with initial infected people randomly scattered;
Experiment C. Activity-space neighborhood with a centralized disease focus;
Experiment D. Activity-space neighborhood with initial infected people randomly scattered.

For each time step t, the dynamics of people interactions were based on the following rule: each individual located at position (i, j) can become infected according to the probability

$$Pr(i, j, t) = \beta \cdot \frac{\text{number of infected neighbors of cell } \textit{(i,j)} \text{ at time } t}{\text{number of neighbors of cell } \textit{(i,j)}},$$

where β is the contagion probability. Implicitly, this formula assumes an idea of time-sharing: the more neighbors an individual has, the less time he spends with each neighbor. Figure 2 illustrates some situations of time-sharing. Each circle in Figure 2 represents one individual, while the edges represent the neighborhood relations. The blues circles correspond to susceptible individuals and the red ones represent the infected individuals. In

cases (a) and (b), the individuals at the top of the graph have the same amount of neighbors and, therefore, their chances to get exposed to the pathogen are the same. However, in case (b), the individual at the top is more likely to become infected than the one in case (a). Regarding the case (c), the individual's chance to be exposed to the pathogen is greater than the one in case (a), since the latter has fewer neighbors. However, the individual at the top in case (c) is less likely to become infected than the one in case (a), since he spends less time with each neighbor. The idea of time-sharing is fundamental in our approach, because it differentiates the risk of being exposed to the pathogen from the risk of contracting it.

Figure 2. Some situations of time–sharing. Each circle represents one individual, while the edges represent the neighborhood relations. The blues circles correspond to susceptible individuals and the red ones represent the infected subjects.

Each iteration corresponds to one time step and the common parameters used in all simulations are showed in Table 1.

Table 1. Parameters used in all simulations.

Fixed Parameters	
Dimension of cellular space (N)	51
Percentage of initial infected people (ρ)	0.05
Contagion probability (β)	0.3

For monitoring the simulation outputs, we compute the time needed for achieving the saturation point (all individuals becoming infected) and spatial indices of segregation. In this epidemiological context, indices of segregation are used to depict the potential contact between infected and susceptible individuals. Global and local spatial indices of segregation are calculated: while global indices summarize the segregation degree of the whole city and can be reported and plotted in graphs, local indices depict segregation as a spatially variant phenomenon. The model reports spatial exposure and isolation indices [12]. The global version of the exposure index of group m to n ($exp_{(m,n)}$) measures the average proportion of group n in the neighborhood of each member of group m. The exposure index ranges from 0 to 1 (maximum exposure) and its formula is:

$$exp_{(m,n)} = \sum_{j=1}^{J} \frac{N_{jm}}{N_m} \cdot \frac{L_{jn}}{L_j},$$

where J is the total number of areal units (cells); N_{jm} is the population of group m in areal unit j; N_m is the population of group m in the study area (lattice); L_{jn} is the population belonging to group n in the neighborhood of j; and L_j is the population in the neighborhood of j. The exposure index values depend on the overall population composition of

the study area. For example, if there is an increase in proportion of group n, the value of $exp_{(m,n)}$ tends to become higher. The spatial isolation index ($isol_m$) is a particular case of the exposure index that expresses the exposure of group m to itself. Both indices also present local versions that can be displayed as maps.

3. Results and Discussions

The behavior of the number of infectious individuals (epidemiologically, the prevalence) over the time steps is shown on Figure 3. Each time series evolves under a sigmoidal-like shape (a logistic function): one first region with a fast growth followed by an stationary trend. On the cases with the randomly scattered initial condition, for both neighborhood types (MS and AS curves), the rapid initial growth is explained by the formation of several distinct and simultaneous foci of disease in the first time steps. On the other hand, for both initial condition configurations, the activity-space neighborhood (corresponding to AC and AS curves) is the fastest way of epidemic spreading.

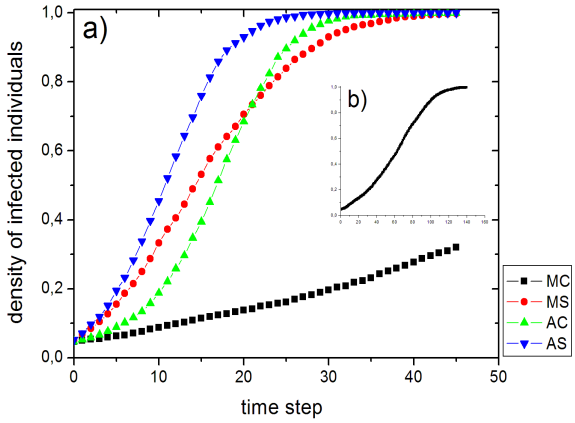


Figure 3. Time series of infected people density. (a) Four time series according with the experiments: MC means Moore neighborhood with concentrated initial infected people; MS means Moore neighborhood with randomly scattered initial infected people; AC means activity-space neighborhood with concentrated initial infected people and AS means activity-space neighborhood with randomly scattered initial infected people. (b) The behavior of MC curve during 139 time steps.

For the four different combinations of neighborhood types and initial spatial distributions, we also compared the dynamics of the density of infected individuals ($densInfec$) with global and local indices of segregation (Figures 4, 5 and 6). The segregation indices considered in the analysis were: isolation of infected individuals ($isolInfec$) and exposure of susceptible to infected individuals ($expSuscInfec$).

In Experiment A, which combines the Moore neighborhood with a centralized initial disease focus, the initial 5% of infected individuals are highly isolated, revealing a global index of isolation of infected individuals of 0.79 (Figure 4(a)). This means that, on average, 79% of the individuals in the surroundings of an infected entity are also infected.

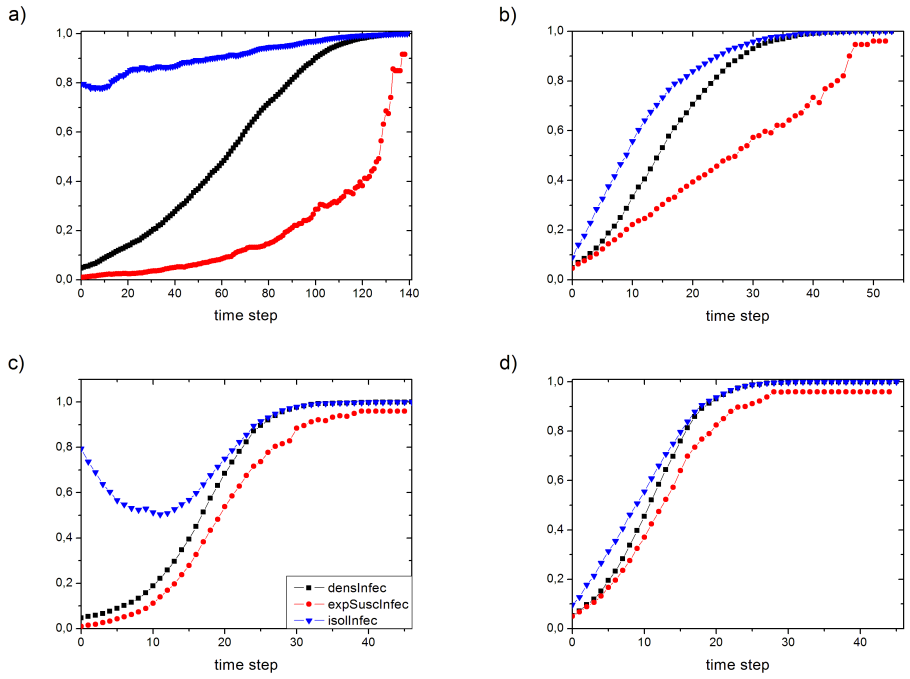


Figure 4. Density of infected individuals ($densInfec$), isolation of infected individuals ($isolInfec$) and exposure of susceptible to infected individuals ($expSuscInfec$) computed for the four experiments based on different combinations of neighborhood types and initial spatial distributions: (a) Moore neighborhood and centralized initial distribution, (b) Moore neighborhood and scattered initial distribution, (c) activity-space neighborhood and centralized initial distribution, and (d) activity-space neighborhood and scattered initial distribution.

On the other hand, the exposure index of susceptible individuals to the infected ones, equal to 0.01, reveals that, on average, only 1% of the individuals in the surroundings of a healthy entity are infected.

The local indices of segregation displayed in Figures 5(a) and 6(a) ($t = 0$) complement this information by showing, respectively, where the infected individuals are initially isolated and the location of a few susceptible individuals with a certain degree of exposure to the infected ones.

As time progresses, the isolation of infected individuals increases slowly, in a linear fashion, and only achieves its maximum value ($isolInfec = 1$) when $t = 139$. It is important to remind that the isolation index necessarily achieves its maximum value at the saturation point, when 100% of the individuals in the surroundings of each infected individual are also infected.

The curve depicting the exposure index of susceptible to infected individuals ($expSuscInfec$) has an exponential-like shape, which increases slowly during the first simulation iterations but accelerates its rhythm as the proportion of infected individuals increases. Figures 5(a) and 6(a) illustrate the spatial pattern of these processes of isolation/exposure, showing how the epidemics dynamics resemble a cluster expansion that

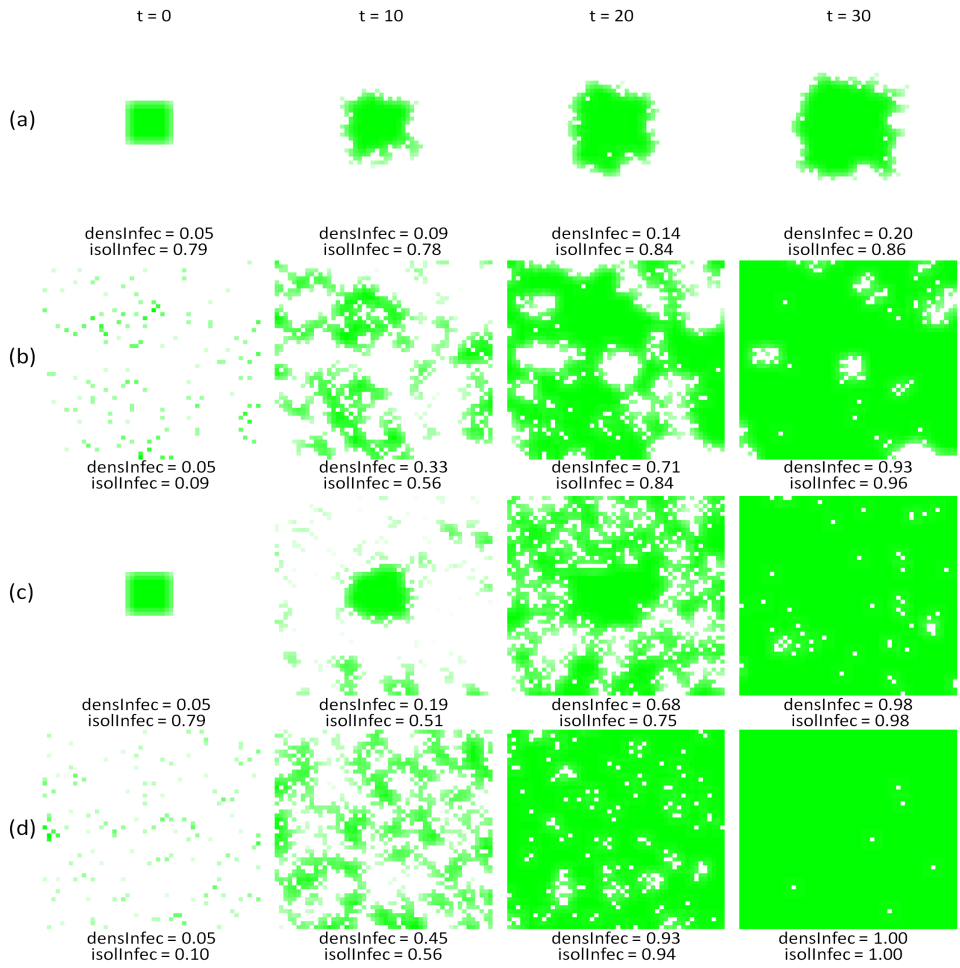


Figure 5. Local indices of isolation of infected individuals computed for four experiments based on different combinations of neighborhood types and initial spatial distributions: (a) Moore neighborhood and centralized initial distribution, (b) Moore neighborhood and scattered initial distribution, (c) activity-space neighborhood and centralized initial distribution, and (d) activity-space neighborhood and scattered initial distribution.

propagates through the lattice. This spreading pattern is consistent for describing processes dictated by contiguous proximity relations.

The Experiment B is similar to the above and also more appropriate to describing dissemination processes ruled by contiguous proximity relations. Nevertheless, in this case, we test the impact of having several initial disease foci randomly scattered in the lattice. Comparing situations A and B in Figure 4, its possible to observe that the scattered initial configuration promoted a strong decrease in the initial isolation of infected individuals ($isolInfec$ decays from 0.79 to 0.09). The exposure of susceptible to infected individuals, on the other hand, increases and reaches the same value as the proportion of infected people ($expSuscInfec = 0.05$), meaning that, on average, a susceptible entity has 5% of infected individuals in its surroundings.

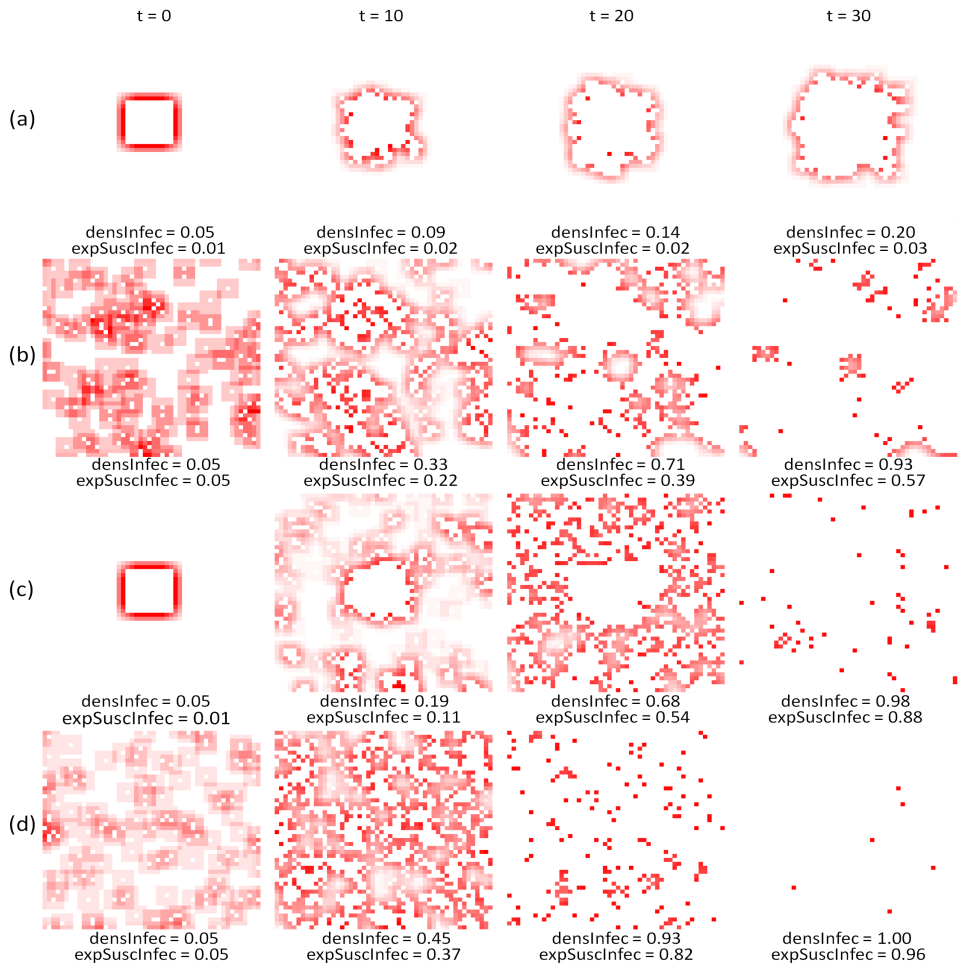


Figure 6. Local indices of exposure of susceptible to infected individuals computed for four experiments based on different combinations of neighborhood types and initial spatial distributions: (a) Moore neighborhood and centralized initial distribution, (b) Moore neighborhood and scattered initial distribution, (c) activity-space neighborhood and centralized initial distribution, and (d) activity-space neighborhood and scattered initial distribution.

Figure 4(b) also shows that the curve describing the isolation index of infected individuals followed a sigmoid shape similar to the one obtained for the density of infected entities. The exposure index of susceptible to infected individuals increased in a linear manner and presents much higher values along the time axis when compared with situation A. In this less-segregated pattern, where the contact between a susceptible and an infected entity becomes more common, it is possible to observe a much faster spread of the epidemics, which saturated at $t = 53$. Figures 5(b) and 6(b) show the local isolation and exposure patterns associated with this type of propagation dynamics.

The Experiment C starts with the same initial condition as the one presented in Experiment A, i.e., with a centralized initial disease focus and a high isolation degree of infected individuals. Nevertheless, it adopts the activity-space neighborhood to model

the interactions between entities, which is a more realistic approach to represent those epidemic processes that depend on direct-contact relations between individuals. Because the activity-space neighborhood includes a non-contiguous component that models those interactions that do not necessarily occur in the surroundings of the individual's residence, the initial isolation of infected individuals strongly decreases during the first iterations of the simulation (Figure 4(c)), when several disease outbreaks appeared (Figure 5(c), $t = 10$) in different points of the lattice. These outbreaks, which are the result of interactions that occur, for instance, during work or school activities, accelerate the dissemination of the epidemic (saturation time equal to 53).

The Experiment D combines the activity-space neighborhood with the initial pattern characterized by several disease foci randomly scattered in the lattice. Comparing with the other experiments, this situation provides the least-segregated patterns between infected and susceptible individuals and, therefore, is also where the saturation point is achieved faster (saturation time equal to 46). Figure 4(d) shows that, in this case, the curves describing the isolation and exposure indices followed the same sigmoid shape as the curve depicting the density of infected individuals. This behavior is the natural trend of both isolation and exposure indices, since an increase in the density of infected individuals also tends to increase the chance of individuals in general (infected or susceptible) to have more infected individuals in their surroundings.

4. Conclusions and Perspectives

This work investigated the effects of neighborhood structures on disease spreading through a compartmental epidemiological CA-model. The paper demonstrated how CA-based models allow the design and exploration of different scenarios of disease transmission dynamics. CA models are able to address the emergence of global spatial patterns of dissemination processes from local interactions between entities and/or individuals. In addition, it provides empirical ways for testing hypotheses and exploring the impacts of public policies.

Considering the concept of neighborhood as an essential aspect to represent interactions between individuals, this work relied on the idea of activity space to explore a neighborhood type that includes not only places surrounding the residential location of individuals, which can be represented by contiguity-based neighborhoods (e.g., Moore), but also additional areas where individuals interact on a daily basis, such as work or school. By combining two types of neighborhood (activity-space or Moore) with two different initial spatial distribution of infected people (centralized or randomly scattered), four different scenarios were simulated and analyzed.

For the analyses, local and global indices of isolation and exposure, normally applied for measuring segregation between social groups, were adopted in this epidemiological context as tools able to depict the spatial arrangement between infected and susceptible individuals as well as the potential contact between them. Using these indices to monitor the simulation scenarios, it was possible to obtain new insights on how epidemics evolve on time and space. The experiments revealed, for instance, how the activity-space neighborhood is related to the appearance of additional disease outbreaks in different areas of the lattice. These outbreaks promote a less-segregated pattern between infected and susceptible individuals and, therefore, a faster spread of the disease.

Such results reinforce the need of considering more realistic neighborhood structures, such as the activity-space neighborhood, to represent epidemic processes that depend on direct-contact relations between individuals. For empirical analysis, the implementation of the activity-space neighborhood would demand the use of additional data that are able to reveal the areas in which people live and interact day-to-day (e.g., origin-destination surveys).

References

[1] Almeida, C. M.; Batty, M.; Monteiro, A. M. V.; Câmara, G.; Soares-Filho, B. S.; Cerqueira, G. C.; Pennachin, C. L. (2003). Stochastic cellular automata modelling of urban land use dynamics: Empirical development and estimation. Computers, Environment and Urban Systems, v. 27, n. 5, p. 481–509.

[2] Almeida, R. M.; Macau, E. E. N. (2011). Stochastic cellular automata model for wildland fire spread dynamics. Journal of Physics. Conference Series (Online), v. 285, p. 012038.

[3] Almeida, R. M. ; Macau, E. E. N. ; França, H. ; Ramos, F. M. ; Carneiro, T. G. S. (2008). Simulando padroes de incendios no Parque Nacional das Emas, Estado de Goias, Brasil. In: X Simpósio Brasileiro de Geoinformática, 2008, Rio de Janeiro - RJ. Anais do X Simpósio Brasileiro de Geoinformática 2008.

[4] Andrade, R. F. S.; Rocha-Neto, I. C.; Santos, L. B. L.; Santana, C. N.; Diniz, M. V. C. et al. (2011). Detecting network communities: An application to phylogenetic analysis. PLoS Computational Biology, v. 7, n. 5. doi:10.1371/journal.pcbi.1001131.

[5] Barker D. P. and Bennett, F. J. (1976). Pratical of epidemiology, Churchill, Livingstone.

[6] Batty, M. (2000). GeoComputation using cellular automata. In: Geocomputation, S. Openshaw and R.J. Abrahart (Eds), New York: Taylor Francis, p. 95–126.

[7] Batty, M.; Xie, Y. (1994). From cells to cities. Environment and Planning B, v. 21, p. 31–48.

[8] Câmara, G. and Monteiro, A. M. V. (2001). Geocomputation techniques for spatial analysis: Are they relevant for health data? Cadernos de Saúde Pública, v. 17, n. 5, p. 1059–1081.

[9] Câmara, G.; Souza, R. C. M.; Pedorsa, B. M.; Vinhas, L.; Monteiro, A. M. V.; Paiva, J. A.; Carvalho, M. T.; Gattass, M. (2000). TerraLib: Technology in support of GIS innovation. II Brazilian Symposium on Geoinformatics, GeoInfo 2000, São Paulo.

[10] Carneiro, T. G. S. (2006). Nested-CA: a foundation for multiscale modeling of land use and land cover change. In: Computer Science Department - INPE, São José dos Campos, SP.

[11] Encinas, L. H.; White, S. H.; Martín del Rey, A.; Sánchez, G. R. (2007). Modelling forest fire spread using hexagonal cellular automata. Applied Mathematical Modelling, v. 31, p. 1213–1227.

[12] Feitosa, F. F.; Câmara, G.; Monteiro, A. M. V.; Koschitzki, T.; Silva, M. P. S. (2007). Global and local spatial indices of urban segregation. International Journal of Geographical Information Science, v. 21, n. 3, p. 299–323.

