GEOINFO 2007

IX Simpósio Brasileiro de Geoinformática

IX Brazilian Symposium on GeoInformatics

Campos do Jordão - SP - 25 a 28 de novembro de 2007

http://www.geoinfo.info

Campos do Jordão, SP.

(22° 44' 22"S ; 45 ° 35' 29"W)

PROGRAMA FINAL

FINAL PROGRAM

Orotour Garden Hotel

Rua Eng. Gustavo Kaiser, 165 – Jaguaribe – 12460-000 – Campos do Jordão, SP

Tel (12) 3662-2833 - http://www.orotour.com.br

Dados Internacionais de Catalogação na Publicação

SI57A Simpósio Brasileiro de Geoinformática (9. , 2007: Campos do Jordão, SP).
Anais do IX Simpósio Brasileiro de Geoinformática, São José dos Campos-Brasil, 25 a 28 nov. 2007.
CD-ROM
ISBN 978-85-17-00036-2

1.Geoinformática. I.Vinhas, Lúbia. II.Costa, Antonio Carlos da Rocha. III.Instituto Nacional de Pesquisas Espaciais.

CDU 681.3.06

Preface

Dear reader,

You have in your hands the Proceedings of GEOINFO 2007, the IX Brazilian Symposium on GeoInformatics.

We would like to thank all Program Committee members and additional reviewers, and to assure you that they made all the efforts to select the best works submitted to the event. However, many other good papers had to be left out, because, obviously, there is a limitation in the conference program size. Together, all those works express the high quality of the research efforts being made in the area, in Brazil.

A total of 82 articles were submitted: 43 full papers and 39 short papers. A total of 28 papers were accepted, 17 full and 11 short papers, thus defining an acceptance rate of 34.1. Among the authors of the accepted papers 17 distinct Brazilian academic institutions and research centers were represented.

We would also like to thank INPE (Brazil's National Institute for Space Research) and SBC (Brazilian Computer Society) for their support to all events of the GeoInfo series. We also thank our sponsors, promoters and supporters, identified in this proceedings volume.

We specially thank Terezinha, Hilcéa, Thanisse, Daniela and Janete, from the organizing committee, for their companionship in this journey.

Enjoy the symposium, and the reading of the papers!

Antônio Carlos da Rocha Costa
GeoInfo 2007 Program Chairs

Cirano Iochpe
GeoInfo 2007 Program Chairs

Lúbia Vinhas
General Chair

GEOINFO 2007
IX Simpósio Brasileiro de Geoinformática
IX Brazilian Symposium on GeoInformatics
Campos do Jordão - SP - 25 a 28 de novembro de 2007

Organização / *Organized by*

INPE

ppginf·ucpel
programa de pós-graduação em informática - esin/ucpel

Informática UFRGS

Patrocínio / *Sponsored by*

INPE

K2

Promoção / *Promoted by*

SBC
Sociedade Brasileira de Computação

Apoio / *Supported by*

SELPER

Comissão Organizadora / Organizing Committee

- Lúbia Vinhas, INPE - Coordenadora Geral
- Antônio Carlos da Rocha Costa, UCPel - Coordenador de Programa
- Cirano Iochpe, UFRGS - Coordenador de Programa
- Gilberto Câmara, INPE
- Daniela Seki, INPE
- Hilcéa Santos Ferreira, INPE
- Janete da Cunha, INPE
- Terezinha Gomes dos Santos, INPE
- Thanisse Silva Braga, INPE

Comissão de Programa / Program Committee

- Antônio Carlos da Rocha Costa, UCPel, Brasil (Chair)
- Cirano Iochpe, UFRGS, Brasil (Co-chair)
- Ana Carolina Salgado, UFPE, Brasil
- Andrea Rodriguez, Universidad de Concepción, Chile
- Andrew Frank, Technical University of Vienna, Austria
- Antônio Machado, GISPLAN, Brasil
- Antônio Miguel Vieira Monteiro, INPE, Brasil
- Armanda Rodrigues, UNL, Portugal
- Camilo Daleles Rennó, INPE, Brasil
- Christelle Vangenot, Swiss Federal Institute of Technology at Lausanne, Switzerland
- Christopher Jones, Cardiff University, United Kingdom
- Cláudio Baptista, UFCG, Brasil
- Cláudio Esperança, COPPE/UFRJ, Brasil
- Clodoveu Davis, PUC-MG, Brasil
- Eduardo Celso G. Camargo, INPE, Brasil
- Eymar Sampaio Lopes, INPE, Brasil
- Frederico Fonseca, Penn State, USA
- Gilberto Câmara, INPE, Brasil
- Isabel Sobral Escada, INPE, Brazil
- João Argemiro Paiva, Oracle Corporation, USA
- Jorge Campos, UNIFACS, Brasil
- José Luiz de Souza Pio, UFAM, Brasil
- José Alberto Quintanilha, USP, Brasil
- Jugurta Lisboa Filho, UFV, Brasil
- Julio Cesar d'Alge, INPE, Brasil
- Karla Albuquerque de Vasconcelos Borges, Prodabel, Brasil
- Kathleen Hornsby, University of Maine, USA
- Laercio Namikawa, INPE, Brasil
- Leila Fonseca, INPE, Brasil
- Lúbia Vinhas, INPE, Brasil
- Marcelino Pereira dos Santos Silva, UERN, Brazil
- Marcelo Tílio Monteiro de Carvalho, PUC-RJ, Brasil
- Marco Antônio Casanova, PUC-RJ, Brasil
- Marcus Vinicius Alvim Andrade, UFV, Brasil
- Maria Cecília Calani Baranauskas, Unicamp, Brasil
- Marilton Sanchotene de Aguiar, UCPel, Brasil
- Mario J. Silva, Universidade de Lisboa, Portugal
- Max Egenhofer, University of Maine, USA
- Miguel Torres, Centro de Investigacion en Computacion, Mexico
- Paulo Justiniano Ribeiro Jr., UFPR, Brasil
- Paulo Roberto Gomes Luzzardi, UCPel, Brasil
- Raul Queiroz Ferreira, PUC-Rio, Brasil
- Renato Martins Assunção, UFMG, Brasil
- Ricardo da Silva Torres, Unicamp, Brasil
- Ricardo Rodrigues Ciferri, UFSCar, Brasil
- Rolf de By, ITC, Netherlands
- Shashi Shekhar, University of Minnesota, USA
- Sergei Levashkine, Centro de Investigacion em Computacion, Mexico
- Silvana Amaral, INPE, Brasil
- Stephan Winter, University of Melbourne, Australia
- Tiago Garcia de Senna Carneiro, UFOP, Brasil
- Valéria Gonçalves Soares, UFPB, Brasil
- Valéria Times, UFPE, Brasil
- Vania Bogorny, Uhasselt, BE
- Virgínia Ragoni de Moraes Correia , INPE, Brasil
- Yola Georgiadou, ITC, Netherlands

Avaliadores Adicionais / *Additional Reviewers*

Claudio Ruschel
Guillermo Hess
Gilberto Ribeiro de Queiroz
Guaraci Erthal
Ilka Afonso Reis
Javier Morales
Marcelo Mettelo
Melissa Lemos
Miguel Fornari
Otto Huisman
Paulo Ribeiro
Robson Fidalgo
Sergio Rosim
Luciano Digiampietri
Leone Fonseca
Luciana Rocha
Wenwu Tang

PROGRAM

Domingo, 25 de novembro - *Sunday, November 25*		
15:00 – 18:00	Entrega de Credenciais *Registration*	Secretaria Geral *General Secretariat*
16:00 – 19:00	Curso de TerraME TerraME Course	Dr. Tiago G. S. Carneiro Dr. Laercio M. Namikawa

TerraME is a development environment for spatial dynamical modelling that supports the concepts of nested cellular automata (nested-CA). TerraME uses a spatial database for data storage and retrieval. A spatial dynamic model is a model whose locations are independent variables. The outcomes of these models are maps that depict the spatial distribution of a pattern or of a continuous variable. TerraME enables simulation in two-dimensional cellular spaces. Among the typical applications of TerraME are land change and hydrological models.

This tutorial provides an introduction to the basic features of TerraME. For a full description, see [1]. The tutorial has four parts. In section 2, we present the TerraME architecture. In section 3, we present the basic commands of the TerraME programming language. In section 4, we show an example of using TerraME for hydrological modelling. In section 5, we show an example of land change modelling. Before using this tutorial, the reader should first install TerraME. Readers interested in an introduction to the principles of modelling should refer to [2] or [3].

1. Carneiro, T., Nested-CA: a foundation for multiscale modeling of land use and land change., in PhD Thesis in Computer Science. 2006, National Institute for Space Research: São José dos Campos, Brazil. p. 109.

2. Odum, H.T., Systems Ecology: An Introduction. 1983, March: John Wiley and Sons.

3. Zeigler, B.P., T.G. Kim, and H. Praehofer, Theory of modeling and simulation. 2005, Orlando, FL, USA: Academic Press, Inc.

Segunda-feira, 26 de novembro - *Monday, November 26*		
08:00 – 08:30	Entrega de Credenciais *Registration*	Secretaria Geral *General Secretariat*
08:30 - 09:00	*GEOINFO´2007 Opening and Welcome Session*	Auditorium
S1 – Digital Image Processing ***Chair: Dr. Lúbia Vinhas***		
09:00 - 09:30	Improvements to Expectation-Maximization Approach for Unsupervised Classification of Remote Sensing	*Thales Korting, Luciano Dutra, Leila Fonseca, Guaraci Erthal, Felipe Silva*

S2 – User Interactions / Applications ***Chair: Dra. Claudia Bauzer Medeiros***		
9:30 – 10:00	Continuous Interaction with TDK: Improving the User Experience in Terralib	*Marcelo Metello, Mário Vera, Melissa Lemos, Leone Masiero, Marcelo Carvalho*
10:00 – 10:30	The Development of a Large Scale Geospatial Telecommunications Application Independent from GIS Proprietary Mechanisms	*Eliane Dias, Geovane Magalhães, Grace Silva*
10:30 – 10:45	Coffee Break	
10:45 – 12:00	**Invited Talk 1 - Dr. Andrea Rodriguez** **Universidad de Concepción, Chile** **Learning to live with spatial inconsistencies** Spatial consistency defines admissible values of spatial data in database and information systems. Although this is a desirable and usually enforced property of geographic information systems, inconsistency in geographic information systems is not necessarily an exception, but a common situation we have to live with. Multiple examples are found around spatial global systems that access multiple and heterogeneous spatial local databases. This talk raises issues about inconsistency handling, which aims at manipulating data despite the fact that databases or information systems may contain inconsistency information. Instead of focusing on modeling and reasoning about consistency, it discusses what we can do when inconsistency exists. The talk takes a database perspective to define consistency in geographic information and concentrates on spatial constraints (topo-semantic and geometric constraints) that describe the valid states of spatial databases. The talk discusses traditional database approaches and their limitations in the context of geographic information. It also proposes to combine structural, functional, and semantics characteristics of spatial information for handling inconsistency.	
12:00 – 14:00	Lunch	
S3 – Ontology / Spatial Databases ***Chair: Dr. Marco A. Casanova***		
14:00 – 14:30	Towards a Geographic Ontology Reference Model for Matching Purposes	*Guillermo Hess, Cirano Iochpe, Silvana Castano*
14:30 – 15:00	Approximate String Matching for Geographic Names and Personal Names	*Clodoveu Davis, Emerson Salles*
15:00 – 15:30	Trajectory Data Warehouses: Proposal of Design and Application to Exploit Data	*Fernando Jose Braz*
15:30 – 16:00	Ecologically-aware Queries for Biodiversity Research	*Celso Gomes, Claudia Medeiros*
16:00 – 16:30	Coffee Break	
S4 – Image Analysis / Applications ***Chair: Dr. Clodoveu Davis, Jr.***		

16:30 – 16:45	Construção de Mosaicos Georreferenciados Usando Imagens Aéreas de Pequeno Formato para SIG	*Natal Henrique Cordeiro, Bruno Motta de Carvalho, Luiz Marcos Garcia Gonçalves*
16:45-17:00	Análise da Complexidade de Texturas em Imagens Urbanas Utilizando Dimensão Fractal	*André Backes, Adriana Bruno, Mauro Barros, Odemir Bruno*
17:00-17:15	Utilização de Imagens de Sensoriamento Remoto de Alta Resolução para Realizar a Contagem de Copas em Povoamento de Eucalyptus spp.	*Frederico Reis, Luciano Oliveira, Luis Marcelo Carvalho*
17:15-17:30	Integração do SGBD Oracle Spatial e do Google Earth para disponibilizar informações relacionadas ao Inventário Florestal de Minas Gerais	*Samuel R. de Sales Campos, Adriana Zanella Martinhago, Thomaz Oliveira, Luca Egas Pietro, Ronaldo da Silva, Aleksander França, Ivayr Farah Netto*
19:30 – 20:30	Welcome cocktail and book launch: "GEOINFORMAÇÃO EM URBANISMO: cidade real X cidade virtual" (Geoinformation in the Urban Planning: real city X virtual city)	

Terça-feira, 27 de novembro - *Tuesday, November 27*		
S5 – Distributed GIS / GIS and Internet ***Chair: Dr. Cirano Iochpe***		
08:30 – 09:00	Federated Spatial Cursors	*Nazario Cipriani, Matthias Grossmann, Daniela Nicklas, Bernhard Mitschang*
09:00 – 09:30	A Service-Oriented Architecture for Progressive Transmission of Maps	*David Costa, Mario Teixeira, Anselmo Paiva, Claudio Baptista*
09:30 – 10:00	An Instance-based Approach for Matching Export Schemas of Geographical Database Web Services	*Daniela Brauner, Chantal Intrator, João Carlos Freitas, Marco Casanova*
10:00 – 10:15	Desenvolvimento de SIG para Web utilizando MDA	*Carlos Mello, Geraldo Zimbrão, Jano Souza*
10:15 – 10:30	O projeto WebMAPS: desafios e resultados	*Carla Macario, Claudia Medeiros, Rodrigo Senra*
10:30 – 10:45	Coffee Break	
10:45 – 12:00	**Invited Talk 2 - Dra. Dina Q Goldin** **Brown University, EUA** Dr. Goldin works with database models and query languages, computing paradigms, and algorithms.	
12:00 – 14:00	Lunch	

S6– Modelling and Representation ***Chair: Dr. Laércio M. Namikawa***		
14:00 – 14:30	Model selection for a class of spatio-temporal models for areal data	*Juan Vivar, Marco Ferreira*
14:30 – 15:00	Rule-based Evolution of Typed Spatio-temporal Objects	*Olga Bittencourt, Gilberto Câmara, Lúbia Vinhas, Joice Mota*
15:00 – 15:15	Coverage representation in TerraLib	*Vitor Dantas, Marcelo Metello, Melissa Lemos, Marco Casanova*
15:15 – 15:30	Representação das Características do Movimento de Objetos Móveis em Mapas Estáticos	*Daniel Cotrim, Jorge Campos*
15:30 – 15:45	Modelagem espacial de florestas estacionais do Domínio do Cerrado no Estado de Minas Gerais utilizando envelope climático	*Gleyce Campos Dutra, Luis Marcelo Tavares de Carvalho, Ary Teixeira de Oliveira Filho*
15:45 – 16:30	Coffee Break	
S7 – Agents / Spatial Analysis ***Chair: Dr. Antonio Miguel Vieira Monteiro***		
16:30 – 17:00	An Architecture Based on Multi-Agent Systems and Geographic Databases for the Development of Georeferenced Ecological and Social Simulations	*Pablo Grigoletti, Antonio Costa*
17:00 – 17:15	Análise espacial da distribuição de Aedes aegypti (Diptera: Culicidade) em diferentes áreas da cidade do Rio de Janeiro	*Izabel Reis, Nildimar Honório, Cláudia Codeço, Christovam Barcellos, Mônica Magalhães*
17:15 – 17:30	Extensão do WEKA para Métodos de Agrupamento com Restrição de Contigüidade	*Carlos Mello, Geraldo Zimbrão, Jano Souza*
Special Session – Brazilian Computer Society (SBC) ***Chair: Dr. Cirano Iochpe (President of SBC)***		
17:30 – 18:30	Meeting of the SBC Special Committee for Geoinformatics.	
20:00 – 22:00	Barbecue	
24:00	Soup Festival	

Quarta-feira, 28 de novembro - *Wednesday, November 28*		
S8 – Computational Geometry / Algorithms ***Chair: Dr. Antônio Carlos Rocha***		
08:30 – 09:00	Polygon Clipping and Polygon Reconstruction	*Leonardo Azevedo, Ralf Güting*
09:00 – 09:30	Weighted Overlay, Fuzzy Logic and Neural Networks for Estimating Vegetation Vulnerability in Minas Gerais, Brazil	*Luis Carvalho, Moises Ribeiro, Luciano Oliveira, Thomaz Oliveira, Julio Louzada, Jose Scolforo, Antonio Oliveira*
09:30 – 10:00	An efficient algorithm to compute the viewshed on DEM terrains stored in the external memory	*Mirella Magalhães, Salles Magalhães, Marcus Andrade, Jugurta Lisboa*
10:00 – 10:30	Comparison of Machine Learning Algorithms for Mapping the Phytophysiognomies of the Brazilian Cerrado	*Luciano Oliveira, Thomaz Oliveira, Carvalho Luis, Wilian Lacerda, Samuel Sales, Adriana Martinhago*
10:30 – 10:45	Coffee Break	
10:45 – 11:30	Closing Session	

Contents

FULL PAPERS

SHORT PAPERS

S4 - Image Analysis / Applications

S5 - Distributed GIS / GIS and Internet

S6- Modelling and Representation

S7 - Agents / Spatial Analysis

FULL PAPERS

Improvements to Expectation-Maximization approach for unsupervised classification of remote sensing data

Thales Sehn Korting[1]
Luciano Vieira Dutra[1], **Leila Maria Garcia Fonseca**[1]
Guaraci Erthal[1], **Felipe Castro da Silva**[1]

[1]Image Processing Division
National Institute for Space Research – INPE
São José dos Campos – SP, Brazil

tkorting, dutra, leila, gaia, felipe@dpi.inpe.br

Abstract. *In statistical pattern recognition, mixture models allow a formal approach to unsupervised learning. This work aims to present a modification of the Expectation-Maximization clustering method applied to remote sensing images. The stability of its convergence has been increased by supplying the results of the well-known K-Means algorithm, as seed points. Hence, the accuracy has been improved by applying cluster validity measures to each configuration, varying the initial number of clusters. High-resolution urban scenes has been tested, and we show a comparison to supervised classification results. Performance tests were also realized, showing the improvements of our proposal, in comparison to the original one.*

1. Introduction

Generally, a color composition of some remote sensing image behaves as a mixture of several colors, which changes gradually according x and y pixel positions. If a specialist performs a manual classification in a certain image, and after views its scatter plot, the classes will appear together, in such a way that linear classification algorithms will not have success when classifying it. Figure 1 shows one example of this idea.

In this Figure, we used 6 classes, namely *Streets, Pools, Roofs, Shadows, Greens*, and *Others*. By visualizing the scatter plots, which draws the pixel occurrence and also pixel class for bands RG, RB and GB, it seams clear that classes named roofs and swimming pools are linearly separable from the rest, as shown in the second scatter plot (Figure 1c). However, the other 4 classes remain together, and it's a challenging task to discover their statistical distributions. Each class can be thought as an independent variable; as they are a fraction of a total (the entire image), it characterizes a mixture model.

One way to estimate mixture models is to assume that data points have "membership" in one of the distributions present in the data. At first, such membership is unknown. The objective is to estimate suitable parameters for the model, where the connection to the data points is represented as their membership in the individual model distributions.

In statistical pattern recognition, such mixture models allow a formal approach to unsupervised learning (*i.e.* clustering) [Figueiredo and Jain 2002]. A standard method to fit finite mixture models to observed data is the *Expectation-Maximization* (EM) algorithm, first proposed by [Dempster et al. 1977]. EM is an iterative procedure which

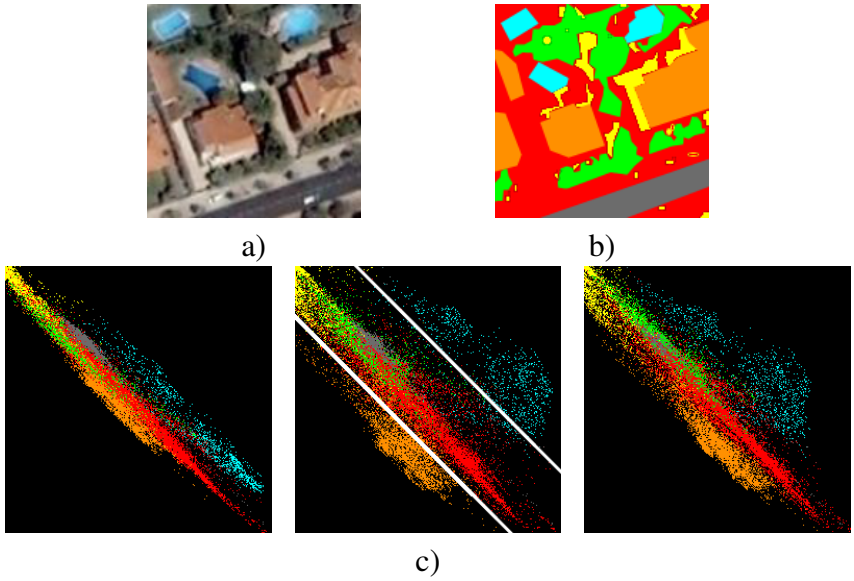


Figure 1. a) Example remote sensing image. b) Manual Classification. c) Scatter Plots of bands RG, RB and GB considering manual classification.

converges to a (local) maximum of the marginal *a posteriori* probability function without manipulating the marginal likelihood $p(\theta|\mathbf{x})$ [Figueiredo 2004]:

$$p(\theta|\mathbf{x}) = p(\mathbf{x}|\theta)p(\theta) \tag{1}$$

where θ is a set of unknown parameters from $\mathbf{x}$. Therefore, EM estimates the components probabilities present in a certain cluster. In our case, the input is composed by the image pixels, and the parameters are mean and variance.

In other words, EM is a general method of estimating the features of a given data set, when the data are incomplete or have missing values [Bilmes 1998]. This algorithm has been used in several areas, such as image reconstruction [Lay and Katsaggelos 1990, Qian and Titterington 1993, Shepp and Vardi 1982], signal processing, and machine learning [Beal and Ghahramani 2003, Guo and Rodriguez 1992, Lawrence and Reilly 1990].

The finite mixture models are able to represent arbitrarily complex probability density functions [Figueiredo 2004]. This fact makes EM approach proper for representing complex likelihood functions, considering Bayesian inference. Being an iterative procedure, the EM method can present high computational cost. So, in this article we present a variation of the EM algorithm, increasing stability and capability, by providing the first set of parameters from K-Means algorithm and performing clustering validation.

The paper is organized as follows. Section 2 starts explaining the EM approach and its application to mixture models, followed by how to estimate the parameters using such method. After, in Section 3 we show our main contribution describing the "improved EM" approach. We discuss the implemented system, divided by modules on the whole process. Section 4 presents some results when applying the method to urban remote sensing images, and a discussion over the performance achieved using the suggested

improvements. In Section 5 we conclude with some remarks about the results and future works.

2. The standard EM algorithm

An image pixel might behave differently if it comes from an edge rather than a smooth region. Therefore, the global behavior is likely to be a mixture of the two distinctive behaviors [Bouman 1995]. The objective of the mixture distributions is to produce a probabilistic model composed of a subclasses set. In our approach, each class is characterized by a set of parameters describing the mean and variance of the spectral components.

EM algorithm is based on the Bayesian theory. We assume the algorithm will estimate M clusters (or classes) $C_j, j = 1, \ldots M$. For each of the N input vectors $\mathbf{x}_k, k = 1, \ldots N$, the algorithm calculates its probability $P(C_j|\mathbf{x}_k)$ to belong to a certain class [Theodoridis and Koutroumbas 2003]. The highest probability will point to the vector's class.

Being an unsupervised classification method, there is no training stage. The image and the number of clusters to be estimated form the input. The attributes-vector is composed of the pixel-value for each band. So, an image with three bands produces a 3D-space for the whole set, and so on.

2.1. Computing EM

The EM algorithm works iteratively by applying two steps: the E-step (*Expectation*) and the M-step (*Maximization*). Formally, $\hat{\theta}(t) = \{\mu_j(t), \Sigma_j(t)\}, j = 1, \ldots M$ stands for successive parameter estimates. The method aims to approximate $\hat{\theta}(t)$ to real data distribution when $t = 0, 1, \ldots$

E-step: This step calculates the conditional expectation of the complete *a posteriori* probability function;

M-step: This step updates the parameter estimation $\hat{\theta}(t)$.

Each cluster probability, given a certain attribute-vector, is estimated as following:

$$P(C_j|\mathbf{x}) = \frac{|\Sigma_j(t)|^{-GB} e^{\eta_j} P_j(t)}{\sum_{k=1}^{M} |\Sigma_k(t)|^{-GB} e^{\eta_k} P_k(t)} \tag{2}$$

where

$$\eta_i = -\frac{1}{2}(\mathbf{x} - \mu_i(t))^T \Sigma_x^{-1}(t)(\mathbf{x} - \mu_i(t))$$

With such probabilities, one can now estimate the mean, covariance, and the *a priori* probability for each cluster, at time $t + 1$, according to Equations 3, 4, and 5:

$$\mu_j(t+1) = \frac{\sum_{k=1}^{N} P(C_j|\mathbf{x}_k)\mathbf{x}_k}{\sum_{k=1}^{N} P(C_j|\mathbf{x}_k)} \tag{3}$$

$$\Sigma_j(t+1) = \frac{\sum_{k=1}^{N} P(C_j|\mathbf{x}_k)(\mathbf{x}_k - \mu_j(t))(\mathbf{x}_k - \mu_j(t))^T}{\sum_{k=1}^{N} P(C_j|\mathbf{x}_k)} \tag{4}$$

$$P_j(t+1) = \frac{1}{N} \sum_{k=1}^{N} P(C_j|\mathbf{x}_k) \tag{5}$$

These steps are performed until reaching the convergence, according the following equation [Theodoridis and Koutroumbas 2003]:

$$\| \theta(t+1) - \theta(t) \| < \varepsilon \tag{6}$$

where $\| \ . \ \|$, in this implementation, is the Euclidean distance between the vectors $\mu(t+1)$ and $\mu(t)$, and ε is a threshold chosen by the user. After the calculations, Equation 2 is used to classify the image. The next section explains the classification in detail.

3. The "improved EM" approach

Figure 2 shows a diagram composed of four modules, presenting our method, according equations presented in the previous section, and with the contributions presented by this paper.

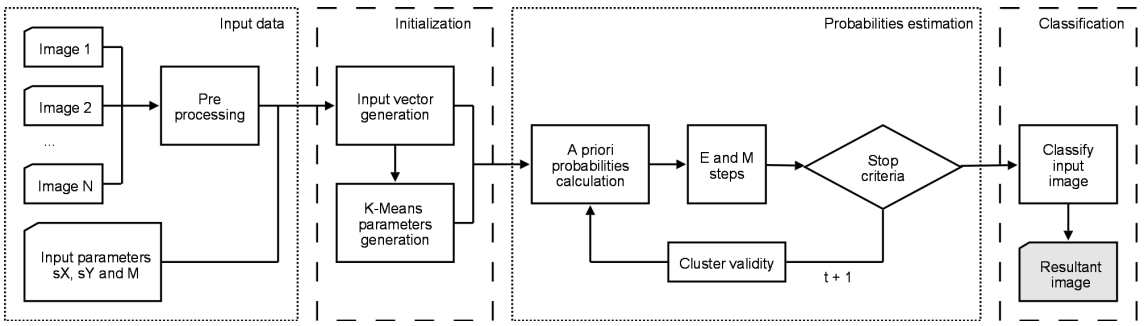


Figure 2. System's diagram.

The implementation follows this script:

Input data: this module deals with N images and the input parameters called sampling rate (sX and sY), on directions x and y. This rate aims to reduce the input data, building the input vector as a fraction of the image pixels. M stands for the number of clusters the algorithm has to estimate. Here we propose a preprocessing stage, removing, from the input data, pixels close to the image border because, because of sensor features, sometimes they are not trustworthy;

Initialization: using the sampling rate, we build the instance set $\mathbf{x}$, and create the θ set, with seed points provided by the K-Means algorithm. On the standard EM implementation, the first set of parameters are randomized, and this is one of the main causes of the high computational cost of this algorithm, and the risk of converging to local minimums;

Probabilities estimation: this module performs the iterative procedure of successive parameter estimation and cluster validity, described below. Such technique aims to certify the number of classes provided by the user, and guarantee that all clusters are distant from each other. While t increases, a test is performed to check if the algorithm has already converged, or a maximum number of iterations have been reached;

Classification: here the final classification is performed. For each of the N pixels $\mathbf{x}_k$ is associated the class with higher probability, that is, find $P(C_j|\mathbf{x}_k) > P(C_i|\mathbf{x}_k), j \neq i$ and classify $\mathbf{x}_k$ as C_j.

The "Initialization" and "Probabilities estimation" modules were adjusted to carry out more stability and capability to the results. We introduced the solution to use K-Means for producing the first set of unknown parameters θ, *i.e.* when $t = 0$. Applying this to the EM approach, we reduce the number of iterations, thus reducing computational time.

Sometimes, the algorithm is not able to converge, during the "Probabilities estimation" module, to the entire set of classes, because of the mixture models natural behavior. On our approach, we modified each iteration of this module by validating the current clustering arrangement. During convergence, if a cluster center is approaching another one, then one of them is randomly modified for the next iteration. This aims to "shake" the values, so that cluster C_j may converge to another class, far from C_i in the attribute space.

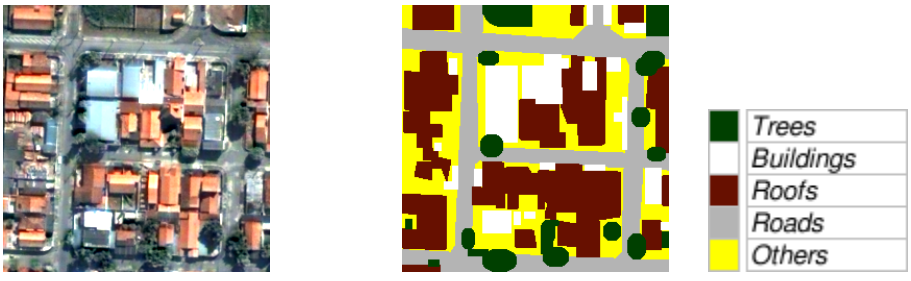
Considering clustering validation, we also perform cluster exclusion when some of them have a low probability. It was implemented because sometimes the user-supplied parameters can have a mistaken number of parameters, or the attributes distribution doesn't allow detecting a certain number of clusters. Through a threshold η, the cluster exclusion is implemented according the following equation:

$$\text{if } P_i(t) < \eta \text{ then exclude cluster } C_i \tag{7}$$

4. Results

This section presents some results, applying the EM algorithm to classify remote sensing images.

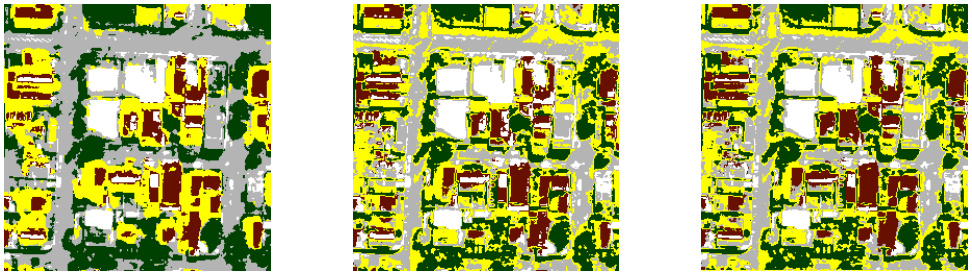
Firstly we have a color composition of an urban area from São José dos Campos – Brazil. Such image was taken in January 2004, from QuickBird, and the composition is R3G2B1. Figure 3 shows the original image and its manual classification, with respect to 5 classes, namely *Trees, Buildings, Roofs, Roads*, and *Others*. In order to analyze the results and compare it with another known methods, we performed the classification using three algorithms: improved EM, KMeans and Maximum Likelihood (ML). The classification results are shown in Figure 4.



a) b)

Figure 3. a) Color composition R3G2B1 of QuickBird scene from São José dos Campos – Brazil. b) Manual classification.