[13] Góes-Neto, A.; Diniz, M. V. C.; Santos, L. B. L.; Pinho, S. T. R.; Miranda, J. G. V.; et al. (2010). Comparative protein analysis of the chitin metabolic pathway in extant organisms: A complex network approach. BioSystems, v. 101, p. 5966.

[14] Hagoort, M.; Geertman, S; Ottens, H. (2008). Spatial externalities, neighbourhood rules and CA land-use modeling. In: The Annals of Regional Science, v. 42, n. 1.

[15] Hethcote, H. W. (2000). The mathematics of infectious diseases. SIAM Review, vol. 42, n. 4, p. 599-653.

[16] Hill, A. L; Rand, D. G.; Nowak, M. A.; Christakis, N. A. (2010). Infectious disease modeling of social contagion in networks. Plos Computational Biology, v. 6, i. 11.

[17] Ierusalimschy, R.; Figueiredo, L. H. et al. (1996). Lua - an extensible extension language. Software: Practice Experience, v. 26, p. 635–652.

[18] Lana, R.; Carneiro, T. G. S.; Honório, N. A.; Codeço, C. T. (2010). Change allocation in spatially-explicit models for *Aedes aegypti* population dynamics. In: XI Brazilian Symposium on Geoinformatics, GeoInfo 2010, Campos do Jordão.

[19] Martin dél Rey, A.; White, S. H.; Sánchez, G. R. (2006). A model based on cellular automata to simulate epidemic diseases. In: Yacoubi, S. E.; Chopard, B.; Bandini, S. (Eds.): ACRI 2006, LNCS 4173, 304–310. Springer-Verlag Berlin Heidelberg 2006.

[20] Massad E.; Menezes, R. X.; Silveira, P. S. P.; Ortega, N. R. S. (2004). Métodos Quantitativos em Medicina, São Paulo, Editora Manole.

[21] Mayhew, S. (2009). A dictionary of geography. New York: Oxford University Press, 560p.

[22] Medeiros, L. C. C.; Castilho, C. A. R.; Braga, C.; Souza, W. V.; Regis, L.; Monteiro, A. M. V. (2011). Modeling the Dynamic Transmission of Dengue Fever: Investigating Disease Persistence. PLOS neglected tropical diseases. v. 5, n. 1, p. e942. doi:10.1371/journal.pntd.0000942.

[23] Monteiro, A. M. V.; Carvalho, M. S.; Assunção, R.; Vieira, W.; Ribeiro, P. J.; Davis Jr, C.; Regis, L. et al. (2009). SAUDAVEL: Bridging the gap between research and service in public health operational programs by multi-institutional networking development and use of spatial information technology innovative tools. Rev. Bras. Biom., v. 27, n. 4, p. 519–537.

[24] Mikler, A. R.; Venkatachalam, S.; Abbas, K. (2005). Modeling infectious diseases using global stochastic cellular automata. In: Journal of Biological Systems, v. 13, n. 4, p. 421–439.

[25] Mohtashemi, M.; Szolovits, P.; Dunyak, J.; Mandl, K. D. (2006). A susceptible-infected model of early detection of respiratory infection outbreaks on a background of influenza. Journal of Theoretical Biology, v. 241, n. 4, p. 954–963.

[26] Nishiura, H. (2006). Mathematical and statistical analyses of the spread of dengue. Dengue Bull, v. 30, p. 51-67.

[27] Oliveira, G. M. B.; Siqueira, S. R. C. (2006). Parameter characterization of two-dimensional cellular automata rule space. Physica D, v. 217, p. 1-6.

[28] Reis, E. A.; Santos, L. B. L.; Pinho, S. T. R. (2009). A cellular automata model for avascular solid tumor growth under the effect of therapy. Physica A: Statistical Mechanics and its Applications, v. 388, n. 7, p. 1303–1314.

[29] Sarkar, P. (2000). A brief history of cellular automata. ACM Computing Surveys, v. 32, n. 1, p. 80-107.

[30] Santos, L. B. L.; Costa, M. C.; Pinho, S. T. R.; Andrade, R. F. S.; Barreto, F. R.; Teixeira, M. G.; Barreto, M. L. (2009). Periodic forcing in a three-level cellular automata model for a vector-transmitted disease. Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics (Print), v. 80, p. 016102.

[31] Stauffer D.; Sahimi, M. (2007). Can a few fanatics influence the opinion of a large segment of a society? European Physical Journal B, 57: 147-152. DOI: 10.1140/epjb/e2007-00106-7.

[32] Von Neumann, J. (1966). Theory of self-reproducing automata. Illinois: A.W. Burks.

[33] White, S. H.; Rey, A. M.; Sánches, G. R. (2007). Modeling epidemics using cellular automata. Applied Mathematics and Computation, v. 186, p. 193–202.

[34] Wimpenny, J. W.; Colasanti, R. (1997). A unifying hypothesis for the structure of microbial biofilms based on cellular automaton models. FEMS Microbiology Ecology, v. 22, n. 1, p. 1-16.

[35] Wolfram, S. (1994). Cellular Automata and Complexity. New York, Addison-Wesley Publishing Company, p. 316.

[36] Santos, R. M. Z. and Coutinho, S. (2001). Dynamics of HIV infection: A cellular automata approach. Physical Review Letters, v. 87, n. 16. doi:10.1103/PhysRevLett.87.168102.

Divide and Segment – An alternative for parallel segmentation

Thales Sehn Korting[1], **Emiliano Ferreira Castejon**[1], **Leila M. Garcia Fonseca**[1]

[1]National Institute for Space Research (INPE) – Image Processing Division (DPI)
12201-010 São José dos Campos – SP – Brazil

{tkorting,castejon,leila}@dpi.inpe.br

Abstract. *Remote sensing images with large sizes are usual. They also include several spectral channels, increasing the volume of information. To get valuable information from data automatically, computers need higher amounts of memory and efficient processing techniques. Segmentation is a key technique to deal with remote sensing. It identifies regions in images. Therefore, it deals with large amounts of information. Even with current computational power, some image sizes exceed the memory limits, which need different solutions. An alternative to overcome such limits is to employ divide and conquer strategy, splitting the image into tiles, and segmenting each one individually. However, arises the problem of merging neighboring tiles and keeping the homogeneity in such regions. In this work, we propose an alternative to create the tiles, by defining noncrisp borders between tiles, but adaptive borders for the tiles. By applying our method, we avoid the postprocessing of neighboring regions, and therefore speed up the final segmentation.*

1. Introduction

Segmentation in remote sensing is a challenging field. Techniques for segmentation are the first step in all analysis tasks. Their results are expected to describe the objects, allowing a deeper interpretation by experts or classification algorithms. The work of [Haralick and Shapiro 1985] defined segmentation as a way to separate the image into simple regions with homogeneous behavior.

In remote sensing, segmentation techniques are not new – see [Bins et al. 1996], [Câmara et al. 1996]. However, the field of GEOBIA (GEographic Object-Based Image Analysis) emerged recently. It makes a link between objects and radiometric properties [Blaschke 2010].

To partition automatically an image into regions, algorithms must consider the context, scale, neighborhood, meaning, and computational resources. However, according to [Wassenberg et al. 2009], good quality results often come at the price of high computational cost.

For example, the collection rate for IKONOS satellite is about 890 megapixels each minute [Dial et al. 2003]; for CBERS-2B is about 120 megapixels each minute. According to [Wassenberg et al. 2009], even a tuned sequential segmentation algorithm is far slower than these rates.

Remote sensing images often present large sizes. The variety of spectral channels, that in one side contain rich information about the land targets, in other side increases the volume of information. Even with current computational power, certain sizes exceed the

memory limits, claiming new solutions. Methods based on divide and conquer strategy arise as an alternative for these limits. Such methods split the image into tiles, and segment each one individually. The problem of this approach is to merge neighboring tiles, keeping the homogeneity in these regions.

In this article we tackle the problem of creating tiles for parallel segmentation. After defining tiles, any algorithm can run on them in an independent way, since the implementation is based on multiprogrammed techniques. We argue that by defining noncrisp borders between tiles, we avoid the postprocessing of neighboring regions, and therefore speed up the final segmentation.

2. Related Work

Several techniques arise from "region growing" strategy, relying on the similarity of near pixels. [Bins et al. 1996] applied this approach in remote sensing images. Their method is based on the likeness between neighboring pixels and the smallest area allowed for a region.

[Baatz and Schape 2000] is another example of region growing technique. In this approach, the algorithm minimizes the average heterogeneity of the regions. The heterogeneity balances the object's smoothness and compactness, resulting in more regular objects. It deals with the standard deviation of pixels for each band as well. We suggest the reading of [Meinel and Neubert 2004] for a comparison of these two algorithms and alternatives for remote sensing segmentation.

According to [Lenkiewicz et al. 2009], parallel architectures are becoming a standard for handling complex operations that need significant computational power. Large size images include medical data sets of magnetic resonance imaging (MRI) [Prassni et al. 2010], or remote sensing hyperspectral and multitemporal images [Valencia et al. 2008, Plaza et al. 2011]. Therefore parallel algorithms arise as an alternative to overcome these limits. Such methods usually split the image into tiles, and segment each one individually. A postprocessing step is necessary to couple bordering regions.

The tiles often have regular sizes to be assigned equally among the processors [Bader et al. 1996]. However, according to [Wassenberg et al. 2009] this is not acceptable because border objects are not correctly handled. A common solution is to adopt overlapping tiles, which is also inadequate because there is no upper bound on the size of objects of interest (e.g. rivers or roads).

The work of [Singh et al. 1999] proposed a parallel method for the seeded region growing algorithm ([Adams and Bischof 1994]), based on spreading seeds in different processes, each one growing in parallel. The authors needed to deal with simultaneous access for the same pixels. To avoid this problem, images were divided in square windows, employing a postprocessing step to join regions.

[Happ et al. 2010] employed the traditional parallel segmentation. Each tile was processed by a different thread, through a sequential algorithm [Baatz and Schape 2000]. This method falls on the same problem of treating boundary segments. The number of boundary objects, which depends on the image, can be prohibitive in certain cases. Therefore this paper deals with adaptive tiles, minimizing the problem of boundary regions.

In the area of merging and mosaicking, [Bagli and Fonseca 2006] presented a

technique for image blending, based on multi-resolution decomposition. The authors defined a cut line, considering texture information from overlapping regions of mosaicking images. The method found automatically the transition zone size and the cut line on satellite and aerial images.

3. Method

Traditional parallel schemes first divide the image into crisp tiles, and then treat bordering regions. Some creates tiles with overlapping regions, but fall into the same problem of postprocessing. We propose to create adaptive tiles using non crisp borders. Figure 1 shows our approach.

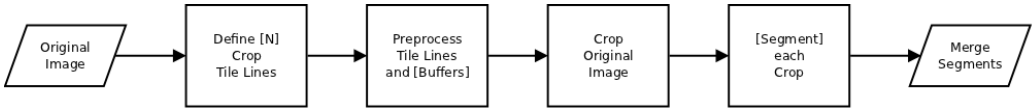


Figure 1. Main diagram from our technique. User defined parameters are in [braces].

We suggest to first define the crisp tiles, and analyze them to adapt the local features, in a strategy of presegmentation. We use two basic parameters , a maximum neighborhood for each pixel in the line ($t1$), transverse to the tile line, and a buffer around the original tile line ($t2$). Figure 2 depicts the approach, where the original tile line is defined as the middle line between the buffer lines. Let the neighborhood of each pixel be called profile. The algorithm is described as follows:

```
1. get pixel in tile line and its profile
2. find border
3. change the tile position to this border
4. assign next pixel to the same border position
5. if there are more pixels in the tile line, back to 1
6. crop image using new tile line
```

To detect the edges, our strategy calculates the first slope of each profile. The maximum absolute value of the slope for each profile points a border. However, alternative methods to detect borders in the profiles can be employed. In this paper, we describe and test only horizontal tile lines. Although the same scheme can be applied if tiles contain vertical lines as well. After adapting the tiles, individual and parallel methods segment each tile. The result is the merging from all tiles.

Noisy tile lines can be created, depending on the right choice of $t1$ and $t2$. The buffer size $t2$ can be defined based on the parameters of average region size from the segmentation algorithms. The choice of proper values depends on the image scale and the dimension of the objects of interest.

4. Results and Discussion

To apply our method, we used the algorithm of region growing [Bins et al. 1996] available at TerraLib C++ library [Câmara et al. 2008]. This library presents parallel methods for segmentation, dividing the image into crisp tiles, and merging neighbors afterwards.

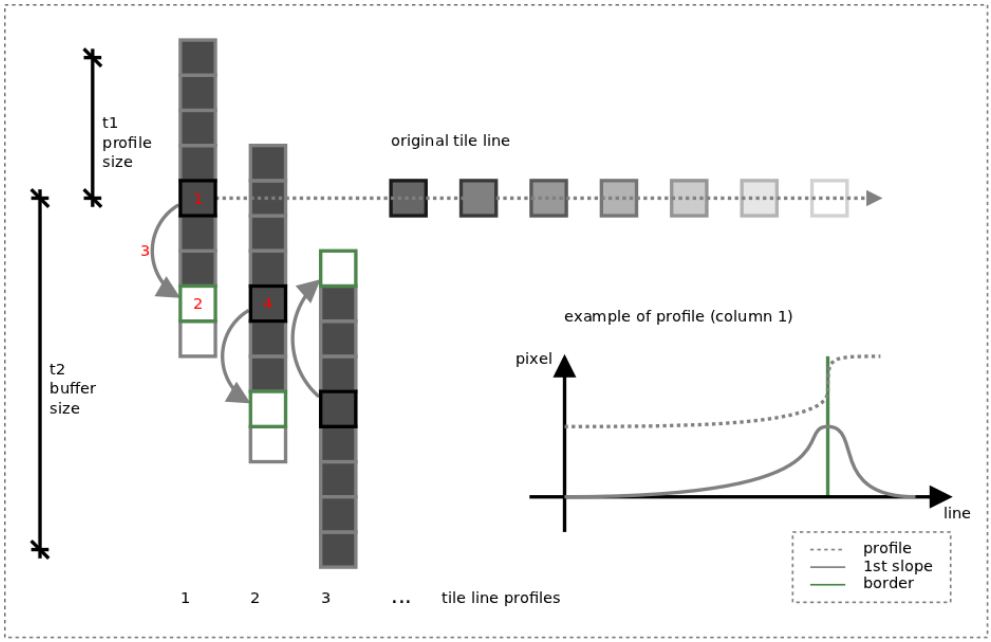


Figure 2. Scheme for adapting the tile lines. Steps 1 to 4 are highlighted (red).

The sequential algorithm (without dividing the image into tiles) is available as well. We compared our approach to the parallel algorithm available in TerraLib, using the same parameters. We used 3 images with different contexts to evaluate the method.

The first example shows a Quickbird image from São José dos Campos, Brazil, obtained in 2005. The size of the image is 1024×1024 pixels. The parameters used to adapt the tile line were $t1 = 8$ and $t2 = 20$. The adapted tile line is shown in Figure 3. The segmentation parameters (see [Bins et al. 1996]) were minimum area of 200 pixels, and Euclidean distance of 80 pixels. To compare the results of the segmentation, Figure 4 shows both approaches (adapted tile line, and crisp tile line with postprocessing). It is possible to see that our approach reduced crispy resultant objects.

The second image is a crop (1000×1000 pixels) of a region in the state of Bahia, Brazil, using satellite CBERS-2B, instrument HRC[1]. The pixels of this image have $2.7m^2$. The choice of parameters was $t1 = 5$ pixels, and a buffer of 100 pixels ($t2$). The adapted tile line is shown in Figure 5. The parameters were minimum area of 50 pixels, and Euclidean distance of 14 pixels. To compare the results of the segmentation, Figure 6 shows both approaches (adapted tile line, and crisp tile line with postprocessing). In this case the tile line has adapted to the river, allowing a better result of segmentation than to use crisp tiles.

The third image is a crop of a Quickbird scene from São Paulo, Brazil, with 1000×1175 pixels. The pixels of this image have $1m^2$. The choice of parameters was $t1 = 6$ pixels, and a buffer of 70 pixels ($t2$). The adapted tile line is shown in Figure 7. The parameters were minimum area of 300 pixels, and Euclidean distance of 50 pixels. To compare the results of the segmentation, Figure 8 shows both approaches (adapted tile

[1]Free remote sensing imagery at `http://www.dgi.inpe.br/CDSR/`.

Figure 3. The adapted tile line (yellow) for the first example. Red and Green lines shows the buffer size (parameter t2), and the Blue line shows the original tile line.

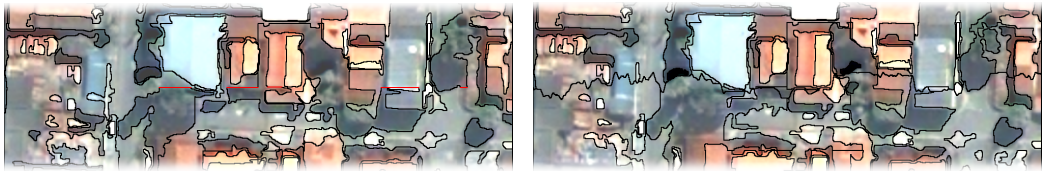


Figure 4. Comparison of approaches for Quickbird scene. Left is the segmentation using crisp tiles. Right is our approach using adapted tile lines.

line, and crisp tile line with postprocessing). One can note the presence of certain crisp polygons, which couldn't be merged due to their spectral differences. However using our alternative, the segmentation achieved a smoothest result between the tiles.

5. Conclusion

This article tackled the problem of defining tiles for parallel segmentation. Current methods create crisp tiles, needing postprocessing steps to get final regions. In certain cases, such methods create inconsistent objects, demanding computational power to deal with bordering regions. Postprocessing detects the bordering regions, and test the best combination of regions to merge. This step aims to keep the consistence of the new regions to specific segmentation parameters, as spectral homogeneity and size.

From the results it is clear to see that new tile borders remain at the end of segmentation. For most of the cases this is the expected result, and should not influence the overall segmentation. However, the shape of certain regions near the adaptive tile lines will not split image targets properly. Therefore dealing with such problem still remain as an open problem, currently unsolved by this method. Future works also include initialization issues, since our method begins in the leftmost pixel. Since this method is extendable to vertical tile lines, next steps also include testing images with horizontal and vertical tile lines. Besides, we aim at defining automatically the parameters ($t1$ and $t2$) based on the specific segmentation parameters.

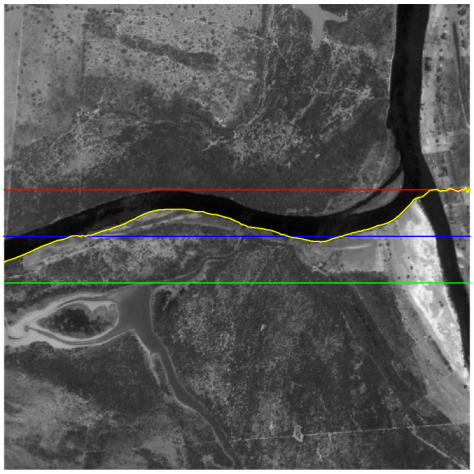


Figure 5. The adapted tile line (yellow) for the second example. Red and Green lines shows the buffer size (parameter t2), and the Blue line shows the original tile line.

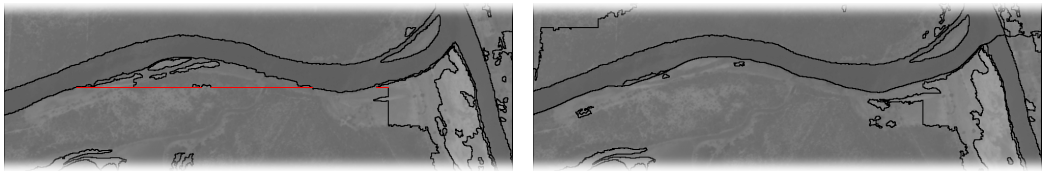


Figure 6. Comparison of approaches for CBERS-2B HRC. Left is the segmentation using crisp tiles. Right is our approach using adapted tile lines.

This work is a first step towards the definition of a method for creating adaptive tiles, developed for the main purpose of dealing with remote sensing images with large sizes. By defining noncrisp borders between tiles, our method avoided the processing of neighboring regions, creating adaptive tiles. The algorithm runs with a complexity of $O(n)$, and was developed using the TerraLib library, in C++.

References

Adams, R. and Bischof, L. (1994). Seeded region growing. *Pattern Analysis and Machine*, 16.

Baatz, M. and Schape, A. (2000). Multiresolution Segmentation: an optimization approach for high quality multi-scale image segmentation. In Wichmann-Verlag, editor, *XII Angewandte Geographische Informationsverarbeitung*, Heidelberg. Herbert Wichmann Verlag.

Bader, D., Jaja, J., Harwood, D., and Davis, L. (1996). Parallel algorithms for image enhancement and segmentation by region growing with an experimental study. *Proceedings of International Conference on Parallel Processing*, pages 414–423.

Bagli, V. and Fonseca, L. (2006). Seamless mosaicking via multiresolution analysis and cut line definition. In *Signal and Image Processing*. ACTA Press.

Figure 7. The adapted tile line (yellow) for the third example. Red and Green lines shows the buffer size (parameter t2), and the Blue line shows the original tile line.

Figure 8. Comparing approaches for the third image. Left is the segmentation using crisp tiles. Right is our approach using adapted tile lines.

Bins, L., Fonseca, L., Erthal, G., and Li, F. (1996). Satellite imagery segmentation: a region growing approach. *Brazilian Remote Sensing Symposium*, 8.

Blaschke, T. (2010). Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(1):2–16.

Câmara, G., Souza, R., Freitas, U., Garrido, J., and Li, F. (1996). Spring: Integrating remote sensing and gis by object-oriented data modelling. *Computers and Graphics*, 20(3):395–403.

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 large-scale environmental and socio-economic applications. *Open Source*, pages 247–270.

Dial, G., Bowen, H., Gerlach, F., Grodecki, J., and Oleszczuk, R. (2003). IKONOS satellite, imagery, and products. *Remote Sensing of Environment*, 88(1-2):23–36.

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*.

Haralick, R. and Shapiro, L. (1985). Image segmentation techniques. *Applications of Artificial Intelligence II., 1985,*, 548:2–9.

Lenkiewicz, P., Pereira, M., Freire, M., and Fernandes, J. (2009). A new 3D image segmentation method for parallel architectures. *2009 IEEE International Conference on Multimedia and Expo*, pages 1813–1816.

Meinel, G. and Neubert, M. (2004). A comparison of segmentation programs for high resolution remote sensing data. *International Archives of Photogrammetry and Remote Sensing*, 35(Part B):1097–1105.