To prove the enhancement on the results, by the use of our improved EM approach, we show on Table 1 the agreement matrices for each of the tested algorithms.



a) b) c)

Figure 4. Classification results using: a) Improved EM, b) KMeans, and c) ML.

By observing the tables we can note that the only algorithm which achieved more than 70% of correct matches was the improved EM method. We should already expect better results than KMeans, since it provides the first set of parameters, and improved EM adjust them in a better way. However, the ML algorithm is supervised, and the training stage was performed with enough samples. Another point that must be taken into consideration is the low matches of the last class, named *Others*. Even being a bad result, this class stands for the less important set of objects in the scene, that even was not classified by the speciallist.

Since the improved EM algorithm got good results for "urban classes", like *Trees* and *Roads*, also better than the other algorithms, we must point out that our approach can be used successfully for such kind of image classification.

Table 1. Agreement Matrices for: a) Improved EM, b) KMeans, and c) ML. Classes are: 1) Trees, 2) Buildings, 3) Roofs, 4) Roads, and 5) Others.

	1	2	3	4	5
1	0,87	0,05	0,05	0,25	0,40
2	0,00	0,57	0,07	0,03	0,03
3	0,00	0,02	0,31	0,00	0,01
4	0,10	0,28	0,05	0,70	0,39
5	0,03	0,08	0,53	0,02	0,17

a)

	1	2	3	4	5
1	0,66	0,07	0,15	0,19	0,30
2	0,01	0,59	0,10	0,06	0,04
3	0,00	0,01	0,35	0,01	0,00
4	0,03	0,23	0,09	0,49	0,25
5	0,30	0,09	0,31	0,26	0,40

b)

	1	2	3	4	5
1	0,67	0,07	0,16	0,18	0,31
2	0,00	0,56	0,06	0,05	0,03
3	0,00	0,01	0,36	0,00	0,01
4	0,03	0,27	0,13	0,48	0,26
5	0,29	0,09	0,29	0,30	0,39

c)

Figure 5a shows a CBERS-2 color composition of bands 2, 3, and 4. Three classes are identified on this image, namely *Urban, Vegetation*, and *Water*. Figures 5b and 5c show, respectively, the scatter plot and the classified image for different classification methods[1]: EM, ML, and Euclidean Distance (ED). We show the scatter plots, so that the reader is able to perceive the mixture model present in such image, and also to draw the classification result, since the classes are exposed on each combination of bands RG, RB and GB. And, in comparison to the other approaches, EM got the smoothest thematic

[1]Software SPRING was used to perform such classifications [Câmara et al. 1996]

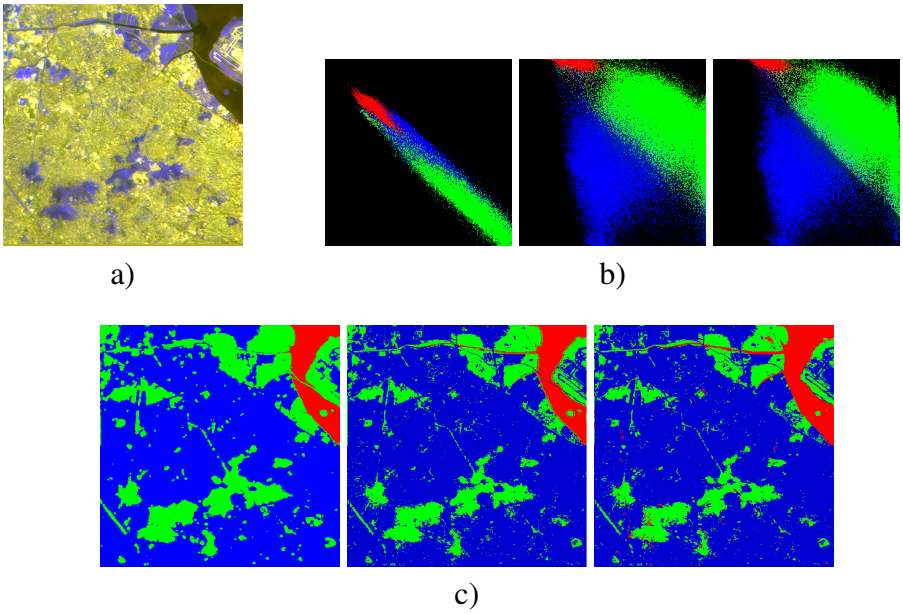


Figure 5. a) CBERS-2 color composition R2G3B4. b) Scatter plots. c) Classification (from left to right) using EM, ML, and ED methods.

map. It is important to point out that both methods (ML and ED) are supervised; however, visually the EM result is more satisfactory, as we can observe comparing results on Figure 5c.

4.1. Discussion

EM algorithm presents some drawbacks. Being a local method, it is sensitive to initialization because the likelihood function of a mixture model is not unimodal [Figueiredo and Jain 2002]. This was the main cause for using K-Means as first set of parameters. For certain mixture types, it may converge to the parameter space boundary, leading to meaningless estimates.

However, to test the performance increase we have performed several tests, using the original EM proposal, and the modified approach. The tests considered processing time until convergence, for both approaches. We used 5 different images and input parameters, so the final increasing in performance is unbiased. Table 2 shows the results, considering image size, number of classes for each one, and computational time until convergence.

Table 2. Comparison between original and improved approaches.

	Image1	Image2	Image3	Image4	Image5
Image size	512×512	512×512	200×200	512×384	264×377
# of classes	4	4	5	6	5
Δt_1 original EM	$467s$	$467s$	$103s$	$402s$	$202s$
Δt_2 improved EM	$140s$	$148s$	$29s$	$105s$	$70s$
$\Delta t_1/\Delta t_2$	3.335	3.155	3.551	3.828	2.885

Calculating the average values for time decrease, showed in Table 2 at the line $\Delta t_1/\Delta t_2$, we reach the value 3.35. This means that our improved approach is around

$3\times$ faster than the original, and considering the showed results, its also more robust to outliers.

Images classified by pixel-based methods (not on region), generally, present a noisy appearance because of some isolated pixels that are misclassified [Guo and Moore 1991]. To fix such problem, we can use some post-classification method. One expects some degree of spatial correlation among neighborhood pixels, so we can remove isolated misclassification, resulting in a smoothed map.

Even becoming faster than the original approach, the EM algorithm is still more expensive than the other methods. It performs calculations of inverse matrix and determinant at each iteration, for the whole set of data. One approach, to reduce the computational demand, is to assume that all covariance matrices are diagonal or that they are equal to each other [Theodoridis and Koutroumbas 2003]. In this case, only one inversion matrix is needed at each iteration step, however, the system loses in generalization.

5. Conclusion

This work has presented an improvement to the EM Clustering Method, by using K-Means results as input, and some cluster validity techniques. Estimating mixture parameters is clearly a missing data problem, where the cluster labels of each observation are unknown [Figueiredo 2004]. The EM algorithm can be adopted, as we have proposed in this work, as a standard choice for this task.

One advantage of the EM algorithm is that its convergence is smooth and not vulnerable to instabilities. However, we have shown that wrong initial parameters might result in meaningless classification. Therefore the proposed approach, which estimates the first parameters using K-Means, increases the resultant accuracy.

In [Starck et al. 1998] they present the recovery of Gaussian-like clusters, applying the *à trous* wavelet. Future works include tests not only with K-Means approach but with this one as well. We also intend to perform another preprocessing stage, searching for outliers and removing them from the whole data set.

We have shown how to implement an EM algorithm and how to apply it to unsupervised image classification. As well, some classification results obtained with the proposed method and others were shown to compare their accuracy. We have implemented the algorithm using TerraLib library [Câmara et al. 2000], which is available for free download at `http://www.terralib.org/`. We also developed a software for unsupervised image classification, available at `http://www.dpi.inpe.br/~tkorting/`.

References

Beal, M. and Ghahramani, Z. (2003). The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian Statistics*, 7:453–464.

Bilmes, J. (1998). A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. *manuscript, International Computer Science Institute*.

Bouman, C. (1995). Cluster: An unsupervised algorithm for modeling gaussian mixtures. *preprint available at http://www. ece. purdue. edu/ bouman/ softare/cluster/manual. pdf*.

Câmara, G., Souza, R., Freitas, U., and Garrido, J. (1996). SPRING: integrating remote sensing and GIS by object-oriented data modelling. *Computers & Graphics*, 20(3):395–403.

Câmara, G., Souza, R., Pedrosa, B., Vinhas, L., Monteiro, A., Paiva, J., Carvalho, M., and Gatass, M. (2000). TerraLib: Technology in Support of GIS Innovation. *II Workshop Brasileiro de Geoinformática, GeoInfo2000*.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood estimation from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

Figueiredo (2004). Lecture Notes on the EM Algorithm. Technical report, Institute of Tele-communication.

Figueiredo, M. and Jain, A. (2002). Unsupervised learning of finite mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(3):381–396.

Guo, G. and Rodriguez, G. (1992). Estimating a Multivariate Proportional Hazards Model for Clustered Data Using the EM Algorithm, with an Application to Child Survival in Guatemala. *Journal of the American Statistical Association*, 87(420):969–976.

Guo, L. and Moore, J. (1991). Post-classification Processing For Thematic Mapping Based On Remotely Sensed Image Data. *Geoscience and Remote Sensing Symposium, 1991. IGARSS'91.'Remote Sensing: Global Monitoring for Earth Management'., International*, 4.

Lawrence, C. and Reilly, A. (1990). An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins: Structure, Function and Genetics*, 7:41–51.

Lay, K. and Katsaggelos, A. (1990). Blur identification and image restoration based on the EM algorithm. *Optical Engineering*, 29(5):436–445.

Qian, W. and Titterington, D. (1993). Bayesian image restoration: an application to edge-preservingsurface recovery. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(7):748–752.

Shepp, L. and Vardi, Y. (1982). Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imag*, 1(2):113–122.

Starck, J., Murtagh, F., and Bijaoui, A. (1998). *Image Processing and Data Analysis: The Multiscale Approach*. Cambridge University Press.

Theodoridis, S. and Koutroumbas, K. (2003). *Pattern Recognition*. Academic Press.

Continuous Interaction with TDK: Improving the User Experience in Terralib

Marcelo Metello, Mário de Sá Vera, Melissa Lemos, Leone Pereira Masiero, Marcelo Tilio Monteiro de Carvalho

Tecgraf—Computer Graphics Technology Group,
Pontifical Catholic University of Rio de Janeiro,
Rua Marquês de São Vicente 225, Gávea, Rio de Janeiro, RJ 22453-900, Brazil

{metello,mvera,melissa,leone,tilio}@tecgraf.puc-rio.br

Abstract. *Historically, visual display has always played a very important role in GIS applications. However, visual exploration tools do not scale well when applied to huge spatial data sets. More recently, faster processing hardware and more sophisticated computer graphics has been used to improve user experience for geospatial data visualization. Some applications have recently introduced the concept that the user should be able to navigate through the data in a smooth and continuous way without being blocked waiting for data to be loaded, even if this means seeing incomplete data for some time. Google Earth and NASA World Wind are recent examples of this concept. This paper describes an architecture incorporated in the TDK (TerraLib Development Kit) open source API to provide support for applications that want this kind of user interaction with Terralib geographical data.*

1. Introduction

In a Geographic Information System (GIS) data visualization plays a fundamental role in the system user experience. From a location based application to a more sophisticated spatial data manipulation graphic tool the user wants to have a pleasant interaction with the data visualization interface in place.

If we consider the dualism of GIS data models where thematic and spatial data are together [Oosterom 1988, Longley et al 2001] we can understand the particularity of the user visual experience in a GIS scenario. The spatial data component requires a special paradigm for visualization in order to give the user a realistic interaction with the system.

Another consideration is the requirement for an efficient processing of data for a pleasant user experience. In terms of data scaling factors GIS definitely falls into one of major high scale heavy databases. So while visualizing a GIS database information the data requests are not as easy to be addressed as in the majority of other fields.

For the past several decades the user experience with GIS data visualization has been restricted to a simple “geographic map like” paradigm. In this legacy model the user navigates over the data by panning over the data with the visualization window. This user interface model was inherited from the standards brought by modern

Graphical User Interfaces solutions [Mesa 1998] from the 80s. The inventors of this GUI paradigm were definitely not aware of the particularities involved in a GIS data visualization application. As a consequence of this non predictable usage of the standard two dimensional window oriented paradigm the GIS data visualization suffers of several potential drawbacks: (i) the data navigation as a blocking prone experience; (ii) the system is not responsive all the time.

One major evolution experienced by the game development technology has been the usage of computer graphics techniques to give the user a navigation experience more realistic. These techniques have been adopted in the GIS community by systems such as Google Earth [Google 2007] and World Wind [NASA 2007] to improve user experience. These systems have implemented what we define here as *continuous interaction* as opposed to *discrete interaction.*

By *discrete interaction* we mean that the user clearly sees that the visualization area "jumps" from one point to another which is perceptively away. On the other side we call *continuous interaction* [Faconti et al 2000, Smith et al 1999] when the user sees the visualization area moving in a smooth way so that the user does not perceive any discontinuity in the visualization area trajectory. For a truly rich navigation experience we consider the use of *continuous interaction* fundamental.

The objective of this work was to provide the open source project TDK (TerraLib Development Kit) [Tilio and Vera 2005, TDK 2007] with support for continuous interaction. TDK is an API build on top of Terralib [Câmara et al 2000] with the purpose of providing components and services to make it easier to implement GIS applications using Terralib. Terralib is a GIS C++ library, available from the Internet as open source, devoted to the development of multiple GIS tools. Its main aim is to enable the development of a new generation of GIS applications, based on the technological advances on spatial databases. The architecture presented here was implemented and incorporated into TDK (in version 2.1).

We followed two strategies to reach the objective: (i) development of a spatial data responsive cache using spatial access methods for taking the spatial proximity in consideration for efficiency and fast data feedback; (ii) absorption of Computer Graphics high performance navigation techniques applied currently to flight simulators and gaming in general. These techniques aim to make full use of all the resources available in the Graphics Processing Unit (GPU) of modern personal computers.

The GPU is a single-chip processor and dedicated graphics rendering device used in a personal computer or game console. Modern GPU's such as the ATI Radeon [ATI 2002] and GeForce [nVIDIA 2002] are designed to handle well 3D computer graphics and since they are optimized for graphics, they make use of parallelization and are extremely fast for operations such as texture mapping, rendering triangles, color operations and interpolation, coordinate system transformation and vector and matrix operations.

The contributions of this paper lie in exploring how data preprocessing and data storage in cache in a format close to the one required by the graphics toolkit (which is *OpenGL* [OpenGL 2007] in this project) may become a viable strategy for rendering maps quickly and make the whole system more user-responsive. We also describe an

implementation effort that was carried out to test the ideas described in this paper and prove that it is possible to provide *continuous interaction* with data provided by Terralib.

The paper is organized as follows. Section 2 describes in detail the system architecture proposed in the paper. Section 3 discusses experimental results obtained with an implementation of the system. Finally, Section 4 contains conclusions and future work.

2. System Architecture

This section shall begin with some definitions followed by the description of all the steps of the rendering process and finally presents the general system architecture.

2.1 Definitions

Terralib defines the concept of *theme*, which was absorbed by *TDK*. In the context of this paper, a *theme* is a set of geographic objects of the same nature plus all style information needed to render each object. Each object has its own geometries (such as points, lines and polygons) and alphanumeric attributes. A visual style can be defined for all objects in a theme, for a group of objects (based on conditions on their attributes) or for an individual object. Every *theme* will have all its objects grouped into *data blocks* by spatial proximity. It allows spatial indexing with more efficiency as compared to treating each object individually. We also call *map* the image created by rendering a vector of themes in a given visualization window.

2.2 The Rendering Process

The Figure 1 shows the dataflow process that begins with the geographic data stored initially in a *Terralib* database. This data is requested according to *map* rendering demands. When loaded, every piece of data is preprocessed into a format friendly to the graphics toolkit. After that the preprocessed data is stored in a visualization cache to be used by the rendering routines and sent to the GPU.

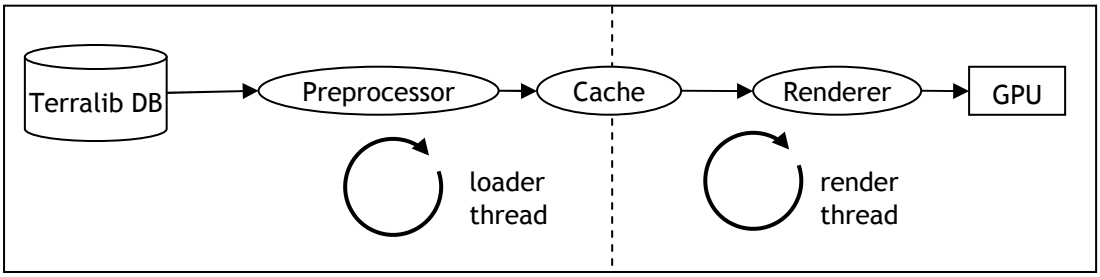


Figure 1. Rendering Process.

2.3 Data Format and Preprocessing

As it was already mentioned, a memory cache is needed in order to render *maps* quickly and make the whole system more user-responsive and interactive. A number of techniques have been developed to meet these requirements [Certain et al, 1996, Matos et al, 1998, Pinheiro et al, 2004].

Our approach was consider that data should be stored in memory in a format as close to the graphics API format as possible in order to improve rendering performance. That's why the data is preprocessed before being stored in the cache. This preprocessing step includes: (i) geographic projection conversion; (ii) polygon tessellation (breaking them in triangles) and (iii) style grouping (putting together geometries that will be drawn with the same style).

The reason for projection conversion is that converting geometries between two geographic projection systems may involve quite complex calculations and it is very costly compared to the total rendering time, therefore converting the geometries in advance and only once helps improving rendering performance. The polygon tessellation process consists in generating a triangle mesh that will cover exactly the same area as the original polygon. Although it is costly, the tessellation is necessary to make full use of all the GPU features such as parallelization and pipelining since its architecture is more prepared to handle triangles. Lastly, style grouping is also an optimization for the GPU because it allows rendering the same data with less state changes, such as changes in the current color, line width, textures and so on.

By doing all preprocessing in advance, every time the system needs to render a *map* it will do it in the fastest possible way if the data is in the memory cache. But what should happen if it is not? From the user point of view, it is highly desirable that the system is responsive and that it gives the most feedback as quickly as possible. This way, in this situation, the system renders whatever is in memory and starts loading the missing parts without blocking the user interaction. The system was designed as multithreaded exactly to handle this case: load data while interacting with the user.

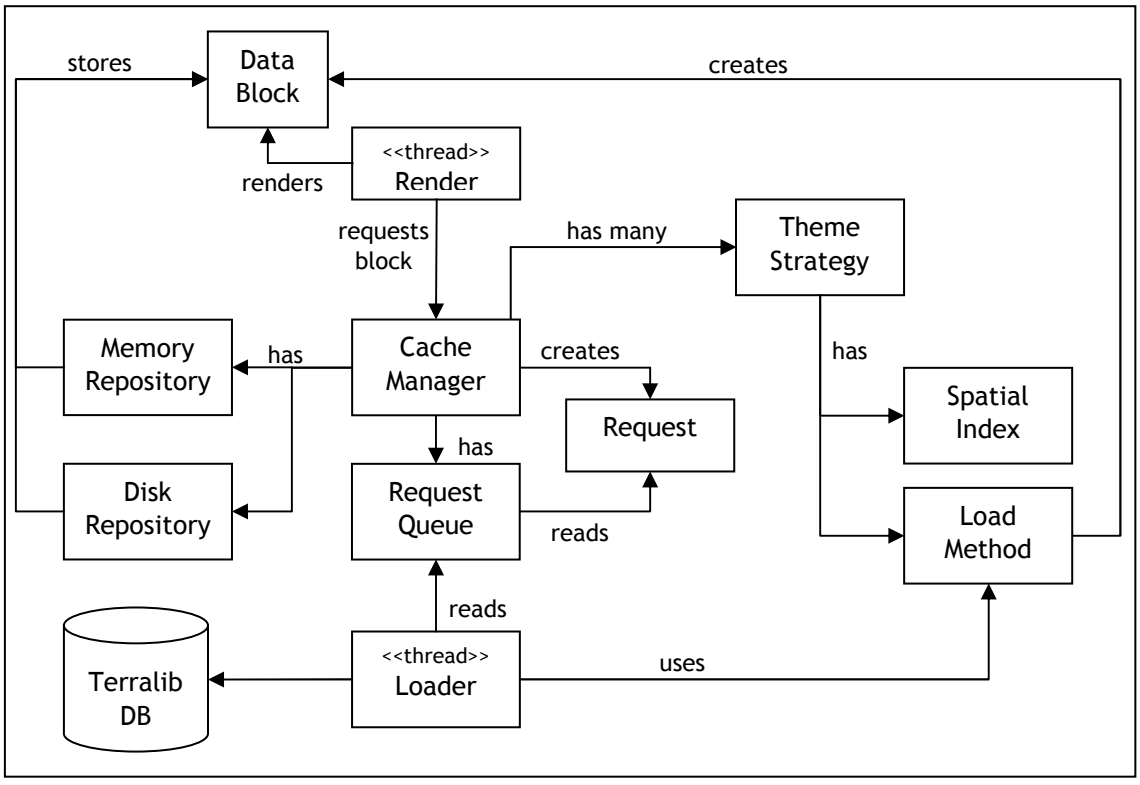


Figure 2. Functional Diagram.

There are currently two threads in the system: the *loader thread* loads data from the database to the cache and the *render thread* renders *maps* and handles user input. These two threads communicate through the cache: the *render thread* reads from the memory cache and the *loader thread* puts data in it. Besides that, when the *render thread* needs data that is not in memory, it queues a request to the *loader thread* using a *request queue*. The *loader thread* continuously checks the request queue looking for new requests.

The functional diagram (Figure 2) shows the main components of the system. The process starts when the *render thread* requests a *data block* to the *cache manager*, who checks whether the block is already in the *memory repository*. If it is, just return it. If it is not, check if it is in the *disk repository*. If it is, load it to the *memory repository* and return it. If it is not, load it from the database to the *memory repository* and return it.

Whenever the *memory repository* is full and a data block needs to be loaded, the *cache manager* should pick one other block in memory chosen by some heuristics (e.g. the block that is not used by the largest period) and save it in the *disk repository*. If the *disk repository* is full, then it needs to pick a block and just delete it to free some space. This way the cache is composed by a three-level hierarchy [Matos et al. 1998] that should, by the use of good heuristics, optimize the whole process of data visualization by using all resources available in an efficient way.

2.4 Immediate Response

In some cases, especially when *continuous interaction* is desired, the *render thread* cannot wait for the loading process to finish, even if it consists of just one block. In this case, it should put a flag named “immediate response” in its request. This flag will make the cache return a block only if it is already in memory. If it is not, it will queue a request for the *loader thread* and return a null block. This is how a behavior like Google Earth or NASA World Wind can be achieved with this cache system. By always returning the request call immediately, the user interaction is guaranteed not to stop, even if all data needed to render the *map* has not been loaded yet. In this case, the user will see an incomplete map while the rest of the data is being loaded.

2.5 Hints from the Render Thread

Imagine the following scenario: the user starts navigating through some path in a faster way than the cache can load all the data along it. Using a naïve strategy, what will happen is that, when the user stops navigating, he will wait for the system to load all data along the path because a lot of requests were queued in the way, even if only a small part of all this data is still visible to him.

The *render thread* can give hints to the cache to help it optimize the loading task. For instance, it can provide the information of a rectangle, inside which is all the data the user is currently visualizing. This information can be extremely interesting to the cache because, if it has not started loading a *data block* and this block is not within the visualization area anymore it does not have to be loaded. In the scenario just mentioned above, after the user stops moving, only the data visible to him will be loaded.

2.6 Points of Flexibility: Data Format and Indexing

The system architecture offers two great points of flexibility (hot spots): the *data block* and the *theme strategy* components.

The *data block* is a class that defines the format in which the geospatial data will be stored in memory plus the preprocessing functions. Any application can write its own *data block* class by implementing a common interface and provide a factory [Gamma et al 1995] of *data blocks* to the *cache manager*. This way the cache engine can hold spatial data in any format. This is very convenient for fast rendering because the data can be stored in the most convenient format for the graphics toolkit.

The other flexibility point is the *theme strategy*. This class implements a common interface and is responsible for the blockage and spatial indexing of a *theme* and the loading method. By blockage of a *theme* we mean the way the objects of a *Terralib theme* will be grouped into blocks (usually by some way that takes spatial proximity into consideration). The spatial indexing strategy is a data structure used to spatially query the data blocks in memory, more specifically it answers questions like, for example, which blocks intersect a window query. The default structure provided is a R-Tree [Oosterom, P.V.,1999], but any application can implement other strategy. The loading method is simply a method that loads a block. This flexibility is important because it allows the use of indexed persistence formats in an efficient way. It also allows any application to load data from data sources other than a *Terralib* database.

3. Results

The *VIPE* [TDK 2007] application is built on top of *TDK* and was used to test the results of the architecture proposed in this paper. This section presents experimental results we have obtained with three distinct versions: *VIPE v0*, *VIPE v1* and *VIPE v2*, this last one was built especially to test the concepts and architecture proposed in this paper. The others versions are explained in better detail later in this section.

Since *VIPE v2* could successfully reproduce the user experience of systems like Google Earth and World Wind in terms of rendering speed and system responsiveness, it would be very interesting to compare it with them. However, we found no direct way to make a full comparison between these systems.

There are basically three measurable factors critical to systems like these: database loading performance, disk loading performance and rendering performance. The database loading performance cannot be compared because Google Earth and World Wind cannot access a *Terralib* database. A disk loading performance comparison could be unfair since in our system the data on disk is already preprocessed and efficiently blocked while the other systems would have to access a common file format like shape files [ESRI 1998] or GML[OGC 2007]. Finally, a rendering performance comparison could also be unfair since the other systems render data in a 3D environment while ours works only in 2D for now, besides there are significant differences in the style representation capabilities for rendering geographic features in each of these systems. Because of all the factors just mentioned, we have decided to compare *VIPE v2* with one other existing application, namely *Terraview v3.1.4* [INPE-

DPI 2007] and two previous versions of *VIPE* (*v0* and *v1*) because all of them are built on top of *Terralib*.

The test set used was a *Terralib* database build from data provided by IBGE (the Brazilian Institute for Geography and Statistics) and consist of the following *themes* (all of them covering all the territory of Brazil): municipal district boundaries(5,621 polygons), permanent rivers (179,701 polylines), highways (30,969 polylines with), cities and capitals (5,771 points). The total amount of coordinates of all these *themes* is 2,903,498. The machine in which the tests were executed was a Pentium Dual Core 3.00 GHz, 1GB RAM with a nVIDIA GeForce 7300 SE/7200 GS graphics card.

The *Terraview* and *VIPE v0* systems are very similar with respect to the rendering process. Neither implements any sort of caching mechanism. Every time they need to render a *map*, they just generate a SQL query to the *Terralib* database to fetch all geometries that intersect the visualization window (i.e. the area being rendered) and render the *map* in a loop consisting of three steps: fetch a geometry from the record set returned from the database, do all the processing needed and finally render the geometry. The loop goes on until there are no more geometries intersecting the visualization area. It becomes clear that it is not possible to compare database loading performance or rendering performance individually because these operations are interleaved in these systems. Therefore, the comparison was made by the whole operation (loading and rendering).

The *VIPE v1* system implements a cache mechanism and it stores raw database data in the *Terralib* format. All the processing is done at rendering time. In order to test the rendering performance with and without the data in the cache, two time measures were taken, one for rendering right after connecting to the database (without any data in the cache) and one right after this first (with all the data in the cache). Special care was taken to configure the cache so that the entire map would fit in memory.

All tests were run multiple times and the average time was recorded. This way was preferred to testing with a profiling tool because we wanted results as close to the user experience as possible. It was interesting to compare different versions of the same system to have a better idea of how the evolutions on the architecture influenced the user experience.

Table 1. Experimental Results Comparison.

System	Average Time for Map Rendering	
	Uncached data	**Cached data**
Terraview	37.38s	-
VIPE v0	53.21s	-
VIPE v1	60.20s	8.14s
VIPE v2	114.40s	0.10s

We observed in Table 1 that both *VIPE v1* and *VIPE v2* performed better when the data was in the cache. This confirms the utility of caching in reducing the time for map rendering.

We can also see that with cached data, *VIPE v2* was much more efficient than *VIPE v1*. This occurred because *VIPE v2* caches preprocessed data in a format very close to the one requested by the graphics toolkit. This test shows us that this sort of preprocessing can bring extreme performance improvement in rendering time, especially in a toolkit that explores well the GPU features such as *OpenGL*.

The performance achieved by *VIPE v2* was enough to implement *continuous interaction* for a reasonably large data set at an acceptable frame rate. It is a consensus in the computer graphics area that the minimum frame rate for real time scientific visualization is about six frames per second, which allows a maximum rendering time of approximately 0.16 seconds.

It is worth mentioning that, during the test with uncached data, *VIPE v2* rendered all data loaded so far about 1,000 times while the others only needed to render once. This is necessary because *VIPE v2* allows continuous interaction. That certainly contributed for it taking more time to finish loading all the data than the others. A positive side of it is that, while it was loading the data, the user navigation was not blocked.

Finally, the systems without caching mechanisms (*Terraview* and *VIPE v0*) seem to load data more quickly than the others. This accounts probably to the fact that their rendering processes do not have the preprocessing and caching steps (illustraded in Figure 1). Both *VIPE v1* and *VIPE v2* need to organize data in memory to take advantage of the spatial dimension of data and therefore, some amount of extra processing is needed. In the case of *VIPE v2*, the data has to be grouped into blocks by spatial proximity and also, a spatial index structure for the data in memory has to be maintained. Besides that, in this test, it rendered all its cached data about 1,000 times.

Since *VIPE v2* rendering time improved approximately by a factor of 80 compared to the *VIPE v1* system (by comparing their average times with cached data), the conclusion we take is that, in GIS, it is worth paying attention to the GPU capabilities and make an effort to provide data in an efficient format to it.

4. Conclusion and Future Work

In this paper we proposed an architecture implemented in TDK that preprocesses data and stores data in a memory cache to reduce the time for rendering maps and to make the GIS applications more user-responsive. This strategy allows the user to navigate through geospatial data in a smooth and continuous way.

We also presented experimental results we have obtained that confirm that this strategy makes a big difference in rendering performance in GIS applications.

This project opens a lot of space for future work. The main lines would be: raster support, optimization, applying these caching techniques to servers and taking advantage of *OpenGL* and the GPU for 3D rendering and navigation.

Although raster data (i.e. data represented by a rectangular grid, such as images) is not supported in this first version of the cache, it would certainly be a very useful feature to add in the nearby future. Raster support would probably require multi-resolution, which means that it would be necessary to manage multiple versions of the same data, one for each different resolution.

With regards to optimization, it would be interesting to study different formats to store spatial data in the database. For instance, the objects could be stored already in blocks, each block in a record of a table. New algorithms for theme blockage and spatial indexing could also be tried. Finally, there is a lot of space for research and improvement in cache management policies and heuristics. One good thing would be to make these policies and heuristics flexibility points so that each application could override the default and write some better suited for its specific use.

One more improvement could be a prefetcher component, who would try to predict what data would be requested to the cache in the near future and load it in advance. It could consider for instance all data around the area currently being visualized.

Since the main purpose of all this system is to render *maps* quickly using cache and GPU, it seems that it would fit perfectly in GIS servers that render lots of maps. By using proper cache management heuristics a server would quite likely improve its performance by making full use of all its resources.

Another challenge, this time much more difficult, would be to present the data from the *Terralib* database in a 3D navigator. Although the task may be done in a simple way just by laying the 2D geometries on a sphere surface, it would be far more interesting to create some sort of translators from a 2D object to 3D and place them in a real 3D environment with terrain modeling, sky, oceans and everything else to make it look nicer.

5. References

ATI (2002). “Radeon 9700”, http://www.ati.com/ .

Bar-Ze’ev, A. (2007). “How Google Earth [Really] works”, http://www.realityprime.com/articles/how-google-earth-really-works.

Câmara, G., Souza, R., Pedrosa, B., Vinhas, L., Monteiro, A., Paiva, J., Carvalho, M., Gattass, M. (2000). “TerraLib: Technology in Support of GIS Innovation”, II Brazilian Symposium on GeoInformatics, São Paulo.

Certain, A., Popovi´c, J., DeRose, T., Duchamp, T., Salesin, D. and Stuetzle, W. (1996). “Interactive multiresolution surface viewing. In SIGGRAPH 96 Conference Proceedings, pages 91–98.

ESRI (1998) “ESRI Shapefile Technical Description. An ESRI White Paper”, http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf.

Faconti, G. and Massink, M. (2000). “Continuity in human computer interaction**”**, *ACM SIGCHI Bulletin*, Vol. 32(4) - Sept./Oct.

Gamma, E., Helm, R., Johnson, R., Vlissides, J. (2005). “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley.

Google (2007) “Google Earth”, http://earth.google.com/.

INPE-DPI (2007) TerraView, http://www.dpi.inpe.br/terraview.

Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W. (2001). “Geographic Information System and Science”, John Wiley.

Matos, A., Gomes, J. and Velho, L. (1998). "Cache Management for Real Time Visualization of 2D Data Sets". In *Proceedings of SIBGRAPI 98*, pages 111–118. SBC - Sociedade Brasileira de Computacao, IEEE Press.

Mesa, A. F. (1998) “A History of the Graphical User Interface”, http://applemuseum.bott.org/sections/gui.html/ .

Nasa (2007) “World Wind”, http://worldwind.arc.nasa.gov/.

NVIDIA (2002) “GeForce FX”, http://www.nvidia.com/.

OpenGL (2007) “The Industry's Foundation for High Performance Graphics”, http://www.opengl.org/.

OGC - Open Geospatial Consortium (2007) “GML - the Geography Markup Language”, http://www.opengis.net/gml/.

Pinheiro, S., Celes, W and Velho, L. (2004). "Um sistema de cache preditivo para o processamento em tempo-real de grandes volumes de dados gráficos". In *SIBGRAPI Workshop of Thesis and Dissertations*.

Smith, S., Duke, D. (1999). “The Hybrid World of Virtual Environments”, Eurographics, Volume 18, Number 3.

S. Sun and X. Zhou, "Semantic Caching for Web-Based Spatial Applications", in Proceeding of APWeb 2005 (LNCS 3399), pages 783-794, March 29- April 1, 2005, Shanghai, China.

TDK (2007). “Terralib Development Kit”, http://www.tecgraf.puc-rio.br/tdk

Tilio, M. And Vera, M.S. (2005). “Desenvolvimento de Aplicativos com a Terralib”, In Bancos de Dados Geográficos, Edited by Casanova, M A, Câmara, G., Davis Jr, C.A., Vinhas, L. Queiroz, G.R., MundoGEO, pages 477-506.

Oosterom, P. V. (1999). “Spatial access methods. Chapter in Geographical Information Systems Principles, Technical Issues, Management Issues, and Applications”, edited by Longley, Goodchild, Maguire en Rhind, John Wiley pages 385-400 (vol.1), 1999.

Oosterom, P. V. (1988). “Spatial Data Structures in Geographic Information Systems”, Computer Science in the Netherlands, pages 463 - 477.

The Development of a Large Scale Geospatial Telecommunications Application Independent from GIS Proprietary Mechanisms

Eliane Z. Victorelli Dias, Geovane Cayres Magalhães, Grace Kelly de C. Silva

Fundação CPqD
Rodovia Campinas/Mogi-Mirim, Km118,5 – CEP 13086-902 – Campinas – SP – Brazil
{eliane,geovane,grace}@cpqd.com.br

Abstract *This article presents the evolution of a very large geospatial database and application developed for telecommunications outside plant management. The needs for the construction of this system and the decisions taken during its development and consolidation phases are described. New challenges found in securing and expanding this type of system in today's market will be listed, as will be the need for a solution independent from the geographical information system (GIS) proprietary mechanisms. The alternatives examined and the approaches chosen to address the main points regarding the system evolution have resulted in a guide to similar initiatives in the area.*

1 Introduction

GIS or geographical information systems were developed using proprietary technology until the mid 90's. The business secret was exactly in how spatial data was stored, recovered and processed [MELO JUNIOR; CANDEIAS, 2005].

Currently, an important strategy in the information technology industry is to become independent from proprietary technology. GIS technology's role has become more important to organizations, and in order to take full advantage of its capabilities, spatial data must be shared and systems must be interoperable. However, this is not an easy goal to achieve, and the concepts, standards and technologies used to implement GIS interoperability have continuously evolved.

At first, data sharing between organizations that had different GIS providers was limited to data conversion. It later evolved to using transfer standards and the utilization of files with open formats. Soon after, some GIS began offering API for direct reading. Finally, backed by several organizations, efforts were made to define standards for geographical information exchange on different levels. These standards apply to issues that go from data format to service integration over the Web.

The GIS products, as well as the applications that make use of them, have been developed on open standards in order to guarantee a high level of interoperability between platforms, databases, and programming languages [ESSID, 2004].

The aim of this article is to demonstrate the impact that the need for interoperability has on a high-end, GIS-based solution that is a standard in its market. It offers a general view of where efforts were concentrated to ensure interoperability in a process of improving a solution for managing telecommunications outside plant.

This article lays out the historical facts on the development of a telecommunications outside plant management system and how it was conceived in alignment with the trends seen in the geospatial research domain. It then goes on to describe the market demands that motivated stepping up to GIS independence and lists the main characteristics of the new solution. Finally, the main issues involved in the process of making the system GIS-independent are set forth, as well as the alternatives that were evaluated and the solutions that were adopted.

2 History

CPqD OSP is a step up from SAGRE, one of the most complete telecommunications outside plant management systems ever developed. The SAGRE system was created to assist the companies that made up the Telebras System in offering better quality services. The goal was to manage the Brazil's entire telecommunications outside network, offering support to planning, design and operation in a more efficient manner.

The regional operators had different cultures, which led to them using outside plants with different technologies, topologies and representations of their facilities on maps. The initial demand for offering support to functions such as design and records on a more consistent basis led to the system's concept of flexibility and reconfigurability.

Throughout its history, the development of SAGRE was pioneering in several ways. Magalhães (1996) described the technological innovations introduced at the system's concept phase. Since then, every large step ahead taken towards making improvements to the system were oriented by trend analysis and up-to-date surveys.

Given the context, the choice for the GIS platform to offer support to the solution was made based on an international bidding process with a strict benchmark for evaluation.

The first versions of the system indicated that the records and design functions required the development of several user interfaces. To increase productivity and make maintenance easier, metadata was used to automatically generate the interfaces [MAGALHÃES; OLIVEIRA; CUNHA, 1995].

Another requirement that set the system apart from others of its kind was the need for offering support to long term project activity transactions and short term operations transactions simultaneously, allowing users to access multiple views of the network. In order to offer support in such scenario, a complex mechanism for long transactions using versions had to be created [DIAS; GRANADO; MAGALHÃES, 1995].

Making it possible to transfer data from a large number of paper or raster maps onto computer storage was, at that time, and continues to be, a huge challenge [MAGALHÃES, 2007]. This process was traditionally based on the proprietary formats of the GIS used, which made its acceptance burdensome and complex. In order to shorten deadlines and reduce costs, a conversion model built on a high-level, business object oriented language was specified. The development of a method for bulk conversion founded on this format gave way to the participation of domestic companies in the process. The initiative also ensured that an excellent-quality database would be put together, as the same integrity rules supported by the application were secured in the bulk conversion.

At the start of the data conversion process, one of the companies showed interest in controlling all the service provision data such that it were integrated to their outside

plant records. To each new telephone installation was designated an ideal copper network using geographical criteria.

SAGRE then went on to become a critical mission system for Telesp, which was later acquired by Telefonica. Other operators, such as Sercomtel and GVT began similar procedures. Simultaneously, the design and inventory functions were upgraded to support Embratel's optical network architecture and to ensure that the Brasil Telecom and the others public phone regulation standards were met.

Integration of records data and that of service provision or other operational information is an excellent strategy, since it not only reduces operational process costs, but also allows the records database to be kept up-to-date on a constant basis.

The system is currently integrated to service order management, customer support and other systems, under environments with very different architectures. Previdelli e Magalhães [2002] describe some of the integration solutions taken, which go from file exchange queues to the use of message-oriented middleware or the provision of services in application server architecture.

Telefonica's outside plant is one of the largest existing spatial databases and includes the entire telephone network in the State of Sao Paulo. There are over 12,000,000 customers and 16,000,000 access lines in its records database. The provision of outside plant facilities is completely supported by the SAGRE system such that it is also integrated to several other operations support systems (OSS). This integration has lowered provision process costs by 40%. Over 200,000 service orders are processed monthly, and over 1,000,000 geographical inquiries are made [FARIAS; MAGALHÃES; PREVIDELLI, 2005].

Recently, management of project activities and the building of networks controlled by SAGRE were integrated into the company ERP. The integration has allowed for the costs of material and labor calculated automatically on project design to be used for the subcontractors payments. The phases involved in each endeavor are controlled in conjunction by the two systems. This way, the life span of projects has been reduced from weeks to days, and records updates have been ensured by eliminating backlog and as-built plants.

In early 2007, Telefonica received the Excellence Award for the Telecommunications Sector from GITA – Geospatial Information & Technology Association [GITA, 2007]. The award was granted in recognition to the quality and efficiency of integration and standardization of different operational procedures, which significantly brought down costs associated with these activities.

3 New Challenges

The new business opportunities for outside plant management products indicate that the market is more competitive than ever. Usually, the telecom companies have highly diversified requirements due to the wide variety of network technologies, services offered and business models found in telecommunications operators. Generally, in order for the system to meet the needs of a new customer, major changes have to be made to its features.

Another important factor in the sale of a product of this sort is the diversity in the magnitude of customers. There is a demand for both operators with millions of terminals and those with a few thousand. In light of this, the platforms used for small operators cannot be as costly as those that offer the power and performance necessary for a bigger customer.

On the other hand, current customers also upgrade their networks and expand their catalog of products and services, requiring major maintenance to all of their operating systems in order to evolve. Converged networks are becoming a reality and need to be managed by systems that allow operational procedures to become more agile and more efficiently utilize the facilities available at the service provider's plant. Networks are heading towards a new architecture with the goal of providing integrated services, something that is currently being called Quad Play, which are based on IP technologies to provide Internet, TV, voice, data and video transmission.

CPqD has oriented one of its lines of research, and consequently, the solutions obtained in its products, to fulfil the needs created by this scenario. The strategy established to keep the SAGRE system competitive and able to cater to the wide array of functional and non-functional demands was the development of a new version with a new architecture. The requirements that guided the definition of the new solution, the CPqD OSP were:

- Allowing for process distribution;
- Performing in accordance with different volumes of data;
- Supporting large variations in the number of users;
- Making integration with external systems easier;
- Yielding low costs for development and maintenance;
- Allowing for new features to be added quickly and;
- Being independent of the GIS platform, SGBD and application server.

4 CPqD OSP – The New Solution

4.1 Architecture

From the technological perspective, the CPqD OSP features the characteristics responsible for the success of the previous product but with new added concepts. As illustrated in Figure 1, CPqD OSP has an n-layer architecture which is based on JavaEE [JAVA, 2007], Open Geospatial Consortium, Inc ® standards [OGC, 2007], RBAC [SANDHU et al., 1996] and LDAP [WAHL et al., 1997], allowing for more flexibility and scalability.

Each layer plays well defined roles and communicates with each other using XML [XML, 2006]. This independence allows the replacement of one layer without affecting the entire architecture.

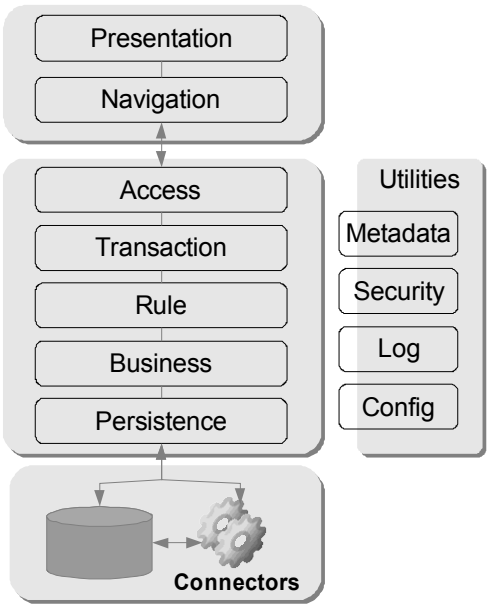


Figure 1: CPqD OSP Architecture

The Presentation and Navigation layers reside on the client side; they can be either a desktop client, web or mobile device. The other layers are on the server side. Access layer is responsible for user access control. Users with different profiles can access specific services and geographic areas. The Transaction layer controls the transaction that takes place when a service is requested, until it is committed or rolled back. Development is concentrated mainly on the Rule and Business layers - the Application Layer - where the business rules are implemented. The Persistence layer uses an enterprise data source concept for technology independent data access.

On top of this architecture was developed a metadata based framework that makes it possible to create solutions in a quicker and more standardized fashion. The goal of using this framework is bringing the focus of the efforts involved in the development to where they truly belong in the creation of a product: describing the domain objects and the rules that control this universe.

SAGRE already made use of metadata to generate its user interfaces, but the OSP solution intensified its use by applying it to data conversion, interface formats and business rules.

Regarding data conversion, Weiss and Dias (2004) defined the TOPML, which is a Geography Markup Language [OGC GML, 2004] application plan, designed to describe data pertaining to a telecommunications outside plant in which geographical information plays an important role.

4.2 Modules

The packaging of CPqD OSP is made up of several different modules (Figure 2) that can be used separately or put together into different combinations, which allow for better suiting each customer’s needs.

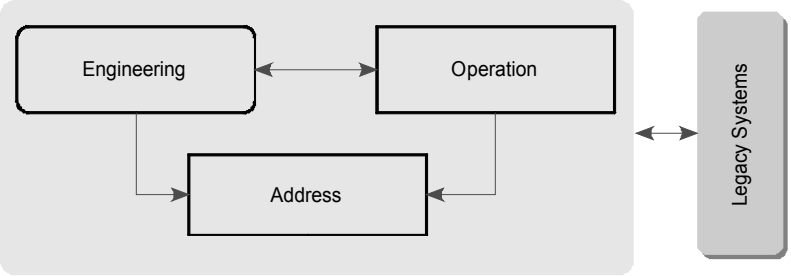


Figure 2: CPqD OSP Modules

The Address component is responsible for the business rules and the persistence of address objects. It offers a very important service of geocoding, which consists in determining a geographical coordinate of an address from a text description.

Also offered are interfaces that represent addressing objects to the outside world and that are utilized in different contexts that demand a wide variety of data recovery services from the component. These interfaces are based on OGC specifications.

In this case, special attention should be drawn to the use of the Filter Encoding Implementation Specification standard [OGC FE, 2005], which defines a format for XML-based filter expressions. These filter expressions restrict property values in order to create a subassembly of a group of objects.

The Engineering operates on top of the very popular Autocad Map CAD/GIS tool [AUTODESK, 2007]. This module covers the main use cases of inventory data maintenance for the outside plant. The inventory includes the elements of the copper, coaxial and optical networks, as well as the elements used in the basic infrastructure for building networks.

This module also offers support to the project activities for network maintenance and expansion. The functionalities provide support to the transitions between the stages in the life cycle of the project and different business and persistence rules are ensured for objects that are part of a project. The address component is used by the engineering module to add locations to network equipment.

The third module, Operation, covers the main use cases of network resource provisioning and trouble management. The business rules involved in registration and maintenance of the products and services are also supported by it. Reference to network information from the provisioning point of view is also offered, as is support to algorithms that determine the feasibility of providing a service to a specific address. Address components are used to associate a network facility to the customer's parcel.

The New Generation Operations Systems and Software (NGOSS) [NGOSS, 2007] specifications, which offers means to assist communications service providers in managing their businesses, including the SID model [SID, 2007] for shared information and the eTom map of telecommunications processes [eTOM, 2005] were widely studied and used as guidelines for the development of these services [SCHMIDT et al., 2007].

As is true for the SAGRE, integration between the Engineering and Operation modules is a factor that sets the CPqD OSP solution apart from others. It allows for provisioning operations to act on current information while viewing the proposed modifications. Aside from this, network modification operations preserve the integrity of the

provisioning data and can provide the information used for planning network cuts. Even though the modules are a part of the same solution, one of the new business requirements was that they offer the capability of being implemented independently.

The guidelines that oriented the development of CPqD OSP were independence from GIS and the use of open standards for interoperability. These requirements affect mainly the presentation and persistence layers. All of these features were highly challenging and will be discussed in detail in the next sections.

5 GIS Independence from Proprietary Mechanisms

On the path to a solution for outside plant management independent of vendors proprietary solutions, the biggest challenge faced was, without a doubt, GIS independence. This did not come as a surprise, as this solution is strongly founded upon spatial information. The impact of this requirement fell upon several important components of the solution, such as the geographical data model, long transaction control, the client application and the rendering mechanism. For every component, several alternatives were analyzed using cost, deadline and architectural criteria. A presentation of the main characteristics of this study follows.

5.1 Persistence Model

An important step for GIS independence consisted in adopting a persistence layer that made the application flexible so that it would support different geographical data models.

The use of a well-defined interface for manipulating geographical data makes the application independent of the means of persistence and changes made to the structure of the geographical data will not affect the application layer.

The spatial data interface defines a structure that describes a spatial object and standardizes access to this data type by the application.

Insertion, removal, modification and reference operations for any spatial object must be carried out in the persistence layer. These operations must read from the persistent means and generate an object that implements the spatial object interface and write onto the persistent means from a spatial object.

Independent alternatives were sought out for the spatial data interface and for the spatial data storage models.

5.1.1 Storage models for spatial data

The data model adopted presumes a geographical object’s data being isolated according to its type. The alphanumerical information and spatial information are stored each in different entities. This allows for several graphical replicas to be represented without creating redundancy of alphanumerical information in the database.

The application of outside plant management commonly resorts to several layouts. Usually, these layouts are in planar coordinate system and the land base data is stored in other spatial coordinate system due to large areas being covered.

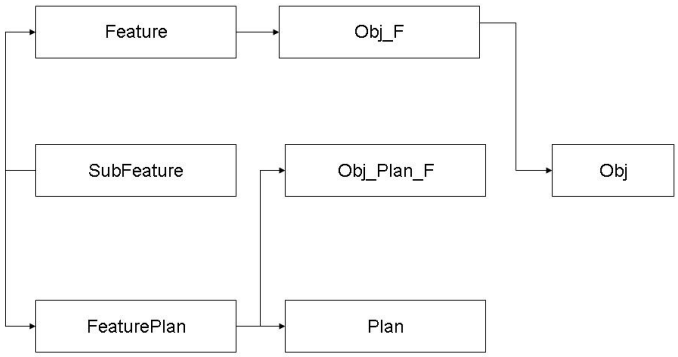


Figure 3: Data Model

5.1.2 Geographical data manipulation interface

For the representation of geographical data, the OGC specifications were quite natural and were followed in defining this interface. Some open source components offered OGC specifications, such as the GeoTools.

However, CPqD already had an interface implementation for the spatial data called SpatialObject, and some applications already made use of this interface. Implementation of SpatialObject made use of concepts very similar to those required for the generic interface. However, it would also be of interest to use the GeoTools library to read the data from the persistent means. A drawback is that the data read by GeoTools is stored in JTS component objects. The first alternative would be to create routines to convert a SpatialObject into a JTS component object and vice-versa. In this case, it would not be possible to develop objects with sub-features containing very complex geometries.

Re-using SpatialObject and making a few changes to its structure was also an option considered. The objects would store coordinates using JTS. It would then be possible to have sub-features with more complex geometries.

The second alternative allowed for more functionality. However, the first alternative was chosen, as it supported the use of sub-features, which was an important requisite, and lowered development costs. This decision was based on factors which indicated that future versions of GeoTools would no longer use JTS.

5.2 Controlling long transactions

The SAGRE solution supports a powerful long transaction mechanism based on version control. However, it is native of the GIS used. Since the goal of the CPqD OSP solution was to be independent from GIS, a new mechanism had to be implemented. The challenge was big, seen as the users would not like to be delivered a product with functionalities that were inferior to those offered by the previous version of the product.

The first alternative examined consisted in implementing all of the project control functionalities as well as version support, similarly to SAGRE. This would, however, be very expensive. In this context, the use of a few available version control components was examined. However, a mandatory requisite was that the project views could be made available in the main base during the engineering phase, before the project was

constructed. The components studied do not offer this facility and developing it based on version mechanisms supported elsewhere would be a risk.

An approach combining cost and offered functionalities was taken for the version mechanisms development. The choice made was to develop a simplified model of project modification control, with which every operation carried out on an object is stored in the same version. Dependencies between objects is less sophisticated than that developed for SAGRE; for example, when an object depends on another it is not possible to determine what change motivated the dependency.

However, the simplified mechanism meets the current needs; its development and maintenance were less complex and it can be easily upgraded in case functionalities offered by a control based on a transaction version becomes necessary.

5.3 Flexibilization of the Engineering Module Client Application

To put it in general terms, the client application carries out the mediation and control of the flow of information between the attribute manager, the map and the application server.

It features functionalities that retrieve geographical data from the map and sends it to the application server and vice-versa. In order to achieve this, a specific client application must be developed depending on the front-end used to represent the geographical data. This is therefore one of the solution's section that is affected most by the requisite of being independent from GIS.

The OGC defines the Web Map Service [OGC WMS, 2004] and Web Feature Service [OGC WFS, 2005] for geographical data exchange between a server and a client application.

The WMS service is a way to produce maps from geo-referenced data and carry out queries on its attributes from information coming in simultaneously from multiple heterogeneous remote servers. The map is the visual representation of the geospatial information, and not the information itself.

The WFS allows a client to retrieve and update geospatial data encoded in Geography Markup Language (GML) from multiple Web Feature Services. The specification defines interfaces for data access and manipulation operations on geographic features, using HTTP as the distributed computing platform. Via these interfaces, a Web user or service can combine, use and manage geodata -- the feature information behind a map image -- from different sources.

A standardized interface had to be defined for the client application just as for the other features of the OSP solution. Both of the specifications mentioned were used in different contexts.

5.4 Rendering

The process of generating a graphical representation of an object is also called rendering. In order for the GIS independence to be fulfilled, it was also necessary to seek out alternatives to the development of a feature rendering module that could be used alongside several graphical devices.

Apart from that, it is quite common for users of a GIS based solution to want to adapt symbols to the visual standards used in their organization. Therefore, the rendering solution also had to be prone to this kind of customization being made by users in their own environments.

It was observed that the drawing process had to be divided into at least two stages to represent objects in different devices, ensuring that the code of rules for rendering could be re-utilized after devices were replaced. The first stage would contain all of the rules on how to draw an object, with no connection to the device on which the drawing would be presented. It was necessary to assume the existence of a group of devices on which previously defined shapes referenced by names could be represented, even if the method for obtaining the representation differed between devices in the group. Therefore, it was presumed that simple entities such as text, lines, polygons and symbols could be represented on any graphical device.

The output of the first stage is a set of entities or drawing primitives called design logical entities, and are not associated to any graphical device. These entities have to be submitted to a process of translation in order to be displayed on the device in question. The translation module, which is device dependant, is called a driver.

Part of the rendering process is run on the server-end, while the rest is run on the client-end. Server processing is responsible for generating the design logical entities, such as defining lines, points, colors and other elements that make up an object. The client is then responsible for transforming logical entities into design entities and has the code specific to the device on which the drawing will be generated. The figure below illustrates how the rendering module works.

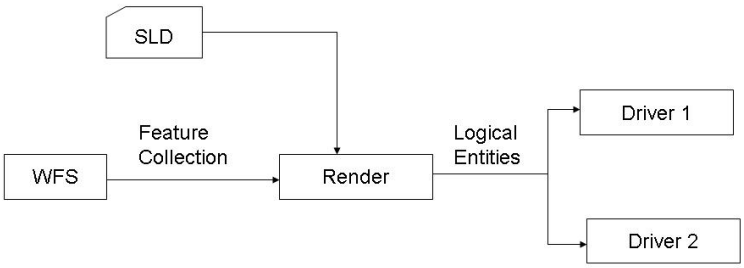


Figure 4: Rendering Module Components

The language adopted for the description of the logical entities was the Styled Layer Descriptor [OGC SLD, 2007], which is an OGC specification for the rendering of features. This standard can be used in any context where there is a need for flexibility on the way objects are displayed and allows for declarative customization.

This standard was chosen for having consolidated characteristics and for representing the results of the experiences of several georeferencing specialists. Besides this, there are open source implementations for its reading and rendering capabilities. However, SLD does not completely suit the needs of the project and an extension of the standard was developed.

Breaking away from the standard makes it difficult to use the tools available on the market that rely on SLD. However, the greatest advantage to this approach is the flexibility for altering post-implementation rendering aspects, enabling users to better customize their own symbols.

6 Conclusion

The changes made to an outside plant management system in order to make it GIS independent have been presented. This is a costly and demanding effort, but this project has proven it feasible.

It was also made clear that the creation of well defined interfaces outlining the functionalities strongly linked to GIS alongside the adoption of interoperability standards are generally the least expensive and most efficient way to achieve this goal.

7 Acknowledgments

A major portion of this article was obtained from internal reports written by the staff at the CPqD's Outside Plant Systems Management, which is comprised of dedicated and highly competent professionals. The technological evolution of a system of such magnitude taking place alongside support and maintenance of critical mission environments is only made possible by the brilliance and team effort of this group.

The authors wish to thank FUNTTEL (Fundo para o Desenvolvimento Tecnológico das Telecomunicações) for their support to this work through Triple Play Project.

8 References

AUTODESK (2007) Autodesk, Inc., Autocad Map. http://www.autodesk.com/autocadmap.

DIAS, E.; GRANADO, S.; MAGALHÃES, G. (1995) "Uso de Versões na Garantia de Consistência em Ambientes Mistos de Projetos e Operação" Anais do X Simpósio Brasileiro de Banco de Dados. Recife, p. 321 - 334.

ESSID, M.; BOUCELMA, O. (2004) "Mediated Geographic Web Feature Service. Anais do Brazilian Symposium On Geoinformatics", Inpe, p. 331 - 342.

eTOM (2005) "Enhanced Telecom Operations Map (eTOM) The Business Process Framework" GB921 v.6.1 - Release 6.0 edition TM Forum.

FARIAS, M.; MAGALHÃES, G.; PREVIDELLI, C. (2005) "Geospatial Database for Operations and Engineering" Oracle Spatial User Conference. http://www.oracle.com/technology/products/spatial/htdocs/spatial_conf_0503_idx.

GITA (2007) "Telefonica S.A. Receives GITA's Excellence Award" http://gita.org/resources/news/press_releases/3_2007-excellence_tsa.asp.

JAVA (2007) Sun Microsystems, Inc. http://www.sun.com/java.

MAGALHÃES, G.; OLIVEIRA, J.; CUNHA, C. (1995) "Métodos de Objetos para Construção de Interfaces Visuais Dinâmicas" Anais do IX Simpósio Brasileiro de Engenharia de Software. Recife, p. 143 - 158.

MAGALHÃES, G. (1996) “O Projeto SAGRE” In: CÂMARA, G.; CASANOVA, M.; HEMERLY, A. Anatomia de Sistemas de Inf. Geográfica. Campinas, Cap. 12.

MAGALHÃES, G. (2007) “A Better Way to Conversion: The Open, Incremental, GIS Free Approach” In: GITA´S ANNUAL CONFERENCE, 30., 2007, San Antonio.

MELO JUNIOR, J.; CANDEIAS, A. (2005) “SIG e sua interoperabilidade utilizando servidores de WEB” Anais XII Simpósio Brasileiro de Sensoriamento Remoto. Goiânia: Inpe, p. 2273 - 2280.

WAHL, M.; HOWES, T.; KILLE, S. (1997) "Lightweight Directory Access Protocol (v3)," IETF RFC 2251; http://www.ietf.org/rfcrfc2251.

NGOSS (2007) “New Generation Operations Systems and Software 6.0 Solution Suite” TM Forum http://www.tmforum.org/tech/NGOSS/1911.

OGC (2007) Open Geospatial Consortium, Org., OpenGIS Specifications. http://www.opengeospatial.org/standards.

OGC FE (2005) “Filter Encoding” Version 1.1 MA: Open GIS Consortium, Inc. < http://www.opengeospatial.org/standards/filter.

OGC GML (2004) “Geography Markup Language Encoding Specification” Version 3.1.1 Open GIS Consortium, Inc. http://www.opengeospatial.org/standards/gml.

OGC LS (2004) “Location Services: Core Services [Parts 1-5]” Version 1.0. MA: Open GIS Consortium, Inc. http://www.opengeospatial.org/standards/olscore.

OGC SLD (2007) “Styled Layer Descriptor Profile of the Web Map Service” Version 1.1 MA: Open GIS Consortium, Inc. http://www.opengeospatial.org/standards/sld.

OGC WFS (2005) “Web Feature Service” Version 1.1 MA: Open GIS Consortium, Inc. http://www.opengeospatial.org/standards/wfs.

OGC WMS (2004) “Web Map Service” Version 1.3 MA: Open GIS Consortium, Inc. http://www.opengeospatial.org/standards/wms.

PREVIDELLI, C.; MAGALHÃES, G. (2002) “Moving Geospatial Applications Towards a Mission Critical Scenario” Procedings of GITA 2002 Tampa.

SANDHU, R.; COYNE, E.J.; FEINSTEIN, H.L.; YOUMAN, C.E. (1996). "Role-Based Access Control Models". IEEE Computer, P. 38-47. IEEE Press.

SCHMIDT Simone et al.(2007) “Mapping OSS Products Using the eTOM Business Processes Framework” Proceedings Of The International Workshop On Telecommunications, Santa Rita do Sapucaí, p.58-65.

SID (2007) “SID Solution Suite” GB922 & GB926 NGOSS Release 7.0 TM Forum http://www.tmforum.org/page32555.

WEISS, G.; DIAS, E. (2004) “Moving Telecom Outside Plant Systems Towards a Standard Interoperable Environment” Lecture Notes In Computer Science, Springer Berlin / Heidelberg, v. 2004, n. 3124, p.1246-1251, 28.

XML (2006) “Extensible Markup Language” Version 1.1 W3C. http://www.w3.org/TR/xml11/.

Towards a Geographic Ontology Reference Model for Matching Purposes

Guillermo Nudelman Hess[1], **Cirano Iochpe**[1,3], **Silvana Castano**[2]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

[2]DICo – Università degli Studi di Milano
20135 Milano – Italy

[3]Procempa – Empresa da Tecnologia da Informação e Comunicação de Porto Alegre
Porto Alegre – RS – Brazil

{hess,ciochpe}@inf.ufrgs.br, castano@dico.unimi.it

Abstract. *Ontologies and geographic ontologies are becoming important research and application fields for the geographic information systems (GIS) community. Although geographic ontologies (also known as spatial-temporal ontologies, geo-ontologies or geospatial ontologies) are becoming popular, a standard and complete model is still missing. The attempts for establishing standards are yet incipient, i.e., do not fulfill the actual needs. In this paper we propose a reference model for developing geographic ontologies, with the specific purpose of matching. The main idea is to extend the model for conventional, non-geographic, ontologies to make it suitable for describing the particularities of the GIS data and the relationships among them.*

Resumo. *Ontologias e, mais especificamente, ontologias geográficas, estão se tornando um importante campo de aplicação e pesquisa relacionado à comunidade de Sistemas de Informações Geográfica (SIG). Embora ontologias geográficas (também conhecidas por ontologias espaciais, ontologias espaço-temporais, geo-ontologias ou ainda ontologias geo-espaciais) vêm tornando-se populares, falta ainda um modelo completo e que seja adotado como padrão. As tentativas para estabelecimento de padrões são ainda incipientes, ou seja, não satisfazem completamente as necessidades atuais. Neste artigo nós propomos um modelo de referência para o desenvolvimento de ontologias geográficas, com o propósito especial de* matching *(casamento). A idéia principal é estender o modelo para ontologias não-geográficas, de modo a fazê-lo adequado para descrever as particularidades dos dados espaço-temporais e os relacionamentos entre eles.*

1. Introduction

Ontologies and geographic ontologies are becoming important research and application fields for the geographic information systems (GIS) community. Due to the particularities of the GIS data - geometry and location [Fonseca et al. 2003], and, eventually, temporal properties as well [Sotnykova et al. 2005], besides the usual descriptive attributes - a simple alphanumeric ontology (called here conventional ontology) is not expressive

enough. It is like in the database field, where there are special types of databases for geographic data, called geographic databases, because conventional databases were not designed for holding these features. The ability to build proper geographic ontologies will facilitate their integration and, subsequently, will advance semantic interoperability, which has been acknowledged as a primary concern in geographic information science nowadays [Tomai and Kavouras 2004].