Plaza, A., Plaza, J., Paz, A., and Sanchez, S. (2011). Parallel Hyperspectral Image and Signal Processing. *IEEE Signal Processing Magazine*, 28(May):119–126.

Prassni, J., Ropinski, T., and Hinrichs, K. (2010). Uncertainty-aware guided volume segmentation. *IEEE transactions on visualization and computer graphics*, 16(6):1358–65.

Singh, D., Heras, D., and Rivera, F. (1999). Parallel Seeded Region Growing Algorithm. In *VIII Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes*, Bilbao, Spain.

Valencia, D., Lastovetsky, A., O'Flynn, M., Plaza, A., and Plaza, J. (2008). Parallel Processing of Remotely Sensed Hyperspectral Images On Heterogeneous Networks of Workstations Using HeteroMPI. *International Journal of High Performance Computing Applications*, 22(4):386–407.

Wassenberg, J., Middelmann, W., and Sanders, P. (2009). An Efficient Parallel Algorithm for Graph-Based Image Segmentation. In *Computer Analysis of Images and Patterns*, pages 1003–1010. Springer.

Uma infraestrutura de dados espaciais para o Projeto GeoMINAS

Lucas F. M. Vegi, Jugurta Lisboa F., Wagner D. Souza, João P.C. Lamas, Glauber L. S. Costa, Wellington M. Oliveira, Rafael S. Carrasco, Tiago G. Ferreira, Joás W. Baia

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

lucasvegi@gmail.com, idegeominas@ufv.br

Abstract. *This article describes the project which aimed to create a new Spatial Data Infrastructure for the GeoMINAS Project, restoring their original data and documenting them by means of metadata, described based on the national metadata standard (MGB Profile). The steps for the implementation of the SDI GeoMINAS are described in detail.*

Resumo. *Este artigo descreve o projeto que teve como objetivo criar uma nova Infraestrutura de Dados Espaciais para o Projeto GeoMINAS, resgatando os seus dados originais e documentando-os por meio de metadados, descritos com base no padrão nacional de metadados (Perfil MGB). Os passos para a implantação da IDE GeoMINAS são descritos detalhadamente.*

1. Introdução

O termo Sistema de Informação Geográfica (SIG) é usado para denotar sistemas capazes de relacionar dados cadastrais/atributos e dados geográficos [NOGUERAS-ISO et al., 2005a]. Este tipo de sistema possibilita o acesso, através de interfaces amigáveis, a visualização de variáveis sobre feições e fenômenos que ocorrem na superfície terrestre. Os SIG vêm tornando-se cada vez mais sofisticados e o avanço nos dispositivos de captura de dados geoespaciais tornou o processo cada vez mais rápido [RAJABIFARD, 2001]. Entretanto, quase todo novo projeto de SIG pode implicar em desenvolvimento a partir do zero, caso não haja suporte a reutilização de base de dados já existentes.

A resposta ao suporte para reutilização de dados espaciais gerados por diversos agentes foi a criação de uma infraestrutura que permitisse seu compartilhamento, ou seja, uma Infraestruturas de Dados Espaciais (IDE). Este termo surgiu em 1993 quando o Conselho de Pesquisa Norte-Americano (*US National Research Council*) estabeleceu a necessidade de acesso padronizado à informação geográfica [MAGUIRE e LONGLEY, 2005]. Para isto é fundamental o conhecimento e existência dos dados sobre os dados: os metadados. Os metadados desempenham uma função primordial em uma IDE, pois eles tornam os dados e serviços geoespaciais acessíveis a usuários e softwares-cliente. Estes são agrupados em catálogos com a finalidade de facilitar o conhecimento e recuperação dos dados, bem como a sua utilização, a partir da obtenção (*download*) do dado ou ainda diretamente da IDE, por meio de serviços Web [NOGUERAS-ISO et al., 2005b].

No Brasil, grande parte dos dados geoespaciais ainda encontra-se dispersa em diversas instituições públicas e privadas gerando assim um conjunto de problemas que poderiam ser, em grande parte, resolvidos, se as instituições que possuem estes dados utilizassem um sistema integrado de catalogação de metadados.

Com a recente iniciativa da Comissão Nacional de Cartografia (CONCAR), de lançar a INDE – Infraestrutura Nacional de Dados Geográficos [CONCAR, 2010], juntamente com a definição do Perfil MGB – Perfil Brasileiro para Metadados Geográficos [CONCAR, 2009], em consonância com o padrão internacional demetadados (Série ISO TC211), iniciativas de IDE nos diversos níveis de abrangência começam a ser reestruturadas.

Este artigo descreve a iniciativa de reestruturação do portal de dados do Projeto GeoMINAS, ou seja, a criação da IDE estadual GeoMINAS[1], para disponibilizar as coleções de dados espaciais disponíveis no antigo site do GeoMINAS, o qual não adotava nenhum padrão para descrição dos metadados, já que na época de seu lançamento ainda não estavam estabelecidos padrões como o CSDGM [FGDC, 1998] e o ISO19115 [ISO, 2003]. Com esta reestruturação os dados passam a ser documentados por metadados elaborados de acordo com o Perfil MGB, da INDE. Esta nova IDE, além de manter os dados do antigo site, visando a possibilidade de execução de operações de análises temporais, está apta a receber novos e atualizados dados de abrangência estadual.

2. Uma IDE para o Projeto GeoMINAS

O site GeoMINAS[2] foi uma iniciativa publicada em 1995, a partir da articulação de um grupo de instituições sediadas no estado de Minas Gerais sendo pioneiro na disponibilização de dados geoespaciais em âmbito estadual. Seus dados foram amplamente utilizados por usuários de todo o Brasil durante seus 15 anos de existência, entretanto devido a falta de apoio político, este site não recebeu atualizações e em abril de 2011 foi retirado do ar.

No primeiro semestre de 2011, como projeto da linha de pesquisa em Bancos de Dados Espaciais, do Programa de Pós-Graduação em Computação da Universidade Federal de Viçosa, foi realizado um trabalho colaborativo que teve como objetivo criar uma nova IDE para o GeoMINAS, resgatando os seus dados originais e documentando-os com base no Perfil MGB. Para tal foi necessário utilizar uma ferramenta capaz de catalogar, recuperar e realizar buscas espaciais nesses metadados.

Para atender essas necessidades foi escolhido o catálogo de metadados GeoNetwork[3]. Nas seções a seguir são apresentadas as etapas da construção da IDE GeoMINAS.

2.1. Configuração do GeoNetwork

O GeoNetwork proporciona a catalogação e o acesso ao conteúdo de diversos tipos de dados, inclusive espaciais, através de metadados, tendo como base os padrões de metadados ISO19115, ISO19139, FGDC e Dublin Core. Este sistema é amplamente utilizado em iniciativas de IDE pelo o mundo, por exemplo, a INDE [CONCAR, 2010] e a *Scottish Spatial Data Infrastructure*[4].

Optou-se por utilizar a versão 2.2 do GeoNetwork, disponibilizada no site do IBGE por esta possuir suporte ao Perfil MGB. Devido a necessidade do GeoNetwork ser

[1]http://www.ide.ufv.br/geominas
[2]http://www.geominas.mg.gov.br
[3]http://geonetwork-opensource.org
[4]http://scotgovsdi.edina.ac.uk/srv/en/main.home

executado em um servidor de aplicação com suporte a Java Web e Servlets, foi utilizado neste projeto o software livre Apache Tomcat.

Foi preciso aumentar a memória heap e a "PermSize" do Tomcat durante a implantação para não haver sobrecargas e consequentemente a paralisação do servidor hospedeiro. Para isso foi executado o software apache6w.exe e na aba "Java" da interface foi necessário inserir um comando após a última linha do "Java Options" e alterar o valor do campo "Maximum memory pool", como mostrado na Figura 1.

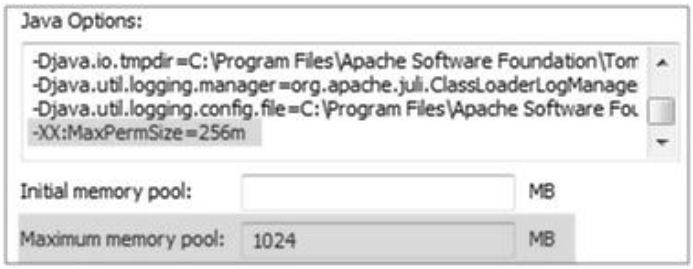


Figura1. Configuração de Memória Heap e PermSize no Apache Tomcat

A Figura 2 mostra o local onde o Tomcat localiza os arquivos do GeoNetwork e como fica o endereço dele para acessá-lo via *browser*. O campo "path" da tag "Context" é o endereço Web para chegar ao GeoNetwork. Considerando que o domínio das IDEs hospedadas no servidor utilizado é "www.ide.ufv.br", quando o usuário acessa "www.ide.ufv.br/geominas" ele entra no ambiente do GeoNetwork. O campo "docBase" é onde se localizam os arquivos do catálogo de metadados dentro do servidor.

```
<Context path="/geominas"
       docBase="D:\Geonetwork\geominas\web\geonetwork"
       crossContext="false"
       debug="0"
       reloadable="false" />
```

Figura 2. Configuração do GeoMINAS no Servidor de aplicação Apache Tomcat

Para adequar a interface da IDE GeoMINAS, foram realizadas modificações nas folhas de estilo "geonetwork.css" e "ext-all.css", presentes respectivamente nos diretórios "geonetwork" e "css". Esses arquivos são responsáveis por configurar as cores da interface do GeoNetwork. Além dos parâmetros das folhas de estilo, algumas imagens presentes no diretório "images" foram substituídas para compor a interface.

O GeoNetwork é integrado a outros dois softwares, o Geoserver e o Intermap. Eles auxiliam na realização de buscas espaciais, exibindo um mapa interativo onde o usuário pode selecionar e ampliar uma região de interesse para a busca de dados e um serviço de mapa para Web conhecido por WMS, onde se pode adquirir várias camadas de mapas de outros ambientes na Internet e ter uma melhor noção de localização.

Durante a implantação do GeoNetwork ocorreram problemas com a exibição do mapa interativo, solucionados após a realização de configurações no arquivo "mapServers.xml", como mostra a Figura 3. Para tal, foi necessário alterar os parâmetros "url" explicitando qual o domínio ou IP do serviço WMS do Geoserver.

A Figura 4 mostra um exemplo de interface da IDE GeoMINAS, após as customizações descritas nesta seção.

```
<mapContexts>
<default name="Layers for default map - DUMMY NAME : note used">
      <server url="http://www.ide.ufv.br/geoserver/wms" type="2">
            <layer name="gn:world"></layer>
      </server>
      <server url="http://www.ide.ufv.br/geoserver/wms" type="2">
            <layer name="gn:gboundaries"></layer>
      </server>
```

Figura 3. Configuração do Intermap e Geoserver

Figura 4. Página dos metadados da IDE GeoMINAS

2.2. Obtenção dos Dados Geoespaciais do GeoMINAS

Como o antigo site GeoMINAS foi retirado do ar, buscou-se encontrar cópias dos dados originais em fontes alternativas. Considerando que os dados foram publicados e disponibilizados livremente para *download* e uso, e ainda que a Universidade Federal de Viçosa foi uma das parceiras no projeto original, entendeu-se que não haveria problemas legais em disponibilizar os dados do GeoMINAS por meio de uma IDE.

Para a obtenção dos dados foram analisadas e utilizadas quatro fontes distintas, as quais possuíam cópias dos dados originais. O primeiro sítio foi um servidor do INPE - Instituto Nacional de Pesquisas Espaciais, onde se obteve um volume de 13Mb de dados. Estes dados estavam disponíveis para serem utilizados como exemplos no software Spring. A segunda fonte de dados foi um site de uma disciplina de banco de dados espaciais, do Departamento de Computação da Universidade Federal de Minas Gerais, onde os dados estavam armazenados em um SGBD PostGIS. A terceira fonte foi uma empresa de Viçosa-MG, chamada iPlanus, onde obteve-se uma cópia parcial dos dados, com volume na ordem de 70Mb. Um dos proprietários desta empresa havia participado do projeto original do GeoMINAS. Por fim, uma cópia completa dos dados foi obtida junto ao Departamento de Solos da UFV, cujo contato também havia participado do projeto original.

2.3. Elaboração dos Metadados no Perfil MGB

Logo após a execução do processo de recuperação dos dados do GeoMINAS, iniciou-se o trabalho de análise e elaboração manual dos metadados. Para tal, foi utilizado o editor de metadados do GeoNetwork.

As informações sobre os metadados originais foram primeiramente coletadas do arquivo de descrição dos dados do site do GeoMINAS, cuja cópia foi obtida juntamente com os dados. Este arquivo de descrição segue uma padronização denominada Kit Desktop Mapping 2.0. O trabalho de edição dos metadados iniciou-se na adaptação destas descrições para o Perfil MGB. O documento descritivo obtido não continha as coordenadas limites e nem o sistema de referência. Assim utilizou-se o sistema *freeware* Quantum GIS para a coleta das coordenadas e o sistema de referência para cada arquivo *shape*. Os títulos dos metadados contidos na descrição do Kit Desktop Mapping 2.0 estavam rotulados com uma sigla que não identificava com clareza o dado com o qual estava relacionado. Assim, a descrição do metadado foi alterada para um rótulo mais significativo que expressa melhor a identificação do dado bem como auxilia na recuperação do mesmo. O nome relativo a um dado espacial "MG.TAB", por exemplo, foi alterado para "Contorno do Estado de Minas Gerais (MG.TAB)". Para não perder a primeira descrição do metadado, manteve-se o seu antigo nome no final da descrição, entre parênteses.

3. Considerações Finais

O acesso a dados geoespaciais é um recurso estratégico para as instituições, tanto públicas como privadas. No entanto, o custo para obtenção e produção de dados geoespaciais é muito alto, quando comparado à aquisição de dados em aplicações não espaciais. Desta forma, o compartilhamento de dados geoespaciais é fundamental para que esses também possam ser utilizados por instituições que não possuem recursos para produzi-los.

Uma Infraestrutura de Dados Espaciais é a resposta tecnológica para minimizar esses problemas. O uso de um padrão de metadados, além de permitir a busca e localização de dados existentes, ainda fornece outras informações. O Projeto GeoMINAS original foi um dos pioneiros no Brasil a mobilizar e reunir um grupo de profissionais de diferentes instituições com um interesse comum e era, até recentemente, uma importante fonte de

dados geoespaciais. No entanto, sua desativação deixou muitos usuários potenciais sem o acesso a este importante repositório de dados.

A IDE GeoMINAS além de prover acesso aos dados do antigo site, reestruturou a documentação dos dados de acordo com o novo padrão de metadados brasileiro, o Perfil MGB. A opção de utilizar o software GeoNetwork possibilitou adicionar ao GeoMINAS novas funcionalidades, pois além de facilitar a localização e permitir que o usuário adquira os dados para seu computador, possui funcionalidades de WebSIG, ou seja, permite ao usuário consultar e realizar análises espaciais simples usando apenas o *browser*.

Com a IDE GeoMINAS o repositório de dados deixa de ser estático e disponibiliza um ambiente onde os usuários podem facilmente publicar seus dados. Como trabalhos futuros, serão desenvolvidos novos serviços Web e pretende-se integrar a IDE GeoMINAS à INDE, como mais um nó da rede de servidores de dados geoespaciais.

Agradecimentos

Projeto parcialmente financiado com recursos da FAPEMIG, CAPES e do CNPq. Os autores também agradecem aos pesquisadores que disponibilizaram suas bases de dados do GeoMINAS, Prof. Clodoveu A. Davis Jr. (DCC/UFMG), Prof. Elpídio I. Fernandes Filho (Departamento de Solos/UFV), Sr. Anderson Meira (iPlanus) e ao INPE.

Referências Bibliográficas

Comissão Nacional de Cartografia (CONCAR). *Perfil de Metadados Geoespaciais do Brasil* – Perfil MGB. Brasília: Ministério do Planejamento, 2009. 194p.

Comissão Nacional De Cartografia (CONCAR). *Plano De Ação Para Implantação da INDE*: Infraestrutura De Dados Espaciais. Brasília: Ministério Do Planejamento, 2010. 203p.

Federal Geographic Data Committee (FGDC). *Content standard for digital geospatial metadata*. Washington: D.C.: Federal Geographic Data Committee, 1998. 78p.

International Organization for Standardization (ISO). *ISO 19115*: geographic information– metadata. Genève, Switzerland, 2003.

Maguire, D. J.; Longley, P. A. The emergence of geoportals and their role in spatial data infrastructures. *Computers, Environment and Urban Systems*, n. 29, p. 3-14, 2005.

Nogueras-Iso, J.; Zarazaga-Soria, F. J.; Muro-Medrano, P. R. *Geographic information metadata for spatial data infrastructures*. New York: Springer, 2005a.

Nogueras-Iso, J.; Zarazaga-Soria, F. J.; Béjar, R.; Álvarez, P. J.; Muro-Medrano, P. R. OGC Catalog Services: a key element for the development of Spatial Data Infrastructures. *Computers & Geosciences*, v.31, p.199-209, 2005b.

Rajabifard, A. Spatial data infrastructures: concept, SDI hierarchy and future directions. *Proceedings of GEOMATICS*, 2001.

Using linked data to extract geo-knowledge

Matheus Silva Mota[1], **João Sávio Ceregatti Longo**[1]
Daniel Cintra Cugler[1], **Claudia Bauzer Medeiros**[1]

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

{matheus,joaosavio}@lis.ic.unicamp.br, {danielcugler,cmbm}@ic.unicamp.br

Abstract. *There are several approaches to extract geo-knowledge from documents and textual fields in databases. Most of them focus on detecting geographic evidence, from which the associated geographic location can be determined. This paper is based on a different premise – geo-knowledge can be extracted even from non-geographic evidence, taking advantage of the linked data paradigm. The paper gives an overview of our approach and presents two case studies to extract geo-knowledge from documents and databases in the biodiversity domain.*

1. Introduction

There is extensive research on extracting geographical knowledge from documents, mostly based on text analysis on documents and textual fields in databases. Basically, those papers try to find geographic references (e.g., matching place names according to a dictionary), and subsequently correlate the text with specific regions, points etc. [Odon de Alencar et al. 2010, Strötgen et al. 2010]. This often implies in issues of georeferencing, token indexing and document corpus analysis algorithms (e.g., using gazeteers [Nadeau et al. 2006] or geographical databases [Gomes and Medeiros 2007]). Furthermore, there are problems related to the heterogeneity of formats, since most of the solutions focus on interoperable formats (e.g., HTML) or some specific format (e.g., PDF).

Our work differs from such research efforts in three directions. (i) In case of documents, rather than analyzing the document itself, our strategy focus on an intermediate and interoperable document descriptor; (ii) instead of focusing only in geographical evidence, we also process other elements that may indirectly be associated with geo-information; (iii) we take advantage of the Open Linked Data Initiative to infer geo-knowledge.

This paper presents our work and two case studies involving our approach for the biodiversity domain. The first study processes research papers concerning biodiversity issues to extract geographic knowledge from non-geographic elements via linked data. The second case study processes a database of metadata of sound recordings of animals connecting the metadata with ontology terms and linked structured data. The first case study uses the notion of document descriptor and linked data, while the second connects metadata fields directly to semantic information.

This work is being developed at *LIS – Laboratory of Information Systems* – at the Institute of Computing, UNICAMP. This paper is organized as follows. Section 2 introduces concepts and related work. Section 3 presents two case studies and our effort to extract geo-knowledge from documents and databases using linked data. Finally, Section 4 presents conclusions and ongoing work.

2. Concepts and Related Work

2.1. Semantic Web, Linked Data and the Linking Open Data Project

The Semantic Web is commonly defined as the Web of Data [Auer et al. 2007]. The main difference between the Web as we know today and the Semantic Web is the focus on the meaning of the data, not only on availability and sharing as before. This information is not related to human consumption, but aims to help machines to understand and consume the information on the World Wide Web [Auer et al. 2007, Berners-Lee et al. 2001].

The notion of *Linked Data* appeared in the Semantic Web context. The term is related to a set of practices for publishing and sharing structured data on the Web. Basically, Linked Data uses the RDF (Resource Descriptor Framework) format to make typed statements that link things [Bizer et al. 2009b, Bizer et al. 2009a]. The 4 "rules" of linked data are: (i) Use URIs as names of things; (ii) use HTTP URIs so that people can look up those names; (iii) when someone looks up a URI, provide useful information; and (iv) include links to other URIs, so they can discover more things [Berners-Lee et al. 2001].

The Linking Open Data (LOD) is a W3C project related to the linked data publishing method. Its main goal is make several open data sets available and connected on the Web (such as DBPedia, Geonames, WordNet, the DBLP bibliography, the GeoSpecies Knowledge Base etc.). To do that, the data sets must publish the data as RDF, using URIs to link resources on/via Web [Auer et al. 2007, Bizer et al. 2009a]. Hence, LOD aims to create a machine consumable interlinked graph. The right part of Figure 1 shows some data sets that are interlinked by the project – 203 data sets, over 25 billion RDF triples interlinked by around 395 million RDF links (as of September 2010).

2.1.1. Semantic Annotations for Geographical Data

The main idea of Semantic Annotations is derived from textual annotations [Oren et al. 2006]. Such annotations can have different objectives and be structured in many forms, e.g., free remarks, tags, floating layers [Euzenat 2002]. There are three main approaches to create annotations: manually [Lesaffre et al. 2003], semi-automatically [Macario and Medeiros 2009] and automatically [Cano 2004].

Annotations are used, among others, to describe a resource and what it represents. Informal ones are usually inserted on documents for human consumption. This hampers computer processing and annotation exchange. Semantic annotations appeared with the purpose of third-party interpretation, providing explicit and machine interpretable semantics, as supported by Semantic Web standards [Euzenat 2002]. Annotations acquire more semantics when they follow structural schemes and relate concepts and relationships between concepts and/or resources. This strategy allows machine consumption, therefore the development of new types of applications [Kiryakov et al. 2004], such as text categorization, content and multimodal information retrieval.

Semantic annotations in the geographic context should also consider the spatial component. Therefore, "the geospatial annotation process should be based on geospatial evidence - those that conduct to a geographic locality or phenomenon" [Macario and Medeiros 2009].

From a high level point of view, Wikipedia, for instance, has some kind of geographic semantic annotations. Latitude and longitude are provided through a coordinate template named *geotag*. Thus, the task to get this kind of information is straightforward. Relying on this fact, [Okamoto et al. 2010] focus on extracting and visualizing spatiotemporal data from Wikipedia. Moreover, [Odon de Alencar et al. 2010] show the feasibility of classifying Wikipedia documents according to their association to places. The method tries to find evidence of localization, searching in the graph formed by links. The authors claim it is almost always possible to say which location is related to a particular document.

2.2. Shadow-driven Document Representation

The SdR – *Shadow-driven Representation* – strategy is based on building an interoperable document descriptor that summarizes elements of a document. A *document shadow* can be seen as a generic structure (well formed XML) that isolates relevant elements of a document from its format, allowing its indexing and annotation/retrieval. These elements (e.g., title, authors, tables, figures, captions, sections) are defined by users: distinct group of users may have different needs on document management. The concept of shadow is akin to that of image descriptor: it summarizes key aspects of a document according to a predefined set of document features (elements of interest).

Figure 1 represents an abstraction of the SdR approach. A shadow element is associated to a fragment of a document (extreme left). The middle of Figure 1 (b) presents an abstraction of the internal structure of a shadow, with contents and structures: here, a document contains pages, which contain paragraphs etc. The production of a document shadow is divided in two steps: (a) Definition, by users. of the elements of interest that should be present in the shadow (the schema); and (b) instantiation of the shadow for each document, based on these elements.

3. Using linked data to extract geo-knowledge: Case Studies

This section presents two case studies where we use Linked Data to extract geo-knowledge from papers on biodiversity (Section 3.1) and from a metadata database (Section 3.2).

3.1. Extracting geo-knowledge from Biodiversity Documents Using Shadows

Rather than analyzing the text directly to extract geo-knowledge, the basis of our strategy focus on extracting this knowledge from shadows. This strategy, developed by us, isolates concerns on document format from query processing itself (and has allowed us to use a single set of code and algorithms to extract geographical knowledge from documents created using Adobe PDF, MS-Word and Open Office).

Instead of restricting ourselves to geographical data, we also process data indirectly associated with geographic references (e.g., images can be connected to their meaning on the semantic web, the authors of a paper can be connected to their birthplaces, conference proceedings can be connected with where the conference took place or the address of the publishers). The middle and right parts of Figure 1 present an abstraction of the connection between a shadow element (addressable via URI) and its meaning in a specific data set of the Linking Open Data Project (LOD).

For instance, consider a paper that contains an image of an animal. The image label identifies the species name. Using the species name, we can find its URI in the

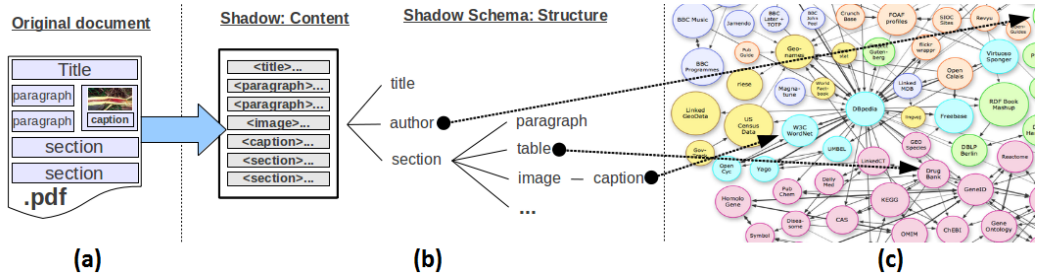


Figure 1. Abstraction of the relation between a shadow, the corresponding document and the link between an element and an external data set

LOD, and therefore additional geo-information about that species which are dispersed over different data sets. The paper's author can also be linked similarly, and so on. Hence, the shadow can now be used to answer queries such as "*what researchers have written papers on animals that are found within X kilometers of their work place?*", or "*group papers by geographic regions of where the species described can be found*", or "*given a document, show where mentioned species can be found*", or even "*which documents mention species that appear in a polygon P?*".

3.2. Using Geo-knowledge and Linked Data to Improve Queries in the Biodiversity Context

Between 1961 and 2010, researchers from Fonoteca Neotropical "Jacques Vielliard" – UNICAMP – have recorded about 20.000 animal sounds, creating the largest collection of animal recordings in the Neotropics. When biologists record animal sounds, a set of metadata related to the recordings is also collected – e.g., air and/or water temperature, weather conditions, country/state/city where the sound was recorded, and so on.

Managing and retrieving animal sounds and their metadata pose countless challenges. We developed a web system where one can, among other functionalities, retrieve animal sounds based on their metadata [Cugler et al. 2011]. Since most of the data were collected in the 70's and 80's, coordinates are not provided. Moreover, even location names may not allow determining these coordinates, since they may contain notes such as "São Marcos Indian Reservation, Namun Kurá Village". One additional issue is that biologists would like to further exploit the metadata, by finding correlations with other facts.

To overcome these challenges, we performed a case study using this collection. Our focus was to use geo-knowledge and linked data as a bridge to connect animal recordings with the LOD data sets (see screen copy of our prototype, Figure 2). Metadata fields are used as concepts to be sought in LOD (in data sets such as DBPedia and GeoSpecies Knowledge Base). Once the concept is found in the LOD, it is possible to extract associated geo-knowledge. For instance, if a metadata field has a city name, then latitude/longitude can be retrieved from LOD, stored in a geographic database and used for other more complex queries. Linked data is able to provide additional information - e.g., region surface, altitude, description, population density, climate and time zone. Other kinds of information can also be retrieved, e.g., details about the animal whose sound was collected.

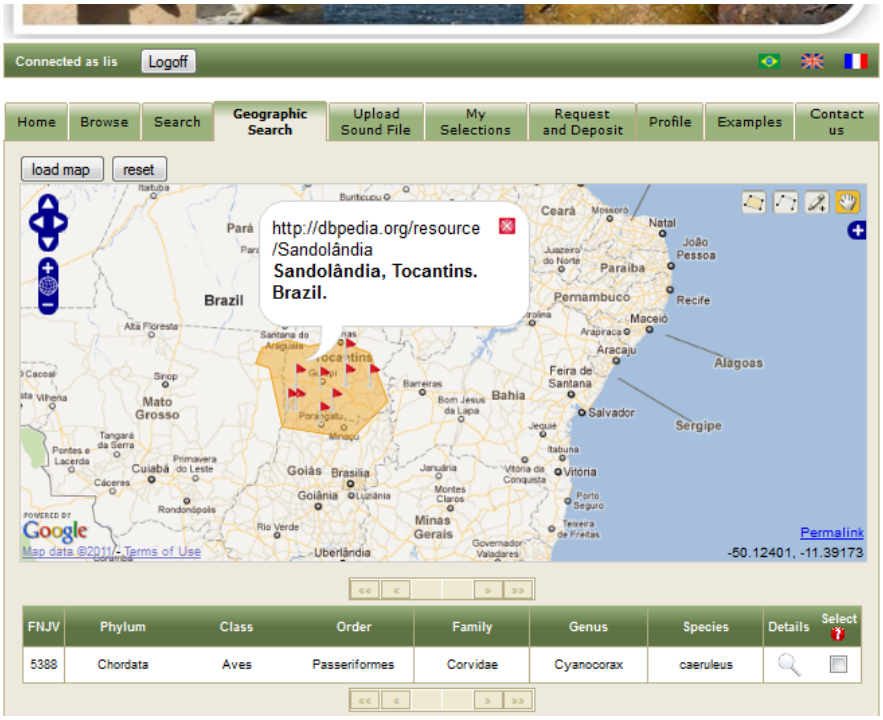


Figure 2. Screen copy of the prototype

The link between recordings and data from the LOD data sets provides plenty of semantic information. The limit of available fields for filtering queries is proportional to the size of the LOD Initiative. If it grows up, so do the filtering possibilities.

4. Concluding Remarks

This paper describes work in progress in combining shadows and semantic information via LOD. It proposes a different approach to extract geo-knowledge from documents and textual fields in databases. Our strategy adopts a notion of "document descriptor" (shadows) to handle the documents independently of file formats. It also takes advantage of the LOD project to extract geo-knowledge from non-geographic information.

We present two case studies where we connect elements of documents (through shadows) and a metadata database in biodiversity with open data sets (such as DBPedia and GeoSpecies Knowledge Base) in order to extract geo-knowledge and allow more sophisticated queries that will go beyond geographic references.

Acknowledgments

Research partially financed by CNPq, FAPESP, CAPES and INCT in Web Science.

References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer Berlin / Heidelberg.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5):28–37.

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

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009b). Dbpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154 – 165.

Cano, P. (2004). Automatic sound annotation. In *In IEEE workshop on Machine Learning for Signal Processing*, pages 391–400.

Cugler, D., Medeiros, C., and Toledo, L. (2011). Managing animal sounds-some challenges and research directions. *Proceedings V eScience Workshop - XXXI Brazilian Computer Society Conference*.

Euzenat, J. (2002). Eight questions about semantic web annotations. *IEEE Intelligent S.*, 17(2):55–62.

Gomes, L. C. and Medeiros, C. B. (2007). Ecologically-aware queries for biodiversity research. In *Proceedings of GeoInfo - Brazilian Geoinformatics Symposium*, pages 73–84.

Kiryakov, A., Popov, B., Terziev, I., Manov, D., and Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1):49 – 79.

Lesaffre, M., Tanghe, K., Martens, G., Moelants, D., Leman, M., Baets, B. D., Meyer, H. D., and Martens, J.-P. (2003). The mami query-by-voice experiment: Collecting and annotating vocal queries for music information retrieval. In *In: Proceedings of the International Conference on Music Information Retrieval*, pages 26–30.

Macario, C. G. N. and Medeiros, C. B. (2009). A framework for semantic annotation of geospatial data for agriculture. *Int. J. Metadata, Semantics and Ontology - Special Issue on "Agricultural Metadata and Semantics"*, 4:118–132.

Nadeau, D., Turney, P., and Matwin, S. (2006). Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Advances in Artificial Intelligence*, volume 4013 of *Lecture Notes in Computer Science*, pages 266–277.

Odon de Alencar, R., Davis, Jr., C. A., and Gonçalves, M. A. (2010). Geographical classification of documents using evidence from wikipedia. In *Proceedings of the 6th Workshop on Geographic Information Retrieval*, GIR '10, pages 12:1–12:8. ACM.

Okamoto, A., Yokoyama, S., Fukuta, N., and Ishikawa, H. (2010). Proposal of spatiotemporal data extraction and visualization system based on wikipedia for application to earth science. In *Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on*, pages 651–656.

Oren, E., Moller, K. H., Scerri, S., Handschuh, S., and Sintek, M. (2006). What are semantic annotations?? Technical report, DERI Galway.

Strötgen, J., Gertz, M., and Popov, P. (2010). Extraction and exploration of spatiotemporal information in documents. In *Proceedings of the 6th Workshop on Geographic Information Retrieval*, GIR '10, pages 16:1–16:8, New York, NY, USA. ACM.

Sistema Interativo para Posicionamento de Observadores em Terrenos Representados por Modelos Digitais de Elevação

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

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

chaulio@hotmail.com,{smagalhaes,marcus}@dpi.ufv.br

Resumo. *Este trabalho apresenta um sistema interativo para o posicionamento de observadores em terrenos representados por modelos digitais de elevação. O sistema é composto por uma interface gráfica na qual o usuário pode carregar o terreno e selecionar uma heurística para o posicionamento automático de observadores sobre este terreno. Então, a solução obtida pela heurística é apresentada ao usuário e ele pode modificar interativamente esta solução incluindo, removendo ou alterando a posição dos observadores. Ao modificar uma solução, o usuário pode visualizar em tempo real os efeitos causados pelas modificações realizadas como por exemplo, o aumento na taxa de cobertura.*

Abstract. *This work presents an interactive system to site observers in a terrain represented by digital elevation models. The system is composed by a graphical interface where the user can load a terrain and select a heuristic for observer siting on this terrain. The interface displays the solution obtained by the heuristic and it allows the user interactively to modify this solution including, removing or changing the position of the observers. The modified solution is displayed on real time and so, the user can see the effects of the modifications such as, the coverage increase.*

1. Introdução

Com a grande disponibilidade de dados geográficos de alta resolução obtidos a partir de tecnologias como LIDAR e IFSAR é importante desenvolver novos métodos em Sistemas de Informação Geográfica [7] (SIG) para processá-los.

Um importante grupo de aplicações de SIG é relacionado a modelagem de terrenos e, dentre os diversos tipos de operações que utilizam modelos de terrenos, uma operação importante é o cálculo do conjunto de pontos do terreno que são visíveis a partir de um ponto particular (o *observador*). Este problema possui diversas aplicações em telecomunicações, planejamento ambiental, navegação de veículos autônomos, etc [8, 10, 1]. Outro problema importante é posicionar um determinado número de observadores com o objetivo de "cobrir o terreno". Tais observadores podem representar, por exemplo, torres de internet, de televisão ou de monitoramento florestal [2, 3].

Este artigo apresenta um sistema interativo para o posicionamento de observadores em terrenos. A idéia básica é utilizar uma interface gráfica que permita que o usuário atue de maneira direta e interativa sobre as soluções obtidas pelas heurísticas de posicionamento fornecidas pelo sistema. Ou seja, o sistema disponibiliza um conjunto de

heurísticas que podem ser utilizadas para a obtenção de uma solução inicial, que é exibida na tela. Então, o usuário pode realizar alterações manuais nesta solução incluindo, removendo ou modificando a posição do(s) observadore(s) com o intuito de melhorá-la. Esta melhora pode ser aumentando a cobertura obtida através da inclusão de novos observadores ou pelo reposicionando os observadores utilizados. Também é possível tentar reduzir o número de observadores mantendo a cobertura próxima do valor desejado.

2. Visibilidade em terrenos

Um *terreno* é uma representação da elevação da superfície terrestre em uma determinada região. Em geral, tal representação é realizada utilizando modelos digitais de elevação. Neste trabalho, o terreno será representado utilizando modelos digitais de elevação *raster* (ou *DEM*) que armazenam a elevação de amostras do terreno regularmente espaçadas[4].

Um *observador* é um ponto do espaço que "deseja" visualizar ou se comunicar com outros pontos do espaço chamados de *alvos*. A notação usual para observadores e alvos é O e T. Os *pontos base* de O e T sao os pontos O_b e T_b do terreno que se localizam abaixo de, respectivamente, O e T. A visão do observador é limitada por um valor R que representa o alcance máximo chamado de *raio de interesse*. Por exemplo, se O representa uma torre de observação, R é a distância máxima que uma pessoa no topo da torre é capaz de ver na ausência de obstruções.

Um alvo T é *visível* a partir de um observador O se, e somente se, T estiver dentro do raio de interesse de O e não houver nenhum ponto do terreno que bloqueia o segmento de reta, chamado de *Linha de Visão* (ou *LOS - line of sight*), que conecta O a T. Veja a Figura 1. Nesta figura, T_2 e T_4 são visíveis a partir de O e T_1 e T_3 não são visíveis.

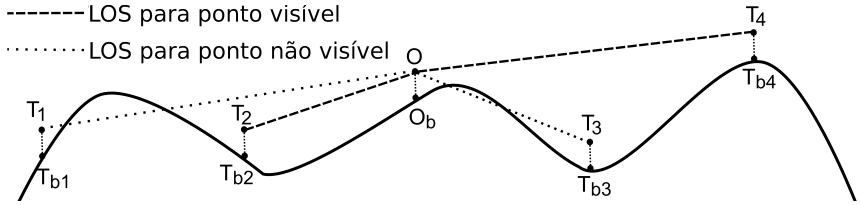


Figura 1. Determinação da visibilidade de pontos utilizando uma *LOS* em uma secção vertical de terreno.

O conjunto V de todos os pontos base que são visíveis a partir de O é chamado de *viewshed* de O. V é normalmente armazenado utilizando uma matriz de bits onde o valor 1 indica que um determinado ponto é visível e o valor 0 indica que o ponto não é visível.

O *índice de visibilidade*, $VIX(O)$ é o número de alvos que são visíveis a partir de O. Este valor, que representa o número de bits 1 em V, normalmente é estimado utilizando uma amostragem de alvos escolhidos dentro do alcance visual de O.

O *viewshed acumulado*, V, de um conjunto de observadores $\mathcal{O} = \{O_i\}$ representa a união dos viewsheds $V(O_i)$, ou seja, é a aplicação da operação binária *OR* nas matrizes de bits destes *viewsheds*. O *índice de visibilidade acumulado*, $VIX(\mathcal{O})$ é o número de alvos no terreno que são visíveis a partir de pelo menos um observador em $\mathcal{O}$. Este valor é normalmente normalizado para representar um percentual da área do terreno.

O *problema de posicionamento de múltiplos observadores* consiste em otimizar a localização de um conjunto de observadores de modo a maximiar o índice de visibilidade acumulado destes observadores[6, 9]. Este problema é NP-Completo [10] e possui várias aplicações práticas como, por exemplo, otimizar o posicionamento de torres de observação, sistemas de radares ou torres de telefonia celular.

As heurísticas descritas na seção 3 consideram uma variação do problema de posicionamento de múltiplos observadores onde o objetivo é minimizar o número de observadores necessários para se cobrir um percentual do terreno. Mais especificamente, dado um conjunto $P = \{P_i\}$ de pontos candidatos a receberem observadores, o problema consiste em encontrar o menor subconjunto $S = \{S_i\}$ de P cujo índice de visibilidade acumulado seja maior ou igual a um determinado valor VIX_{min}.

3. Heurísticas para o posicionamento de observadores

Em [9] são propostas duas novas heurísticas para resolver a variação do problema de posicionamento de múltiplos observadores descrita acima. Ambas se baseiam no uso de uma busca local para estender ou adaptar outras heurísticas, como o método *Site* proposto por Franklin e outros [5] para aproximar a solução do problema de posicionamento de múltiplos observadores e o método GRASP [11], uma metaheurística geral de otimização.

O método *Site* utiliza uma abordagem gulosa para construir, de forma iterativa, a solução (representada pelo conjunto S) da seguinte forma: dado um conjunto $P = \{P_i\}$ de pontos candidatos a receberem observadores, em cada passo, o ponto P_i que contribuir mais para o aumento do índice de visibilidade de S é inserido em S. Este processo é executado enquanto $VIX(S)$ for menor do que VIX_{min}.

Em [9], este método foi estendido com uma busca local para tentar melhorar a solução parcial obtida em cada passo do método guloso. Mais especificamente, dada uma solução parcial S obtida em uma iteração do método guloso, é criada uma vizinhança contendo todas as soluções que estiverem "próximas" a S e, então, soluções "melhores" são procuradas nesta vizinhança. Uma solução S' é considerada vizinha a uma solução S se as duas soluções possuirem o mesmo número de observadores e houver apenas um observador em S que seja diferente dos observadores de S'. Além disso, a distância entre os observadores diferentes de S e S' deve ser menor do que um dado limite.

O método GRASP, originalmente proposto por Resende e outros [11] consiste em criar uma solução inicial utilizando um método construtivo e, então, melhorar essa solução com uma busca local. Em [9], esta heurística foi adaptada para resolver o problema de posicionamento de múltiplos observadores.

Conforme descrito em [9], as soluções obtidas pela heurística baseada em GRASP e pela heurística baseada no método guloso melhorado com busca local utilizam, em média, 7% menos observadores do que as soluções obtidas pelo método *Site*. Vale mencionar que, em alguns casos de teste o método guloso melhorado com busca local apresenta resultados melhores do que o método GRASP e, em outros casos de teste, a heurística GRASP apresenta resultados melhores.

4. O sistema

Como dito anteriormente, não há uma heurística mais adequada para o posicionamento de observadores e, portanto, o objetivo desse trabalho é desenvolver um sistema interativo

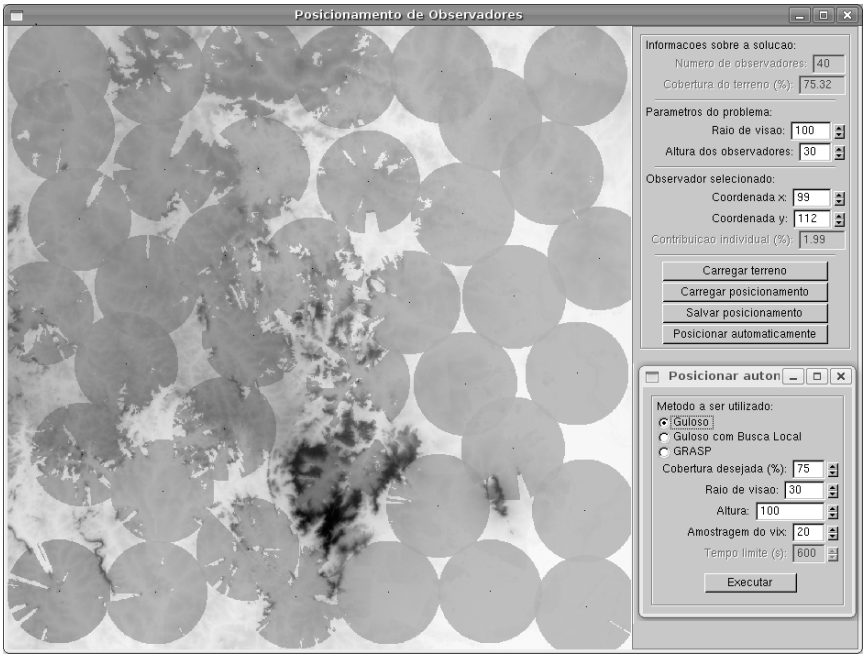


Figura 2. Janela principal do sistema após um terreno ser carregado.

que permita ao usuário selecionar a heurística a ser utilizada e, considerando a solução obtida, ele possa modificar interativamente esta solução incluindo, removendo ou modificando a posição do(s) observadore(s) com o intuito de melhorar a solução. Esta melhora pode ser devido ao aumento da cobertura obtida incluindo-se novos observadores ou reposicionando-se os observadores utilizados ou então devido à redução do número de observadores mantendo-se a cobertura próxima do valor desejado.

A interface do sistema foi desenvolvida na linguagem C++ utilizando a biblioteca Qt. Inicialmente, é apresentada uma janela onde o usuário deverá fornecer os parâmetros para o sistema sendo que o primeiro deles é o arquivo contendo a matriz de elevação que representa o terreno. Então, a imagem deste terreno é mostrada na janela de visualização e então, o sistema exibe um menu *pop-up*, no canto inferior direito, onde o usuário deverá escolher a heurística a ser utilizada para obtenção da solução inicial cujas opções são: *Guloso*, *GulosoBL* e *GRASP*. Neste menu, além de selecionar a heurística, o usuário deverá informar os seguintes parâmetros: percentual da cobertura desejada, raio de visão do observador (em metros), altura (em metros) acima do solo dos observadores e dos alvos e o tempo máximo (utilizado como critério de parada pela heurística *GRASP* - esse parâmetro só é solicitado caso seja selecionada a heurística *GRASP*). Após clicar no botão *Executar*, a solução obtida é exibida na janela de visualização. A Figura 2 apresenta a solução obtida pela heurística *Guloso*.