According to Spaccapietra et al. [Spaccapietra et al. 2004] space and time can meet ontologies in three different ways: (1) as the spatial domain specifying space, spatial elements and spatial relationships, or as the temporal domain, specifying time, temporal elements and temporal relationships; (2) as the implicit background to an application domain that relies on geographical data or; (3) to enrich the description of the concepts in the ontology, to represent their spatial and temporal location, in the same way spatio-temporal data models support the description of spatial and temporal features in spatio-temporal databases.

Although ontologies are being widely used by the GIS community, there is still a lack for an actual geographic ontology model (also know as spatial ontology, geo-ontology, spatio-temporal ontology or geospatial ontology). That is, the ontologies proposed and used at the moment are designed for conventional (descriptive), non-spatial purposes and the particularities of the geographic data, such as the geometry, temporality and topological relations are missing or poorly described. There are already some standard proposals (ISO 19109 and GML OWL encoding), but they have some limitations in terms of expressiveness, validation or easiness to use or extend. In other words, the attempts for establishing standards are yet incipient, i.e., do not fulfill the actual needs. The main limitation of these proposals is that they do not really hold the semantics of a geographic ontology. Instead, they basically define the syntax and the names for some geographic elements.

There are many fields in which geographic ontologies can be applied, such as building a common ground for describing the geographic phenomena, spatial reasoning, semantic annotation of maps, geographic information integration and retrieval, and so on. In this paper we are proposing a geographic ontology model specially designed for the purpose of geographic ontology matching, which is quite different from conventional ontology matching [Hess et al. 2007b]. This ontology is part of a wider project, in which we are developing a methodology for matching geographic ontologies at both the concept-level [Hess et al. 2006] and the instance-level [Hess et al. 2007a].

The reminder of this paper is structured as follows. In Section 2 we present some related work which try to define geographic ontology models. Our proposal for a geographic ontology reference model is presented in Section 3. Then, in Section 4 we show an example of an ontology built based on our reference model. Finally, conclusions and future work are discussed in Section 5.

2. Related Work

Maedche and Staab [Maedche and Staab 2000] state that an ontology should comprise the following: (a) Concepts, (b) the Lexicon, (c) Relations and (d) Axioms. Concepts are an integral part of an ontology as they stand for mental things of all possible things [Tomai and Kavouras 2004]. The Lexicon comprises the descriptions of the con-

cepts, i.e., their definition in natural language. The semantic relations link two concepts in hypernym/hyponym relation and in the meronym/holonym relation as well. The relation as semantic properties refer to the properties of the concepts in the ontology. The axioms refer to constraints imposed on concept or relations.

Tomai and Kavouras [Tomai and Kavouras 2004] extend Maedche and Staab's definition of ontology by defining the components of a geographic ontology. They basically create some semantic properties to be associated to a concept when it represents a geographic concept: Spatiality, Temporality, Nature, Material/cover, Purpose and Activity. The first two are the ones that actually characterize a geographic ontology. Spatiality covers the relative spatial properties of the concept, such as topology, location, and the internal spatial properties, such as size and shape. Temporality is divided into time (period or instant) and condition/status.

Casati, Smith and Varzi [Casati et al. 1998] separate a geographic ontology in two parts: objects and relations. The geographic objects are specialized into physical, such as mountains, rivers and forests, and human, such as countries, cities, and so on. A geographic object is composed by a number of descriptive attributes and by a border. The relations can be of type mereology, location or topology. In a mereology association, a geographic object A *is part of* a geographic object B. The location relation associates a geographic concept with a set of coordinates, and a topology relation spatially associates two geographic concepts. Souza et al. propose an ontology to represent contextual information in geospatial data integration [Souza et al. 2006]. The ontology is composed by 5 contexts, as the authors present. Each one of it stores some kind of information. The main two are the *DataContext* and *AssociationContext*. The *GeospatialEntity* is the main concept of the *DataContext*, and contains the properties for geometric representation, location and some metadata. The *AssociationContext* has the information about the spatial association of the concepts and the semantic associations (degree of similarity) as well [Souza et al. 2006]. As weak points of these works we can point the absence of temporal aspects and the impossibility of representing non-geographic concepts.

Fu et al. [Fu et al. 2005] developed a geo-ontology restricted to geographic places, such as cities, countries, districts and so on. Each concept is described in terms of its names (can be multiple), geometry (called footprints by the authors) and some metadata. Furthermore, each place may be related to another by only one relation, the *containment relation*. Kolas et al. [Kolas et al. 2006] propose an architecture for *Geospatial Semantic Web* [Egenhofer 2002]. They define 6 ontologies, and one of them, called *Base Geospatial Ontology* is of interest in the context of this paper. It forms the ontological foundation of geospatial information by mapping some GML's elements to OWL, in order to link the geographic data with knowledge outside the geospatial realm [Kolas et al. 2006].

SWETO-GS [Arpinar et al. 2006] is a spatio-temporal ontology with three dimensions, namely thematic, spatial and temporal. The thematic dimension contains the concepts of a general domain such as people, places and organizations, or for a specific domain such as travel and transport. In that dimension there are both geographic and non-geographic concepts. The geospatial dimension stores the spatial data and relationships. The concepts are described in terms of their coordinates, translated from the thematic dimension. The temporal dimension stores the temporal relations that may occur between concepts. Finally, some metadata can be associated to the SWETO-GS ontology.

Bittner and Smith propose an ontological theory which contains resources to describe geographic processes and the concepts that participate therein [Bittner and Smith 2003]. For that purpose two (sub-)ontologies are presented, one describing the concepts with their properties, called SNAP, and one describing the processes and their parts and aggregates, called SPAN. SNAP entities are described in terms of their properties, spatial relations and conventional relations, while SPAN entities are described also considering time.

3. The reference model

Any ontology can be defined as a 4-tuple $O =< C, P, I, A >$, where C is the set of concepts, P is the set of properties, I is the set of instances, and A is the set of axioms [Scharffe and de Bruijn 2005][1]. A concept $c \in C$ is any real world phenomenon of interest to be represented in the ontology and is defined by the term t that is used to nominate it. The name of a concept is given by the unary function $t(c)$. A property $p \in P$ is a component that is associated to a concept c with the goal of characterizing it, but is defined outside the scope of a concept. It can be a data type property, which means that its range is a data type, such as string, integer, double, etc. or an object type property, meaning that the allowed range values are other concepts. A data type property can be viewed as a database attribute, while an object type property is like a database relationship.

The context of a concept $c \in C$ is defined as the set of properties P (each one given by the unary function $p(c)$) related to it, as well as by the set of axioms A, representing the generalization/specialization relations as well as the restrictions (each one given by the unary function $x(c)$). Formally, the context of a concept can be defined as:

$$ctx(c) =< t(c), \{p(c)\}, \{x(c)\} >$$

An instance $i \in I$ is a particular occurrence of a concept c, with values for each property p associated to the concept and an unique identification. Thus, an instance may be defined as

$$i =< t(c), t(i), VP >,$$

where $t(c)$ is the concept being instantiated, $t(i)$ is the instance unique identifier (name) and VP is the set of values for the properties belonging to the context of the instantiated concept.

At last, an axiom describes an hierarchical relationship between concepts, or provides an association between a property and a concept (through the property domain or through a concept restriction), or associates an instance with the concept it belongs.

3.1. Geographic concept

A spatio-temporal object (STOBJ) as defined by Xu et al. [Xu et al. 2006] has spatial and temporal properties as well. The former encompass geometries and the spatial relationships such as distance, position, topological, and so on. Temporal properties are,

[1]This definition is based on the OKBC model [Chaudhri et al. 1998]. In the original work, instead of P (properties) it was R (relations)

basically, instant and period. Based on these properties, a spatio-temporal ontology is a normative system describing spatio-temporal objects and relationships between them.

Following this premise, a conventional ontology is not expressive enough to handle the particularities of geographic phenomena. Thus, we define a geographic ontology, which is an extension of a conventional ontology. It is also a 4-tuple $O =< C, P, I, A >$, where C is the set of concepts, P is the set of properties, I is the set of instances, and A is the set of axioms.

A concept $c \in C$ is classified into domain concept, such as a *River*, a *Park* or a *Building*; geometry concept, such as *Point*, *Line* or *Polygon*; or time concept, specialized in *instant* and *period*. Furthermore, a geographic domain concept gc is a specialization of a domain concept representing a geographic phenomenon, as depicted in Figure 1. A geographic domain concept is defined as being a domain concept with an axiom saying that it must be associated to, at least, one geometry concept, through a geometric relationship property, which is explained in the following. The geometry plays a fundamental role on defining the possible spatial relationships the concept may have.

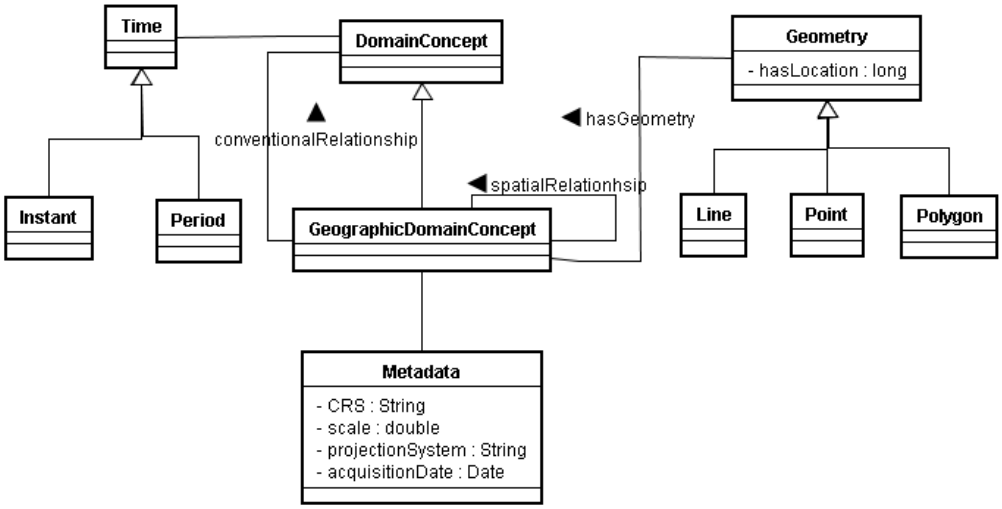


Figure 1. Types of concepts of the geographic ontology reference model

3.2. Properties in a geographic ontology

In an ontology a property can be defined by itself, i.e., outside the context of a concept. However, for matching purposes, a property is relevant when associated to a concept, directly in its domain or through a restriction. For this reason the property is always considered into the context of a concept. In a geographic ontology, each property $p \in P$ can be of one of five possible types: conventional, spatial relationship, geometric relationship, positional or temporal. Formally, it can be defined as a 4-tuple

$$p =< t(p), pd, minCard, maxCard >,$$

where $t(p)$ is the function which gives the property's name, pd is the property domain (detailed in the following) and $minCard$ and $maxCard$ are, respectively, the property's minimum and maximum cardinalities.

A conventional property may be even a data type property or an object type property. In the first case it represents an attribute of a domain concept. In the second case it represents an association between a domain concept (geographic or not) with a non-geographic domain concept, which we call conventional relationship (cr).

An attribute $a \in P$ is a special type of property to which the minimum and maximum cardinalities are not relevant, and the domain is a data type (dtp), such as string, integer, and so on. Thus, it can be simplified as a tuple of the form

$$a =< t(p), dtp >$$

A conventional relationship, on the other hand, is a property $p =< t(p), pd, minCard, maxCard >$ with the restriction that the property's domain pd is a domain concept, identified as $t(c_x)$. Furthermore, the concept identified by $t(c_x)$ cannot be a geographic domain concept, i.e., $cr = (p \in P|(c_x : \neg gc))$

A spatial relationship property sr (topological, directional or metric) is always an object type property $p =< t(p), pd, minCard, maxCard >$, and represents an association between two geographic domain concepts, i.e, can appear only in the context of a geographic domain concept. The spatial relationships have a pre-defined semantics and are already standardized in the literature [Egenhofer and Franzosa 1991] and by the Open GIS Consortium (OGC). Formally, we define a spatial relationship as $sr = (p \in P|(c_x : gc))$

A geometric relationship property ge (always an object type property $p =< t(p), pd, minCard, maxCard >$) is an association between a geographic domain concept with a geometry concept geo, i.e., $ge = (p \in P|(c_x : geo) \wedge minCard = 1)$

A positional property pos is a data type property that must be associated to a geometry concept, to give its location (set of coordinates).

Finally, a temporal relationship property tr is an association between a domain concept and a time concept, i.e., $tr = (p \in P|(c_x : time))$

These relationships allow one to answer queries such as:

- With which instances i_x a given instance i has borders;
- Which concepts gc_x may cross the concept gc;
- How far one instance i is from an instance i_x.

3.3. Geographic and geometry concepts as axioms

The set of axioms A describes the hierarchical (IS-A) relationships between concepts as well as provides associations between properties and concepts, and relates instances with the concepts they belong to. A hierarchy $h \in A$ is a binary relation of type $h(c, c_x)$, where c_x is the superclass of the concept c.

With the definitions above, we can now formally define a geographic concept gc through a restriction axiom, as:

$$gc = (c \in O|\exists p \in P \wedge p : ge \wedge t(p) = "hasGeometry" \wedge minCard = 1),$$

where ge is the geometric property.

A geometry concept can also be formally defined as a concept with a restriction axiom stating it must have associated at least one positional property pos.

$$geo = (c \in O | \exists p \in P \wedge p : pos \wedge t(p) = "hasLocation" \wedge minCard = 1),$$

where pos is the positional property.

3.4. Geographic region and instance

An instance of a concept is defined by the property values associated to a concept. An instance $i \in O$ is a triple of the form $i =< t(c), t(i), VP >$, where

- $t(c)$ is the name of the concept being instantiated.
- $t(i)$ is the unique identifier of an instance (instance name).
- VP is the set of values for the properties. Each one of the elements is represented by the binary function $vp(t(p), val)$, where $t(p)$ is the property name and val is the value associated to the property for that instance.

A geographic instance gi is an extension of an instance i. As a geographic instance must be associated to, at least, one instance of a geometry concept, the value of the positional property (*hasLocation*) gives the spatial position (coordinates) of that geographic instance. A geographic instance $gi \in O$ is, thus, a 4-tuple of the form
$gi =< t(c), t(i), VP, vMD >$, where

- $t(c)$ is the name of the concept being instantiated.
- $t(i)$ is the unique identifier of an instance (instance name).
- vP is the set of values for the properties. Each one of the elements is represented by the ternary function $vp(t(p), val)$, where $t(p)$ is the property name and val is the value associated to the property for that instance.
- vMD is the set of metadata values associated to the instance. Each one of the metadata values is represented by a binary function $vmd(t(md), val)$, where $t(md)$ is the metadata and val is the value set for that geographic instance.

Georeferencing is the set of geographic coordinates of the vertices or planar coordinates in a given coordinate system. Additionally, it has the information of the cartographic projection. It applies only to the geographic instances, not to the concept definitions. The georeferencing information is stored by the metadata, and is thus given to an instance by the vMD component (association holding between a geographic instance gi and the metadata).

The set of instances of a concept c is given by I, and thus $i \in I$. Figure 2 shows graphically the types of instances we can have in our ontology and how one relates to the other. It is important to notice, however, that *GeographicRegion*, *RegionRepresentation* and *Metadata* are not concepts described in an ontology, but concepts belonging to the reference model for matching purposes.

As it is possible to infer, $I =< R, \{i\} >$, where R is a new ontology element we are introducing into our ontology. It represents the region covered by the set of instances stored in the ontology. Furthermore, it generalizes the metadata values associated to the instances. The *GeographicRegion* plays an important role in the matching process for which this reference model is designed to. Basically, the two main reasons for creating the notion of geographic region are:

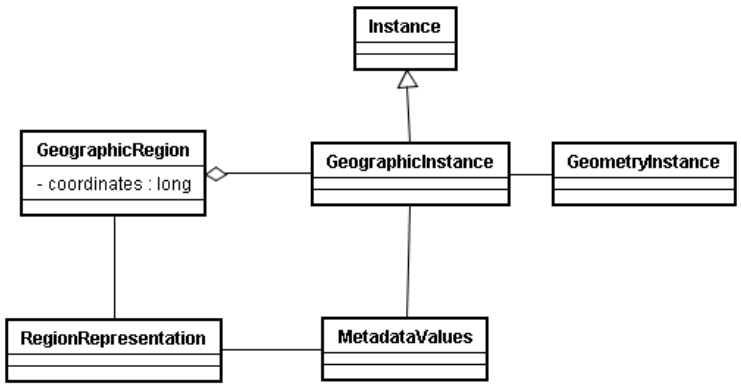


Figure 2. Types of instances of the geographic ontology reference model

- To create the notion of region similarity, and, as a consequence, to measure how similar are two ontologies not at the instance granularity, but at the instance set granularity;
- To accelerate the process of instance similarity assessment by eliminating the pairs of instances which are geographically disjoint.

3.5. Metadata

As already stated, the $metadata$ class and its instantiation $metadaValues$ do not represent concepts and instances defined by the ontology designer. Instead, they provide additional information about them, such as the coordinate reference system, the projection scale, the data's capture date, among others. These information is crucial in the matching process, in order to avoid incorrect interpretations due to differences on the metadata. For example, a concept that may be associated to a geometry specialization of $point$ in a low detailed scale, such as 1:500.000. However, the same concept may be associated to a $polygon$ concept if in a more detailed scale, such as 1:25.000. The same applies to instances. The values for an instance's coordinates vary if are described using $< latitude, longitude >$ reference system or if they are described using $polar$ coordinate reference system.

3.6. Operations

In the reference model we are proposing in this research two operations may be performed over an ontology: the creation/insertion of concepts and the definition of hierarchical relationships between concepts.

As our target is the matching of two geographic ontologies, it may happen that at least one of the input ontologies contains only the explicit definition of the instances, i.e., without the explicit declaration of the concept they instantiate and its structure. In this case new concepts are added to the input ontology through the information extracted from the instances. The concept name is obtained from the $t(c)$ component of the instance triple $i =< t(c), t(i), VP >$ (or 4-tuple $gi =< t(c), t(gi), VP, vMD >$ in case of a geographic instance) and the properties are given by the union of the properties the instances have values for.

Once the concepts are defined from their instances, it is possible to define their names ($t(c)$) and properties (attributes and relationships between concepts as well), but

not the hierarchy among them. For that purpose, the operation of taxonomy definition is performed, consisting on searching into a reference ontology, also defined according to the reference model proposed here, to identify the hierarchical relationships among the concepts created for the input ontology.

4. Example of an ontology based on the reference model

In this section we present a geographic ontology we developed based on the reference model we proposed. The example, however, does not exploit all the expressiveness power of the reference model, specially in terms of the temporal aspects. Furthermore, the metadata do not appear explicitly in the ontology. Figure 3 graphically depicts the concepts and the instances of the ontology, as well as the properties associated to the concepts. It is important to notice that both geographic and non-geographic concepts can be defined using the reference model.

The rectangles with continuous lines represent concepts, the ellipses the properties representing attributes associated with a concept and the dashed rectangles the instances belonging to a concept. The arcs linking two concepts correspond to the properties which represent relationships holding between them, while the isa labeled arrows are the taxonomic relationships (axioms) between two concepts, in which one is the specialization of the other.

Figure 4 presents the encoding of the example ontology described in a generic language, somehow structurally similar to description logics (DL). However, it is important to clarify that the syntax is completely different from DL and is not from any existing language. We try to use the DL format with a natural language syntax, just to formalize a little the ontology.

5. Conclusions and future directions

Ontologies are becoming the standard mean to describe resources to be shared with semantics. With geographic information is not different. Many ontologies are being proposed. However, due to the absence of a wide accepted standard reference model for a spatio-temporal ontology, the comparison of the concepts and instances of these ontologies is a very hard, time consuming and error prone task. As these ontologies may be defined using different models, before matching their concepts and instances, it is necessary to identify the corresponding elements used in their definition and how they relate to each other. In other words, a meta-matching of the ontologies' models is needed.

To solve this gap, we proposed here a geographic reference ontology model, for the specific purpose of geographic ontology matching. By extending the formally defined model it is possible to create concepts, hierarchies, properties and associate them to concepts and instances, just as for a conventional, non-geographic ontology. Then, we defined some properties and axioms specific for the geographic field, such as geometry, spatial relationships (topology, directional and metric), spatial position and temporality. Finally, we established the association between instances and metadata, which are fundamental information for the geographic data. The presented example showed how to use the reference model to define a geographic ontology.

As future works we plan to develop a more sophisticated ontology based on the reference model, in order to make use of all its potentiality, including geographic concepts

and non-geographic concepts, temporal properties and more spatial relationships. We also plan to use more the metadata component of the instances. Furthermore, the current reference model allows the definition of temporal concepts, but not temporal properties. We thus plan to extend it to support temporality for properties.

6. Acknowledgments

This research is founded by the Brazilian agency CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

References

Arpinar, B., I, Sheth, Amit, Ramakrishnan, Cartic, Usery, L., E, Azami, Molly, Kwan, and Mei-Po (2006). Geospatial ontology development and semantic analytics. *Transactions in GIS*, 10(4):551–575.

Bittner, T. and Smith, B. (2003). Granular spatio-temporal ontologies. In *Proceedings of the AAAI Spring Symposium on Foundations and Applications of Spatio-Temporal Reasoning (FASTR).*

Casati, R., Smith, B., and Varzi, A. C. (1998). Ontological tools for geographic representation. In Guarino, N., editor, *Formal Ontology in Information Systems*, pages 77–85. IOS Press.

Chaudhri, V. K., Farquhar, A., Fikes, R., Karp, P. D., and Rice, J. (1998). OKBC: A programmatic foundation for knowledge base interoperability. In *AAAI/IAAI*, pages 600–607.

Egenhofer, M. J. (2002). Toward the semantic geospatial web. In Voisard, A. and Chen, S.-C., editors, *ACM-GIS*, pages 1–4. ACM.

Egenhofer, M. J. and Franzosa, R. D. (1991). Point set topological relations. *International Journal of Geographical Information Systems*, 5:161–174.

Fonseca, F. T., Davis, C. A., and Camara, G. (2003). Bridging ontologies and conceptual schemas in geographic information integration. *GeoInformatica*, 7(4):355–378.

Fu, G., Jones, C. B., and Abdelmoty, A. I. (2005). Building a geographical ontology for intelligent spatial search on the web. *Databases and Applications (DBA2005)*, pages 167–172.

Hess, G. N., Iochpe, C., and Castano, S. (2006). An algorithm and implementation for geoontologies integration. In *Prof. of VIII Brazilian Symposium on Geoinformatics (GEOINFO'06)*.

Hess, G. N., Iochpe, C., and Castano, S. (2007a). Geographic ontology matching with ig-match. In Kollios, G., Papadias, D., and Zhang, D., editors, *SSTD*, Lecture Notes in Computer Science. Springer.

Hess, G. N., Iochpe, C., Ferrara, A., and Castano, S. (2007b). Towards effective geographic ontology matching. In *Second International Conference on GeoSpatial Semantics (GeoS 2007).*, Lecture Notes in Computer Science. Springer.

Kolas, D., Dean, M., and Hebeler, J. (2006). Geospatial semantic web: architecture of ontologies. In *2006 IEEE Aerospace Conference*.

Maedche, A. and Staab, S. (2000). Semi-automatic engineering of ontologies from text. In *Proceedings of the Twelfth International Conference on Software Engineering and Knowledge Engineering (SEKE'2000)*.

Scharffe, F. and de Bruijn, J. (2005). A language to specify mappings between ontologies. In Chbeir, R., Dipanda, A., and Yétongnon, K., editors, *SITIS*, pages 267–271. Dicolor Press.

Sotnykova, A., Cullot, N., and Vangenot, C. (2005). Spatio-temporal schema integration with validation: A practical approach. In Meersman, R., Tari, Z., Herrero, P., Mendez, G., Cavedon, L., Martin, D., Hinze, A., Buchanan, G., Perez, M. S., Robles, V., Humble, J., Albani, A., Dietz, J. L. G., Panetto, H., Scannapieco, M., Halpin, T. A., Spyns, P., Zaha, J. M., Zimanyi, E., Stefanakis, E., Dillon, T. S., Feng, L., Jarrar, M., Lehmann, J., de Moor, A., Duval, E., and Aroyo, L., editors, *OTM Workshops*, volume 3762 of *Lecture Notes in Computer Science*, pages 1027–1036. Springer.

Souza, D., Salgado, A. C., and Tedesco, P. A. (2006). Towards a context ontology for geospatial data integration. In Meersman, R., Tari, Z., and Herrero, P., editors, *OTM Workshops (2)*, volume 4278 of *Lecture Notes in Computer Science*, pages 1576–1585. Springer.

Spaccapietra, S., Cullot, N., Parent, C., and Vangenot, C. (2004). On spatial ontologies. In *VI Brazilian Symposium on GeoInformatics (GEOINFO 2004)*.

Tomai, E. and Kavouras, M. (2004). From "onto-geonoesis" to "onto-genesis": The design of geographic ontologies. *GeoInformatica*, 8(3):285–302.

Xu, W., Huang, H.-K., and Liu, X.-H. (2006). Spatio-temporal ontology and its application in geographic information system. In *Fifth International Conference on Machine Learning and Cybernetics*, pages 1487–1492.

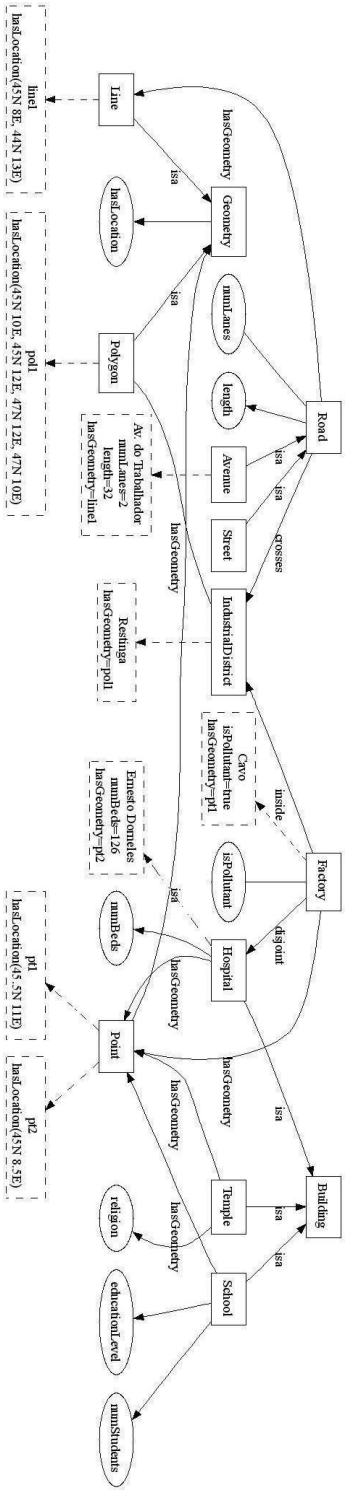


Figure 3. An example of ontology O

C = $Road, Street, Avenue, Factory, IndustrialDistrict, Building,$
$Hospital, Temple, School$ (domain)
$Geometry, Line, Polygon, Point$ (geometry)

P = $numLanes, length, isPollutant, numBeds, religion,$
$educationLevel, numStudents$ (conventional)
$disjoint, crosses, inside$ (spatial)
$hasGeometry$ (geometric)
$hasLocation$ (positioning)

A = $isa(Polygon, Geometry)$
$isa(Point, Geometry)$
$isa(Line, Geometry)$
$isa(School, Building)$
$isa(Hospital, Building)$
$isa(Temple, Building)$
$isa(Street, Road)$
$isa(Avenue, Road)$
$hasGeometry(Road, Line)$
$hasGeometry(IndustrialDistrict, Polygon)$
$hasGeometry(Factory, Point)$
$hasGeometry(Temple, Point)$
$hasGeometry(School, Point)$
$hasGeometry(Hospital, Point)$
$crosses(Road, Some(IndustrialDistrict))$
$disjoint(Hospital, Factory)$
$inside(Factory, Some(IndustrialDistrict))$
$isPollutant(Factory, boolean)$
$educationLevel(School, String)$
$numStudents(School, integer)$
$religion(Temple, String)$
$numBeds(Hospital, integer)$
$numLanes(Road, integer)$
$length(Road, double)$
$hasLocation(Geometry, double)$

I = $instanceOf(pol1, Polygon)$
$hasLocation(pol1,' 45N, 19E, 45N12E, 47N12E, 47N10E')$
$instanceOf(pt1, Point)$
$hasLocation(pt1,' 45.5N11E')$
$instanceOf(pt2, Point)$
$hasLocation(pt2,' 45N8.5E')$
$instanceOf(line11, Line)$
$hasLocation(hasLocation,' 45N8E, 44N13E')$
$InstanceOf(AvenidaDoTrabalhador, Avenue)$
$numeLanes(AvenidaDoTrabalhador, 2)$
$hasGeometry(AvenidaDoTrabalhador, line1)$
$instanceOf(Restinga, IndustrialDistrict)$
$hasGeometry(Restinga, pol1)$
$instanceOf(Cavo, Factory)$
$isPollutant(Cavo, true)$
$hasGeometry(Cavo, pt1)$
$instanceOf(ErnestoDorneles, Hospital)$
$numBeds(ErnestoDorneles, 111l)$
$hasGeometry(ErnestoDorneles, pt2)$
$inside(Cavo, Restinga)$
$crosses(AvDoTrabalhador, Resting)$

Figure 4. Example of ontology defined according to the proposed reference model

Approximate String Matching for Geographic Names and Personal Names

Clodoveu A. Davis Jr.[1], Emerson de Salles[1]

[1]Instituto de Informática – Pontifícia Universidade Católica de Minas Gerais
Rua Walter Ianni, 255 – 31980-110 – Belo Horizonte – MG – Brazil
clodoveu@pucminas.br, hemer80@gmail.com

Abstract. *The problem of matching strings allowing errors has recently gained importance, considering the increasing volume of online textual data. In geotechnologies, approximate string matching algorithms find many applications, such as gazetteers, address matching, and geographic information retrieval. This paper presents a novel method for approximate string matching, developed for the recognition of geographic and personal names. The method deals with abbreviations, name inversions, stopwords, and omission of parts. Three similarity measures and a method to match individual words considering accent marks and other multilingual aspects were developed. Test results show high precision-recall rates and good overall matching efficiency.*

1. Introduction

The problem of matching strings allowing errors and other kinds of discrepancies has been studied for some time (Navarro 2001), but still presents some interesting and challenging issues. Its importance is growing, since large volumes of textual data have become available through the Internet. Applications of approximate string matching nowadays include some important areas in computing, such as information retrieval, digital libraries, ontology integration and computational biology. In many of those applications, such tools are needed whenever the input data are uncertain, semi-structured, or simply reflect the various views people have on their surrounding environment.