Após realizar o posicionamento automatizado, as informações sobre a solução gerada são exibidas no canto superior direito onde o usuário pode realizar ajustes na solução modificando estes valores que são: o número de observadores utilizados, o percentual de cobertura alcançado, as coordenadas e a taxa de contribuição de um observador selecionado. Além disso, neste painel também há alguns botões que permitem que usuário

realize ações como: carregar um novo terreno, carregar um posicionamento anteriormente calculado, salvar o posicionamento obtido ou obter um novo posicionamento.

Quando o usuário modifica algum parâmetro (por exemplo, quando ele modifica a posição de um observador) a solução apresentada na janela de visualização é automaticamente atualizada para exibir a nova solução obtida com estes parâmetros.

Um observador pode ser selecionado clicando-se em seu *viewshed* (o *viewshed* de um observador selecionado é colorido com a cor azul). Após selecionar um observador, o usuário pode excluí-lo da solução atual utilizando a tecla R do teclado ou modificar a sua posição no terreno arrastando-o com o mouse ou digitando no painel as suas coordenadas.

Ao mover um observador, a visualização da solução é modificada em tempo real na janela de visualização de forma que o usuário possa ver a cobertura do terreno à medida em que o observador é deslocado. Vale mencionar que, por motivos de eficiência, o *viewshed* acumulado que é exibido na janela não é recalculado completamente para cada posição do terreno na qual o usuário passa com o mouse. Mais especificamente, quando um observador é selecionado pelo usuário, o *viewshed* acumulado V dos outros observadores da solução é calculado e, à medida em que o usuário move o observador selecionado, a imagem na janela de visualização é atualizada. Por exemplo, se o movimento é feito arrastando-se o mouse, a cada evento gerado pela biblioteca gráfica utilizada, o *viewshed* acumulado V é redesenhado na tela e, então, o *viewshed* do ponto correspondente à nova posição do mouse é desenhado de forma sobreposta a V.

O sistema também permite que o usuário adicione novos observadores à solução corrente. Para isso, o usuário deve pressionar a tecla A e clicar com o botão esquerdo do mouse na posição desejada do terreno.

5. Conclusão e trabalhos futuros

Neste trabalho foi apresentado um sistema interativo para o posicionamento de observadores em terrenos representados por modelos digitais de elevação que permite ao usuário a seleção da heurística a ser utilizada para a obtenção de uma solução inicial. Então, o usuário pode tentar melhorar esta solução incluindo, removendo ou reposicionando observadores. Esta melhora pode se dar pelo aumento do percentual de cobertura do terreno através da inclusão de novos observadores ou pelo reposicionamento de observadores em regiões com baixa cobertura. Outra opção é a remoção de observadores com pequena contribuição permitindo a redução do número de observadores e mantendo-se a cobertura próxima do valor desejado. Também é possível avaliar o impacto de uma modificação na altura dos observadores. Um pequeno aumento na altura pode levar a consideráveis alterações na solução: pode permitir a redução do número de observadores necessários para se obter a mesma cobertura ou pode levar a uma melhoria na cobertura obtida com os mesmos observadores. Ou até mesmo, a alteração na altura dos observadores pode fazer com que um novo conjunto de observadores alcance um melhor resultado.

Assim, pode-se concluir que este sistema pode auxiliar bastante o processo de obtenção da solução visto que o usuário pode atuar de maneira decisiva neste processo. Ele também auxilia no processo de tomada de decisão pois o usuário pode analisar de maneira objetiva vários cenários. Note que em situações onde um observador representa um item com alto custo de instalação, por exemplo, uma antena de telefonia celular, as melhorias podem significar uma considerável redução de custos.

Como trabalhos futuro, pretende-se melhorar os recursos visuais da interface incorporando recursos que permitam a visualização tridimensional de modo que o usuário possa visualizar a solução de diferentes ponto de vista. Uma outra funcionalidade a ser incluída é a opção de sobreposição de diferentes mapas, por exemplo, mapas urbanos, imagens de satélites, etc. Também pretende-se incluir a opção que permita ao usuário alterar a altura de um único observador.

Adicionalmente, as janelas da interface serão adequadas aos padrões definidos pelo sistema de informações geográficas GRASS de modo que o sistema proposto possa ser disponibilizado como um módulo deste SIG.

Agradecimentos

Este trabalho foi parcialmente financiado pela FAPEMIG - Fundação de Amparo à Pesquisa do Estado de Minas Gerais e pelo CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico.

Referências

[1] Marcus V. A. Andrade, Salles V. G. Magalhães, Mirella A. Magalhães, W. Randolph Franklin, and Barbara M. Cutler. Efficient viewshed computation on terrain in external memory. *GeoInformatica*, 2009.

[2] Boaz Ben-Moshe. *Geometric Facility Location Optimization*. Phd thesis, Ben-Gurion University, Israel, Department of Computer Science, 2005.

[3] Yehuda Ben-Shimol, Boaz Ben-Moshe, Yoav Ben-Yehezkel, Amit Dvir, and Michael Segal. Automated antenna positioning algorithms for wireless fixed-access networks. *Journal of Heuristics*, 13(3):243–263, 2007.

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

[5] W. Randolph Franklin. Siting observers on terrain. In Dianne Richardson and Peter van Oosterom, editors, *Advances in Spatial Data Handling: 10th International Symposium on Spatial Data Handling*, pages 109–120. Springer, 2002.

[6] Young hoon Kim, Sanjay Rana, and Steve Wise. Exploring multiple viewshed analysis using terrain features and optimisation techniques. *Computers & Geosciences*, 30:1019–1032, 2004.

[7] Robert Laurini and Derek Thompson. *Fundamentals of Spatial Information Systems*. Academic Press, 1992.

[8] Z. Li, Q. Zhu, and C. Gold. *Digital Terrain Modeling - principles and methodology*. CRC Press, 2005.

[9] Salles V. G. Magalhães, Marcus Vinícius Alvim Andrade, and Chaulio Ferreira. Heuristics to site observers in a terrain represented by a digital elevation matrix. In *GeoInfo*, pages 110–121, 2010.

[10] G. Nagy. Terrain visibility. *Computers and Graphics*, 18:763–773, 1994.

[11] Mauricio G. C. Resende. Greedy randomized adaptive search procedures (grasp). *Journal of Global Optimization*, 6:109–133, 1999.

Um método para o cálculo da barragem necessária para gerar um reservatório com um determinado volume

Rodolfo Ladeira[1], **Salles Magalhães**[1], **Marcus Andrade**[1], **Mauricio Gruppi**[1]

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

rodolfo53821@hotmail.com,{smagalhaes,marcus,mgruppi}@dpi.ufv.br

Abstract. *This work describes a new method to compute the height and the extension of a dam which have to be build to generate a reservoir with a given volume. This method simulates the flooding process to compute the flooded region incrementally. Its main advantage is that the dam height and extension is obtained as a consequence of the flooding process.*

Resumo. *Este trabalho descreve um novo método para a determinação da altura e extensão da barragem a ser construída num ponto de um terreno para se gerar um reservatório com uma determinada capacidade dada. Este método consiste em simular um processo de alagamento onde a região alagada é calculada de forma incremental. A sua principal vantagem é que os dados da barragem são obtidos como consequência do processo de inundação.*

1. Introdução

A água é um recurso natural essencial à vida humana e sua disponibilidade vem se reduzindo devido a inúmeros fatores como o aumento da área de agricultura irrigada, o crescimento populacional e consequentemente, o aumento do consumo urbano, desmatamento de regiões de nascentes, etc [2, 3]. Diante disso, o gerenciamento adequado dos recursos hídricos tem se tornado cada vez mais importante e, dentre os vários elementos envolvidos nesse gerenciamento, temos o processo de regularização de vazões que consiste em adotar medidas para manter a disponibilidade de água mesmo em períodos de estiagem e queda da vazão em cursos d´água. Geralmente, este processo de regularização é realizado pelo represamento das águas através da construção de barragens em trechos bem determinados dos cursos d´água naturais. Os reservatórios têm por objetivo acumular parte da água disponível nos períodos chuvosos para compensar as deficiências nos períodos de estiagem, exercendo assim um efeito regularizador das vazões naturais.

Mais especificamente, a regularização das vazões por meio da construção de barragem (formação de reservatório) visa atingir os seguintes objetivos: o atendimento às necessidades do abastecimento urbano ou rural (irrigação); o aproveitamento hidroelétrico (geração de energia); a atenuação de cheias (combate às inundações); o controle de estiagens; o controle de sedimentos; a recreação; e, também, permitir a navegação fluvial.

Conforme descrito em [3], uma questão importante no processo de regularização é a definição da capacidade (o volume máximo) do reservatório a ser construído. Esta capacidade depende da altura da barragem e das características topográficas da região onde a barragem é construída. Todo reservatório possui um nível máximo e um nível mínimo operacional que correspondem, respectivamente, à altura máxima e mínima que

a água pode alcançar quando as águas se elevam ou abaixam em condições normais de operação. A Figura 1 apresenta um exemplo de barragem com estes dois tipos de nível operacional. Note que nem toda a água armazenada em um reservatório (a capacidade) está disponível para o uso, pois parte desta água é o volume morto que é formado pelo volume armazenado abaixo do nível mínimo de operação e é destinado a acomodar a carga de sedimentos afluentes ao reservatório durante a sua vida útil. Assim, o volume útil é a diferença entre o volume máximo (capacidade) e o volume morto.

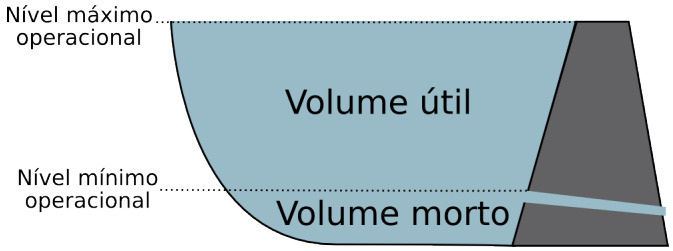


Figura 1. Definição do volume de um reservatório.

Uma forma de calcular a capacidade de um reservatório, descrita em [3], consiste em utilizar um mapa topográfico (em escala adequada) para primeiro obter a relação cota (altura) versus área alagada, isto é, para determinar a área delimitada pelas curvas de nível relativas a cada cota. Então, a relação cota versus capacidade é obtida integrando-se a curva cota versus área. Esta integração é realizada numericamente, determinando-se os volumes entre duas curvas de nível consecutivas, sendo que este volume é obtido de forma aproximada multiplicando-se a média das áreas correspondentes às curvas de nível consecutivas pela diferença entre as cotas dessas curvas.

No método descrito em [1, 6], a capacidade do reservatório é calculada utilizando terreno representado por um modelo digital de elevação (MDE) e, uma vez definida a posição, a extensão e a altura da barragem, o primeiro passo é determinar o conjunto de pontos do terreno que serão alagados devido à construção daquela barragem. Então, supondo que o nível máximo da água no reservatório é dado pela altura da barragem então a capacidade do reservatório é determinada fazendo-se o somatório do volume de água que pode ser armazenado em cada célula da matriz. Este volume é dado pelo produto entre a área da célula vezes a diferença entre a altura da barragem e a elevação do terreno naquela célula. A principal dificuldade deste método é a obtenção da região alagada e, conforme descrito em [6], este processo é realizado utilizando uma busca em largura no terreno a partir do ponto onde a barragem será construída.

Note que nesses métodos, a capacidade do reservatório é obtida a partir da posição, altura e extensão da barragem e portanto, para se gerar um reservatório com uma determinada capacidade é necessário efetuar um processo iterativo que primeiro define a barragem e depois verifica se a capacidade do reservatório gerado alcança o volume desejado.

Neste trabalho é apresentado um novo método onde não é necessário a realização deste processo iterativo cuja execução requer um tempo considerável. O método proposto é dividido em duas etapas. Na primeira etapa, o método RWFlood [4] é utilizado para calcular a rede de drenagem do terreno. Na segunda etapa, é utilizado um processo que simula a inundação do reservatório e determina de forma incremental a região alagada

e seu volume. Dessa forma, a altura e extensão da barragem necessária para gerar um reservatório com a capacidade desejada é obtida como consequência do próprio processo de inundação.

2. Determinação da barragem e da área alagada

Dado um terreno representado por um modelo digital de elevação, suponha que se deseja construir uma barragem num determinado ponto p deste terreno tal que o reservatório gerado seja capaz de armazenar um volume maior ou igual a k. O objetivo é determinar a extensão e a altura da barragem de modo que o reservatório tenha a capacidade desejada.

2.1. Obtenção da rede de drenagem

É importante ressaltar que o ponto p onde se posicionar a barragem deve ser um ponto sobre a rede de drenagem do terreno, isto é, deve ser um ponto em algum rio. Assim, o primeiro passo é a obtenção da rede de drenagem do terreno que será obtida utilizando o método RWFlood, proposto por Magalhães e outros [4], que se baseia em simular o processo de inundação para obter esta rede de drenagem. Na verdade, este processo de inundação é utilizado para se obter a direção de fluxo sendo que a idéia é supor que o terreno é uma ilha, isto é, que o terreno está totalmente cercado por um oceano, e que o nível deste oceano vai se elevando de modo a inundar este terreno. Note que o fluxo da água num terreno segue um caminho inverso ao processo de inundação, isto é, as primeiras células a serem inundadas (onde a água entra no terreno) correspondem às células onde a água escoa para fora do terreno (são a foz dos rios); as próximas células a serem inundadas (vizinhas às primeiras) serão as penúltimas antes das fozes e assim por diante. Portanto, o processo de inundação permite obter a direção de fluxo, pois esta direção corresponde à direção contrária àquela da água do oceano inundando o terreno.

Resumidamente, este processo de inundação é simulado inicializando-se o nível da água como sendo igual à elevação do(s) ponto(s) mais baixo(s) na borda do terreno, ou seja, estes são os primeiros pontos a serem inundados e eles são inseridos em uma fila de prioridade onde o topo contém o ponto de menor elevação. Então, o ponto p no topo da fila é removido e, dentre os seus oito vizinhos, aqueles que ainda não foram visitados (inundados) são inseridos na fila. Porém, se a elevação de um ponto q a ser inserido na fila for menor do que o nível da água (isto é, menor do que a elevação do ponto p) então a elevação de q é aumentada, o que corresponde a inundar o ponto q que passa a ser um ponto já visitado[1]. Note que neste momento a direção de fluxo do ponto q pode ser definida como sendo para o ponto p.

Após o cálculo da direção de fluxo, o algoritmo RWFlood calcula o fluxo acumulado no terreno utilizando uma estratégia baseada em ordenação topológica. Conceitualmente, a ideia é supor a existência de um grafo onde cada vértice representa uma célula do terreno e há uma aresta ligando um vértice v a um vértice u se, e somente se, a direção de escoamento de v aponta para u. Os vértices são inicializados com 1 unidade de água e o processamento se inicia num vértice v cujo grau de entrada é 0. Este vértice é marcado como visitado e, supondo que v direciona o fluxo para o vértice u, então o fluxo do vértice v é adicionado ao fluxo atual do vértice u. Além disso, a aresta que conecta o vértice v ao

[1]Este processo de elevação do nível da água equivale à remoção das depressões utilizada pela maioria dos métodos de obtenção da direção de fluxo.

vértice u é removida reduzindo assim o grau de entrada do vértice u - este vértice u será processado (visitado) quando o seu grau de entrada se tornar 0.

Como apresentado em [4], o método RWFlood pode ser implementado de forma bastante simples e eficiente (com complexidade linear em relação ao tamanho do terreno) chegando a ser 100 vezes mais rápido do que os principais métodos descritos na literatura. Esta eficiência se deve principalmente ao fato de que não é necessário pré-processar o terreno para eliminar as depressões visto que elas são naturalmente removidas durante o processo de alagamento. Além disso, este método também é capaz de processar grandes terrenos com mais de 10^9 células.

2.2. Determinação da altura e extensão da barragem e da área alagada

Após a obtenção da rede de drenagem e da escolha do ponto onde instalar a barragem, o próximo passo é a determinação da altura e extensão da barragem e da área alagada de modo que o reservatório possua a capacidade desejada. Por definição, a barragem será orientada perpendicularmente à direção de fluxo no ponto definido para o posicionamento da barragem. Além disso, a princípio, todos os pontos do terreno nesta linha perpendicular poderiam fazer parte da barragem mas, caso desejado, é possível limitar a extensão máxima da barragem.

Assim, dados o terreno, sua rede de drenagem, o ponto p onde posicionar a barragem e a capacidade desejada k para o reservatório, então a altura e a extensão são determinadas utilizando uma adaptação do processo de inundação descrito anteriormente. Neste caso, o processo de inundação é iniciado no ponto p' vizinho a montante de p e a fila de prioridades Q, organizada com o ponto de menor elevação no topo, irá armazenar os pontos do terreno que estão na iminência de serem inundados, ou seja, os pontos que são adjacentes à região já inundada. Inicialmente, esta fila é inicializada contendo apenas o ponto p' e o nível h da água é inicializado com a elevação de p'. Então, o processo de inundação prossegue removendo o ponto do topo da fila que passa a ser um ponto pertencente à região alagada. Seja q este ponto. Assim, a área da célula correspondente ao ponto q é adicionada à área total da região alagada e consequentemente, os pontos vizinhos a q passam a ser pontos na iminência da inundação - portanto, devem ser inseridos na fila Q. Além disso, caso a elevação de q seja maior do que h, que é o nível atual da água, então este nível é elevado fazendo h passar a ser igual à elevação de q. Então, a capacidade do reservatório é atualizada somando-se à capacidade atual o volume adicionado por esta elevação que é dado pelo produto da área já alagada pela diferença entre o novo nível e o nível anterior. Caso contrário, se a elevação de q é menor do que h então o ponto q é inundado e volume desta inundação é adicionado à capacidade do reservatório. Este volume é dado pela área da célula correspondente ao ponto q pela diferença entre h e a elevação de q . Este processo é repetido até que a capacidade do reservatório passe a ser maior do que k e também até que a elevação do ponto no topo da fila seja maior do que h.

Finalmente, a extensão (efetiva) da barragem é determinada obtendo-se os pontos na linha de posicionamento que irão fazer parte desta barragem. Estes pontos são os pontos nesta linha que são adjacentes a algum ponto que pertence à área alagada.

É importante ressaltar que há alguns casos especiais a serem tratados tais como: o processo de inundação pode atingir células pertencentes a outras bacias (neste caso, não é possível criar um reservatório com a capacidade desejada) e o ponto selecionado para o

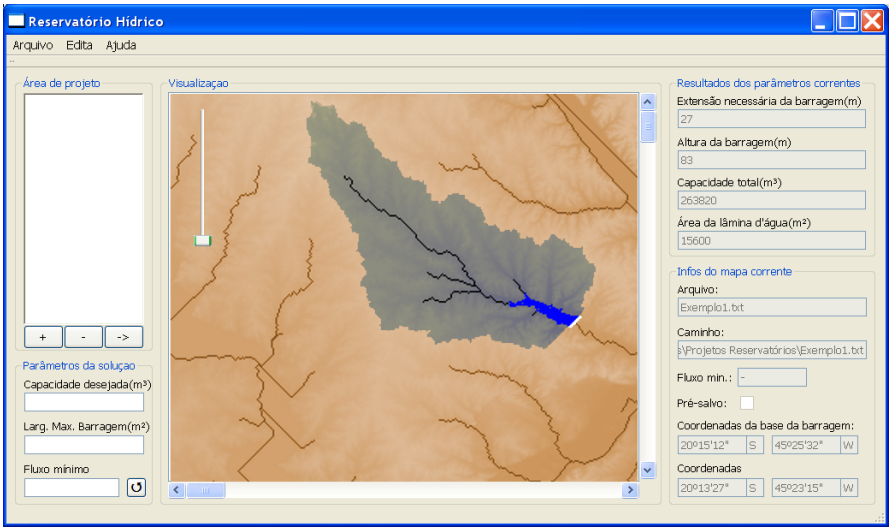


Figura 2. Interface para utilização do método de determinação da barragem

posicionamento da barragem não pode ser um ponto na borda do terreno.

3. Interface para uso do método

Para auxiliar a utilização do método, foi desenvolvida uma primeira versão de interface gráfica, onde o usuário pode definir os parâmetros para o algoritmo e visualizar os resultados. Veja Figura 2. Esta interface foi desenvolvida em C++ utilizando o framework Qt [5] e pode ser compilada para diversas plataformas.

O primeiro passo é fornecer a matriz de elevação que representa o terreno e então, o método computa a rede de drenagem e apresenta a imagem do terreno com esta rede sobreposta. Assim, o usuário pode selecionar o ponto onde ele deseja posicionar a barragem clicando com o mouse sobre um ponto desta rede. Neste momento, ele também deve fornecer a capacidade desejada, em m^3, para o reservatório a ser criado.

Fornecidos estes elementos, o método é executado e os resultados são apresentados na tela. Na imagem podem ser visualizadas a linha que define a barragem e a região alagada que é apresentada em azul. Além disso, a área da bacia de contribuição em relação ao ponto selecionado para o posicionamento da barragem também é apresentada, neste caso, corresponde à região sombreada na figura.

Os valores referentes aos resultados também são apresentados nesta interface, isto é, a interface exibe o valor da altura e da extensão da barragem em metros, a área da região alagada em m^2 e a capacidade do reservatório, em m^3.

Finalmente, através desta interface o usuário pode redefinir alguns parâmetros utilizados pelo método como a extensão máxima permitida para barragem cujo valor padrão é toda a extensão do terreno e também, o valor mínimo do fluxo acumulado para que um ponto faça parte de um rio (este valor é utilizado pelo algoritmo RWFlood).

4. Conclusões e trabalhos futuros

Foi apresentado um método para cálculo da altura e extensão da barragem a ser construída para que seja gerado um reservatório com um determinado volume dado. O método simula de forma incremental o processo de alagamento da região onde o reservatório será gerado. Dessa forma, quando o a capacidade desejada é alcançada, o método pode ser interrompido e, com isso, obtêm-se informações sobre as dimensões da barragem.

Para facilitar o uso do método, foi desenvolvido uma interface que permite ao usuário definir o ponto onde a barragem deve ser construída e visualizar as informações relativas a esse posicionamento.

Como trabalhos futuros, pretende-se efetuar uma melhor avaliação do desempenho e da qualidade dos resultados obtidos pelo método proposto comparando-os com os obtidos por outros métodos.

Além disso, pretende-se utilizar este método para desenvolver um sistema que permita determinar a posição mais adequada, num trecho de rio, para se posicionar a barragem. Este processo deve incluir uma função que compute o "custo"da construção da barragem e do reservatório onde serão avaliados o custo econômico (preço da construção, desapropriação da área alagada, reconstrução de estradas, etc) e o custo social (remoção de população, impacto ambiental, etc).

Agradecimentos

Este trabalho foi parcialmente financiado pela FAPEMIG - Fundação de Amparo à Pesquisa do Estado de Minas Gerais e pelo CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico.

Referências

[1] *ArcGIS tutorial*, http://www.esri.com/software/arcgis/index.html (acessado em agosto de 2011).

[2] J. M. Bravo, W. Collischonn, C. E. M. Tucci, and J. V. Pilar. Otimização de regras de operação de reservatórios com incorporação da previsão de vazão. *Revista Brasileira de Recursos Hídricos*, 13:181–196, 2008.

[3] A. R. Barbosa Jr. Hidrografia aplicada. Notas de aula, Universidade Federal de Ouro Preto, UFOP, 2010.

[4] S. V. G. Magalhães, M. V. A. Andrade, W. R. Franklin, and G. C. Pena. Faster and simpler terrain flow accumulation based on raising the water level. In *submetido para 15th AGILE International Conference on Geographic Information Science (AGILE 2012)*.

[5] Nokia. *Qt designer manual*, http://doc.qt.nokia.com/3.3/designer-manual.html (acessado em agosto de 2011.

[6] J. R. C. Souza, M. V. A. Andrade, and K. Nogueira. Heurística para o posicionamento de reservatórios d'àgua. In *Anais do XI Simposio Brasileiro de Geoinformatica*, 2010.

Algoritmo de Simplificação de TIN para Aplicações de Hidrologia

Eduilson Lívio N. da C. Carneiro[1,2], Laércio M. Namikawa[1], Gilberto Câmara[1]

[1]Divisão de Processamento de Dados – Instituto Nacional de Pesquisas Espaciais - INPE
São José dos Campos – SP – Brasil

[2]Departamento de Informação, Ambiente, Saúde e Produção Alimentícia - DIASPA – Instituto Federal de Educação, Ciência e Tecnologia do Piauí - IFPI
Teresina – PI - Brasil

{eduilson@ifpi.edu.br, laercio@dpi.inpe.br, gilberto.camara@inpe.br}

Abstract. *This paper describes an algorithm for TIN simplification based on the values of slope and aspect of the triangles. Most simplification algorithms use the vertical distance as their simplification criteria. But the vertical distance does not guarantee the preservation of features such as slope and aspect. Most applications using TIN, such as hydrology applications, apply these data in their computation. We considered that maintaining slope and aspect in TIN simplification, preserve of the terrain morphology, improving results in applications.*

Resumo. *Este artigo descreve um algoritmo para simplificação de TIN baseado nos valores de declividade e orientação de vertente dos triângulos. Muitos dos algoritmos de simplificação utilizam a distância vertical como critério para simplificação. Porém a distância vertical não garante a preservação de características como declividade e orientação de vertente. Algumas aplicações, que utilizam TIN, fazem uso desses dados em suas fórmulas, as aplicações de hidrologia são exemplos disso. Consideramos que a manutenção das características de declividade e orientação de vertente em simplificação de TIN mantém a morfologia do terreno, melhorando os resultados nas aplicações.*

1. Introdução

O crescimento na disponibilidade de dados de terreno tem permitido a geração de modelos de terreno mais complexos e com maior qualidade. Desta forma, os pontos utilizados no modelo afetam diretamente a qualidade do modelo. Modelos criados com um número maior de pontos tornam-se mais complexos e com melhor resolução. Porém, o aumento na quantidade de dados demanda maior custo de processamento e armazenamento.

Para reduzir a quantidade de dados, mantendo a qualidade dos modelos e melhorando a eficiência da aplicação, os algoritmos de simplificação têm sido objeto de estudo nas últimas décadas. Esses algoritmos consistem da seleção de um subconjunto

dos pontos utilizados para gerar modelos mais simples, contudo, que mantenham as características desejadas pela aplicação.

De acordo com a literatura, existe uma extensa pesquisa sobre geração e gerenciamento de modelos de terreno simplificados. Podemos encontrar um estudo com a revisão e comparação dos algoritmos de simplificação em [Garland, 1999; Lindstrom; Pascucci, 2002]. Os modelos de simplificação podem ser divididos em dois grupos: modelos de simplificação para malhas regulares e modelos de simplificação de TIN. Neste trabalho temos interesse somente em simplificação de TIN.

Diferentes abordagens para simplificação de TIN têm surgido nos últimos anos. As principais dividem-se em eliminação de vértices [Schroeder *et al.*, 1992] e eliminação de arestas [Yang *et al.*, 2005]. Em geral, esses métodos utilizam a distância vertical como critério para seleção dos pontos, gerando um modelo com distribuição de pontos espacialmente regular. Porém, importantes feições do terreno não podem ser mantidas com as operações de simplificação do modelo.

Muitas aplicações que fazem uso de modelos de terreno utilizam as características de declividade e orientação de vertente em seus resultados [Rennó, 2005; Sanyal; Lu, 2006]. Este trabalho apresenta um algoritmo para simplificação de TIN utilizando como critério de seleção dos pontos os valores de declividade e orientação de vertente dos triângulos adjacentes ao ponto.

2. Modelos de Simplificação

2.1. Eliminação de vértices

A eliminação de vértice é executada, basicamente, em três passos:

a) Seleção do vértice para remoção.

b) Deleção de todos os triângulos adjacentes ao vértice eliminado.

c) Construção da triangulação cobrindo o espaço deixado pelo passo “b”.

O ponto chave desse algoritmo é o critério de seleção do vértice para remoção. Existem diferentes critérios de seleção e o critério baseado na distância vertical do vértice ao plano em que ele incide é um dos mais utilizados [Schroeder *et al.*, 1992] [De Floriani *et al.*, 1998]. Nesse critério, se o vértice estiver dentro de uma distância específica para o plano médio (Figura 1), o algoritmo remove-o, caso contrário, ele permanece.

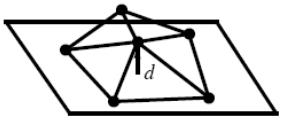


Figura 1 – Distância vertical do vértice ao plano médio
Fonte: adaptado de Schroeder [1992]

2.2. Eliminação de arestas

O processo de eliminação de aresta segue dois passos:

a) Seleção da aresta candidata

b) Validação da eliminação da aresta

Nos algoritmos de eliminação de arestas [Xu *et al.*, 2006; Yang *et al.*, 2005], a entre são os critérios de seleção da aresta e o cálculo de validação da eliminação das arestas. Por exemplo, Garland e Heckbert [1997] utilizam a medida do erro quadrático, Yang et al. [2005] utiliza a distância da aresta para o plano médio, onde a aresta dentro de uma distância mínima para o plano médio será eliminada primeiro. O cálculo do plano médio segue o proposto por Schroeder et al. [1992].

3. Algoritmo Proposto

Diferentes abordagens para simplificação de TIN utilizam a medida de erro associada à distância vertical como critério de seleção para remoção de vértices ou arestas. Como resultado, essas abordagens utilizam avaliação pela distância vertical, a qual avalia somente características geométricas dos modelos de terreno.

Como alternativa propomos um algoritmo que avalia a importância de um ponto baseado nas medidas de declividade e orientação de vertente dos triângulos adjacentes ao vértice. Com isso, buscamos preservar as características morfológicas do terreno. Para tal é criado um *ranking* dos pontos do TIN, onde os vértices com menor valor no *ranking* seriam os primeiros a serem eliminados.

O algoritmo para cálculo do *ranking* é executado como segue:

a) Criar uma triangulação Delaunay para os pontos apresentados.
b) Para cada vértice do TIN fazer:
   i. Buscar todos os triângulos adjacentes.
   ii. Calcular a declividade, área e orientação de vertente dos triângulos adjacentes.
   iii. Calcular a declividade no vértice pela média das declividades dos triângulos adjacentes ponderada pela área dos respectivos triângulos. A área do triangulo é utilizada para que os triângulos maiores, que cobrem uma área maior do terreno, tenham maior importância, já que esse modelo tende a concentrar grande quantidade de pontos nas regiões do terreno com grandes mudanças de curvatura.
   iv. Encontrar a maior diferença entre a declividade do vértice e dos triângulos adjacentes. (DifSlope).
   v. Encontrar a maior diferença entre os ângulos de orientação de vertente dentre os triângulos adjacentes. (DifAspect).
   vi. Calcular o valor do *ranking* do vértice pela soma do DifSlope com o DifAspect.

Após a criação do *ranking*, a criação de TIN com diferentes níveis de detalhes é feita partindo dos pontos com maiores valores no *ranking*. A seguir, apresentamos os testes para avaliar a qualidade do algoritmo.

4. Resultados Experimentais

Para avaliar a qualidade do algoritmo proposto, comparamos os resultados com os obtidos na simplificação de TIN pelo modelo *Multi-Triangulation* (MT). Puppo [1998] faz uma apresentação formal e detalhada do MT, que são implementados codificados na

biblioteca *Multi-Tesselation* [Magillo, 2005]. Os testes com o MT utilizaram o critério de distância vertical para seleção dos pontos a serem eliminados.

Utilizamos um conjunto de dados com 69.222 pontos que representam a topografia da região de San Bernardino (cortesia do *U.S. Geological Survey*). Criamos modelos simplificados com 70% dos pontos originais (com 48.455 pontos), 50% (34.611 pontos) e 30% (20.767 pontos) e comparamos os modelos simplificados com o modelo original. Avaliamos o RMSE (*root-mean-square error*) para as diferenças de altitude, declividade e orientação de vertente. Além disso, utilizamos o modelo de potencial de erosão e deposição [Mitasova *et al.*, 1996] e a grade de fluxos acumulados para modelos hidrológicos.

Na Tabela 1 temos o RMSE para as diferenças de altitude, declividade e orientação de vertente entre os modelos. Podemos notar que o algoritmo proposto apresentou menor erro para quase todos os critérios. No entanto, o erro aumentou na avaliação de diferença de altitude quando o modelo foi reduzido para 20.767 pontos, o que ocorreu também na avaliação de erro de declividade.

Tabela 1. RMSE Altitude, Declividade e Orientação de Vertente

RMSE Altitude			
	48.455 pontos	34.611 pontos	20.767 pontos
MT	8,16	8,02	8,56
Algoritmo Proposto	3,25	6,55	12,30
RMSE Declividade			
	48.455 pontos	34.611 pontos	20.767 pontos
MT	5,67	5,59	5,85
Algoritmo Proposto	2,07	3,94	6,49
RMSE Orientação de Vertente			
	48.455 pontos	34.611 pontos	20.767 pontos
MT	27,85	30,91	36,84
Algoritmo Proposto	11,42	18,49	29,80

Na Tabela 2 temos o RMSE para as aplicações de potencial de erosão/deposição e para a grade de fluxos acumulados. O algoritmo proposto obteve menor erro em todas as avaliações. Verifica-se que mesmo onde o erro de altitude e declividade foram maiores que os apresentados pelo MT, na avaliação dos modelos de erosão e fluxos acumulados, o algoritmo proposto obteve menor erro.

Tabela 2. RMSE Potencial de Erosão/Deposição e Grade de Fluxos Acumulados

RMSE Potencial de Erosão/Deposição			
	48.455 pontos	34.611 pontos	20.767 pontos
MT	0,76	0,74	0,79
Algoritmo Proposto	0,20	0,36	0,52
RMSE Grade de Fluxos Acumulados			
	48.455 pontos	34.611 pontos	20.767 pontos
MT	1558	1549	1637
Algoritmo Proposto	716	1028	1361

O algoritmo proposto calcula o *ranking* dos vértices de forma estática, enquanto o MT avalia a eliminação de pontos dinamicamente, recalculando o erro associado a cada vértice toda vez que um vértice é eliminado. A fim de avaliar como se comportaria o MT utilizando o mesmo critério de eliminação de pontos utilizados por nosso algoritmo, refizemos os testes e comparamos os resultados do MT utilizando distância vertical e o MT utilizando o método de cálculo do *ranking,* e avaliamos os resultados (Tabela 3).

Tabela 3. RMSE Altitude, Declividade, Orientação de Vertente, Potencial de Erosão/Deposição e Grade de Fluxos Acumulados

RMSE Altitude			
	48.455 pontos	34.611 pontos	20.767 pontos
MT – distância vertical	8,16	8,02	8,56
MT - *ranking*	7,87	7,65	8,06
RMSE Declividade			
	48.455 pontos	34.611 pontos	20.767 pontos
MT – distância vertical	5,67	5,54	5,85
MT - *ranking*	5,36	5,10	5,34
RMSE Orientação de Vertente			
	48.455 pontos	34.611 pontos	20.767 pontos
MT – distância vertical	27,85	30,91	36,84
MT - *ranking*	26,66	26,79	30,44
RMSE Potencial de Erosão/Deposição			
	48.455 pontos	34.611 pontos	20.767 pontos
MT – distância vertical	0,75	0,75	0,79
MT - *ranking*	0,74	0,71	0,69
RMSE Grade de Fluxo Acumulados			
	48.455 pontos	34.611 pontos	20.767 pontos
MT – distância vertical	1558	1549	1637
MT - *ranking*	1470	1561	1593

Os resultados na Tabela 3 mostram que o MT, utilizando o método de *ranking*, resultou em simplificações com menores erros.

6. Conclusão

Simplificações que melhor representem as características de declividade e orientação de vertente são mais úteis às aplicações do que as que utilizam a distância vertical, devido à diminuição da quantidade de pontos para representação dos modelos de terreno sem perder a qualidade. Assim, este trabalho apresenta um algoritmo para simplificação de TIN baseado em declividade e orientação de vertente dos triângulos e faz comparação com um algoritmo de simplificação que utiliza a distância vertical como critério.

Nota-se que mesmo onde os erros de altitude (12,3 metros) e declividade (6,49 graus), são maiores que os apresentados pelo MT (8,56 metros e 5,85 graus, respectivamente), na avaliação utilizando os modelos de erosão e fluxos acumulados, o algoritmo proposto obteve um erro menor.

Apesar do nosso algoritmo não ter apresentado melhores resultados em todos os níveis de simplificação em relação ao erro de altitude, que normalmente é o parâmetro utilizado para avaliar modelos de terreno, o algoritmo mostrou-se adequado para simplificar malhas utilizadas pelas aplicações apresentadas.

O algoritmo proposto ainda possui algumas melhorias que poderão ser implementadas, como, por exemplo, avaliar a possibilidade de acrescentar pesos diferentes aos dados de declividade e orientação de vertente no cálculo do *ranking*.

Referências

De Floriani, L.; Magillo, P.; Puppo, E. Efficient implementation of multi-triangulations. In: Visualization '98, 1998, Research Triangle Park, NC, USA. IEEE, p. 43-50.

Garland, M. Multiresolution Modeling: Survey & Future Opportunities. In: EUROGRAPHICS '99 - State of the Art Report (STAR), 1999, Aire-la-Ville (CH). p. 111-131.

Garland, M.; Heckbert, P. S. Surface simplification using quadric error metrics. In: SIGGRAPH '97, 1997 p. 209-216.

Lindstrom, P.; Pascucci, V. Terrain simplification simplified: a general framework for view-dependent out-of-core visualization. **IEEE Transactions on Visualization and Computer Graphics**, v. 8, p. 239-254, 2002.

Magillo, P. **The MT (Multi-Tesselation) Library**. Genova, Italia, 2005. Disponível em: www.disi.unige.it/person/MagilloP/MT/. Acesso em: jun/2009.

Mitasova, H.; Hofierka, J.; Zlocha, M.; Iverson, L. R. Modelling topographic potential for erosion and deposition using GIS. **International Journal of Geographical Information Systems**, v. 10, n. 5, p. 629-641, 1996.

Puppo, E. Variable Resolution Triangulations. **Computational Geometry: Theory and Applications**, v. 11, n. 3-4, p. 219-238, 1998.

Rennó, C. D. Eliminação de áreas planas e extração automática de linhas de drenagem em modelos digitais de elevação representados por grades triangulares. In: XII Simpósio Brasileiro de Sensoriamento Remoto, 2005, Goiânia, Brasil. p. 2543-2550.

Sanyal, J.; Lu, X. X. GIS-based flood hazard mapping at different administrative scales: A case study in Gangetic West Bengal, India. **Singapure Journal of Tropical Geography**, v. 27, n. 2, p. 207-220, 2006.

Schroeder, W. J.; Zarge, J. A.; Lorensen, W. E. Decimation of triangle meshes. **Computer Graphics**, v. 26, n. 2, p. 65-70, 1992.

Xu, k.; Zhou, X.; Lin, X.; Shen, H. T.; Deng, K. A multiresolution terrain model for efficient visualization query processing. **IEEE Transactions on Knowledge and Data Engineering**, v. 18, n. 10, p. 1382-1396, 2006.

Yang, B.; Shi, W.; Li, Q. A Dynamic Method for Generating Multi-Resolution TIN Models. **Photogrammetric Engineering and Remote Sensing**, v. 71, n. 8, p. 917-926, 2005.

Proceedings XII GEOINFO, November 27-29, 2011, Campos do Jordão, Brazil. p 135-140.

CLASS-CHASE: Um Algoritmo para Classificação de Tipos de Padrões de Perseguição em Trajetórias de Objetos Móveis

Fernando de Lucca Siqueira[1], **Vania Bogorny**[1]

[1]Departamento de Informática e Estatística (INE), Universidade Federal de Santa Catarina (UFSC)
Florianópolis, Brasil

fernandols@inf.ufsc.br, vania.bogorny@ufsc.br

Abstract. *Tracking technology like GPS and cell phones generates huge amounts of spatio-temporal data, that are typically large and confuse. These data are called trajectories of moving objects. There are several studies in the literature that discover types of movement patterns like flocks, avoidance and leadership. A pattern that has received little attention is chasing. Chasing can be applied in many domains like human behavior analysis, criminal record and animal hunting. This article proposes the definition of different types of chasing and an algorithm to compute chasing patterns. Experimental results show that the method correctly classifies the type of chasing.*

Resumo. *Tecnologias de rastreamento como GPS e celulares geram grandes quantidades de dados espaço-temporais, que são volumosos e confusos. Estes dados são chamados de trajetórias de objetos móveis. Estudos investigam padrões de movimento como flocks, avoidance e leadership. Um novo tipo de padrão ainda pouco explorado é o padrão de perseguição, que pode ser aplicado em várias áreas como análise de comportamento humano, rastreamento de criminosos e caçada de animais. Este artigo propõe a definição de diferentes tipos de padrões de perseguição e um algoritmo para classificá-los. Experimentos iniciais mostram que o método classifica corretamente o tipo de perseguição.*

1. Introdução

Dispositivos móveis que coletam as trajetórias de seus indivíduos, como GPS e telefone celular, geram enormes quantidades de dados espaço-temporais chamados de trajetórias de objetos móveis. Estes dados, entretanto, são volumosos e confusos, necessitando de métodos e ferramentas inteligentes para extrair informações úteis destes dados. Existem vários domínios de aplicação que fazem uso de dados espaço-temporais como previsão de tempo, tráfego urbano, desastres naturais, mobilidade urbana e jogos eletrônicos.

Vários trabalhos na área de trajetórias tem focado na descoberta de novos padrões de movimento. Laube [Laube et al. 2005] foi um dos pioneiros, definindo quatro tipos de padrões de movimento: Convergência (trajetórias diferentes convergindo para uma mesma região em tempos diferentes), Encontro (trajetórias diferentes localizadas na mesma região ao mesmo tempo), Flock (grupo de trajetórias andando juntas) e Liderança (grupo de trajetórias andando juntas com um objeto liderando). Cao [Cao et al. 2006] apresentou a idéia de sequências de eventos periódicos, episódios frequentes, onde busca trajetórias que permanecem juntas durante um período de tempo definido por uma janela temporal. Lee [Lee et al. 2008] propõe um método para classificar sub-trabjetórias de

acordo com seu objetivo como, por exemplo, barcos que param em áreas de pesca são classificados como barcos pesqueiros e barcos que param em portos de contêiner são classificados como barcos de carga.

Uma área de pesquisa ainda pouco explorada é a análise de comportamento em trajetórias. Ao contrário dos trabalhos citados anteriormente, que focam na geometria dos padrões, a análise de comportamento tenta explicar como o objeto agiu, dando mais semântica ao padrão. Alvares [Alvares et al. 2011] propôs um algoritmo para identificar trajetórias que desviam de certos objetos como, por exemplo, um suspeito desviando câmeras de segurança ou postos policiais. Baglioni [Baglioni et al. 2009] classifica trajetórias com base nos locais que o objeto frequenta como, por exemplo, trajetórias que passam por hotéis e pontos turísticos são trajetórias de turistas. Legendre [Legendre et al. 2006] define o movimento dos objetos a partir de regras de comportamento. Por exemplo, para andar sozinho o objeto deve desviar de obstáculos e de outros objetos.

Siqueira [Siqueira and Bogorny 2011] foi o primeiro trabalho a definir formalmente uma perseguição, assim como o algoritmo TRA-CHASE para identificar padrões de perseguição entre trajetórias, considerando tempo, distância e velocidade. Perseguição entre trajetórias pode ser utilizada em diversos tipos de aplicação como monitoramento de pessoas, análise de crimes, comportamento de animais, jogos de computador, etc. O problema dos padrões de perseguição encontradas em [Siqueira and Bogorny 2011] é que uma perseguição é genérica, podendo identificar padrões falso-positivos como, por exemplo, duas pessoas andando no mesmo caminho sem o conhecimento uma da outra ou carros em uma rodovia. Nestes casos a perseguição é coincidência e acidental, mas o método irá identificar este padrão.

Uma perseguição pode ter características diferentes que variam de acordo com seu objetivo e com o comportamento de ambas as trajetórias durante e após a perseguição. Dessa forma, este artigo propõe a extensão do trabalho de [Siqueira and Bogorny 2011] com a definição de diferentes tipos de padrões de perseguição, assim como um algoritmo para identificar o tipo da perseguição, com o objetivo de aumentar a certeza que o padrão encontrado foi realmente uma perseguição. O restante do artigo está organizado da seguinte forma: a Seção 2 apresenta os tipos de perseguição, a Seção 3 descreve o algoritmo proposto CLASS-CHASE, a Seção 4 apresenta um experimento realizado para validar o método e a Seção 5 apresenta a conclusão do artigo.

2. Tipos de Perseguição

Um padrão de perseguição pode apresentar diversas características, sendo necessária a classificação de diferentes tipos. Este trabalho define cinco tipos de perseguição (espionagem, captura, assalto, caçada e guia), analisando o comportamento das trajetórias durante e após a perseguição.