Approximate string matching techniques are frequently required in geotechnologies and in applications that have to deal with much semantic uncertainty. Consider, for instance, the response of a person to the simple question "Where do you live?" Depending on the context of the conversation, the answer may be the name of a city, a state, a country, or even an address, with details such as a house number and a postal code. Anyway, the description certainly includes references to one or more place names, which can be ambiguous and contain subtle and misleading errors. Many ambiguities result from the reuse of the names of places that are famous elsewhere ("Paris, Texas", as opposed to "Paris, France"), while errors may be caused by spelling difficulties (place names that include foreign words or proper nouns), language or local particularities (phonetic alphabets and use of accent marks). String matching is also needed in geographic ontology integration, geographic information retrieval (Jones, Purves et al. 2002; Borges, Laender et al. 2007), and other semantic-related subjects.

Our interest in approximate string matching lies on the applications that use place names and their variations. Since personal names are frequently used as place names,

our interest extends towards them as well[1]. Research on retrieval of personal names and proper nouns is quite active, with many recent works (Barcala, Vilares et al. 2002; Cohen, Ravikumar et al. 2003; Patman and Thompson 2003; Minkov, Wang et al. 2005; Christen 2006). Previous work on gazetteers (Souza, Davis Jr. et al. 2005), geographic information retrieval (Borges, Laender et al. 2003; Delboni, Borges et al. 2007), geocoding (Davis Jr., Fonseca et al. 2003) and address matching (Davis Jr. and Fonseca 2007) has shown that the use of approximate matching algorithms can be an important asset whenever the input data have been manually fed or obtained directly from natural language text. This work, therefore, presents string matching techniques that seek better results and more flexibility when dealing with geographic and personal names.

The remainder of this paper is organized as follows. Section 2 introduces the approximate string matching problem in more detail and discusses existing techniques. Section 3 studies specific aspects of matching personal and geographic names, and presents our approach to the problem. Section 4 shows experimental results. The paper is concluded in Section 5, which also presents some of our priorities for ongoing work.

2. Problem statement and related work

The problem of approximate string matching is a traditional one in computer science. For recent surveys of initiatives on this subject, see (Navarro 2001) and (Christen 2006). In an informal way, approximate string matching corresponds to finding out if a given string *S* matches a pattern *P*, within a given similarity threshold δ. This threshold can be given as a maximum number of allowable errors, or as a normalized measure, a number between 0 and 1, with 0 meaning a mismatch and 1 meaning an exact match.

Since we are interested in personal and geographic names, our review of related work pays more attention to techniques that work better when both the string and the pattern are short. Some approximate string matching techniques and algorithms are more suited to seek the presence of a short pattern in a (presumably long) text. Given the objective of this paper, we will not discuss this variation further. However, we only point out that this kind of matching is important for geographic information retrieval, in algorithms that try to determine the geographic context in a natural-language text, based on landmark names and on expressions that indicate locality (Borges, Laender et al. 2007).

More formally, approximate string matching can be stated as in Definition 1.

> ***Definition 1.*** *Let S and P be strings of characters, i.e. sequences of symbols from a finite alphabet* Σ. *P is called the pattern, against which S will be compared. We say that S matches P when* $f(S,P) < \delta$, *where* $f : \Sigma \times \Sigma \rightarrow [0,1]$ *is a distance function, which quantifies the similarity between S and P and* δ *is a similarity threshold.*

In other words, *f* indicates how close *S* is from *P*. Several metrics have been proposed for the similarity, depending on the algorithm (Cohen, Ravikumar et al. 2003).

There are two main types of approximate matching techniques: *phonetic matching* and *pattern matching*. Phonetic matching considers the similarity of letters in a string as to the usual sounds they produce in English, and generates a code from any given word. In

[1] An assessment of the street names catalog for the city of Belo Horizonte, Brazil, showed that about 67% of the names have more than one word. Of those, about 65% are references to personal names.

general, similarly sounding words produce equal codes, but subtle spelling differences can keep phonetic matching from identifying many similarities. With these and other limitations, some studies (Zobel and Dart 1995; Christen 2006) have shown that phonetic matching is systematically outperformed by pattern matching techniques as to its matching efficiency, even though phonetic methods tend to run faster. There are also hybrid methods, combining string distance measures and probabilistic interpretations of pattern matching results (Pfeifer, Poersch et al. 1996; Cohen, Ravikumar et al. 2003).

Pattern matching relies on comparing characters from each string, to detect points in common and to quantify their similarity based on that. Some of the most important pattern matching techniques are based on edit distance. One of such techniques (Levenshtein 1965) determines the total number of operations (insertions, deletions, or substitutions) required to transform *S* into *P*. This number is called the *edit distance* or the *Levenshtein distance* (*LD*) between *S* and *P*. Considering that each operation has a cost of 1, similarity can be calculated as

$$f_{LD}(S,P) = 1.0 - \frac{LD(S,P)}{\max(|S|,|P|)} \tag{1}$$

Variations of LD consider the transposition of characters as another operation, with unit cost, or costs that depend on similarity criteria applied to individual characters (Navarro 2001). For instance, for optical character recognition, the comparison between similar letters such as “i” and “l” can have a lower cost. For manual input, a lower substitution cost can be assigned to pairs of letters that are adjacent on the keyboard.

The Levenshtein edit distance algorithm is usually implemented using a matrix $L[|S|+1,|P|+1]$ of integers, filled out in a row-wise traversal to the right, as in Eq. 2:

$$L_{i,j} = \begin{cases} L_{i,0} = i \\ L_{0,j} = j \\ if\_S_i = P_j\_then\_L_{i,j} = L_{i-1,j-1} \\ if\_S_i \neq P_j\_then\_L_{i,j} = \min(L_{i-1,j}, L_{i,j-1}, L_{i-1,j-1}) + 1 \end{cases} \tag{2}$$

As a result, $LD(S,P) = L_{|S|,|P|}$. If the “+1” clause at the last row of equation 2 is changed to a function, it is possible to consider varying costs according to the type of operation or to the similarity between pairs of characters.

Another similarity metric, which is not based on an edit distance model, has been proposed by Jaro and later refined by Winkler (Jaro 1995; Winkler 1999). The Jaro-Winkler algorithm is bases on the number of common characters and the order in which they appear, increasing the similarity value in case there is a match on the initial characters of the string. We decided not to use this metric, partly because some works have shown that their results are better suited to matching short strings (for instance, last names) (Cohen, Ravikumar et al. 2003) and because the Levenshtein distance would suit our ideas for multiword strings better. There is, however, an adaptation of the Winkler technique in which all possible permutations of the words in a multiword string are considered (Christen 2006). Our method can deal with inversions, but is able to avoid performing the permutations, as we will show next.

3. Matching Personal and Geographic Names

Matching personal or geographic names can be defined as the process of determining, within a given level of certainty, whether two strings correspond to the same person or place. Matching names is a challenging task, mainly because of spelling variations and some widely used practices, such as abbreviation. This is further complicated by the adoption of different ordering and by the varying importance assigned to parts of the name, based on cultural differences (Patman and Thompson 2003). Such variations usually keep exact matches from being effective. Spelling variations are common in personal names, both in given names and surnames, as well as in cross-language adaptations of place names. For instance, London is referred to as "Londres" in Portuguese, while "Lisboa" is referred to as "Lisbon" in English. Spelling variations also arise from the transliteration of words and names from different alphabets, as in Chinese, Russian, and other languages, to the Roman alphabet. Another source of variations is the intentional abbreviation and inversion of names, as in the indication of authorship in bibliographic references. In such sources, parts of names are also occasionally omitted.

Spelling variations often cause problems when, for instance, names are dictated over the phone for manual input. Christen (2006) includes both spelling variations and manual input among the primary sources of errors in personal names. Errors on common words can be detected using a dictionary, but detecting errors on names requires a different approach. In spite of spelling and presentation format difficulties, personal names are frequently used to designate places. Even though we do not have detailed data on that, we suspect, from personal observation in Brazil and abroad, that personal names are present on a large share of urban place names. Naming geographic features after people as a form of homage is an established tradition, hence names such as *Magellan strait*, *Weddell sea*, or *Hudson bay*, and, more recently, names assigned to features in other planets, such as *Tycho crater*, on the Moon, or *Maxwell Montes*, on Venus.

We observe that personal and geographic names share some common characteristics. Words are usually small, ranging from three or four characters up to no more than a hundred characters. Special characters, such as accent marks, are used depending on the language, and sometimes are omitted. Abbreviations are common, mostly on middle names. Words can occasionally be inverted or even omitted. Stopwords (such as "of" in English, or "de" in Portuguese) are often present in full names, but are frequently omitted in practice. Titles or professional descriptions are used along with personal names, usually preceding them (as in "Presidente Kubitschek"), and titles can also be abbreviated ("Pres. Vargas"). Such observations are mostly from practice, but we have been able to confirm most of them during our experiments on preliminary test data.

In the next section, we will present our approach to matching personal and geographic names, in which we address all of the above characteristics, by adapting existing techniques and adding heuristics of our own.

3.1 Matching single words

For single words matching, we implemented a variation of the Levenshtein edit distance method. Two word strings are considered to match if their similarity measure f is greater than a threshold δ (Eq. 1). We can determine whether this similarity can be reached using two pre-tests in sequence. If the difference in length of S and P exceeds

the allowed number of errors, no further comparison is necessary, and *S* cannot match *P*. In other words, if $abs(|S|-|P|)/\max(|S|,|P|) > 1.0-\delta$ there is a mismatch on the basis of the *length test*. The other test is based on the *bag distance* (Bartolini, Ciaccia et al. 2002), a linear-time test that generates a lower bound for the edit distance. If we put together two sets of individual characters, *X* and *Y*, each of which formed with the character from *S* and *P*, respectively, then $bag(S,P) = \max(|X-Y|,|Y-X|)$, where the minus sign indicates the set difference operation. It has been shown that $bag(S,P) \le LD(S,P)$, and therefore we can discard the possibility of a match whenever $bag(S,P)/\max(|S,|P|) > 1.0-\delta|$ (*bag test*) (Bartolini, Ciaccia et al. 2002). The bag test allows us to discard mismatches before determining the actual edit distance at a much higher computational cost, as shown next.

Our variation of the Levenshtein edit distance calculation consists in incorporating a practical scheme for matching accent-marked letters and special characters. Characters are organized into groups, so that characters belonging to the same group are considered to match, as formally described in Definition 2.

> ***Definition 2.*** *Let c_1 and c_2 be two characters from the alphabet Σ, and let g_1, g_2, …, g_n be n sets of at least 1 character, in which $g_i \cap g_j = \varnothing, \forall i \neq j$. We say that c_1* matches *c_2* under *g_i when there is a group g_i which contains both c_1 and c_2, i.e.,* $c_1 \overset{g_i}{\equiv} c_2 \Leftrightarrow g_i \supset \{c_1, c_2\}$

Thus, equivalent characters are organized into groups, as shown in Table 1. This strategy allows us to prepare several sets of groups, according to the requirements of a matching effort and the characteristics of source and pattern data. For instance, case-sensitive groups, phonetic groups, or accent-mark-sensitive groups can be prepared and used with no code modification. Figure 1 shows examples of case- and accent-mark-sensitive (a) and insensitive (b) matching using the Levenshtein matrix, in the comparison of the names of two Brazilian cities, Paranaguá (PR) and Paranapuã (SP).

Table 1 - Groups of characters

Id	*Group*
L1	a, á, ã, à, ä, â
L2	e, é, è, ë, ê
L3	i, í, ì, ï, î
L4	o, ó, õ, ò, ö, ô
L5	u, ú, ù, ü, û
L6	n, ñ
L7	c, ç
...	one group for each consonant (lowercase)

Id	*Group*
U1	A, Á, Ã, À, Ä, Â
U2	E, É, È, Ë, Ê
U3	I, Í, Ì, Ï, Î
U4	O, Ó, Õ, Ò, Ö, Ô
U5	U, Ú, Ù, Ü, Û
U6	N, Ñ
U7	C, Ç
...	one group for each consonant (uppercase)

		P	a	r	a	n	a	g	u	á
	0	1	2	3	4	5	6	7	8	9
P	1	0	1	2	3	4	5	6	7	8
a	2	1	0	1	1	2	2	3	4	4
r	3	2	1	0	1	2	3	3	4	5
a	4	3	1	1	0	1	2	3	4	4
n	5	4	2	2	1	0	1	2	3	4
a	6	5	2	3	1	1	0	1	2	2
p	7	6	3	3	2	2	1	1	2	3
u	8	7	4	4	3	3	2	2	1	2
ã	9	8	4	5	3	4	2	3	2	**1**

(a) f_{LD} = 0.89

		P	a	r	a	n	a	g	u	á
	0	1	2	3	4	5	6	7	8	9
P	1	0	1	2	3	4	5	6	7	8
a	2	1	0	1	1	2	2	3	4	5
r	3	2	1	0	1	2	3	3	4	5
a	4	3	1	1	0	1	2	3	4	5
n	5	4	2	2	1	0	1	2	3	4
a	6	5	2	3	1	1	0	1	2	3
p	7	6	3	3	2	2	1	1	2	3
u	8	7	4	4	3	3	2	2	1	2
ã	9	8	4	5	3	4	2	3	2	**2**

(b) f_{LD} = 0.78

Figure 1 - Accent-mark-insensitive (a) and sensitive (b) edit distance

In the definition of the groups and in the implementation, we used the Unicode standard set of characters. Notice that most information retrieval and text mining efforts usually pre-process the input strings, eliminating uppercase characters, accent marks and other special characters. We decided not to do so, since uppercase is often used as a way to distinguish proper nouns, and since accent marks can be decisive in determining a match, depending on the language.

One possible difficulty in using our method is the determination of the similarity threshold δ. We can understand more easily how to choose a value for δ if we think in terms of number of allowable errors. Considering *S* and *P* to have the same length, the maximum (integer) number of allowable errors is equivalent to $\lfloor (1.0 - \delta) \cdot |S| \rfloor$. In order illustrate that, we performed a frequency distribution analysis of word lengths in a data set containing 9,222 personal names, extracted from BDBComp (Brazilian Digital Library on Computing)[2]. After separating 26,924 words from names, we observe that most names have between two and four words. Almost 75% of them have between 4 and 9 characters (Table 2). The large number of 1-character words reflects the use of abbreviations in BDBComp. Therefore, using a threshold $\delta = 0.75$, it means we allow for a maximum of one error in a 4- to 7-letter word, and 2 errors for a 8- to 11-letter word. This threshold seems adequate for personal names, and is left here as a suggestion. In our method for multiword matching, presented in the next section, this threshold applies to individual words taken separately, not to the full string.

Table 2 – Frequency distribution of BDBComp name lengths and number of words

Length	# names	%
1	4383	16,3
2	1377	5,1
3	469	1,7
4	2080	7,7
5	4688	17,4
6	4727	17,6
7	4638	17,2
8	2587	9,6
9	1281	4,8
10	426	1,6
11	182	0,7
12	52	0,2
13	19	0,1
14	8	0,0
15	7	0,0
TOTAL	26924	100,0

# words	#names	%
2	3894	42,2%
3	2922	31,7%
4	1760	19,1%
5	553	6,0%
6	81	0,9%
7	11	0,1%
8	1	0,0%
TOTAL	9222	100,0%

3.2 Matching multiword strings

The first step for matching multiword strings is dividing them into words, using whitespace characters as delimiters, such as blanks, hyphens and other symbols. Points are preserved as the last character in the preceding word, since they can indicate abbreviations. We can also opt to preserve or to eliminate stopwords. Stopwords constitute another way to differentiate between very similar names, and therefore we prefer to preserve them in most situations. However, some sources intentionally leave them out, as in the case of names in bibliographic references, so our implementation allows treating or discarding stopwords as an option. Our matching strategy then proceeds in four phases: (1) checking for standard abbreviations, (2) checking for non-standard abbreviations, (3) word-by-word matching and (4) verifying inversions.

[2] http://www.lbd.dcc.ufmg.br/bdbcomp/bdbcomp.jsp

First, each word in the string is tested against a list of known abbreviations. The list contains pairs of the type <*abb*, *val*>, where *abb* is the standard abbreviation (for instance, "Pres"), and *val* is its meaning, spelled completely (as in "President"). If an *exact* match is found between a word in *S* and an abbreviation from the list, it is replaced by its full spelling. Our intention is to expand abbreviated titles, which are quite common preceding personal names and in some kinds of place names, to their full description. We do not expect to find many false matches, i.e., words that coincide with standard abbreviations but have a meaning of their own (as in someone whose name is "Pres"). The possibility of such coincidences should be assessed by the user, who could then leave conflicting abbreviations out of the list.

Next, non-standard abbreviations are verified. Candidates are 1-character capitalized words and words that end with a point. Such abbreviations are compared to each word in the pattern, and a similarity measure is then calculated as the number of matching characters of the abbreviation $S[i]$ divided by the number of characters in the candidate word from the pattern $P[j]$, where $X[k]$ denotes the kth word of string *X* (Eq. 3).

$$f_{NSA}(S[i], P[j]) = \frac{|S[i]|}{|P[j]|} \tag{3}$$

Unless the similarity threshold is set too low, any non-standard abbreviations in *S* will not find a match with regular names. We assume, heuristically, that the abbreviation has been used in order to save space or typing effort; therefore, we expect a large difference in size between the abbreviated word and its expected match in the pattern. On the other hand, it is likely that a non-standard abbreviation will reproduce the first characters from the corresponding word. We consider this case to be a match if all the characters in the abbreviated word are *equal* (i.e., no approximation is allowed) to the same number of characters at the beginning of the pattern word, which is where the characters that form an abbreviation are usually taken from. Even though in most cases name abbreviations involve simply an initial, with this heuristic we expect to be able to match unusual abbreviations or abbreviations that have been left out of the standard list.

In the third stage, we perform word-by-word matching, using a strategy that is similar to LD calculation (Eq. 2), modified to allow for inversions and to provide a similarity measure. Our matching algorithm uses a matrix $W[|S|_W, |P|_W]$, where $|X|_W$ denotes the number of words in string X. The matrix is filled out in a row-wise traversal to the right, making $W_{i,j} = f_{LD}(S[i], P[j])$ if $S[i]$ is a regular name, or $W_{i,j} = f_{NSA}(S[i], P[j])$, if $S[i]$ is a non-standard abbreviation. The value of f_{LD} is determined using the process described in the previous section. When $S[i]$ is a name, after each row *i* is complete, we identify the column *j* at which the value of the similarity function is maximum. If this value exceeds the similarity threshold δ, a match exists between $S[i]$ and $P[j]$. For the processing of the next row, the word $P[j]$ is left out of the similarity comparisons if the match was exact. At the end of the process, we select the best match for each word in the pattern, considering a valid match only when (1) the similarity threshold has been reached, or (2) there is a match with a non-standard abbreviation.

Denoting as *v* the number of matching words, we propose three similarity measures for multiword string matching. The first, f_{MW}, is calculated dividing the sum of the similarity values found for each matching word by the number of matching words, giving an

idea about the average similarity of words in each string (Eq. 4). The second, $f_{VM,}$ indicates the fraction of the words from the pattern for which a match has been found (Eq. 5). The third measure, f_{INV}, indicates the occurrence of inversions, and is calculated as follows. The order in which words from the string match words from the pattern is generated and analyzed, counting the number of times in which the sequence is broken. The f_{INV} similarity is then calculated by establishing a penalty for each inversion, corresponding to the number of inversions (n_I) divided by the number of matching words (Eq. 6).

$$f_{MW}(S,P) = \frac{\sum_{j=1}^{|P|_W} \left[\max_{i=1}^{|S|_W} \left(f_{LD}(S[i],P[j]),\ f_{NSA}(S[i],P[j]) \right) \right]}{v} \tag{4}$$

$$f_{VM}(S,P) = \frac{v}{\max\left(|S|_W , |P|_W \right)} \tag{5}$$

$$f_{INV}(S,P) = 1.0 - \frac{n_I}{v} \tag{6}$$

The similarity values can be used separately or combined with a weighted average. Weights are assigned according to the characteristics of the matching effort. For instance, when matching full names to bibliographic references, inversions are expected, so the inversion index can receive a lower weight than the other two measures. Equation 7 shows the calculation of the overall similarity f, where w_{MW}, w_{VM}, and w_{INV} are respectively the weights for word, valid matches, and inversions, and $w_{MW} + w_{VM} + w_{INV} = 1$.

$$f(S,P) = w_{MW}\, f_{MW}(S,P) + w_{VM}\, f_{VM}(S,P) + w_{INV}\, f_{INV}(S,P) \tag{7}$$

Figure 2 shows the comparison of two names, considering δ = 0.75, case- and accent-mark-sensitivity. In the first row, the only the first words match, with a similarity of 0.857 (one error in seven characters). The other words from the pattern do not match the first word from the string; a similarity measure does not have to be calculated, since the comparisons fail either the length test or the bag test. Further comparisons only have to be made on "Antônio", first word from the pattern, if an exact match has not been found. In the second row, "C." is a non-standard abbreviation, and is compared against each word from the pattern. A match occurs with "Carlos", for which "C." is a possible initial. However, "Carlos" is not taken out of future comparisons, since a better match can occur with some other word. In the remaining two rows, the edit distance has to be calculated only twice. Since all words from the pattern found a match, f_{VM} = 1.0, and f_{MW} = 0.706, the average of the values in bold in Figure 2. Matches are in order (1<2<3<4), so f_{INV} = 1.0 also.

	Antônio	**Carlos**	**de**	**Souza**
Antonio	**0.857**	*	**	*
C.	**	**0.167**	0.000	0.000
de	**	**	**1.000**	**
Sousa	*	*	***	**0.800**

(*) discarded: bag test; (**) discarded: length test; (***) discarded: previous match

Figure 2 -- Multiword string matching example

Figure 3 shows a similar example. The value for f_{MW} is 0.664, indicating a penalty for the unmatched second name. The value for f_{VM} is now 0.75, for three matches out of four words. No inversions are found (1<3<4), so f_{INV} = 1.0.

	Antônio	Carlos	de	Souza
Antonio	**0.857**	*	**	*
Coelho	*	*	**	*
de	**	**	**1.000**	**
Sousa	*	*	***	**0.800**

(*) discarded: bag test; (**) discarded: length test; (***) discarded: previous match

Figure 3 – Another multiword string matching example

Figure 4 shows an example in which stopwords have been omitted from the string, but are present on the pattern. There are also inversions, as in a bibliographic citation. The value of f_{MW} is 0.278, a low value caused by the uncertainty associated with the matching of the initials. Since there are again three matches in four words, f_{VM} is 0.75. One inversion is found (4>1<3), so $f_{INV} = 0.67$.

	Antônio	Carlos	de	Souza
Sousa	**	*	**	**0.800**
A.	**0.143**	0.000	0.000	**
C.	0.000	**0.167**	0.000	0.000

(*) discarded: bag test; (**) discarded: length test; (***) discarded: previous match

Figure 4 -- Matching with inversions

Similarity values can be used to rank the strings against the pattern, either individually or by combining the measures, as in Equation 7. Table 3 shows the overall similarity values for the three examples, considering equal weights for w_{MW}, w_{VM} and w_{INV}.

Table 3 – Similarity values compared

	f_{MW}	f_{VM}	f_{INV}	f
Antonio C. de Sousa	0.706	1.000	1.000	**0.902**
Antonio Coelho de Sousa	0.664	0.750	1.000	**0.805**
Sousa, A. C.	0.278	0.750	0.670	**0.566**

4. Experimental results

We performed two experiments to assess the efficiency of the proposed method. One compared a bibliographic database to a list of personal names from a Web page, and the other compared manually input street names to an official thoroughfare catalog.

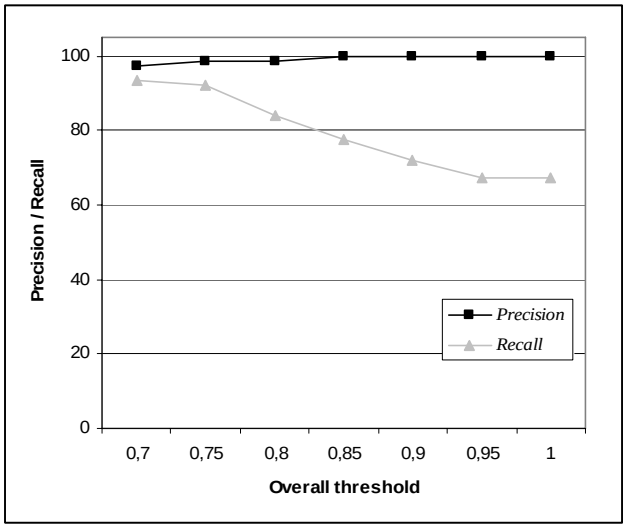
4.1 First experiment: personal names

A list of 85 personal names of Brazilian researchers on Computer Science has been extracted from a Web page in which the results of a grant bid (Edital 01/2007[3]) were published. Names from this list were compared against 9,222 author names from the already mentioned Brazilian Digital Library on Computing (BDBComp), which currently holds data on over 5,200 works published in national journals or conferences. Since both sources contain names of people from the same research community, we expected many matches, i.e., people that received a grant would probably have some paper published in a Brazilian journal or conference. Since CNPq names probably came from the Brazilian national curriculum vitae database, we expected full correctness, because names are ultimately input by the researchers themselves. Manual inspection showed

[3]http://efomento.cnpq.br/efomento/divulgacao/divulgacaoResultados.do?metodo=propostas&codigoLinhaFomento=58&seqChamada=17&idComite=CC

several minor and possibly intentional accent mark oversights, but no abbreviations. In BDBComp, however, abbreviations are common, since it is a bibliographic database.

CNPq names were matched against BDBComp names under the following setup: (1) inversions were not considered (f_{INV} = 1.0 in all comparisons), (2) stopwords, accent-mark- and case-sensitivity were enabled, and (3) δ = 0.85. An overall similarity value *f(S,P)* was calculated as the simple mean of f_{MW}, f_{VM} and f_{INV}. Matches found were then manually checked for false positives and false negatives. Results were summarized using the precision and recall metrics from Information Retrieval (Baeza-Yates and Ribeiro-Neto 1999). Figure 5 shows a precision-recall graphic with *f(S, P)* varying from 0.70 to 1.00. Precision indicates the percentage of correct matches within all matches obtained, while recall indicates the percentage of expected matches that was achieved by the method. Notice that, as the threshold increases, precision rises (i.e., only closer matches are accepted), but recall drops.



f(S, P) threshold	Precision (%)	Recall (%)
0,70	97,3	93,4
0,75	98,6	92,1
0,80	98,5	84,2
0,85	100,0	77,6
0,90	100,0	72,0
0,95	100,0	67,1
1,00	100,0	67,1

Figure 5 - Precision/recall results

Considering *f(S, P)* as a similarity measure, we allowed as many matches names within BDBComp as possible, within the threshold. With this, in most cases multiple matches were obtained. In the lower threshold experiments, some names correctly matched as many as seven BDBComp names, indicating the occurrence of many spelling and abbreviation variations of the same person's name in BDBComp. Figure 6 shows the variation of the average number of correct and incorrect matches per CNPq name. These results indicate that our method could be used to cluster variations of the same name in BDBComp, thus improving the results of author queries to the digital library.

f(S, P) threshold	Correct	Incorrect
0,70	2,77	0,89
0,75	2,77	0,41
0,80	2,38	0,05
0,85	2,08	0,00
0,90	1,69	0,00
0,95	1,22	0,00
1,00	1,20	0,00

Figure 6 – Average number of correct and incorrect matches per name

4.2 Second experiment: geographic names

In a second experiment, we compared a set of a hundred street names against Belo Horizonte's official thoroughfare catalog. The street names were randomly selected from a list of 4,700 manually typed records, as part of a data collection effort. Street names from the list included many problems for geographic name matching, such as abbreviations, omission of parts, and misspellings. All selected street names and were manually geocoded by technicians from PRODABEL, the municipal IT company, who have much experience with local place names. They were allowed to use other data to resolve ambiguities manually. As a result, 87 of the 100 street names were recognized, while the remaining 13 either were unrecognizable, or from nearby cities.

Street names were then matched against the thoroughfare catalog using our method, with case- and accent-mark insensitiveness, stopwords and abbreviations considered, inversions allowed, and $\delta = 0.85$. We used the full street name string, including the thoroughfare type ("Rua", "R.", "Av.", and so on). Under these parameters, 62 of the 87 names were matched correctly, and only 4 were incorrect matches. We achieved a precision of 94% and a recall of 71%, which is similar to the results in Figure 5, but parsing out the thoroughfare type should lead to better results. However, if we considered only the street name, and allowed for case and accent marks, only 32 correct matches would remain, and the recall rate would drop to 35%. Allowing for approximate word matches, inversions, and abbreviations doubled the recall rate in this experiment.

5. Conclusions and future work

Approximate string matching for personal and geographic names is an important and useful technique, with applications in geographic information science, information retrieval, and other areas. We are particularly interested in using the proposed method for multilingual gazetteers, geocoding, geographic information retrieval, and records linkage. The adaptations we propose for individual word matching considering case- and accent-mark-sensitivity can also be used in applications such as multilingual ontology integration and natural language processing.

Our experiments, although preliminary, have demonstrated the validity of the proposals and ideas presented in this paper. We consider this line of work promising, even though more tests with a larger volume of data are required, in order to adequately assess the computational efficiency of the method and to compare it to other proposals. An experiment in records linkage, involving large volumes of data from the health sector, is being prepared. The integration of the techniques presented in this paper to a previous work (Davis Jr. and Fonseca 2007) is also planned.

References

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). Modern Information Retrieval, Addison-Wesley.

Barcala, F. M., Vilares, J., Alonso, M. A., Graña, J. and Vilares, M. (2002). Tokenization and proper noun recognition for information retrieval. 13th International Workshop on Database and Expert Systems Applications, 246-250.

Bartolini, I., Ciaccia, P. and Patella, M. (2002). String Matching with Metric Trees Using an Approximate Distance. Proceedings of the 9th International Symposium on String Processing and Information Retrieval, Springer-Verlag.

Borges, K. A. V., Laender, A. H. F., Medeiros, C. B., Silva, A. S. and Davis Jr., C. A. (2003). The Web as a Data Source for Spatial Databases. V Brazilian Symposium on GeoInformatics (GeoInfo 2003), CD-ROM, Campos do Jordão (SP).

Borges, K. A. V., Laender, A. H. F., Medeiros, C. M. B. and Davis Jr., C. A. (2007). Discovering geographic locations in Web pages using urban addresses. 4th Workshop on Geographic Information Retrieval, to appear, Lisbon, Portugal.

Christen, P. (2006). A Comparison of Personal Name Matching: Techniques and Practical Issues. Sixth IEEE International Conference on Data Mining - Workshops, 290-294, Hong Kong, IEEE.

Cohen, W. W., Ravikumar, P. and Fienberg, S. E. (2003). A Comparison of String Distance Metrics for Name-Matching Tasks. Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web, 73-78.

Davis Jr., C. A. and Fonseca, F. T. (2007). "Assessing the Certainty of Locations Produced by an Address Geocoding System." Geoinformatica 11(1): 103-129.

Davis Jr., C. A., Fonseca, F. T. and Borges, K. A. V. (2003). A Flexible Addressing System for Approximate Urban Geocoding. V Brazilian Symposium on GeoInformatics (GeoInfo 2003), CD-ROM, Campos do Jordão (SP).

Delboni, T. M., Borges, K. A. V., Laender, A. H. F. and Davis Jr., C. A. (2007). "Semantic Expansion of Geographic Web Queries Based on Natural Language Positioning Expressions." Transactions in GIS 11(3): 377-397.

Jaro, M. A. (1995). "Probabilistic linkage of large public health data files." Statistics in Medicine 14.

Jones, C. B., Purves, R., Ruas, A., Sanderson, M., Sester, M., Van Kreveld, M. and Weibel, R. (2002). Spatial Information Retrieval and Geographical Ontologies: an overview of the SPIRIT project. The 25th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval (SIGIR 2002), Tampere, Finland.

Levenshtein, V. I. (1965). "Binary Codes Capable of Correcting Spurious Insertions and Deletions of Ones." Probl. Inf. Transmission 1: 8-17.

Minkov, E., Wang, R. C. and Cohen, W. W. (2005). Extracting Personal Names from Email: Applying Named Entity Recognition to Informal Text. Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), 443-450, Vancouver, Canada.

Navarro, G. (2001). "A Guided Tour to Approximate String Matching." ACM Computing Surveys 33(1): 31-88.

Patman, F. and Thompson, P. (2003). Names: a new frontier in text mining. ISI 2003, 27-38, Springer.

Pfeifer, U., Poersch, T. and Fuhr, N. (1996). "Retrieval effectiveness of proper name search methods." Information Processing and Management 326(667-679).

Souza, L. A., Davis Jr., C. A., Borges, K. A. V., Delboni, T. M. and Laender, A. H. F. (2005). The Role of Gazetteers in Geographic Knowledge Discovery on the Web. 3rd Latin American Web Congress (LAWeb 2005), 157-165, Buenos Aires, Argentina.

Winkler, W. E. (1999). The state of record linkage and current research problems, Internal Revenue Service: http://www.census.gov/srd/www/byname.html.

Zobel, J. and Dart, P. (1995). "Finding Approximate Matches in Large Lexicons." Software - Practice and Experience 25(3): 331-345.

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 61-72.

Trajectory Data Warehouses: Proposal of Design and Application to Exploit Data

Fernando J. Braz[1]

[1]Department of Computer Science – Ca' Foscari University - Venice - Italy

fbraz@dsi.unive.it

Abstract. *In this paper we are interested in storing and perform OLAP queries about various aggregate trajectory properties. We consider a data stream environment where a set of mobile objects send the data about its location in a irregular and unbounded way, the data volume is stored in a centralized and traditional DW with pre-computed aggregations values (preserving the trajectories privacy). We present an application developed to receive the stream data set, to store and compute the pre-aggregation values and to present final results in order to reveal the knowledge about the trajectories.*

1. Introduction

The data warehouse traditional model can be resumed as a subject-oriented data collection integrated from various operational databases. On a data warehouse model, the data are summarized and aggregated in a multidimensional way in order to facilitate access and data analysis. A Data warehouse provides an integrated environment by extracting, filtering, and integrating relevant information from various data sources. A Data Stream environment presents some characteristics that may improve the difficulty to build and maintain a data warehouse. The data arriving on an unpredictable and continuous rate, the larger data volume and the resource constraints (memory, processing) are some of these main characteristics. The data warehouse traditional model must be adapted in order to work in agreement with these constraints.

The development of new technologies for mobile devices and low-cost sensors results in the possibility of storing larger data volumes about trajectories of moving objects. These data volumes could be stored on a multidimensional model in order to allow an accuracy analysis, it can be defined as a trajectory data warehouse. The goal is to store, manage and analyze the trajectories data in a multidimensional way. The trajectory can be represented by position (X and Y) and time data. A set of observations represents data about several moving objects positions. The trajectory data warehouse has two main problems: the loading phase and the computing of measures. The trajectory data of moving objects arrive in an unbounded and unpredictable way, it is a characteristic that must be considered in the building of a multidimensional data warehouse model. The loading phase has to receive and process the data volume considering the available resources and the irregular rate of arriving the data. We consider a data warehouse model where the identifier of the trajectories is abstracted in favour of aggregate information concerning global properties of a set of moving objects. The aggregated information stored in each cell of the DW model can be used to reveal knowledge of the objects. It can be done by the usage of the OLAP operators, these results can be used as input for subsequent analyses.

In this paper we present an application developed to receive the stream data set, to store and compute the pre-aggregation values and to present final results in order to reveal the knowledge about the trajectories. This application works in a data stream environment, it can receive the data stream volumes (*loading phase*), compute and store the aggregation values (*computing measures phase*) in a trajectory data warehouse. The application was built considering the model proposed in [Braz et al. 2007], we have used a traditional DW system in order to store the Trajectory DW. In the Section 2 we present a briefly review about Trajectory DW model, the main problems and the DW model used in order to store the trajectories are also presented. The application is detailed in the Section 3. Finally, in the Section 4 we draw some conclusions and possible future research topics.

2. The Trajectory DW Model

There are some proposals of spatial data warehouses [Han et al. 1998],[Rivest et al. 2001],[Marchant et al. 2004],[Shekhar et al. 2001], but none of these proposals work with objects moving in a continuous way in time. However, in the building of a warehouse for trajectories, it is a crucial issue. The movement of a spatio-temporal object *o* - i.e., it *trajectory* - can be represented by a finite set of *observations*,i.e. a finite subset of points taken from the actual continuous trajectory. This finite set is called a *sampling*.

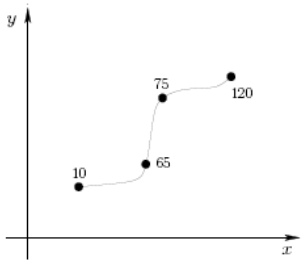


Figure 1. Trajectory with a sampling

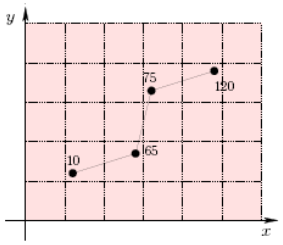


Figure 2. Linear interpolation

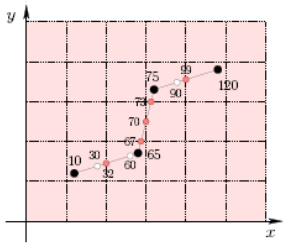


Figure 3. Interpolated trajectory with spatial and temporal points

The Figure 1 represents a trajectory of a moving object in a two dimensional space. Each point of the trajectory is represented by a tuple *(id, x, y, t)* corresponding to an object *id* in a location *(x, y)* at time *t*. There are some situations where we have to reconstruct the trajectory of the moving object from its sampling, e.g., when one is interested in computing the cumulative number of trajectories in a give area. In [Braz et al. 2007] the proposal is to use *linear local interpolation* in order to do it, assuming the movement of the objects between the observed points happens with constant speed, in a straight way. A Trajectory Data Warehouse (*TDW*) has to capable to store a stream of samplings, process the data volume, compute and store the measures in order to provide an environment to analyze the information about the objects. The aggregations measures are crucial in order to do it. However, in a data stream environment, where the data arrive in an irregular and unpredictable way it is specially difficulty. In the *loading phase* the available resources are a very important constraint, it is necessary to develop some mechanism in order to limit the consumption of the resources and to improve the performance of the process. Besides, there is another important phase: *computing measures*, several measures can be

computed in agreement with different complexity spaces. Therefore, the *TDW* must be modeled considering all these issues. In this application we have used the model proposed in [Braz et al. 2007].

2.1. Loading Problem

The loading begins at the base cells of the base cuboid, with suitable sub-aggregate measures, from which starting to compute super-aggregated functions. Considering the data stream characteristic, we have to limit the amount of buffer memory needed to store information about active trajectories. In agreement with [Braz et al. 2007] we consider a trajectory ended when, for a long time interval, no further observation has been received. Given the observations for a trajectory shown in the Figure 1, a possible reconstructed trajectory using linear interpolation is shown by the Figure 2, where we also illustrate the discretized 2D space. With the linear interpolation is possible to infer additional spatio-temporal locations of intermediate points, these points occur between two known trajectory observations. Updating the data warehouse on the basis of each single observation, the *measures* ($M_1, \ldots, M_k$), possibly corresponding to the four observations of our example, the Table 1 shows the base cells.

Table 1. Cells representation - for each observation

Time label	X	Y	T	M_1	...	M_k
10	[30,60)	[30,60)	[0,30)	...	...	...
65	[60,90)	[30,60)	[60,90)	...	...	...
75	[90,120)	[90,120)	[60,90)	...	...	...
120	[120,150)	[90,120)	[60,120)	...	...	...

Table 2. Sequence of segments composing the interpolated trajectory, and the base cells that completely include each segment.

($\mathbf{t}_i$, $\mathbf{t}_{i+1}$)	X	Y	T	M_1	...	M_k
(10,30)	[30,60)	[30,60)	[0,30)	...	...	...
(30,32)	[30,60)	[30,60)	[30,60)	...	...	...
(32,60)	[90,120)	[30,60)	[30,60)	...	...	...
(60,65)	[90,120)	[30,60)	[60,90)	...	...	...
(65,67)	[90,120)	[30,60)	[60,90)	...	...	...
(67,70)	[90,120)	[90,120)	[60,90)	...	...	...
(70,73)	[120,150)	[90,120)	[60,90)	...	...	...
(73,75)	[120,150)	[120,150)	[60,90)	...	...	...
(75,90)	[120,150)	[120,150)	[60,90)	...	...	...
(90,99)	[120,150)	[120,150)	[90,120)	...	...	...
(99,120)	[150,180)	[120,150)	[90,120)	...	...	...

Before the interpolation some base cells could be traversed by the trajectories but, since no observation falls in them, they not appear in the *fact* table, the solution proposed in [Braz et al. 2007] is to consider the additional interpolated points for each cell traversed by a trajectory. The interpolation is computed considering the intersections between the trajectory and the border of the spatio-temporal cells. The Figure 3 shows a trajectory

considering the additional interpolated points. The interpolated points, associated with temporal labels 30, 60, and 90, have been added to match the granularity of the temporal dimension. In fact, they correspond to cross points of a temporal border of some 3D cell. The points labeled with 32, 67, 70, 73, and 99, have been instead introduced to match the spatial dimensions.

After the including of these additional interpolated points, we have further 3D base cells in which we can now store significant measures associated with the trajectory of the given object. The new points subdivide the interpolated trajectory into small segments, each one completely included in some 3D base cell. Therefore we can now update a cell measure on the basis of a single trajectory segment. The Table 2 shows the sequence of edges composing the interpolated trajectory of Figure 3, and the base cell which the edge belongs to.

2.2. Aggregation Problem

A typical measure in a trajectory data warehouse can represent any interesting property about the trajectories in a spatio-temporal interval. In [Gray et al. 1997] the authors present a classification of aggregate functions based on the space complexity for computing a super-aggregate starting from a set of sub-aggregates previously computed:

- *Distributive*: The super-aggregates can be computed from the sub-aggregates.
- *Algebraic*: The super-aggregates can be computed from the sub-aggregates together with a finite set of auxiliary measures .
- *Holistic*: The super-aggregates cannot be computed from the sub-aggregates, not even using any finite number of auxiliary measures.

Table 3. Numeric measures

m1	*numobs*	*Number of observations in the cell*
m2	*trajinit*	*Number of trajectories starting in the cell*
m3	*presence*	*Number of trajectories in the cell*
m4	*distance*	*Total distance covered by trajectories in the cell*
m5	*speed*	*Average speed of trajectories in the cell*
m6	v_{max}	*Maximum speed of trajectories in the cell*

The Table 3 shows the measures that are considered in the application. The computation of the super-aggregates for the measures *m1, m2, m4* and *m6* uses *distributive* aggregate functions. After the loading of the base cells with the exact measures is possible to accumulate the measures by usage the function *sum* (*m1, m2* and *m4*) and *max* (*m6*). However, the super-aggregate for the measure *m5* is *algebraic*, it is necessary to compute an auxiliary measure in order to compute the aggregate function. A pair $\langle distance, time \rangle$ must be considered, where *distance* is the measure *m4* and *time* is the total time spent by trajectories in the cell. For a cell *C* arising as the union of adjacent cells, the cumulative function performs a component-wise addition, producing a pair $\langle distance_f, time_f \rangle$, therefore the average speed in *C* is computed by $distance_f / time_f$. The aggregate function for *m3* is *holistic*, then is necessary to compute the measure in an approximated way. In this application we have used the approach presented in [Braz et al. 2007], the approach will be presented in the following.

The *Presence* function represents the count of the *distinct* trajectories crossing a given cell. Since this function has to deal with the issues related to counting *distinct* trajectories, it is a sort of `COUNT_DISTINCT()` aggregate, and thus a *holistic* one. We exploit alternative and *non-holistic* aggregate functions to compute *Presence* values that *approximate* to the exact ones. These alternative functions only need a few/constant memory size for maintaining the information – i.e., the M-tuple – to be associated with each base cell of our data warehouse, from which we can start to compute a super-aggregate. The two approximate functions we will consider are the following:

1. *Presence*$_{Distributive}$: We assume that the only measure associated with each base cell is the exact (or approximate) count of all the *distinct* trajectories crossing the cell. Therefore, the super-aggregate corresponding to a roll-up operation is simply obtained by summing up all the measures associated with cell. This aggregate function may produce very inexact approximation of the true *Presence*. Because we may count multiple times the same trajectory. We do not have enough information in the base cell that permit us to perform a *count distinct* when rolling-up.

2. *Presence*$_{Algebraic}$: Each base cell stores an M-tuple of measures. One of these is the exact (or approximate) count of all the *distinct* trajectories touching the cell. The other measures are used when we compute the super-aggregate, in order to correct the errors introduces by function *Presence*$_{Distributive}$ due to the duplicated count of trajectory presences.
   More formally, let $C_{x,y,t}$ be a generic base cell of our cuboid, where x, y, and t identify intervals of the form $[l, u)$, in which we have subdivided the spatial and temporal dimensions. The associated measures are thus $C_{x,y,t}.presence$, $C_{x,y,t}.crossX$, $C_{x,y,t}.crossY$, and $C_{x,y,t}.crossT$.
   $C_{x,y,t}.presence$ is the count of all the *distinct* trajectories crossing the cell.
   $C_{x,y,t}.crossX$ is the number of *distinct* trajectories crossing the *spatial* border between $C_{x,y,t}$ and $C_{x+1,y,t}$.
   $C_{x,y,t}.crossY$ is the number of *distinct* trajectories crossing the *spatial* border between $C_{x,y,t}$ and $C_{x,y+1,t}$.
   Finally, $C_{x,y,t}.crossT$ is the number of *distinct* trajectories crossing the *temporal* border between $C_{x,y,t}$ and $C_{x,y,t+1}$.
   In order to compute the super-aggregate corresponding to two adjacent cells with respect to a given dimension, namely $C_{x',y',t'} = C_{x,y,t} \cup C_{x+1,y,t}$, we can compute it as follows:

$$\begin{aligned} &\textit{Presence}_{Algebraic}(C_{x,y,t} \cup C_{x+1,y,t}) = \\ &= C_{x',y',t'}.presence = \\ &= C_{x,y,t}.presence \ + \ C_{x+1,y,t}.presence \ - \ C_{x,y,t}.crossX \end{aligned} \tag{1}$$

   Moreover, if we need to update the other measures associated with the $C_{x',y',t'}$ for subsequent aggregations, we have:

$$C_{x',y',t'}.crossX = C_{x+1,y,t}.crossX$$
$$C_{x',y',t'}.crossY = C_{x,y,t}.crossY + C_{x+1,y,t}.crossY$$

$$C_{x',y',t'}.crossT = C_{x,y,t}.crossT + C_{x+1,y,t}.crossT$$

Then, the idea is to compute a *holistic* measure by the usage of the *distributive* and *algebraic* measures. Therefore, the final result of the measure is an approximated value, computed in agreement with the limited resources presented in a data stream environment.

3. The Application

We have developed our application using the synthetic datasets generated by the traffic simulator described in [Brinkhoff 2000]. These data are stored in the Trajectory Data Warehouse (*TDW*) model presented in the Section 2. The measures stored in the *TDW* can be used to discover interesting phenomena of the trajectories. The application tries to solve the both problems: loading and aggregation, which were presented in the Section 2.1 and 2.2.

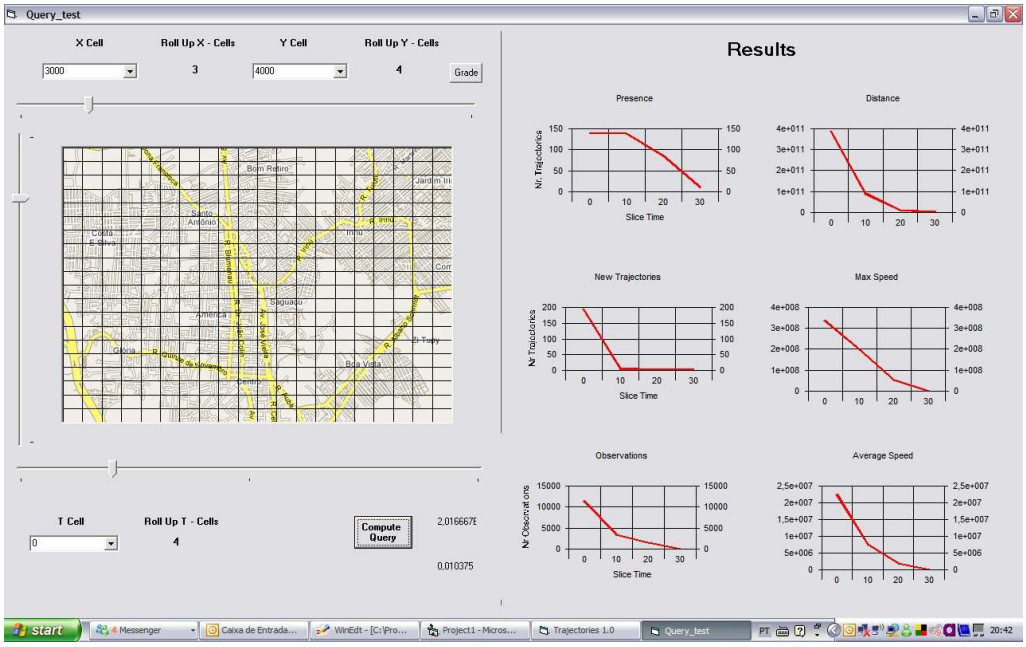


Figure 4. Results Interface

The Figure 4 shows the interface where is possible to visualize the values of the measures computed considering a cell selected by the user. The result presents the evolution of the values of measures in a range of time, in the same visualization is possible to define different values of roll-up operations. The roll-up operation can be defined by the usage of the slider controls over the map, a more detailed explanation will be presented in the next sections.

The *TDW* was implemented in a traditional data warehouse tool, we have used the *MS SQL SERVER 2005*. The *TDW* was modeled in agreement with the *star* model [Kimball 1996], with a fact table and three dimension tables (*X* and *Y* spatial dimensions and *T* temporal dimension). The structure of these tables can be found in the Tables 4, 5 and 5. The Figure 5 shows a schema of the our application: a bottom level where is the *TDW* and the *buffer table*; and a first level where works the loading and aggregation

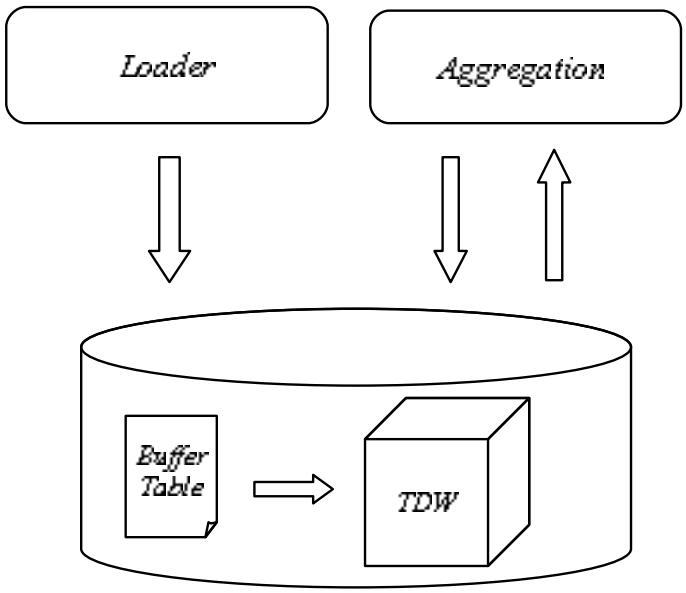


Figure 5. Application Schema

components. The arrows represent the data communication among the components, these components will be described in following sections. The application was built with the *Visual Basic* language, the application allows the user to access the *DTW*. A lot of settings can be done: the definition of the level of the granularity of the trajectory data warehouse, to control the loading of the *TDW* and computation of the measures and aggregations are some of the possibilities of the usage.

Table 4. Fact Table

tid	*time foreign key*
xid	*X spatial foreign key*
yid	*Y spatial foreign key*
numobs	*Number of observations in the cell*
trajinit	*Number of trajectories starting in the cell*
vmax	*Maximum speed of trajectories in the cell*
distance	*Total distance covered by trajectories in the cell*
time	*Total time spend by the trajectories in the cell*
presence	*Number of trajectories in the cell - distributive*
xborder	*Number of trajectories crossing the x cell border*
yborder	*Number of trajectories crossing the y cell border*
tborder	*Number of trajectories crossing the t cell border*
speed	*Average speed of trajectories in the cell*

Each tuple stored in the *fact table* represents a summarization of the measures that are delimited by the borders of the *cell*. The *base cell* are delimited by the *tid, xid* and *yid* values. The measures presented in the Table 4 are detailed in the Section 2.2. The measures *presence, xborder, yborder* and *tborder* are necessary in order to compute the *holistic presence* measure. These measures are specially important when is necessary to compute the *roll-up* operations, this procedure will be explained in the Section 3.2.

Table 5. X or Y Dimension Table

xid	primary key
xl1	first level of hierarchy
xl2	second level of hierarchy

Table 6. T Dimension Table

tid	primary key
tl1	first level of hierarchy
tl2	second level of hierarchy

3.1. Loader Component

The loader component also is responsible by the settings of the environment in order to receive the data volume. This component loads the data volume into a buffer table (see Table 7). We are considering that this process happens in an unpredictable and unbounded way, therefore we have to store packages of data into a buffer table. This procedure permits to release space in the buffer table, it can be done by the exclusion of the tuples of the trajectories ended.

Table 7. Buffer Table

oid	*Object Identifier*
xvalue	*X spatial value*
yvalue	*Y spatial value*
tvalue	*T time value*
dift	*Time variation between two consecutive positions*
difx	*X spatial variation between two consecutive positions*
dify	*Y spatial variation between two consecutive positions*
dist	*Distance covered between two consecutive positions*
vel	*Speed between two consecutive positions*
idrow	*Identifier of the row*
timestamp	*Timestamp of the observation*

Through the loader component is possible to define the details of the environment and to compute the interpolation procedure. In order to compute the interpolation and to load the *TDW* is necessary to define two very important parameters:

- *Granularity level*
- *Dimension hierarchical level*

The definition of the granularity level is necessary in order to define the regular grid which divides the spatio-temporal environment. This procedure is done before the loading the *TDW*, because in the *TDW* schema we have computed the measures for each cell. Therefore, the definition of the cells is the first step in the loading process. After the definition of the granularity level is possible to define the hierarchy level of the dimension tables, this procedure also can be done by the loader component.

The Figure 6 shows a visualization of the interface available in order to define those settings. These procedures are executed just one time, before the beginning of the loading the data warehouse. After the definition of the settings explained above, the interface permits to start the process of the receiving the data stream values and loading the *TDW*. The Algorithm 1 presents the basic procedures in order to complete the loader process, where C_{cur} represents the *current base cell*, C_{prev} the previous *base cell* stored in the *buffer* related to the same trajectory, and IP represents the set of *base cells* computed by the interpolation process. Using the setting values already defined, the loader

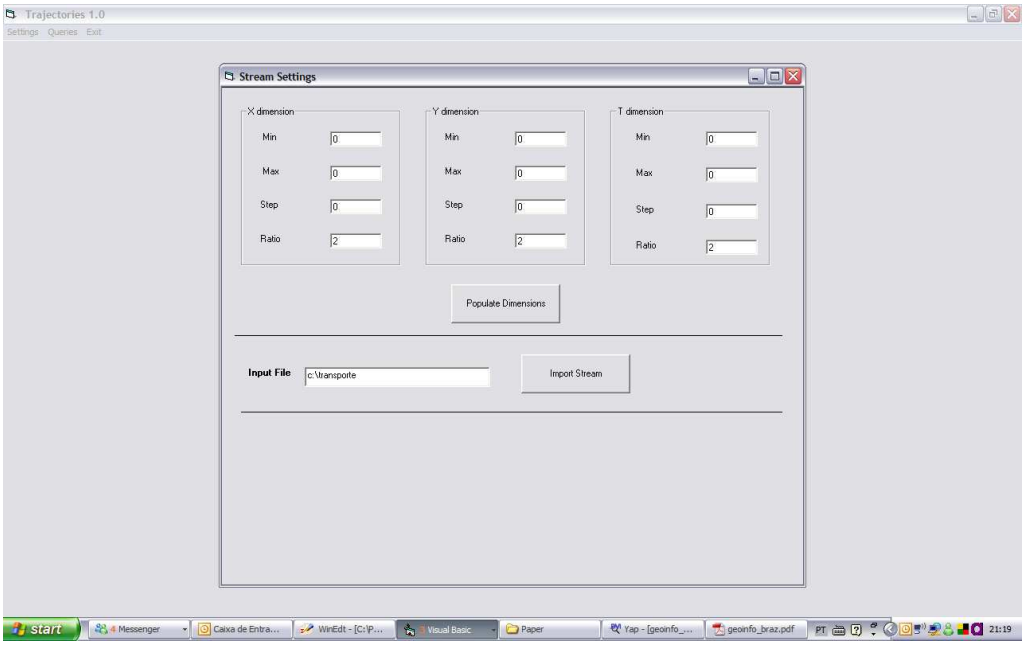


Figure 6. Loading Settings

Algorithm 1 Loading

Input: {*Stream SO of observations - id, x, y, t*}
Output: {*Fact Table FT*}
$FT \longleftarrow \emptyset$
$buffer \longleftarrow \emptyset$
repeat
  $obs \longleftarrow next(SO)$
  $C_{cur} \longleftarrow findCell(obs.x, obs.y, obs.t)$
  if $obs.id \notin buffer$ **then**
    $insertbuffer(obs.id)$
  end if
  $C_{prev} \longleftarrow findCell(obs.x, obs.y, obs.t)$
  $IP \longleftarrow interp(C_{cur}, C_{prev})$
  $ct \longleftarrow 1$
  repeat
    if $IP[ct] \notin FT$ **then**
      $insert(IP[ct], FT)$
    else
      $update(IP[ct], FT)$
    end if
    $ct \longleftarrow ct + 1$
  until $ct < IP.numpoint$
until

component gets the active values in the *buffer table* and performs the procedures in order to load the *TDW*. In the Section 2.1 we have described the concept of *interpolation* used in order to load the *TDW*. The *buffer table* is used in order to implement that procedure. For each *active* row of the *buffer table* the application has to find the related *cell* in the *TDW*. If exists a row in the fact table for that *cell*, the values of *distributive* measures (e.g *numobs, trajinit*) can be updated, else a new row will be inserted in the fact table. It is the procedure when is not necessary to compute the interpolation. However, there are some measures which is necessary to compute the interpolation. In this case the procedure must use the identifier of the active trajectories, their last processed point, the cell such point belongs to, and the speed in the segment ending at such a point. Using that set of values is possible to determine the additional points in order to represent the intersections among the trajectory and the borders of the cells. The additional points can be computed considering the constant speed of the trajectory and the spatio-temporal granularity. In the Algorithm 1, the functions $interp$ is responsible to compute the additional points of the interpolation process. For each new point computed by the interpolation process happens the task of searching the related *cell* in the *TDW*. When there is the base *cell* in the *fact table* the related measures must be updated. Otherwise, the procedure is to insert a tuple with the new values of the measures.

3.2. Aggregation Component

The *aggregation problem* was explained in the Section 2.2. The *distributive* and *algebraic* measures can be solved without problem, but the situation is totaly different when is necessary to compute aggregation measures. In that case there is not a precisely result, just an approximation value can be computed.

In our application the pre-aggregated values are stored in the *TDW*. These pre-aggregated values are computed in the *loading* phase. However, when is necessary to compute some query or to perform a roll-up operation, the application uses the *aggregation component*. The Figure 4 shows the results of a query defined by the boundaries of the cell selected on the map. The results are represented by the charts (right side) where is possible to verify the evolution of the measures for each value of time dimension.

The user can choose the query *base cell* either by the usage of the combo-boxes or by clicking on the map. The map is divided into a regular grid, this division is done in agreement with the granularity defined by the user. In some cases just a simple query in the data warehouse can solve the query, for example when the query is limited in a only one *base cell*, it is possible because the tuples in the *TDW* stores the pre-aggregated values. However, when is necessary to compute a roll-up operation a simple search in the *fact table* is not sufficient. In these cases the solution can not be found by the usage of the relational operators (*Select, Update...*), then the application must use a lot of *stored procedures* developed in order to solve these situations. The computation of the *holistic* measures also is done by the usage of the *stored procedures*. For example, to compute the *holistic presence* measure we need the *distributive* measures *presence* and the other ones *algebraic* measures: *xborder, yborder* and *tborder*. It is a complex task, because is necessary to determine the direction of the *roll-up* operation. If the *roll-up* happens just in the *X-dimension* the only ones measures used will be the *distributive presence* and the other one *algebraic* measure: *xborder*, the same procedure happens for the another dimensions. There is another one more complex query: the *roll-up* in *X, Y* and *T* dimensions, again this

situation is solved by the usage of the *stored procedures*. All the *stored procedures* are invocable by the *loading component* and executed in the *TDW*, the results of the queries are presented in the *Results* area in the query form of the application (see Figure 4).

4. Conclusions and Future Work

In this paper we have presented an application in order to implement a *TDW* considering the constraints to load the data warehouse and compute the aggregation values. The application is a first step in order to try solving the problems of loading and aggregation in a *TDW* environment. The data cube model adopted is very simple, it is just a contribution in order to improve the discussion about the problems to implement a *TDW* model. However, this model can be extended to more general situations.

The current stage of the application can solve some problems of a trajectory data warehouse environment. However, the dimensions that we have used are very simple, a possible future work could be to sophisticate the hierarchy of the dimensions. The loading phase is an opened problem, we have limited the loading phase considering a linear interpolation, but is possible to find some topological situations (e.g roads, bridges) where is very difficult to do this interpolation because of the some constraints in the movement of the object.

In this work we have used the *roll-up* operation, however could be interesting to offer mechanisms in order to compute other operators such as drill-down, pivot, slice and dice. Therefore, the development of a query language using OLAP operators also is a possible point to research.

Another interesting area of additional research is to develop another more complex measures in order to provide values to discover patterns or trend of the trajectories. To compute values to support the data mining tasks is a very interesting point of future research.

References

Braz, F., Orlando, S., Orsini, R., Raffaetà, A., Roncato, A., and Silvestri, C. (2007). Approximate aggregations in trajectory data warehouse. In *Proc. of ICDE Workshop STDM*, pages 536–545.

Brinkhoff, T. (2000). Generating network-based moving objects. In *SSDBM '00: Proceedings of the 12th international Conference on Scientific and Statistical Database Management*, page 253, Washington,DC,USA. IEEE Computer Society.

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., and Pirahesh, H. (1997). Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. *Data Mining and Knowledge Discovery*, 1:29–54.

Han, J., Stefanovic, N., and Kopersky, K. (1998). Selective materialization: An efficient method for spatial data cube construction. *PAKDD'98*, pages 144–158.

Kimball, R. (1996). *Data Warehouse Toolkit*. John Wiley.

Marchant, P., Briseboi, A., Bedard, Y., and Edwards, G. (2004). Implementation and evaluation of a hypercube-based method for spatiotemporal exploration and analysis. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59:6–20.

Rivest, S., Bedard, Y., and Marchand., P. (2001). Towards better support for spatial decision making: Defining the characteristics of spatial on-line analytical processing(solap). *Geomatica*, 55(4):539–555.

Shekhar, S., Lu, C., Tan, X., Chawla, S., and Vatsavai, R. (2001). *Map Cube: Avisualization Tool for Spatial Data Warehouse*, chapter Geographic Data Mining and Knowledge Discovery. Taylor and Francis.