Para definir o tipo de perseguição são analisadas as *regiões* de parada e baixa velocidade das trajetórias onde ocorreu a perseguição e como elas se comportam durante e após essas *regiões*. Por exemplo, o tipo de perseguição *espionagem*: Um espião ou um detetive persegue seu alvo evitando ser visto. Para isso, ele não deve alcançar a vítima, mantendo sempre uma certa distância durante a perseguição. Portanto, quando o alvo da perseguição parar de se movimentar ou estiver a uma velocidade muito baixa, o espião

deve parar no mesmo tempo, mas em local diferente, evitando assim ser visto, como demonstra a figura 1(a), onde os pontos de parada nos instantes 6 e 15 são disjuntos no espaço e se sobrepõe no tempo.

O padrão de perseguição de *captura* é diferente, conforme pode ser observado na figura 1(b). O perseguidor tenta capturar seu alvo, então sua velocidade deve ser similar ou maior do que a velocidade do alvo, a fim de alcançá-lo. Quando ambas as trajetórias se encontram, há uma abordagem do perseguidor a sua vítima, caracterizada por uma parada, e depois disso elas seguem juntas. Por exemplo, um policial perseguindo um criminoso, prendendo-o, e levando-o para a delegacia.

A maior característica do tipo de perseguição de *assalto*, ilustrada na figura 1(c), é o comportamento das trajetórias após o encontro (assalto). Uma trajetória persegue o alvo por um tempo até alcançá-lo, onde ambas ficam paradas por um determinado período de tempo na mesma região. Após a parada, cada trajetória segue para um caminho diferente. Por exemplo, um assaltante perseguindo uma vítima, realizando o roubo e logo após foge.

A *caçada* é similar a captura. O perseguidor tenta alcançar o alvo movendo-se em sua direção, mas a grande diferença é que uma vez que ambos os objetos se encontram, permanecem na mesma região por um tempo e depois o perseguidor continua sua trajetória, enquanto o alvo permanece imóvel ou tem o fim de sua trajetória, como demonstra a figura 1(d). Este padrão pode caracterizar, por exemplo, a caça de um animal por outro ou um assassinato.

Já o tipo de perseguição de *guia* é mais similar ao de espionagem. Nem toda perseguição tem uma má intenção, podendo então existir casos onde o comportamento é consentido. Um exemplo é alguém servindo de guia. Caso uma pessoa não saiba como chegar em um local, outra pode se habilitar a mostrar o caminho, pedindo para segui-la até o local pretendido. Neste tipo de perseguição ambas as trajetórias tem velocidade similar, e ao final das trajetórias, as mesmas se encontram e param na mesma região ao mesmo tempo.

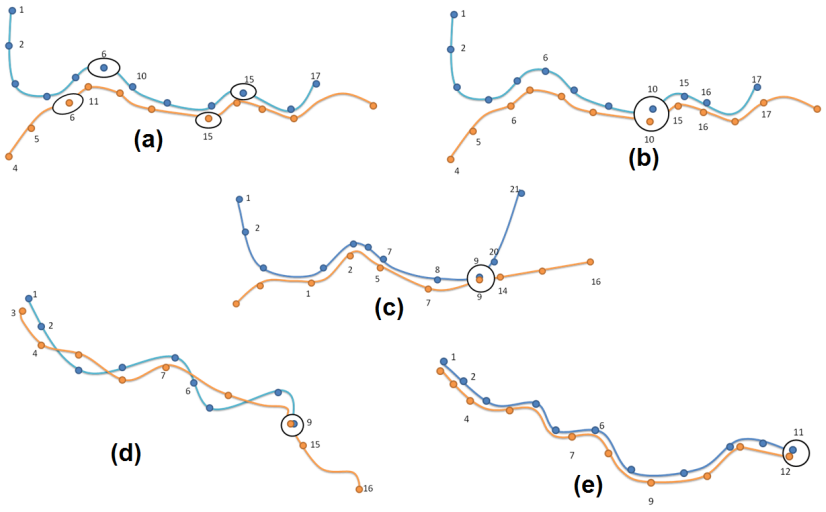


Figure 1. Tipos de perseguição (a) espionagem (b) captura (c) assalto (d) caçada (e) guia.

3. CLASS-CHASE: Algoritmo de classificação de padrões de perseguição

Esta seção apresenta o pseudo-código do algoritmo CLASS-CHASE (Listing 1). O algoritmo leva como entrada um conjunto de trajetórias. A saída do algoritmo é o conjunto de padrões de perseguição classificados. Primeiramente o método verifica se houve perseguição no conjunto de dados de entrada com a função genérica TRA-CHASE (linha 6), que foi definida por [Siqueira and Bogorny 2011].

O segundo passo é classificar os padrões de perseguição, que é a contribuição deste artigo. Para cada perseguição (linha 8) é aplicado o método CB-SMoT [Palma et al. 2008], que utiliza a idéia de stops e moves de [Spaccapietra et al. 2008], onde stops são regiões importantes da trajetória em que o indivíduo permanece por um certo período de tempo. Cada stop gerado pelo método CB-SMoT é uma região da trajetória onde o objeto permaneceu imóvel ou com baixa velocidade. Esse algoritmo foi utilizado para identificar as áreas em que os objetos permanecem imóveis ou se encontram. O algoritmo calcula os stops das trajetórias com perseguição(linhas 11 e 12) e, para cada stop encontrado, é verificado se o tempo dos stops se intercepta (linha 15). Este passo serve para ter certeza de que ambos os stops aconteceram em intervalo de tempo semelhante, evitando assim comparar stops em tempos diferentes. Importante notar que a idéia não é comparar o tempo exato e sim se ocorreram em *intervalos similares*.

Listing 1. CLASS-CHASE pseudo-código

```
1  Entrada:
2          T: conjunto de trajetórias
3  Saída:
4          chase: conjunto de perseguições classificadas
5
6  chase = TRA-CHASE(T);  //método que encontra padrões de perseguição genéricos
7
8  Para cada c ∈ chase
9    t1 = c.alvo;
10   t2 = c.perseguidor;
11   stops1= CBSMoT(t1);  //gera stops da trajetoria 1
12   stops2= CBSMoT(t2);  //gera stops da trajetoria 2
13   Para cada s1 ∈ stops1
14     Para cada s2 ∈ stops2
15       Se (verificaIntervaloTempo(s1,s2))
16         Se (s1.nome = s2.nome)
17            Se(andamJuntas(t1.depoisDeS1, t2.depoisDeS2))
18             P = captura;
19            Senão Se(separam(t1.depoisDeS1, t2.depoisDeS2))
20             P = assalto;
21            Senão Se(terminaDepois(t1,s1) e terminaDepois(t2,s2))
22             P = guia;
23            Senão Se(terminaDepois(t1,s1))
24             P = caça;
25         Senão Se (stopDurantePerseguicao(s1, c) e stopDurantePerseguicao(s2, c))
26          P = espionagem;
27     Fim para cada
28   Fim para cada
29   c.tipoPerseguicao = P;
30 Fim para cada
31
32 Retorne chase;
```

Após encontrar os stops, o algoritmo verifica se ambos os stops tem o mesmo nome (linha 16), pois stops de mesmo nome ocorreram na mesma região. Se após o stop as trajetórias passam a andar juntas (linha 17), o padrão é definido como um padrão de perseguição de captura. Caso esse comportamento não tenha sido observado é feito o teste de padrão de perseguição de assalto, onde é verificado se as trajetórias se separaram após o stop (linha 19). O tipo de perseguição de guia é identificado quando ambas as trajetórias terminam após o stop com a função terminaDepois(t,s) (linha 21), que retorna verdadeiro caso a trajetória t tenha seu final no stop s. Para o tipo caça, apenas a trajetória

alvo $t1$ deve ter seu fim após seu stop (linha 23). Quando o par de stops analisados não tem o mesmo nome, ou seja, não estão no mesmo espaço, o algoritmo verifica se ambos stops ocorreram durante a perseguição (linha 25), caracterizando assim um padrão de espionagem. Por fim, o algoritmo define o tipo de cada padrão de perseguição e retorna o conjunto de padrões de perseguição classificado.

4. Experimento Preliminar

Por limitações de espaço, esta seção apresenta experimentos para validar o tipo de padrão de espionagem, com um conjunto de dados gerado em Florianópolis, sendo pontos coletados a cada 2 segundos. Nesse conjunto, dois indivíduos, sem o conhecimento um do outro, receberam um aparelho de GPS. O primeiro indivíduo foi instruído para se dirigir a um local e aguardasse novas instruções. A cada local ele aguardava uma nova instrução, dizendo o próximo lugar a se dirigir. O segundo indivíduo foi instruído a perseguir o primeiro, mantendo sempre uma distância para evitar ser percebido, simulando uma espionagem.

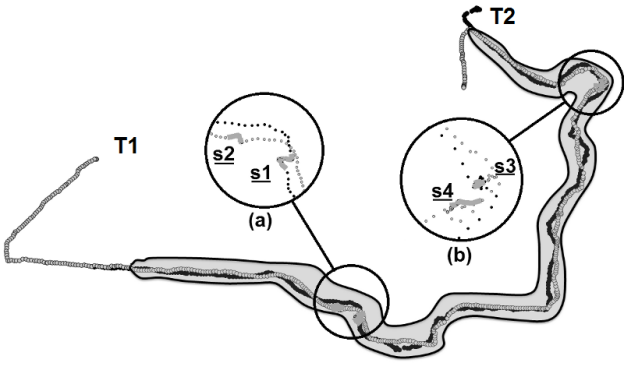


Figure 2. Stops encontrados na perseguição e sua localização nas trajetórias.

O primeiro passo do método CLASS-CHASE gerou o padrão de perseguição ilustrado na figura 2, onde a trajetória mais clara, representada como $T1$, persegue a trajetória alvo mais escura, representada como $T2$. Na região cinza ocorreu o padrão de perseguição.

Table 1. Stops encontrados pelo método CB-SMoT nas trajetórias T1 e T2

ID Stop	ID Trajetória	Nome	Início	Final
s1	T2	0unknown	2011-01-27 19:30:07	2011-01-27 19:34:09
s2	T1	1unknown	2011-01-27 19:30:21	2011-01-27 19:34:23
s3	T2	2unknown	2011-01-27 19:42:01	2011-01-27 19:44:01
s4	T1	3unknown	2011-01-27 19:42:33	2011-01-27 19:44:33

O segundo passo foi classificar o tipo de perseguição. O método CB-SMoT encontrou 4 stops, 2 em cada trajetória, como ilustra a tabela 1. A figura 2 ilustra espacialmente onde ocorreram os stops. Na tabela 1, a trajetória $T2$ teve um stop $s1$ durante o tempo 19:30:07 e 19:34:09. A trajetória $T1$ teve um stop $s2$ durante o tempo 19:30:21 e 19:34:23. Note que os stops de cada trajetória tem nome diferente, pois aconteceram em locais distintos, porém em intervalo de tempo muito similar, caracterizando uma perseguição do tipo espionagem. Essa mesma característica foi observada entre os stops $s3$ e $s4$.

5. Conclusão e Trabalhos Futuros

Este trabalho estende o trabalho de [Siqueira and Bogorny 2011] para classificar diferentes tipos de padrões de perseguição. Foram definidas cinco classes de padrões de perseguição entre trajetórias. Também foi definido um algoritmo, chamado CLASS-CHASE, para identificar o tipo de padrão de perseguição. O algoritmo utiliza o conceito de stops e moves para identificar regiões onde as trajetórias permanecem por um certo tempo. Até o presente momento, o algoritmo foi testado com um conjunto de dados simulando uma perseguição de espionagem. Trabalhos futuros incluem a validação de todos os tipos de perseguição, bem como a análise da complexidade e escalabilidade do algoritmo.

6. Agradecimentos

Os autores agradecem o CNPQ, a Fapesc e a UFSC pelo apoio financeiro a esta pesquisa.

References

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

Baglioni, M., de Macêdo, J. A. F., Renso, C., Trasarti, R., and Wachowicz, M. (2009). Towards semantic interpretation of movement behavior. In Sester, M., Bernard, L., and Paelke, V., editors, *AGILE Conf.*, Lecture Notes in Geoinformation and Cartography, pages 271–288. Springer.

Cao, H., Mamoulis, N., and Cheung, D. W. (2006). Discovery of collocation episodes in spatiotemporal data. In *ICDM*, pages 823–827. IEEE Computer Society.

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

Lee, J.-G., Han, J., Li, X., and Gonzalez, H. (2008). *raClass*: trajectory classification using hierarchical region-based and trajectory-based clustering. *PVLDB*, 1(1):1081–1094.

Legendre, F., Borrel, V., de Amorim, M. D., and Fdida, S. (2006). Modeling mobility with behavioral rules: The case of incident and emergency situations. In Cho, K. and Jacquet, P., editors, *AINTEC*, volume 4311 of *Lecture Notes in Computer Science*, pages 186–205. Springer.

Palma, A. T., Bogorny, V., and Alvares, L. O. (2008). A clustering-based approach for discovering interesting places in trajectories. In *ACMSAC*, pages 863–868, New York, NY, USA. ACM Press.

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

Spaccapietra, S., Parent, C., Damiani, M. L., de Macedo, J. A., Porto, F., and Vangenot, C. (2008). A conceptual view on trajectories. *Data and Knowledge Engineering*, 65(1):126–146.

Identificando Comportamentos Anômalos em Trajetórias de Objetos Móveis

Eduardo M. Carboni and Vania Bogorny

INE – Departamento de Informática e Estatística
Universidade Federal de Santa Catarina(UFSC) – Florianópolis – SC – Brasil
eduardo@inf.ufsc.br, vania@inf.ufsc.br

Abstract. *Mobile Devices such as GPS and mobile phones generate a new data type called trajectory. Several studies are looking for patterns in trajectories, but only a few have focused on the discovery of behavior patterns. This paper proposes a method for analyzing the behavior of moving objects along their trajectories, based on characteristics as abrupt accelerations, abrupt decelerations, and abrupt changes of direction in high speed. The method was evaluated with experiments on real data and presented good results.*

Resumo. *Dispositivos móveis tais como GPS e celulares geram um novo tipo de dado chamado de trajetória. Vários estudos buscam descobrir padrões em trajetórias, mas poucos têm focado na descoberta de padrões de comportamento. Este artigo propõe um método para análise de comportamento do objeto móvel ao longo de sua trajetória, com base em características como acelerações bruscas, desacelerações bruscas e mudanças bruscas de direção em alta velocidade. O método é avaliado através de experimentos com dados reais que apresentaram bons resultados.*

1. Introdução e Motivação

Dispositivos móveis vêm ficando cada vez mais populares devido a sua redução de preço. Com isso, está surgindo um grande volume de dados gerados por estes dispositivos, representando "as pegadas" do objeto ou o caminho que o objeto percorreu. Dados que representam o caminho percorrido pelo objeto são representados na forma de *tid*, *x, y* e *t*, onde *tid* é o identificador do objeto, *x* e *y* representam as coordenadas geográficas e *t* é o tempo no instante da coleta do ponto. O conjunto destes pontos gera um novo tipo de dado espaço-temporal chamado de *trajetória*.

Nos dias atuais, existem muitos estudos que exploram dados de trajetórias. Alguns trabalhos procuram por regiões importantes ao longo das trajetórias Alvares [Alvares et al. 2007], Palma [Palma et al. 2008], Rocha [Rocha et al. 2010]. Outros trabalhos tentam classificar as trajetórias em tipos específicos, como trajetórias de turistas, trajetórias de viajantes Baglioni [Baglioni et al. 2009], trajetórias que desviam de determinados locais ou objetos Alvares [Alvares et al. 2011], perseguição entre trajetórias Siqueira [Siqueira et al. 2011], etc.

Pode-se destacar o estudo realizado por Alvares [Alvares et al. 2007], que encontra regiões importantes em trajetórias utilizando a intersecção do trajeto com regiões geográficas importantes como hotéis, restaurantes, etc. Existe também o estudo

realizado por Palma [Palma et al. 2008], que retorna como regiões importantes das trajetórias aquelas com velocidade baixa, encontrando regiões com engarrafamentos, por exemplo. Ainda, é importante destacar o modelo proposto por Rocha [Rocha et al. 2010], que encontra regiões importantes com base na mudança de direção de um objeto como, por exemplo, a mudança de direção que barcos pesqueiros realizam durante a atividade de pesca.

No que se refere à análise do comportamento do objeto através de sua trajetória, ainda existem poucos estudos. Em Baglioni [Baglioni et al. 2009], é analisado o comportamento do objeto através da análise da origem, destino e locais percorridos pelas trajetórias. Com base nisso, as trajetórias são classificadas em trajetórias turísticas, de trabalho, entre outras. Em Alvares [Alvares et al. 2011], é verificado o comportamento do objeto e definido um algoritmo para verificar se a trajetória está desviando de algum objeto específico como, por exemplo, câmeras de segurança. Em Siqueira [Siqueira et al. 2011], são comparados os comportamentos entre trajetórias, com o objetivo de descobrir se uma trajetória está perseguindo outra trajetória.

Dentre os estudos destacados, nenhum deles classifica os objetos móveis de acordo com o seu comportamento ao longo da trajetória *do mesmo objeto*. Assim, este artigo apresenta um método para analisar as trajetórias de objetos móveis, encontrar regiões de comportamento anômalo e classificar os objetos com comportamentos normais e anormais, analisando acelerações bruscas, desacelerações bruscas e mudanças bruscas de direção. Com essa abordagem é possível monitorar veículos de transporte com o objetivo de descobrir se o condutor possui um comportamento anômalo, pois no caso de uma empresa de transporte coletivo, por exemplo, se um veículo percorre curvas em alta velocidade, desacelera e acelera bruscamente com freqüência, pode gerar desconforto aos passageiros. Outro ponto de destaque é no transporte de alimentos. Segundo informações da ANVISA, cerca de 30% de frutas e verduras transportadas são perdidas por esmagamentos durante o transporte. No caso de transporte de produtos perigosos, este trabalho é interessante para traçar o perfil do condutor através da análise de trajetórias passadas, buscando evitar acidentes futuros.

O restante do artigo está organizado da seguinte forma: a seção 2 apresenta o método proposto. A seção 3 apresenta parte dos experimentos realizados para validar o trabalho. A seção 4 descreve a conclusão e os objetivos futuros.

2. O Método Proposto

O método proposto percorre os pontos de uma trajetória em busca de regiões que apresentam comportamentos de aceleração brusca, desaceleração brusca ou mudança brusca de direção. Uma aceleração é tida como uma sequência de pontos onde a velocidade do objeto aumenta. Já uma *aceleração brusca* é definida como uma sequência de pontos com velocidade crescente e a velocidade aumenta mais que um determinado percentual. Porém, quanto maior for a velocidade, menor será seu percentual de aceleração. Por exemplo, um veículo com velocidade de 1 Km/h passando para 2 Km/h em um segundo tem um aumento percentual de 100% em sua velocidade, mas esta não é considerada uma aceleração brusca, enquanto um veículo à 100 Km/h passar para 120 Km/h teria um aumento de 20% da velocidade, podendo ser considerado um aumento brusco.

Para resolver a situação do percentual de aumento da aceleração, foi criada uma tabela com graus de importância (pesos) para faixas de velocidade, a fim de utilizar estes pesos para identificar movimentos bruscos. Estes pesos foram gerados a partir de experimentos realizados com veículos de passeio e estão ilustrados na tabela 1.

Tabela 1 - Faixas de Velocidade e Pesos.

Velocidade: Km/h	Peso (grau de importância):
> 0 e < 1	20
>= 1 e < 3	10
>= 3 e < 10	4
>= 10 e < 30	3
>= 30 e <70	2
>= 70	1

A velocidade de um ponto, multiplicada pelo peso correspondente e multiplicada pelo percentual de aceleração, definirá o aumento mínimo da velocidade do ponto seguinte para que seja caracterizada uma aceleração brusca. Por exemplo, quando um objeto móvel estiver a uma velocidade de 2 Km/h (terá peso 10 pela tabela), um aumento de 20% na aceleração corresponderia a um aumento mínimo da velocidade para 6 Km/h (velocidade atual 2 + (2*10*20%)) para caracterizar uma aceleração brusca.

A região de uma trajetória que apresenta uma seqüência de pontos com velocidade decrescente é um local que apresenta desaceleração. Se esta região apresentar uma desaceleração maior que determinado percentual, ela é definida como um local com *desaceleração brusca*. O mesmo problema encontrado com as acelerações acontece com as desacelerações, pois uma desaceleração terá um percentual de perda de velocidade maior quando estiver a uma velocidade menor. A tabela com graus de importância é a tabela 1 demonstrada anteriormente, e os pesos também são multiplicados ao percentual de desaceleração.

Ainda existe a situação onde um objeto móvel muda bruscamente a sua direção. Isto ocorre quando o objeto percorre uma região mudando a direção à uma velocidade considerada alta para a diferença das direções durante o trajeto. Também foi criada uma tabela com faixas de mudanças de direção e um limite de velocidade aceitável, mas por limitações de espaço esta tabela não será detalhada.

Uma vez definidos os pesos para os movimentos bruscos é possível definir um algoritmo para extrair estas informações das trajetórias. O algoritmo (resumido na figura 1) recebe como entrada as trajetórias, a tabela de pesos, e os percentuais do aumento da aceleração, desaceleração e mudança de direção (linhas 1, 2 e 3). Em seguida, para cada trajetória do conjunto (linha 4) o algoritmo identifica as regiões com comportamentos anormais (linha5). O resultado é armazenado criando um buffer ao redor de todas as regiões anômalas (linha 6), gerando polígonos, a fim de facilitar a visualização do resultado e verificar se existe sobreposição de regiões com comportamento anômalo entre diferentes trajetórias.

Uma vez gerados os polígonos, estes são comparados com regiões de comportamento anômalo intra trajetórias e entre trajetórias de diferentes objetos (linha 7 e 8), para descobrir trajetórias que apresentam regiões de comportamento anômalo no

seu trajeto (intra trajetória), e verificar também se essas regiões acontecem em trajetórias de outros objetos (entre trajetórias). O fato de um objeto móvel ter regiões (subtrajetórias) com comportamento anômalo na sua trajetória pode não representar mau comportamento, mas podendo ser forçado a isto devido algum evento externo como um buraco na rodovia, a interdição da rodovia para fins de obras, um animal atravessando a pista, etc.

Buscando diferenciar o comportamento anômalo forçado por eventos externos de comportamento anômalo causado pela negligência do objeto móvel, é verificado se mais de uma trajetória tem o mesmo comportamento anômalo no mesmo local (linha 7 e 8 do algoritmo). Quando várias trajetórias têm comportamento anômalo na mesma região, é provável que o problema seja uma causa externa (natural). Neste caso o comportamento não é classificado como anômalo.