Ecologically-aware Queries for Biodiversity Research

Luiz Celso Gomes Jr[1], **Claudia Bauzer Medeiros**[1]

[1]Instituto de Computacao – UNICAMP
Caixa Postal 6176 – 13081-970 – Campinas – SP – Brazil

{luizcelso,cmbm}@ic.unicamp.br

Abstract. *To carry ecologically-relevant biodiversity research, one must collect chunks of information on species and their habitats from a large number of institutions and correlate them using geographic, biologic and ecological knowledge. Distribution and heterogeneity inherent to biodiversity data pose several challenges, such as how to find and merge relevant information on the Web, and process a variety of ecological and spatial predicates. This paper presents a framework that exploits advances in data interoperability and Semantic Web technologies to meet these challenges. The solution relies on ontologies and annotated repositories to support data sharing, discovery and collaborative biodiversity research. A prototype using real data has implemented part of the framework.*

1. Introduction

Biodiversity is an outstanding example of a scientific domain that deals with heterogeneous datasets and concepts from many areas. Biodiversity studies rely on models to define species richness, abundance, endemism, distribution and so forth. To create the models, species occurrence data must be obtained from diverse institutions, and be combined with other kinds of data, such as phylogenetic data (describing evolutionary relations), taxonomic data for nomenclature, data describing ecological correlations among species and geographic data depicting habitat conditions.

Typically, biodiversity information systems provide support to queries that are centered on the so-called *collection* or *occurrence* records, managed by museums or by research institutions. An occurrence record stores data on some kind of observation of living beings – it includes data on species' taxonomic classifications, location where the species were observed or collected, by whom, when and how. Additional data sources include geographical data (e.g. on habitats, or climate variables), and several kinds of annotations. The most common queries on such systems concern species' spatial distribution in a given area. Other queries may demand sets of occurrence records that satisfy a given predicate, or computation of aggregate functions over such records. Scientists may also want to find out more about specific geographic areas (e.g., rainfall or temperature patterns), thereby being able to compute climate models, or run simulations on habitat variables.

Query predicates, in these systems, can be classified into two categories: those that involve operations that are typically computed by standard DBMS mechanisms and those that involve computing spatial predicates. The latter either requires extended DBMS capability e.g., using PostGIS or, more commonly, a GIS. Thus, end-user requests in a typical biodiversity information system are solved by combining spatial correlations to

functions used in a DBMS. This, however, only supports a subset of the functionality demanded by bio-scientists.

These end-users also need more complex computations, e.g., requiring spatio-temporal query processing, such as deriving co-occurrence of species in a given space-time frame. Such processing is seldom supported. Other predicates involve ecological relations among species, e.g., predator-prey or parasitic relationships. Such relationships are not stored, and must be deduced by the scientist after performing a sequence of queries and simulations. Most times, scientists have to invest a considerable amount of time, and perform many manual tasks, to obtain the needed data.

This paper proposes a framework to fill this gap. Besides supporting the more usual kinds of query predicates, it also allows computation of ecological predicates, by combining stored and derived data and ontologic information, for distributed data repositories. This framework has been partially implemented using data from the Institute of Biology, UNICAMP, within an eScience biodiversity project[Medeiros et al. 2007].

The main contributions of the paper are, therefore: (i) the specification of a framework that allows scientists to pose semantically rich queries, encompassing taxonomic, ecological and geographic predicates and (ii) the validation of the framework by the partial implementation of a prototype, using real data.

2. Related work

2.1. Biodiversity research

Research in biodiversity is devoted to understanding the diversity of life and trying to find ways to preserve it. Biodiversity is, however, a complex subject. To begin with, estimates for the total number of species in the planet range to up to 80 million [Wilson 1999] — the bulk of this amount yet to be discovered. Moreover, to undertake biodiversity studies, scientists have to take into account species interactions, both among species and with their environment.

The major interactions between *pairs* of species include competition, predation and mutualism [Morin 1999]. Many more complex interactions can be derived from these elementary processes. Food chains, for example, are pathways of nutrient flow through a sequence of species arranged according to their predator-prey interactions. Another important concept in ecological research is that of *taxonomic relations*, which forms the foundation that enables scientists to properly interpret each other's work [Wilson 1999].

Species interactions with the environment are assessed by combining geographic and ecological data. Therefore, finding and accessing geographical data becomes critical in biodiversity research [Guralnick and Neufeld 2005]. Geographic constraints related to natural conditions (e.g. climate and relief) and human activities (e.g pollution) have direct impact in species richness and distribution. *Species occurrence* data, which also contains geographic information, is the basic unit of information for biodiversity measurements, as mentioned in Section 1. They allow studies on species distribution patterns, thereby supporting efforts on conservation initiatives.

2.2. Biodiversity data sharing

Work on biodiversity involves scientists from many fields, and requires combining a variety of distributed heterogeneous data sources on the Web. Geospatial Web services and

exchange standards for occurrence records are important elements in promoting biodiversity data integration and interoperability among systems.

Data sharing and integration is often based on geographic coordinates. Thus, geospatial Web services are considered in many solutions [Guralnick and Neufeld 2005]. The Open Geospatial Consortium (OGC) [Open Geospatial Consortium Inc. (OGC)] is an international organization that leads the development of standards for interoperability among geospatial applications. The consortium defines the Web Feature Service (WFS) [OGC 2005b] specification to provide a standardized means to access geospatial data encoded in the Geographic Markup Language (GML) [OGC 2003]. GML, also from OGC, is an XML-based standard for the transport and storage of geospatial information. The WMS (Web Map Service) specification defines means to produce two-dimensional maps from geospatial data.

There are many initiatives to leverage sharing and interoperability of species occurrence data. Darwin Core [Taxonomic Databases Working Group (TDWG)] is an XML-based standard that defines the necessary elements to describe species occurrence data, constituting the first step towards data interoperability. Infrastructures for sharing biodiversity data on the Internet (such as Species Analyst [Species Analyst project]) rely on exchange standards and transmission protocols to build an interconnected network of data providers. A scientist interacts with such systems by indicating target sites and data sources and posing queries through a standard interface. Queries are usually limited to textual predicates and return raw occurrence records to the user. Such infrastructures do not allow more elaborate queries, and it is up to the scientists to perform any kind of semantic post-processing.

2.3. Ontologies

Ontologies are being used in Computer Science to formalize shared conceptualizations within communities. An ontology organizes concepts to convey semantic information and to allow new knowledge to be inferred [Gruber 1995].

The Semantic Web initiative is pushing forward the use of ontologies to provide the Web with a machine-understandable metadata framework, fostering interoperability. The World Wide Web Consortium (W3C) is the main player in the Semantic Web initiative. W3C specified the Web Ontology Language (OWL) [Daconta et al. 2003], the standard for ontology specification. OWL is based on the Resource Description Framework (RDF) [Daconta et al. 2003], which is a general-purpose language to represent and correlate Web resources.

W3C is developing the SPARQL query language for querying RDF data [Seaborne and Prud'hommeaux 2006]. A SPARQL query is formatted in terms of RDF triple patterns. Queries are evaluated via pattern matching between the query expression and the RDF graph.

Many biodiversity projects have begun to explore the use of ontologies to allow data sharing on the Web. The SPIRE project [Parr et al. 2006] is investigating how Semantic Web technologies can be applied to the biodiversity domain. The project is developing ontologies for taxonomic, ecological and niche modeling concepts, and is producing tools based on the ontologies. Among the tools is an on-line query form that allows users to submit SPARQL queries. Query results return fragments of the ontologies, ex-

pressed in OWL. There is no attempt to retrieve other kinds of biodiversity-related data available in Web repositories.

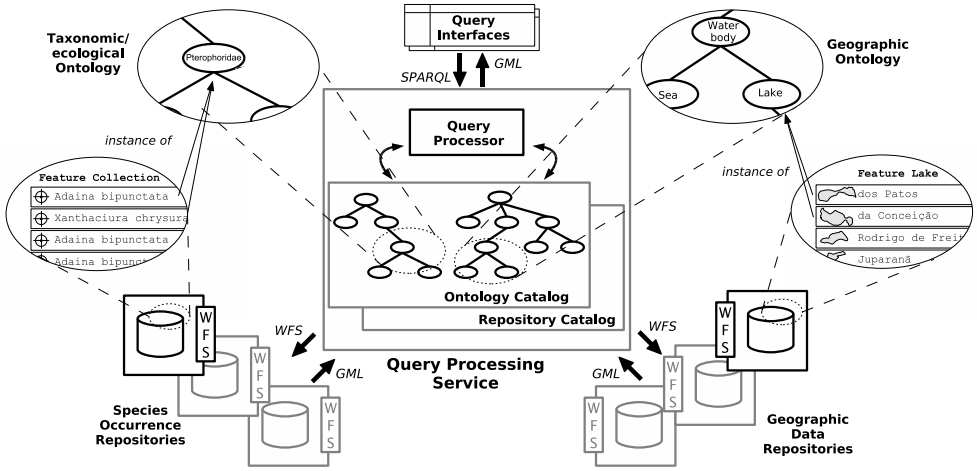


Figure 1. Overview of the interactions among the architecture's elements

3. Ecologically-aware queries

This section presents the architecture of our infrastructure for processing ecological queries. It integrates all trends presented in the previous section: it employs (i) domain ontologies to provide a global model of the data to be shared, (ii) standards to access remote data repositories, and (iii) a combination of spatial, textual and ecological predicates to process ecologically-aware queries.

3.1. Architecture overview

The architecture is composed of three elements: (i) query interfaces, where users pose biodiversity queries, (ii) a query processing service, that processes queries received from the interfaces and (iii) distributed repositories, from where the query service retrieves data. Figure 1 presents a high level view of these elements and their interactions. Query interfaces are applications tailored to specific goals (e.g., predict species occurrence, establish conservation priorities) and users (e.g., biologist, ecologist). User queries at the interface are translated to SPARQL and forwarded to the query processing service.

This query processing service (center of the figure) is the main element of the architecture. Its role is to disambiguate predicates with help of ontologies, to find the appropriate data in distributed Web repositories, process these data, and return the results to the users. The repositories (left and right bottom) are databases published by research groups and institutions. There are two types of repositories: those that hold occurrence records, and those that hold data on geographic objects such as lakes, countries or biomes. The figure shows examples of data published by the institutions. Occurrence and geographic data records are georeferenced (i.e. associated with geographic coordinates).

The figure also shows that the query processor makes extensive use of ontologies to expand terms and to process predicates. The ontology on the left contains taxonomic and ecological information. Its expanded view shows the *Tephritidae* concept (the family of insects that includes fruit flies). The ontology on the right contains geographic information, with *Water Body* and *Lake* concepts in the expanded view. As shown by the

arrows among these detailed views, repositories' contents are associated - in a conceptual level - with ontology elements.

3.2. The query processing service

The query service is composed of a query processor and catalogs. The **query processor** – see Figure 1 – receives SPARQL queries from query interfaces (whose design is outside the scope of this paper). The processor's output is a GML file that can be used to generate maps at the interfaces.

Query processing requires internal data structures, stored in **catalogs**. The term "catalog" was adopted to establish an analogy with standard DBMS query processing mechanisms, where catalogs store information such as database schemas or data allocation properties [Elmasri and Navathe 1994]. The service's catalogs are used by the processor in tasks such as expanding query terms and finding target repositories. There are two kinds of catalog: Domain Ontology Catalog and Repository Catalog. Their contents are expected to be consistent – i.e., there is no conflicting information.

The **Domain Ontology Catalog** stores the ontologies containing taxonomic/ecological and geographic concepts. Its content is provided by research communities. It is used by the query processor to expand queries and process ecological predicates. The taxonomic/ecologic ontology contains assertions such as "*Adaina bipunctata* (a butterfly species) is a subclass of *Pterophoridae* (a family) that preys on plant species *Chromolaena squalida*". The geographic ontology holds taxonomic classifications of geographic phenomena, such as "concept *Lake* is-a *Water Body*".

The **Repository Catalog** plays the role of an "index" to biodiversity data sources on the Web. It contains entries registered by trusted institutions and research groups. As depicted in Figure 2, each such entry is composed of four main fields: the repository type, its URI, a geographic bounding box, and a set of ontologic annotations from the Ontology Catalog. The *type* field indicates whether the Web repository contains information on occurrence or geographic phenomena. The *bounding box* defines the geographic region for which the repository can provide data. The ontologic annotations qualify the contents of a repository. Repository registering assumes that occurrence data records are compliant to the Darwin Core standard [Taxonomic Databases Working Group (TDWG)]. All repositories must be compliant with the WFS standard, thus standardizing interfaces and providing means to apply geographic filters in data retrieval.

Type	URI	Bbox	HasDataAbout
occurrence	http://plants.org/wfs	-46,-18 -43,-16	Chromolaena_squalida, Mikania_purpurascens
occurrence	http://butterflies.org/wfs	-47,-12 -42,-15	Pterophoridae
occurrence	http://flowers.org/wfs	-43,-16 -27,-18	Asteraceae
geographic	http://ibge.gov.br/wfs	-74,4 -26,-35	State
geographic	http://ibama.gov.br/wfs	-74,4 -33,-35	LandBiome

Figure 2. Entries in the repository catalog

3.3. Query processing

Figure 3 shows the sequence of phases in query processing. The processor receives an extended SPARQL query (Phase A) and returns a GML file containing the desired data (Phase C).

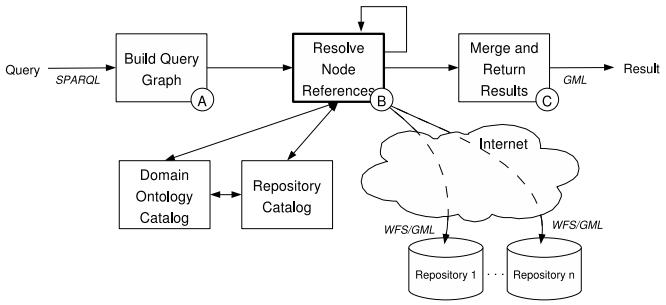


Figure 3. Query processing phases

The framework is strongly based on ontology processing. Ontologies and their elements intervene at each step of query processing. For this reason, the solution for query execution favors structures to process ontologies – i.e., all intermediate results are used to create, match and expand graphs. The three main phases are:

A) Build Query Graph: Analyse the input query, and build the corresponding graph. The graph generated is a straightforward materialization of the graph implicitly expressed in the query: in a query graph $G(V, E)$ for a query Q, (i) $u \in V \Leftrightarrow u$ is subject or predicate of Q and (ii) $(u, v) \in E \Leftrightarrow$ there is a predicate in Q associating the subject of u and the object of v. The graph's vertices and edges are labeled with the URIs expressed in the SPARQL query.

B) Resolve Node References: Iteratively process the query graph, resolving undefined elements. First, the framework's internal catalogs are checked; next, WFS requests are sent to the appropriate Web repositories to retrieve records. The result is a graph, or set thereof, extended with data retrieved from the repositories.

C) Merge and Return Results: Process the contents of the graph(s) resulting from phase B and translate them into GML. The resulting file is returned to the interface level.

Algorithm 1 Process leaf-branches

Require: query graph *G*
Ensure: All graph nodes are resolved
while *G* has leaf-branches to be resolved **do**
  b ⇐ highest priority unresolved branch
  if *priority*(*b*) = 1 **then** {*b* can be resolved locally}
    update query graph
  else if *priority*(*b*) = 2 **then** {*b*'s resolution requires data from catalog}
    resolve using Ontology Catalog data
    apply results to the query graph, updating priorities
  else {*b*'s resolution requires data from repositories}
    simplify spatial predicates
    determine repositories to query, using Repository Catalog
    assemble and submit WFS queries to repositories
    apply results to the query graph, updating priorities
  end if
end while

Step B is the most complex, and is subdivided into several steps according to Algorithm 1 (error conditions are omitted). We name a *leaf-branch* a set composed of one single-degree vertex (a leaf), its incident edge, and the edge's other vertex (hereafter referred to as the branch's *base*). More formally, a leaf-branch B in a query graph G(V,E) may be defined as

$$B = \{(u, v, (u, v)) : u, v \in V \wedge (u, v) \in E \wedge degree(u) = 1\}$$

The algorithm is applied iteratively to each leaf-branch of the graph until the graph is completely resolved. It is suitable to connected, acyclic graphs (trees). The resolution of a leaf-branch comprises analyzing the predicate expressed in the edge, processing this predicate according to the object encoded in the leaf, and applying the results to the branch base. For this reason, we employ the term branch (rather than just leaf) resolution, to indicate the processing to be performed. At the end of a branch processing, its leaf is eliminated and its base contains the results of the processing. The algorithm uses a table [Gomes Jr 2007] to assign priorities to each leaf-branch according to their type. Priority 1 (the highest priority) branches are resolved locally (only by rearranging the query graph), priority 2 branches need the ontologies in the catalog, and lower priority branches (3 and 4) need remote queries to repositories. The goal of this strategy is to postpone costly operations until there is more information to filter intermediate results, avoiding retrieval of unnecessary data. The key steps are described in the following. For more details, the reader is referred to [Gomes Jr 2007].

Obtain highest priority branch (line 2): Chooses one leaf-branch among those with the highest priority, which is to be resolved in the subsequent steps of the algorithm.

Update query graph (line 4): For priority 1 branches, the resolution consists of a simple manipulation in the query graph (e.g. pruning). These branches are handled first, since they do not demand processing data.

Resolve using Ontology Catalog data (line 6): For priority 2 branches, the resolution consists on getting the needed information from the Ontology Catalog.

Simplify spatial predicates (line 9): The resolution of branches with priority higher than 2 involves retrieving data from Web repositories. Whenever a branch bearing spatial predicates enters this step, these predicates can be pre-processed to simplify data retrieval e.g., redundant predicates can be excluded [Rodríguez et al. 2003]. A deeper study on which optimizations may be done in this step is still in progress. The subsequent steps of the algorithm consider that geographic predicates have been pre-processed to restrict the spatial extent of queries submitted to repositories.

Determine repositories to query (line 10): Checks the Repository Catalog for a list of repositories that may provide instances regarding the current branch. This is processed by matching the branch's contents with the type, ontologic annotations and eventually the bounding boxes in the Repository Catalog.

Assemble and submit WFS queries (line 11): Assembles WFS queries tailored to each repository identified in the previous step. Asynchronously submits these queries to the appropriate repositories.

Apply results to graph (lines 7 and 12): Translates into graph representation

results from the queries to the Ontology Catalog or repositories. Updates priorities.

4. Example

Let us now consider the following query: "return all *occurrence records* of species that are *preyed on by* the species *Adaina Bipunctata* and have been found *in São Paulo State's Atlantic Rainforest Biome*". This query contains ecological (prey on), spatial (in) and taxonomic predicates (*species = Adaina Bipunctata*). Additional spatial predicates are defined by naming geographic areas (São Paulo, Atlantic Rainforest). The processing of taxonomic and ecological predicates is based on the ontologies. The processor deals with spatial relations by building geographic filters to retrieve data in the repositories.

```
PREFIX te: <http://. . ./webios/taxo_eco.owl#>
PREFIX geo: <http://. . ./webios/geographic.owl#>
PREFIX sr: <http://. . ./webios/spatial_relation.owl#>
SELECT ?occurrence
WHERE { te:Adaina_Bipunctata te:predatorOf ?species .
        ?occurrence a ?species .
        ?occurrence sr:within geo:Sao_Paulo .
        ?occurrence sr:within geo:Atlantic_Rainforest .
```

Figure 4. Example of query using SPARQL syntax

Figure 4 shows the corresponding query in syntax that is compatible with SPARQL. In the code, the prefixes *te* and *geo* respectively stand for the taxonomic/ecological and geographic domain ontologies, which are to be used to process the query. The prefix *sr* indicates spatial predicates. Accepted spatial expressions are those specified by OpenGIS Spatial Filter Implementation [OGC 2005a], themselves representing the standard binary relationships found in the literature (e.g., [Rodríguez et al. 2003] – such as within, overlaps or disjoint). Keyword *a* is the standard syntax for "instance of" relationships in SPARQL. SPARQL queries provide access to multiple name spaces via the FROM clause; however, all found examples in the literature (and in Web sites) pressupose that there is a possibility of constructing a single ontology graph to be queried from the name spaces. Also, they do not allow accessing multiple ontologies at a time. Thus, this request needs to be decomposed into several queries. To do this, we start by building a query graph.

Figure 5 shows intermediate states of the query graph during the processing of the example query. Figure 5(1) depicts the graph in the beginning of the first iteration of Algorithm 1, which is the original graph built in Phase A. Leaf-branches that are candidates for resolution are highlighted. In this case, the left branch has higher priority and is resolved in this iteration. Figure 5(3) represents the third iteration of the algorithm. The left branch bears now the result of Iteration 1, obtained from the ontology repository (lines 5-7 in Algorithm 1): the ecological ontology states that species *Chromolaena squalida* and *Trichogonia villosa* are preyed on by *Adaina Bipunctata*. By the same token, the middle branch bears the result of Iteration 2 (omitted in the Figure), showing that the geometry for the concept "São Paulo state" is now known. This geometry was retrieved from a geographic Web repository by means of a WFS query execution (lines 8-12 in Algorithm 1).

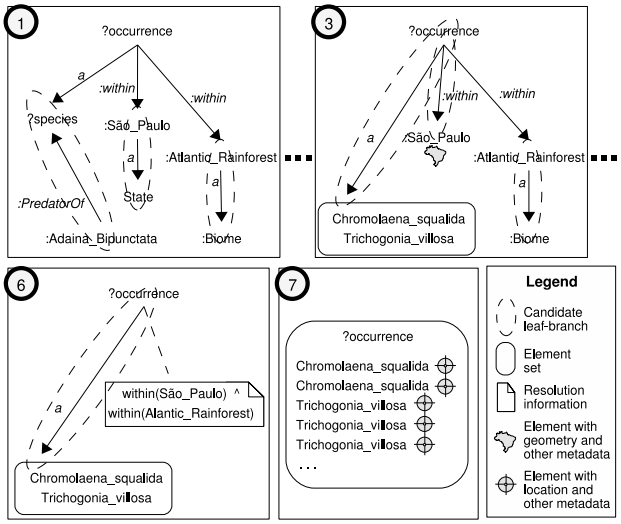


Figure 5. Sequence of states of the query graph for the example query (Figure 4) in successive iterations. Arrows denote the semantics of the predicates and do not imply any orientation to the graph.

Figure 5(6) shows the initial state of the query graph before the last iteration. The graph has been reduced to only one branch. This branch has all information needed to obtain the remote data expressed in the original query: retrieved records must be instances of species *Chromolaena squalida* and *Trichogonia villosa* and must be restricted to the geographic region determined by the intersection of the geometries of São Paulo State and Atlantic Rainforest. With this information, the processor can assemble WFS queries (such as the one shown in Figure 6 - left) and submit them to repositories that, according to the Repository Catalog, may provide the required data (lines 8-12 in Algorithm 1). Figure 5(7) shows the final state of this last iteration. The graph variables are completely resolved, bearing occurrence records of the species requested.

```
<wfs:GetFeature . . . >
 <wfs:Query typeName="plantsorg:species">
  <Filter>
   <And>
    <Or>
     <PropertyIsEqualTo>
      <PropertyName>ScientificN</PropertyName>
      <Literal>Chromolaena_squalida</Literal>
     </PropertyIsEqualTo>
     <PropertyIsEqualTo>
      <PropertyName>ScientificN</PropertyName>
      <Literal>Trichogonia_villosa</Literal>
     </PropertyIsEqualTo>  . . .
    </Or>
    <Within>
     <PropertyName>the_geom</PropertyName>
     <gml:Polygon> . . .
      <gml:coordinates . . . > -46.469289,-18.895586
          -44.87035,-18.66422 . .
    </Within>
   </And>
  </Filter> </wfs:Query> </wfs:GetFeature>
```

```
<wfs:FeatureCollection . . . > . . .
 <gml:featureMember>
  <lis:webios fid=webios.4">
   <lis:the_geom> . . .
    <gml:Point>
     <gml:coordinates . . . >-44.7196,-23.3099 . . .
   <lis:ScientificName>Trichogonia_villosa . . .
   <lis:Collector>A. M. Almeida, U. Kubota . . .
  </lis:webios> </gml:featureMember>
 <gml:featureMember>
  <lis:webios fid=webios.6">
   <lis:the_geom>  . . .
    <gml:Point>
     <gml:coordinates . . . >-44.8341,-23.2024 . . .
   <lis:ScientificName>Chromolaena_squalida . . .
   <lis:Collector>E. P. Anseloni, J.C. Silva . . .
  </lis:webios> . . .
 </gml:featureMember> </wfs:FeatureCollection>
```

Figure 6. (left) Part of a WFS query to retrieve certain species within a given area; (right) GML results for the WFS query containing species occurrence data

The corresponding WFS query (Figure 6 - left) is constructed and sent to the appropriate service. The result is a GML file (Figure 6 - right), corresponding to phase C of the algorithm, and is returned to the query interface.

We have implemented parts of a prototype for the query service. We are using

Jena RDF framework [HP Labs] to process (simplified) SPARQL queries and GeoServer WFS implementation [GeoServer Project] to publish repositories.

We have also developed a graphical interface which takes advantage of WFS and WMS services to support user queries. Figure 7 shows a screen copy of this interface. The left part displays a dynamic tree view containing an excerpt of the ecological ontology, which the user can investigate by hierarchical navigation. Points on the map show locations of observations recorded in occurrence records. The window below the map lets end-users define temporal predicates and desired features - in this case, it shows that points display insect information. When the user clicks a point in the map, a query is sent to the species occurrence repositories and returns details on the corresponding record(s). This interface was implemented using Dojo and MapBuilder widget/AJAX toolkits. Dojo is a toolkit that provides richer user interaction and simplifies AJAX programming (it was used, for example, the dynamic tree view). MapBuilder is a toolkit that provides widgets for map interaction. It is responsible for WMS map presentation and WFS query manipulation in the application.

5. Concluding remarks

This paper proposed an architecture for data sharing and retrieval to support biodiversity research. The approach relies on combining information stored in remote data repositories with ecological and geographic ontologies designed by domain experts. Query processing relies on these ontologies, which embed geographic and ecological relations. This extends present biodiversity system mechanisms by supporting a combination of standard spatial and complex ecological predicates.

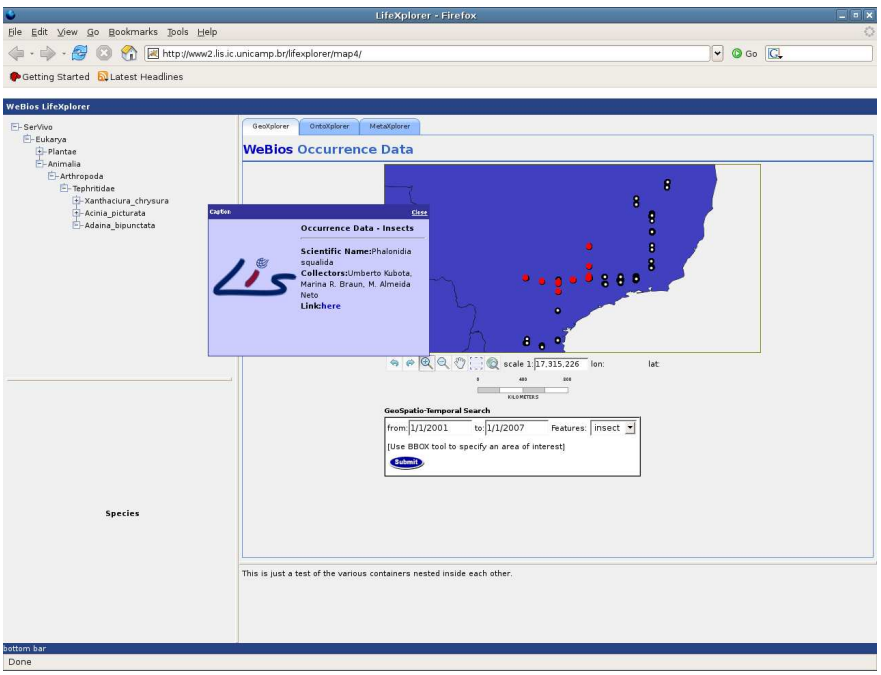


Figure 7. Screen copy of the visualization tool using WFS and WMS service implementations

The approach to conciliate the centralized ontological model and the underlying relational data at the repositories contrast with other strategies that aim at deriving onto-

logical models from relational schemas (e.g. [Laborda and Conrad 2006]). We provide a loosely coupled association between domain specific ontologies and repository data. The ontologies and the repositories are independently developed and can be used in other scenarios. This approach simplifies management of distributed repositories and provides higher flexibility to changes in the centralized model; both characteristics are important in the context of biodiversity data sharing.

Though inspired in the biodiversity research domain, we believe that the architecture could be generalized to encompass data in other scientific fields, provided the appropriate ontologies are available. Present work includes defining a comprehensive set of "typical" user queries, together with end users, to test the effectiveness of the proposed framework. Another issue is query performance. Our implementation favors processing via RDF graph management, to take advantage of our ontology structures, and their processing using SPARQL mechanisms. This kind of processing, however, is less efficient, space and time-wise, to process standard predicates. Thus, for large result datasets, a hybrid mechanism is being envisaged, combining SQL and SPARQL.

Acknowledgements: This research was financed by an eScience grant from Microsoft Research, Redmond, and by Brazilian funding agencies CAPES, CNPq and FAPESP.

References

Daconta, M. C., Obrst, L. J., and Smith, K. T. (2003). *The Semantic Web : A Guide to the Future of XML, Web Services, and Knowledge Management.* Wiley.

Elmasri, R. and Navathe, S. B. (1994). *Fundamentals of Database Systems, 2nd Ed.* Benjamin/Cummings.

GeoServer Project. GeoServer web site. http://geoserver.sourceforge.net (Feb 07).

Gomes Jr, L. C. (2007). Uma Arquitetura para Consultas a Repositórios de Biodiversidade na Web (An architecture to query biodiversity data on the Web). Master's thesis, UNICAMP. Supervision C. B. Medeiros.

Gruber, T. (1995). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *Int Jn of Human-Computer Studies*, 43(5-6):907–928.

Guralnick, R. and Neufeld, D. (2005). Challenges Building Online GIS Services to Support Global Biodiversity Mapping and Analysis: Lessons from the Mountain and Plains Database and Informatics project. *Biodiversity Informatics*, 2:56–69.

HP Labs. Jena website. http://jena.sourceforge.net/ (accessed February 26, 2007).

Laborda, C. P. and Conrad, S. (2006). Bringing Relational Data into the SemanticWeb using SPARQL and Relational.OWL. In Barga, R. S. and Zhou, X., editors, *ICDE Workshops*, page 55.

Medeiros, C. B., Torres, R., Falcao, A., Lewinsohn, T., and Prado, P. (2007). The WeBIOS Project. http://www.lis.ic.unicamp.br/projects/webios (Jun 07).

Morin, P. (1999). *Community Ecology*. Blackwell Science.

OGC (2003). Geography Markup Language (GML) 3.0. https://portal.opengeospatial.org/files/?artifact_id=7174 (accessed February 26, 2007).

OGC (2005a). Filter Encoding Implementation Specification 1.1.0. http://www.opengeospatial.org/standards/filter (accessed February 26, 2007).

OGC (2005b). Web Feature Service (WFS) Implementation Specification. http://portal.opengis.org/files/?artifact_id=8339 (accessed February 26, 2007).

Open Geospatial Consortium Inc. (OGC). OGC website. http://www.opengeospatial.org (Jun 07).

Parr, C., Parafiynyk, A., Sachs, J., Ding, L., Dornbush, S., Finin, T., Wang, D., and Hollander, A. (2006). Integrating ecoinformatics resources on the semantic web. In *WWW '06: Proc 15th international conference on World Wide Web*, pages 1073–1074.

Rodríguez, M. A., Egenhofer, M. J., and Blaser, A. D. (2003). Query Pre-processing of Topological Constraints: Comparing a Composition-Based with Neighborhood-Based Approach. In *Proc SSTD*, volume 2750 of *LNCS*, pages 362–379.

Seaborne, A. and Prud'hommeaux, E. (2006). SPARQL Query Language for RDF. W3C working draft, W3C. http://www.w3.org/TR/2006/WD-rdf-sparql-query-20061004/.

Species Analyst project. Species Analyst website. http://speciesanalyst.net (Feb 07).

Taxonomic Databases Working Group (TDWG). Darwin Core 2 Review. http://darwincore.calacademy.org (Feb 07).

Wilson, E. (1999). *Biological Diversity: The Oldest Human Heritage*. New York State Museum.

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 85-96.