Entrada
1 - *T* // Conjunto de Trajetórias
2 - *C* // Tabela de Pesos
3 - Pace, Pdes, Pdir // Percentuais aceitáveis de aceleração (Pace), desaceleração (Pdes) e de mudança de direção (Pdir)
Método
4 - para cada trajetória *t* em *T*
5 - para cada 3 pontos, identifica regiões com aceleração, desaceleração e direção anômala e adiciona em *a*
6 - para todos *a*, cria buffers em suas áreas
7 - para todas regiões *a*
8 - para cada *a* de uma trajetória t_x que intercepte *a* de uma trajetória t_{x+n}, adiciona em *A*
Saída
9 - *a* // Conjunto de regiões com comportamento anômalo de todas as trajetórias
10 - *A* // Conjunto de regiões com comportamento anômalo em locais que se repetem em trajetórias distintas

Figura 1 - Pseudocódigo do algoritmo

3. Experimentos Parciais

Para a validação do método foram realizados experimentos com diferentes conjuntos de dados. Neste artigo, por limitações de espaço, serão mostrados resultados com um conjunto de dados coletado por um GPS/Celular em intervalos de 1 em 1 segundo, no estado de Santa Catarina.

Para evitar a geração de regiões de comportamento anômalo em locais com falhas na coleta dos pontos, decidiu-se analisar os pontos pelo menos de 3 em 3, ao invés de analisar o comportamento a cada 2 pontos, já que os dados analisados foram gerados de segundo em segundo. Contudo, estudos estão sendo realizados considerando mais pontos. Foram realizados experimentos distintos em relação aos percentuais de aceleração, desaceleração e mudança de direção aceitável.

Dentre as trajetórias analisadas, foram selecionadas algumas em que os locais onde ocorreram comportamentos anormais eram conhecidos (semáforos, algumas curvas e origens e destinos das trajetórias).

A figura 2 (lado esquerdo) ilustra uma trajetória real na cidade de Florianópolis, onde foram destacadas as regiões com comportamento anômalo previamente conhecido. Existem três regiões com desaceleração brusca e uma com mudança brusca de direção. A mudança brusca de direção é representada por um polígono cinza escuro,

representando uma curva realizada em alta velocidade. As regiões com desacelerações bruscas estão demonstradas pelos polígonos em branco, que representam o momento em que o objeto vai entrar na rodovia de alto fluxo de veículos, necessitando diminuir a velocidade para esperar os outros veículos passarem e não haver colisão, um semáforo e o destino da trajetória.

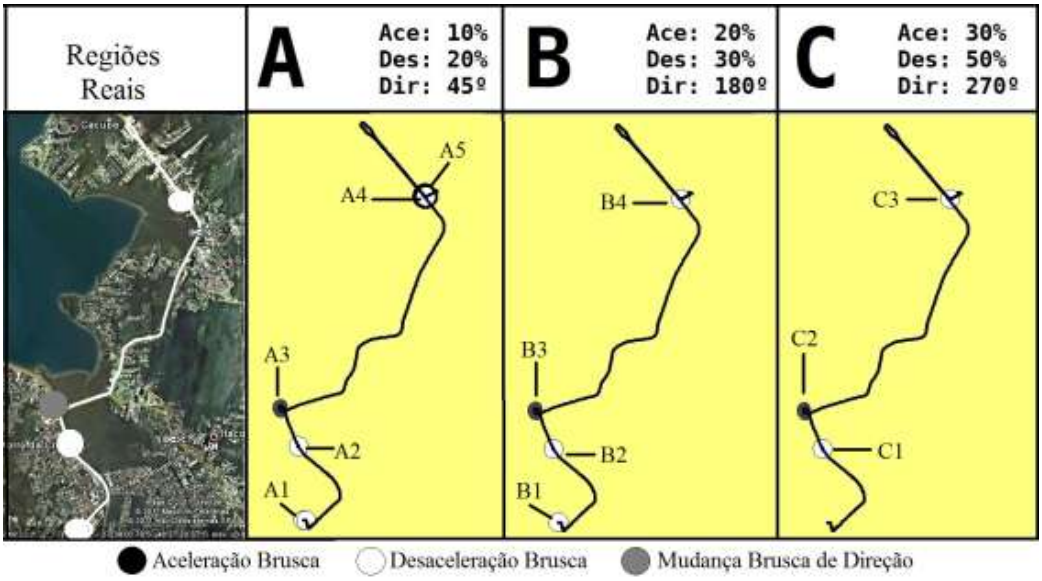


Figura 2 - Trajetória com regiões de comportamento anormal

Tendo o conhecimento das regiões de comportamento anômalo ao longo da trajetória, foram realizados experimentos com percentuais diferentes para aceleração brusca, desaceleração brusca e mudança brusca de direção.

Os melhores valores encontrados foram de 20% para o primeiro comportamento, 30% para o segundo e 180% para o terceiro, conforme pode ser visto na figura 2 (*B*). Comparado os experimentos *B* e *C,* tem-se que o último não encontrou o destino da trajetória como uma desaceleração brusca, ou seja, a região B1. Já se comparados os experimentos *A* e *B*, tem-se que o primeiro retornou uma região de aceleração brusca que não existiu (*A5*), localizada bem próxima a região de desaceleração brusca *A4*.

Nas comparações entre trajetórias que passaram no mesmo local, com o objetivo de descobrir se esses locais de comportamento anômalo se repetem em várias trajetórias, o resultado foi que as regiões que apresentaram comportamento anômalo em uma trajetória, também o apresentaram em muitas outras. Assim, é possível inferir que este não é um comportamento particular da trajetória, mas sim de várias trajetórias, sendo um problema provavelmente relacionado à rodovia, e não particular do objeto móvel.

4. Conclusão e Trabalhos Futuros

Este trabalho propôs classificar o comportamento de objetos móveis em normal ou anormal, com base em acelerações bruscas, desacelerações bruscas e mudanças bruscas de direção ao longo das trajetórias. Foram definidas tabelas de pesos para análise de movimentos bruscos e foi criado um algoritmo que retorna as regiões (subtrajetórias) de

acelerações bruscas, desacelerações bruscas e mudanças bruscas de direção. Foram realizados experimentos com dados reais com o objetivo de definir o comportamento dos objetos e validar este trabalho. O algoritmo encontrou as regiões onde houve comportamento anômalo. Também foi verificado se os comportamentos anormais se repetiam em mais de uma trajetória na mesma região. Com isso, foi possível verificar se o objeto móvel tinha um comportamento anormal em virtude da maneira como dirige ou se a razão está relacionada a fatores externos como as condições da rodovia. Em relação à trabalhos futuros, pretende-se definir novas medidas que garantam com mais certeza se um indivíduo tem ou não comportamento anormal durante sua trajetória.

Agradecimentos

Esta pesquisa foi parcialmente financiada pela FAPESC, CNPQ e Universidade Federal de Santa Catarina.

Referências

Alvares, L. O., Bogorny, V., Kuijpers B., de Macêdo, J. A. F., Moelans, B., Vaisman, A. A. (2007). A model for enriching trajectories with semantic geographical information *15th International Symposium on Advances in Geographic Information Systems* (ACM-GIS), p.162-169.

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

Baglioni, M., de Macêdo, J. A. F., Renso, C., Trasarti, R., Wachowicz, M. (2009) Towards Semantic Interpretation of Movement Behavior. In Sester, M., Bernard, L., and Paelke, V., editors, *AGILE Conf.*, Lecture Notes in Geoinformation and Cartography, pages 271-288. Springer.

Palma, A. T., Bogorny, V., Kuijpers, B., Alvares, L. O. (2008). A Clustering-based approach for discovering interesting places in trajectories *23rd Annual Symposium on Applied Computing*, (ACM-SAC), p.863-868.

Rocha, J. A. M. R., Times, V. C., Oliveira, G., Alvares, L. O., Bogorny, V., (2010). DB-SMoT: A Direction-based spatio-temporal clustering method, *Fifth IEEE International Conference on Intelligent Systems* (IEEE IS), p.114-119.

Siqueira, F. L., Bogorny, V. (2011). Discovering Chasing Behavior Patterns in Moving Object Trajectories", *T. GIS*, 15(5): 667-688.

Postgis Raster Plugin for Quantum GIS

Maurício Carvalho Mathias de Paulo[1,2], **Lubia Vinhas** [2]

[1]Diretoria de Serviço Geográfico – DSG
Quartel General do Exército, Bloco "F", 2o Piso, Ala Norte
CEP:70630-901 – SMU – Brasília – DF, Brasil

[2]Instituto Nacional de Pesquisas Espaciais – INPE
Caixa Postal 515 – 12245-970 – São José dos Campos - SP, Brasil

{mauricio, lubia}@dpi.inpe.br

Abstract. *This article describes a graphic tool implementation to upload and visualize raster data stored in a PostgreSQL database with PostGIS extension. The implementation was done as a Quantum GIS plugin called WktRaster, released as an open source software. The plugin is one of the earliest efforts to explore raster data storage, visualization and processing using the PostGIS Raster extension in a Geographic Information Systems' environment. The main goal is to assist users in exploring the database design flexibility that PostGIS raster type introduces.*

1. Introduction

In the GIS (Geographic Information Systems) community there has been a long debate about how to represent spatial information. Two different approaches coexist, the discrete (object) representation in which spatial data is represented by vector data structures and the continuous (field) representation in which spatial data is represented by raster data structures. Cadastral data (such as a map of census sectors of a city) is an example of vector data and satellite imagery is an example of raster data. In their daily work, GIS users deal with both vector and raster data.

DBMS (Database Management Systems) have evolved to be capable of managing spatial data in a user friendly and efficient way providing spatial abstract data types, indexing mechanisms and functions tailored for spatial data, generically called spatial extensions. One of the most known open source object-relational DBMS is PostgreSQL, that has a spatial extension called PostGIS [PostGIS 2011] . PostGIS' spatial types represent vector data in conformance with OGC (Open Geospatial Consortium) and ISO (International Standards Organization) standards, allowing the storage and manipulation of spatial data using the SQL (Structured Query Language). Even though this is a reality for vector data, there is still a lack of standards for raster data processing and storage in spatial extensions.

Initially PostGIS only had vector data types. A first attempt to support raster data in PostGIS (called PGRaster) was discontinued and in 2010 the Postgis Raster project replaced it's predecessor [PostGIS 2011]. Although PostGIS Raster is an ongoing project, PostGIS version 2.0 is going to include raster support [Obe and Hsu 2011]. This article discusses PostGIS Raster and describes a tool developed to upload raster data to PostGIS as well as to retrieve and visualize the data stored in database, that was released as a plugin for Quantum GIS, a free and open source GIS.

This article starts with an introduction on geographic raster database storage. Section 3 describes the concepts applied in PostGIS Raster. Section 4 describes the Quantum GIS plugin implemented to manage PostGIS Raster layers. Section 5 outlines an example usage of SQL queries to perform processing and visualization of raster data stored using PostGIS' functions.

2. Related work

The main demands that drive implementations are visualization and processing efficiency, interoperability and adequacy to client-server capabilities. The approaches for storing and processing raster data might vary but the partitioning of raster in tiles and maintenance of resolution pyramids are found in most implementations of raster storage.

The partitioning of a raster in tiles, usually of fixed size (for example 128 x 128 cells), allows the indexing of each raster part independently, resulting in efficiency gain when just a raster partition should be processed or visualized. Resolution pyramids are multi-level auxiliary structures that store downsampled (and also tiled) versions of the original raster data. The bottom level contains the original resolution, while higher levels contain the subsequent lower resolution versions. This feature is especially useful for visualization purposes, when applications can retrieve the raster at a level according to a desired zoom level [Vinhas et al. 2003].

One approach to store raster data is using files in a file system. This is the principle of many softwares compliant with the Web Map Tile Service (WMTS) such as *gdal2tiles* [Masó et al. 2010]. In this case, the raster data server is the file system itself. The main advantage is that the data is served the same way that it is stored. One disadvantage is lack of integration with processing functions so the server works only as a repository for raster data.

Some implementations focuses on using the DBMS as a repository for the metadata about the raster, and the data itself is stored as a long binary with no semantics. This approach is similar to the file system's, with the advantage that it maintains vector and raster data stored in the same DBMS. There might be some efficiency loss due to the interaction with the DBMS for data retrieval before being decoded and handled by client applications.

Performance improvement can be achieved by specialized raster DBMS such as PARADISE and RASDAMAN, however, as non-standard servers, they increase the management needs of GIS applications working on top of them [Imran 2009]. An intermediate approach is to construct tiling and multi-pyramids using BLOB (Binary Large Object) type, available on most object-relational DBMS, and adequate indexing and compression mechanisms for efficient data retrieval. The TerraLib library follows this approach providing inside the library data upload and retrieval methods using SQL [Vinhas et al. 2003].

The GeoRaster feature of Oracle Spatial (the spatial extension of Oracle DBMS) allows user to store, index, query, analyze, and deliver raster data and associated metadata. Similarly there is the RasterLite extension for SQLite. Although with different implementations both aimed at the same goal: provide a raster data type that can be handled similarly to the vector data types. The concept implemented by Oracle's GeoRaster doesn't follow the expected behavior. Instead, an auxiliary table using the SDO_Raster

type is created in order to store the gridded data that was assigned to a row in a GeoRaster table. This means that the object and the data are not stored together as expected [Oracle Corporation 2011].

PostGIS Raster follows the same concept of similarity to vector data types behavior, but the raster type objects are complete self-describing values in a table's column. The project also offers a set of SQL functions (like ST_Intersects) to operate seamlessly on vector and raster data. As an example, consider the computation of an elevation profile following a vector feature (for example a road) from a grid (or raster) dataset with elevation values.

3. PostGIS Raster

3.1. Storage

PostGIS Raster aims to implement a minimalistic yet complete raster data structure in the database. There is a single raster type that can be used to define a database entity's attribute. In other words, PostGIS Raster allows the definition of a table where one (or more) columns is of RASTER type. No restriction is implicitly imposed on the raster instances in the column. This means that each row may contain raster tiles that may have different sizes, overlap or snap to different grids [Refractions Research 2011]. Another PostGIS Raster's feature is the option to maintain the raster data itself in it's original data file outside the DBMS, while a representation is stored in the database for organization and storage purposes.

This implementation is flexible enough to allow distinct uses of the Raster type to support the concept of coverages. For example, one row in a table can represent a complete image, a tile of an image or one image that is part of a large mosaic of images. These possibilities fit most users' needs.

3.2. Loading data

Once the Raster type is available in the spatial extension, it is necessary to have tools to insert data into the DBMS. PostGIS 2.0 provides a Python script ("raster2pgsql.py") to upload raster data to a database with PostGIS Raster installed. The script depends only on GDAL [Warmerdam 2008] but has no graphic interface. Sometimes it's difficult for the user to call the script with the right combination of parameters in order to make the best use of PostGIS Raster storage flexibility.

GDAL's command line utilities are able to read raster data stored in database and save it back as a raster file, although, in order to facilitate raster data loading and retrieval from PostGIS database, a graphic tool integrated to a GIS was developed and is described in the next section.

4. The Quantun GIS Plugin for PostGIS Raster

Quantum GIS (or simply QGIS) is an open source GIS able to visualize, analyze and edit vector and raster data. Since it's earliest versions it has been able to handle vector data stored in PostGIS with no raster support. In order to read and write raster data Quantum GIS uses GDAL that since version 1.6 includes a PostGIS Raster driver. Only since version 1.8 the driver became stable and capable of reading irregularly tiled layers

[Refractions Research 2011]. Since GDAL is currently the only library able to read PostGIS's Raster layers and is actively supported by the PostGIS Raster team, Quantum GIS' support for PostGIS Raster was a natural transition.

QGIS Plugin API (Application Programming Interface) provides support for users to add customized tools for specific functionalities The recommended environment for creating plugins is using Python programming language and Qt interface toolkit. Once a plugin is developed and tested it can be published in QGIS' plugin repository[1] maintained by it's development team.

4.1. Loading

QGIS PostGIS Raster Plugin's first goal is to provide a graphical user interface to facilitate the uploading of single and/or multiple raster data to PostGIS Raster, as can be seen in Figure 1.

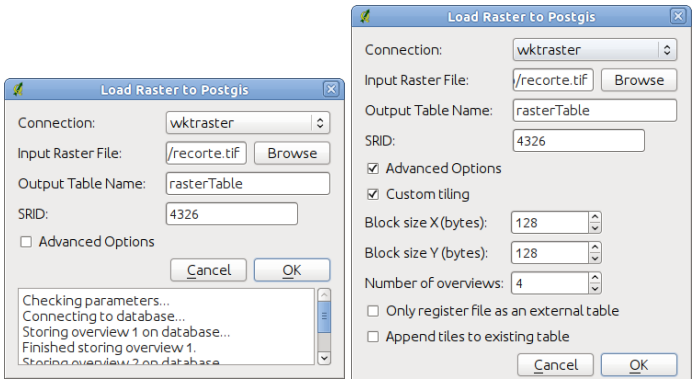


Figure 1. QGIS Plugin interface to upload raster to PostGIS

Connection information is read from QGIS' settings. By default the interface exposes only the minimum parameters needed to upload a single raster file to one table. The interface allows the user to select the number of levels, in the multi-resolution pyramid, to be built. The command line script did not have this functionality, so if a user wanted to create four levels the script would need to be called four times.

Another important parameter that the plugin deals with is the Spatial Reference Identifier (SRID) in which the raster is georeferenced. This is a unique integer identifier, given by a specific authority that should be registered in the database. OGC recognizes the EPSG (European Petroleum Survey Group) as the standard authority to provide SRIDs and some raster formats (for example GeoTiff) include this information. The plugin is able to extract this information and if it is available, avoids the task of finding it manually.

The "Advanced Options" allow users to select the table configuration that should be used, permitting the construction of mosaics and coverages in one or multiple tables. For example, in order to build a mosaic in a single table the user should append many files to one existing table. Since the tiles are self-descriptive the GDAL driver is able to read the mosaic as a single raster coverage.

[1] http://pyqgis.org/

4.2. Visualization

Once a PostGIS database is populated with raster data, users want to explore and visualize them in QGIS. The PostGIS Raster plugin also supports this as shown in Figure 2. Initially the plugin lists all tables that contains raster columns.

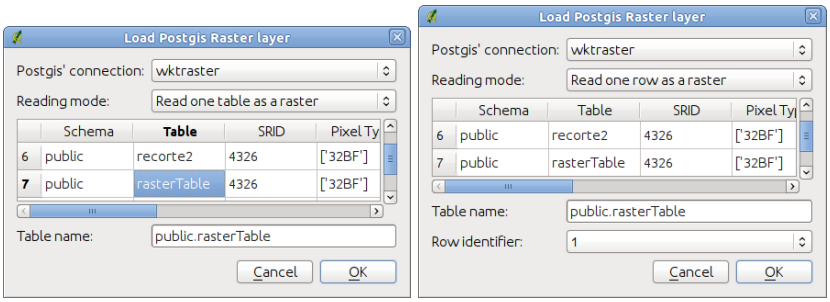


Figure 2. Interface to add a PostGIS raster layer to a Quantum GIS project

The "Reading mode" options allows the user to select how to interpret raster table's arrangement in the database. Since one table can store one raster grid (tiled or untiled) or a mosaic of raster data, this option gives interpretation flexibility.

Choosing the option "Read one row as a raster" the program reads raster data from a single row. It can be a whole image (in a mosaic) or a single tile from a tiled raster table. Whenever the user wants to see the data stored in a table as whole, the option "Read one table as a raster" can be chosen. In this option if each row represents one image, the mosaic is going to be shown. This option also allows vizualization of whole images when a table represents a tiled image. Since some tables can store huge mosaics, the plugin offers the option to "Read the table's vector representation". This allows the user to know what regions each row covers before visualizing the desired area.

4.3. Raster metadata reading

In order to identify the available raster tables, the implementation uses PostgreSQL's internal tables together with the PostGIS Raster's metadata table. This approach allows the plugin to list tables that contain raster columns even if there is no row in the metadata table describing it. This feature makes it possible to explore every storage option described in Section 4.2.

Since PostGIS Raster type is defined as a new PostgreSQL type, it's definition is stored in the "pg_type" table. The query in Listing 1 makes use of this concept by searching every table's attribute for the raster type identifier. This is accomplished by joining the "pg_class" and the "pg_attribute" tables that respectively store tables and attributes lists.

Listing 1. Query to list every existing raster table

```
SELECT relname, attname FROM pg_class AS cla JOIN pg_attribute AS att
    ON att.attrelid = cla.oid AND att.atttypid = 'raster'::regtype ;
```

5. Raster analysis using SQL

Using the WktRaster plugin together with other plugins available it is possible to process raster and vector data on databases with PostGIS enabled. A valid procedure for processing data is to use the "CREATE TABLE AS" statement. There must be an unique integer

column named "rid" to load the table as a Quantum GIS' raster layer and a column named "rast" that contains the raster tiles. Using this syntax the user can store the results of a query as a new raster table and open it for visualization using the WktRaster plugin on Quantum GIS.

Listing 2 shows an example SQL query that applies a threshold of height's above 300m to a raster elevation table. The result of this query is a table containing two columns. One column contains the raster data and the other column contains the unique integer.

Listing 2. Example raster processing using the WktRaster plugin

```
CREATE TABLE testeraster AS SELECT rid, st_mapalgebra(rast, 'CASE WHEN
    rast BETWEEN 0 and 500 THEN 0 WHEN rast BETWEEN 500 and 1000 THEN
    rast ELSE 0 END') as rast from recorte2;
```

6. Conclusion

The plugin developed allows Quantum GIS' users to upload and visualize raster data stored in PostgreSQL databases with PostGIS enabled. This is one of the first implementations that explores the georeferenced raster storage options of PostgreSQL in a GIS environment. This opens path for raster and vector processing using the SQL language.

The plugin explores many of the possible storage options available through the raster type. This flexibility allows easy image mosaicking and improves raster database design options.

The raster visualization speed can be tunned using tiling and pyramids. This can be a workaround for the current GDAL's driver speed as it is in early development and lacks many features that are still being implemented.

References

Imran, M. (2009). Extending an open source spatial database with geospatial image support: An image mining perspective.

Masó, J., Pomakis, K., and Julià, N. (2010). Opengis web map tile service implementation standard.

Obe, R. O. and Hsu, L. S. (2011). *PostGIS in Action*. Manning Publications Co.

Oracle Corporation (2011). Georaster overview and concepts. `http://download.oracle.com/docs/html/B10827_01/geor_intro.htm`, [accessed on Aug 9].

PostGIS (2011). Postgis. `http://www.postgis.org`, [accessed on Aug 9].

Refractions Research (2011). Chapter 8, raster reference. `http://postgis.refractions.net/documentation/manual-svn/RT_reference.html`, [accessed on Aug 9].

Vinhas, L., de Souza, R., and Câmara, G. (2003). Image data handling in spatial databases. In *V Brazilian Symposium on Geoinformatics, Campos do Jordão, Brazil*. Citeseer.

Warmerdam, F. (2008). The geospatial data abstraction library. *Open Source Approaches in Spatial Data Handling*, pages 87–104.

Index of authors