Federated Spatial Cursors

Nazario Cipriani, Matthias Grossmann, Daniela Nicklas, Bernhard Mitschang

[1]Universität Stuttgart, Institute of Parallel and Distributed Systems
Universitätsstraße 38, 70569 Stuttgart, Germany

{cipriani, grossmann, nicklas, mitschang}@ipvs.uni-stuttgart.de

Abstract. *The usage of small mobile devices for data-intensive applications becomes more and more self-evident. As a consequence we have to consider these devices and their inherent characteristics in future system designs, like the limitations of memory and communication bandwidth. For example, when querying data servers for information, a mobile application can hardly anticipate the size of the result set. Our approach is to give more control over the data delivery process to the application, so that it can be adapted regarding its device status, the costs and availability of communication channels, and the user's needs. This paper introduces a flexible and scalable approach by providing spatially federated cursor functionality. It is based on an open federation over a set of loosely coupled data sources that provide simple object retrieval interfaces.*

1. Introduction

With the growth of online accessible data and information systems, the need for integration architectures is increasing. As can be seen in the Web 2.0 trend, more and more information is provided by autonomous data sources, like web sites, Wikis, or web services. To uniformly integrate this information for application use is a very cumbersome, and, at large, almost impossible task. However, when focusing on a certain application domain, we can exploit common characteristics to provide integrated views: e.g., WWW search engines integrate their search results based on rankings that represent the relevance to the user's query. In the Nexus project, we target the upcoming application domain of location-based information services and pervasive computing. Here, new data-intensive applications emerge, which support their users with the right information at the right time and right place, i.e., providing on demand what fits best to the user's current situation [Dey and Abowd 1999]. They often rely on large-scale information systems, where the data is scattered across a multitude of data sources ranging from web sites over digital libraries and geo-information systems to sensors and other stream-based sources. Our integration approach is based on an open federation over a set of loosely coupled data sources that provide simple object retrieval interfaces.

In contrast to conventional distributed database systems, the partitioning of the information is unknown. There is no closed-world assumption, since data providers can dynamically connect and disconnect from the platform. Also, there can be multiple representations of real-world entities in several data providers. Our open federation differs also from conventional federated database systems: since it is based on simple object retrieval and does not provide the full-fledged SQL function set, it does not have to materialize the whole result set within the federation layer when integrating the results from the data providers.

This allows us to develop a scalable algorithm for object retrieval that works on partial results from data providers. It is based on the concept of a federated cursor. Cursors are a long-known database concept that allows an application to piece-wise retrieve tuples of a result set [Date 2000]. This is especially beneficial if the applications run on resource-limited devices, which typically retrieve information over a costly wireless communication channel. Such devices are often used in the areas of location-based services and pervasive computing.

The remainder of the paper is structured as follows: we overview related work in Chapter 2. Chapter 3 describes the Nexus platform which provides an integrated view over geographic data sources in the application domain of location-based and context-aware services. In Chapter 4, we introduce the cursor concept, and in Chapter 5 we present a flexible and efficient federation strategy also covering histograms. Our prototype implementation is evaluated in Chapter 6. Finally, in Chapter 7 we conclude the paper with an outlook on future work in this area.

2. Related Work

There has been some work addressing the problem of efficiently processing and incrementally retrieving partial results. [Haas et al. 1999] try to speed up data intensive applications needing fine-grained object access by loading the cache of the system with relevant objects. The decision of what objects are relevant is taken by the frequency applications access objects. However, this technique does not consider multiple representations of the same object containing incomplete or partial information distributed over several data sources. In this case one has to find and fetch all representations of an object in order to get a complete and consistent object representation.

Garlic [Josifovski et al. 2002] is a platform for federated data management of relational data sources based on IBM DB2. For the incremental retrieval of the result set two possibilities are described. One is to materialize the entire result of each data source. The other is the retrieval of the data using the cursor mechanism. Each time *Fetch* is invoked, one data element a time is retrieved from the data source. Here, no possibility of sophisticated retrieval of the result sets is mentioned. The possibility of incomplete partial results is also not taken into consideration.

In Disco [Tomasic et al. 1996] the problem of dealing with unavailable data sources is addressed. The selected approach uses a *partial evaluation* semantics to return partial answers to queries. Here the portions of the query that could not be answered are mapped back to OQL and integrated with the portions of the query being answered by data providers. It is neither described how exactly the results are retrieved from the data providers nor how the partial results of the portions of the query are integrated to a single answer.

Information Manifold [Levy et al. 1995, Levy et al. 1996] deals with the efficient query processing in a distributed environment involving a large number of data sources. They use descriptions of the data sources for a given query to identify relevant sources, query these sources and finally collect the complete result from these partial results. The query processing engine tries to recognize sources providing redundant information and prunes them. No integration of the partial results or further computations are made. This has to be done by the inquiring application. Also, no further reflection on alternative

retrieval mechanisms were made.

There also exist several mediator-based systems like TSIMMIS [García-Molina et al. 1997] or MedMaker [Papakonstantinou et al. 1996]. However, we focus on location-aware applications using location-based data and provide domain-specific operators and optimizations.

3. The Nexus Platform

The Nexus platform is a federated open system for location-based applications [Nicklas and Mitschang 2004]. As depicted in Figure 1, the Nexus architecture is built up in three tiers: applications, a federation tier containing Nexus nodes and a service tier consisting mainly of context servers, which provide stored or sensed data. Context servers must implement a predefined interface, through which they are contacted by Nexus nodes, and they must register at the Area Service Register (ASR), announcing the area they offer data for. Otherwise the implementation of a context server is not restricted, thus it can easily be tailored to the needs of different kinds of data like positions of vehicles (high update rates) or geometries of buildings (large data volumes) [Grossmann et al. 2005]. Being an open system, adding new context servers to the Nexus platform is not restricted. In particular, it is possible that the data of a new context server overlaps with existing ones in both its service area and content, which can lead to multiply represented objects (MReps). When integrating different result sets from different context servers, Nexus nodes try to detect such multiple representations based on location-based criteria and merge them into a single object [Volz and Walter 2004, Volz 2006]. In the following, the term *federation* is used as a synonym for Nexus nodes.

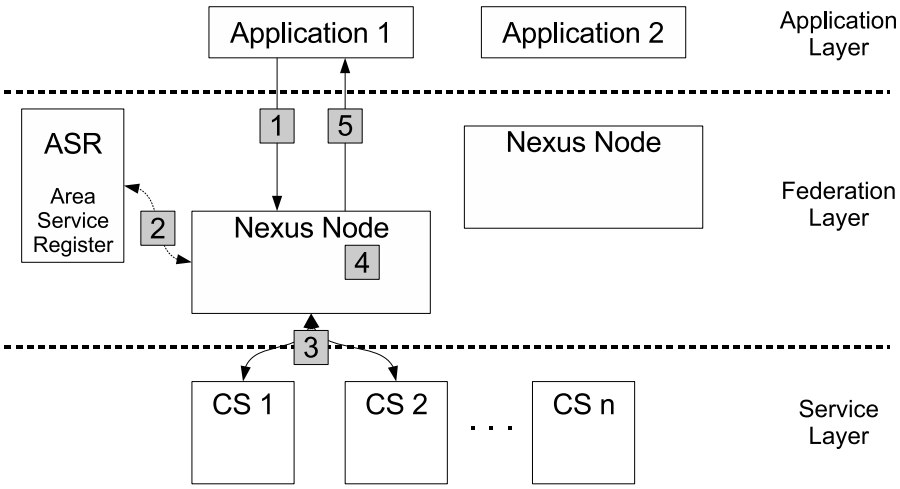


Figure 1. Overview of the original Nexus architecture

The Nexus platform uses a request-response protocol in which queries typically contain a spatial restriction. The processing model is depicted in Figure 1:

1. An application sends a query like *Menu and position of all restaurants closer than 1 mile to my current position* to an arbitrary Nexus node.
2. The Nexus node determines the relevant context servers by an ASR lookup based on the spatial restriction and the queried object type. In the example above the

spatial restriction corresponds to *closer than 1 mile to my current position* and the object type to *restaurants*.
3. The Nexus node forwards the query to those context servers. The context servers process the query and send back their results.
4. The Nexus node integrates the context servers' results. It detects and merges multiply represented objects (MReps). For this, domain-specific methods are used that exploit the spatial structure of the data: only objects in a spatial vicinity are considered candidates for being MReps.
5. The Nexus node returns the integrated result to the application.

Initially, this processing model computes and forwards the full result to the requesting application, which has no control over the data transfer. Obviously, when transferring large results, this can overburden resource-limited devices. For this reason we developed a better suited processing model based on the federated cursor concept.

4. The Federated Nexus Cursor Concept

The main idea for this federated cursor approach is to

- request only context servers that actively contribute
- process only the necessary result data
- support temporarily disconnected applications
- support mobility of applications (free choice of Nexus node).

In the past, cursors were used to bridge the so-called "impedance mismatch" between database systems and programming languages. The cursor allows conventional programming languages to cope with tuple-wise processing by providing a pointer to the actual tuple to be worked up.

The idea has been retained but applied to the domain of a federated, context-aware platform. Here, the cursor concept is used for retrieving partial results of spatial queries in order to prevent memory overflow and to save communication bandwidth, thus bridging the "resource mismatch" between often resource-limited mobile devices and the 'unlimited' server infrastructure.

Using a cursor, an application does not have to wait until the entire result is transferred before processing it. Depending on the type of connection there may be unwanted disconnections: the larger the result, the higher the risk that the full result never reaches the querying application. Also, certain network paths may be expensive.

One way to overcome this problem is to partition the spatial query into many small disjoint sub-queries and post each sub-query subsequently. However, this approach requires knowledge about the results' size of each sub-query which (without the necessary information) cannot be clearly predicted. Thus, the way the initial query should be partitioned is unclear. With a cursor no partitioning is necessary. Instead the result is partitioned with respect to the application needs, which is especially useful if the result is suitably ordered.

In the subsequent chapters the general concept of a federated, status-conscious cursor is introduced, which is used to efficiently retrieve objects over distributed data sources by exploiting the spatial nature of the data.

4.1. Query Processing Sequence

Up to now, a Nexus application posts a spatial query and receives an answer consisting of the query result. The new cursor-based processing model is three-phased. This is analogous to the cursor processing as described in [Date 2000]. The application has to post a query associated with a cursor on the query's result. After that, the application can start to piece-wise process the result. In the end, the result is deleted, if either its lifetime has expired or the application signals that it is no longer needed.

4.1.1. Phase 1: Initialization Phase

In the initialization phase preparations for the next phase (delivery phase) are made. The necessary steps are as follows (cf. Figure 2):

1. An application sends a spatial query to an arbitrary Nexus node and additionally asks the federation to create a cursor on the query's result.
2. The Nexus node determines the relevant context servers by an ASR lookup based on the spatial restriction and the object type in the query.
3. The Nexus node forwards the query to those context servers, which process the query and send their results back. Additionally, the federation sends back an ID (called Nexus Session Locator, see below) of that cursor to the application.

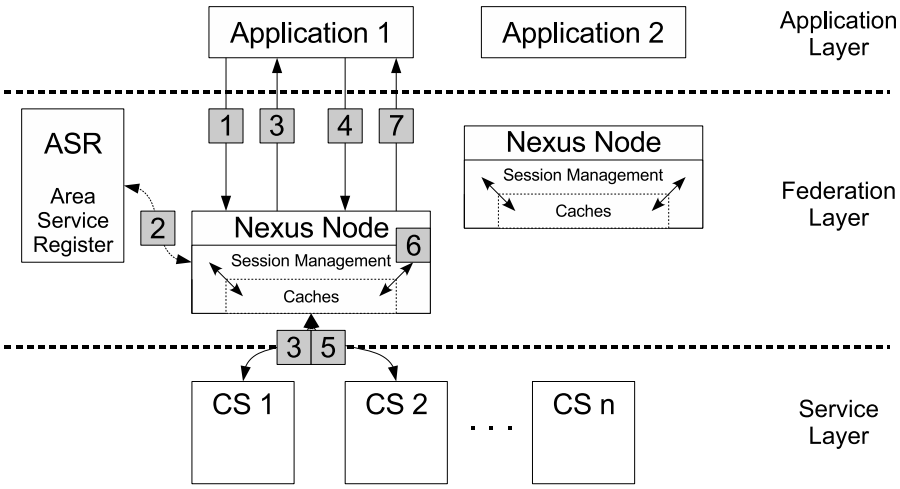


Figure 2. Overview of the new architecture

4.1.2. Phase 2: Delivery Phase

If the initialization phase has been successfully completed, the application is able to send cursor operations on its cursors to the federation to give access to the result data piece by piece. This phase is called delivery phase. The necessary steps are as follows:

4. The application posts a *next* operation stating the next elements pertaining to a certain result identified by an ID.

5. The federation looks for the result belonging to the ID and prepares the objects that go to the result set. Objects have to be retrieved from the context servers (if they are not already in cache).
6. Multiply represented objects have to be detected and merged. This operation can reduce the number of objects.
7. The result set is sent back to the application.

The application repeats the delivery phase until the end is reached or it does not need any further elements and decides to finish the retrieval. If the application signals that decision, the federation enters the termination phase.

4.1.3. Phase 3: Termination Phase

This final phase is entered if the lifetime of the result has expired or if the application signals the federation that it does not need more elements. The resources connected to the ID are released.

4.2. Session Management

For identifying sessions within the Nexus platform we introduce the so-called Nexus Session Locator (NSL). A NSL consists of two parts: a basic service part, which encodes the Nexus node the session was created on and thus holds the session information and a session identifier (SID). The hosting node is encoded within the NSL to support distributed session management: since we want to support mobile devices, an application can change its Nexus node. Using the NSL, a Nexus node that receives a cursor query for a cursor that it does not host can easily forward that query to the correct node. If a mobile device changes the Nexus node during operation, the new node has to retrieve the specific application information from the relevant Nexus node encoded in the NSL in order to be able to process the request adequately. For this, there are two possibilities: one is to transfer all relevant information to the new node (incrementally or at once) and to replace the part of the NSL storing the host with the new host address. The other is to always forward the query to the original node.

5. Federated Processing Strategies

After introducing the general federated cursor concept, we discuss federated processing strategies. In order to optimize query processing, the context servers should support a cursor concept, too. It is an optional feature of a context server. Without that functionality the entire result from each context server has to be transferred to the federation. In Figure 2, the context servers are also extended by a session management in order to be able to hold application specific information. In that way, the results can be kept at each context server locally and objects just needed will be transferred to the federation.

To provide optimal response times for applications, the federation should precache partial results [Haas et al. 1999]. There exist several ways to do this. The naive approach is to query all relevant context servers and cache all results locally. This approach has the advantage that there is no more communication overhead between the federation and the context servers and long latencies for query answering are avoided. But it suffers

from high memory consumption within the federation layer and a long initialization phase since the results of all context servers must be fetched.

To reduce the memory consumption at the federation layer, spatially portioned queries can be sent to the context servers. Here the initialization phase consists of the non-trivial problem of partitioning the query. Objects may be queried that currently are not needed and in worst case never needed and it must be taken care that all multiply represented objects are present for the merge operation at processing time.

Both solutions sketched above are not recommendable. One suffers from memory consumption in the federation layer. The other suffers from communication overhead between the federation and context servers and could also miss information for some objects. So there is a trade off between memory consumption and the system load.

5.1. Cache Histograms

A major feature of the Nexus federation is the merging of multiply represented objects (MReps). To correctly perform this operation also in the cursor mode, we have to pre-cache partial results in a way that all candidates for a MRep-merge are present whenever this operation is carried out. The naive way would be to pre-cache the whole result at federation level. However, this introduces an often unnecessary memory usage at federation level and communication overhead between federation and context servers, particularly if an application does not retrieve the whole result.

We use cache histograms to solve that problem in an efficient way. A single cache histogram represents the query-dependent frequency distribution of the resulting objects based on a sorting criterion (i.e., the distance from a geographical point). Cache histograms are provided by each context server. A cache histogram consists of a set of cache histogram entries. Each cache histogram entry consists of a bucket value which indicates how the partial result of a context server was sorted and the amount of occurrences of that bucket within that partial result. A bucket here refers to a discrete point in the sorting domain and not to an interval as usual.

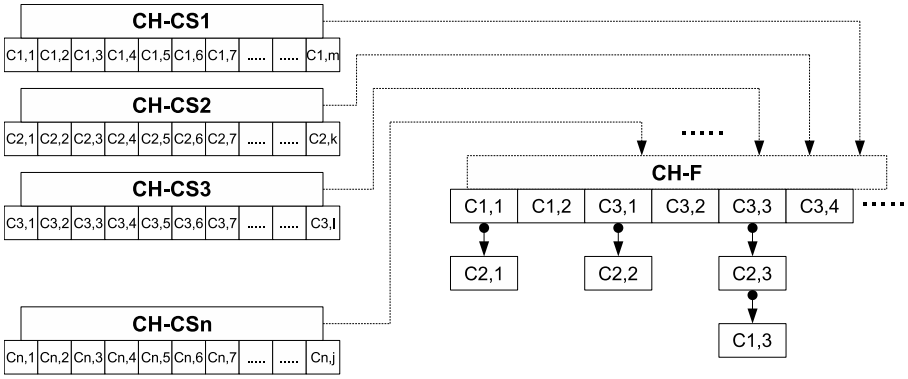


Figure 3. Federated cache strategy using cache histograms

As shown in Figure 3, each context server delivers a cache histogram (CH-CS1 to CH-CSn) which is spatially sorted. $C_{a,b}$ corresponds to a cache histogram entry and gives the value of the cache entry and its frequency of occurrence. In our example the value refers to the distance of an object to the reference point, e.g., $C_{1,1}$, with a value of

<17,5>, addresses 5 nearest objects from context server 1 with a distance of 17 to that reference point.

If cache histograms supplied by the corresponding context servers are not already sorted by the bucket value, the federation has to do it by itself. That may occur if the cache histogram is created before sorting the partial result or the context server does not support sorting at all. Usually the context servers support result sorting. If no sorting criterion is specified, the federation has to sort it. The federation now merges the cache histograms delivered by each data context server into a federated cache histogram in order to get an overall overview (CH-F) of all data sources involved in the incremental retrieval process. The most important information at this point is the order in which the context servers should be queried, what context servers have to be queried and the quantity of objects (which is encoded in the cache histogram entries) to query the context servers for.

Since there may be multiple representations for the same real world entity, there can be objects with the same sorting value in different cache histogram. In that case these entries are stored as a linked list as shown in Figure 3. Elements in the linked list potentially represent the same object. All these objects must be transferred to the federation in order to guarantee a lossless merge. Whether two or more entries in the linked list represent the same object is decided by the federation's merger algorithm. Here $C_{1,1}$ and $C_{2,1}$ got the same bucket value and are thus stored as linked list. Taking for example $C_{1,1}$ with a value of <17,5> and $C_{2,1}$ with a value of <17,3>, the federation would first ask context server 1 for the next 5 objects and then context server 2 for the next 3 objects each with a distance of 17 to the reference point.

Listing 1 shows the cache histogram algorithm in pseudocode. It is used by the federation to build up a federated cache histogram.

Listing 1. The cache histogram algorithm

```
// application sends query to system
receive application query

// determine the relevant context servers
ask ASR for relevant context servers

// answer
send NSL to application

// send query to all necessary sources
for each context server do
    forward query
    // get cache histograms from each context server
    receive the cache histogram
    // eventually sort them
    if cache histogram not sorted
        sort cache histogram

// merge the cache histograms
merge cache histograms to federated cache histogram
```

5.2. The Retrieval Process Using Cache Histograms

Internally, the cursor is split in an horizontal (H) and vertical (V) component. The H component traverses the cache histogram from left to right, the V component from top to bottom. The algorithm is shown in Figure 4 for a *next* operation. The initial state of the algorithm is displayed in the upper left. The H component corresponds to the current

cursor position. The V component indicates the position within the linked list of elements with the same bucket value.

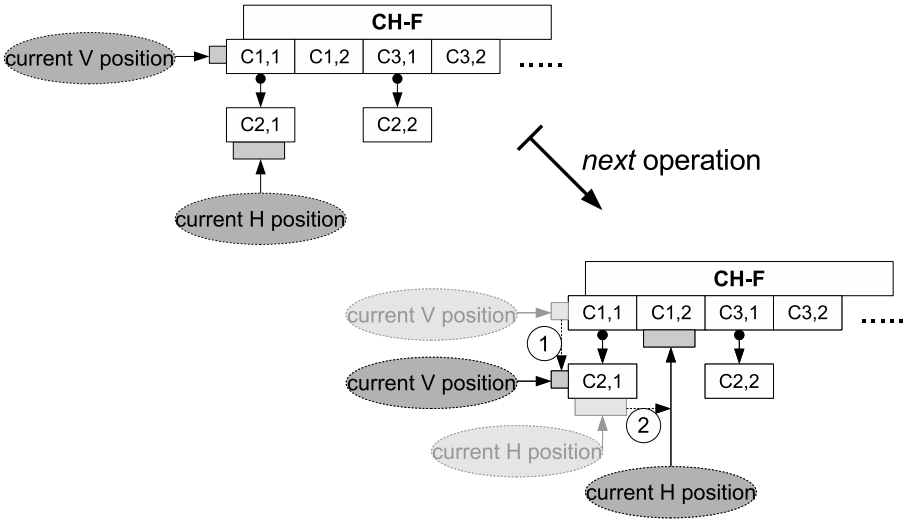


Figure 4. One cache histogram retrieval step

On the bottom right side the next two steps of the algorithm are displayed. First all elements in the linked list have to be processed (step 1), to ensure that all representations of the same object are retrieved.

The next step is to move the H-position by one to the right (step 2). If the linked list here has also got more than one element then step 1 is repeated. Otherwise step 2 is repeated.

The algorithm ends filling the federation caches if the amount of objects needed to answer the previously posted query is reached or if there are no more objects to retrieve. In the last case, a message stating that there are no more objects is sent to the inquiring application.

The algorithm works in an efficient way in terms of memory consumption and network load because only relevant context servers are queried for objects, irrelevant context servers are not considered as the federation only retrieves objects as a result of *next* operations. Furthermore no multiply represented objects are missed. Listing 2 shows a simplified version of the retrieval process using cache histograms.

Listing 2. Retrieval process algorithm using cache histogram

```
// application requests next N objects
K := number of objects in output buffer
PL := []
do N - K times
    // output buffer does not contain enough objects
    if V-component points to cache histogram entry
        P := context server in current cache histogram entry
        M := bucket size in current cache histogram entry
        if P in PL
            increment number of objects to fetch from P by M
        else
            append P to PL
            set number of objects to fetch from P to M
        move V-component one step down
    else
```

```
        move H-component one step right
OL := []
for each P in PL do
    retrieve the given number of objects from P
    append objects to OL
merge objects in OL
append OL to output buffer
remove first N objects from output buffer
send removed objects to application
```

6. Experience and Evaluation

Considering scenarios where mobile devices are forced to piecewise retrieve result sets due to memory limitations of the device, extending the Nexus platform by cursors is clearly an enhancement. Without a cursor, such devices would have to send the same query multiple times to a Nexus node, receive the complete result set each time but only process the appropriate subset and ignore the rest, which is obviously inferior wrt. overall query processing time, data volume transferred and overall energy consumption. In order to assess the overhead involved with the cursor concept, we conducted a suite of experiments to show that the additional overhead caused by the cursor management and histogram calculations is comparatively small.

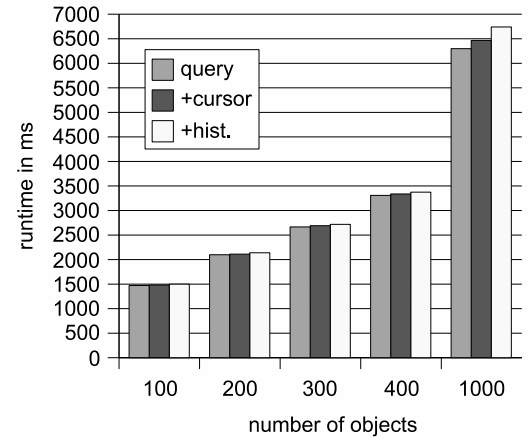


Figure 5. Run times for different result set sizes

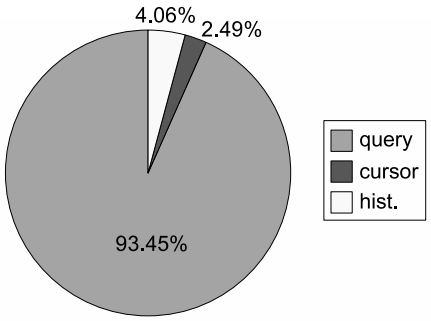


Figure 6. Runtime fractions for cursor management and histograms of 1000 objects query

The context server used for the experiments is implemented in Java and was running on a SUN Blade 2000 with two 1.2 GHz UltraSPARC III CPUs and 6GB of RAM. IBM DB2 8.1.3 was used as the backend system for storing the data. The database contained 3380 objects in total. Figure 5 shows the runtimes of a nearest-neighbor-query with sorting by distance from a reference point. We varied the number of objects to retrieve between 100 and 1000. *query* refers to query processing alone, *+cursor* additionally creates a cursor and *+hist.* furthermore computes a histogram. Figure 6 shows the fractions of the runtime required for processing the query, creating a cursor and computing a histogram for the 1000 objects query. The extra overhead is below 7%. This fraction is even lower for smaller result sets, approximately between 0.5% and 2%.

7. Conclusion

In this paper, we introduced a generic federated cursor concept, explained the underlying idea, and applied it to an integration architecture, the Nexus platform. The main idea is to request just those context servers that actively contribute and to process only necessary result data in order to reduce memory consumption and transmission volume over the network.

Multiple object representation is an additional problem to deal with since no information should be missed in order to achieve a lossless merge of multiple representations. It was solved by introducing the novel approach of cache histograms representing a query-dependent frequency distribution of the resulting objects based on some sorting criterion.

Finally, temporarily disconnected applications due to faulty mobile connections are supported. This problem was solved using sessions. In this way applications are able to reconnect at later time and at arbitrary connection points.

Our prototype evaluation and measurements indicated that the overhead introduced by the cursor concept is in the low percentage range. This is clearly acceptable in view of the benefits the cursor concept introduces to the applications and the federation layer, e.g. availability, disconnection and partial evaluation.

However, the approach suffers from assumptions that may not always be correct. First, it assumes that stored values of objects are exact. That implies that the ordering is always the same for each object and the corresponding bucket, but that is not always correct. For example, if the objects are sorted by distance to the application's reference point there could be some divergence between the position values or between the calculated distances to that reference point. The introduction of an interval could eliminate the error and minimize communication overhead between federation and context servers. Such an interval can be calculated on statistical values the federation (or some other component) has to collect in advance. Here the problem consists of dynamically finding convenient ranges for each bucket.

References

Date, C. J. (2000). *An Introduction to Database Systems*. Addison Wesley Longman, 17th edition.

Dey, A. and Abowd, G. (1999). Towards a better understanding of context and context-awareness. Technical Report GIT-GVU-99-22, Georgia Tech GVU.

García-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J., Vassalos, V., and Widom, J. (1997). The TSIMMIS approach to mediation: Data models and languages. *Journal of Intelligent Information Systems*, 8(2):117–132.

Grossmann, M., Bauer, M., Hönle, N., Käppeler, U.-P., Nicklas, D., and Schwarz, T. (2005). Efficiently managing context information for large-scale scenarios. In *3rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2005), 8-12 March 2005, Kauai Island, HI, USA*, pages 331–340. IEEE Computer Society.

Haas, L. M., Kossmann, D., and Ursu, I. (1999). Loading a cache with query results. In *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*, pages 351–362, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Josifovski, V., Schwarz, P., Haas, L., and Lin, E. (2002). Garlic: a new flavor of federated query processing for DB2. In *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 524–532, New York, NY, USA. ACM Press.

Levy, A. Y., Rajaraman, A., and Ordille, J. J. (1996). Querying heterogeneous information sources using source descriptions. In *VLDB '96: Proceedings of the 22th International Conference on Very Large Data Bases*, pages 251–262, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Levy, A. Y., Srivastava, D., and Kirk, T. (1995). Data model and query evaluation in global information systems. *Journal of Intelligent Information Systems - Special Issue on Networked Information Discovery and Retrieval*, 5(2):121–143.

Nicklas, D. and Mitschang, B. (2004). On building location aware applications using an open platform based on the NEXUS augmented world model. *Software and System Modeling*, 3:303–313.

Papakonstantinou, Y., García-Molina, H., and Ullman, J. (1996). MedMaker: A mediation system based on declarative specifications. In *Proceedings of the Twelfth International Conference on Data Engineering, February 26 - March 1, 1996, New Orleans, Louisiana*, pages 132–141. IEEE Computer Society.

Tomasic, A., Raschid, L., and Valduriez, P. (1996). Scaling heterogeneous databases and the design of Disco. In *ICDCS '96: Proceedings of the 16th International Conference on Distributed Computing Systems (ICDCS '96)*, pages 449–457, Washington, DC, USA. IEEE Computer Society.

Volz, S. (2006). An iterative approach for matching multiple representations of street data. In *Proc. of the JOINT ISPRS Workshop on Multiple Representations and Interoperability of Spatial Data*, volume XXXVI Part 2/W40.

Volz, S. and Walter, V. (2004). Linking different geospatial databases by explicit relations. In *Proceedings of the XXth Conference of ISPRS '04*.

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 97-108.

A Service-Oriented Architecture for Progressive Transmission of Maps

David Cavassana Costa[1], Mario Meireles Teixeira[1], Anselmo Cardoso de Paiva[1], Claudio de Souza Baptista[2]

[1]Departamento de Informática – Universidade Federal do Maranhão (UFMA)
Av. dos Portugueses SN – 65.085-580 – São Luis – MA – Brazil

[2]Departamento de Sistemas e Computação
Universidade Federal de Campina Grande (UFCG) – Campina Grande, MA – Brazil

david@gia.deinf.ufma.br, {mario, paiva}@deinf.ufma.br,
baptista@dsc.ufcg.edu.br

Abstract. *The Internet creates an environment suitable to spatial data share, allowing the users to transmit, visualize, manipulate and interact with them. This environment not only allows new opportunities for geospatial data use, but also introduces new problems that must be managed to permit these data to be used in an effective and useful way. One of such problems is related to the use of these data with small network bandwidth. This work presents an architecture based on Gis Services for the progressive transmission of vector maps in the Internet, anticipating the rasterization process in the server side, thus reducing the amount of data to be transmitted to the client. The architecture proposed uses all the advantages of Gis Services, that are becoming a standard in the construction of Gis applications based on services.*

1. INTRODUCTION

Visualization plays an important role for a better understanding of phenomena in several areas of knowledge and it is so with geographic data. Geographic maps are used long ago for viewing spatial data, helping us to better understand the relationships among those data. In the field of cartography, the visualization process must be formalized through the definition of rules and principles, once different kinds of data can be viewed in different ways [Cecconi 2003]. Such methods and techniques must be used as to optimize their use and validate the data consistency.

Geographic Information Systems (GIS) are computational tools used either to make available and/or analyze information associated with position/localization over the Earth Surface. Currently, the Internet has become a huge content publishing media; being thus a favorable environment for the GIS users to exchange data, perform analyses and present geographic results. Geographic information in the Internet has rapidly evolved with the development of Web technologies.

Internet creates a new geospatial data sharing environment, where data suppliers turn available their geographical databases in a way analogous to the Web pages' textual information, allowing the users to use the Web for data transferring and utilize them for visualizing, analysis and/or manipulation [Bertolotto 1999]. This new configuration supplies new opportunities, for both, the public and the commercial domains, of using

geographic data sets. On the other side, new problems arise from such architecture, for example, the availability of large amounts of data stored in repositories lacking efficient transfer methods.

Among the emergent technologies one can cite the Web Services, which allow the building of Web applications flexible, inter-operational and reusable, enabling interactions between those applications. Web Services are software modules identified by an URI that offer services to remote applications called consumers, using the Internet as communication channel [W3C 2002], [W3C 2004].

GIS Services are services of Geographic Information Systems implemented by means of Web Services that perform a specific GIS function which can be integrated as part of one or more applications [Yumin 2004]. GIS Services is often associated to the Web Services based GIS.

Even with the growing Internet users' access to a bigger bandwidth, there are still situations where the transmission cost is a critical factor, as in the case of wireless networks or dialed access, especially in under developed countries or localities where there is no chance of using a dedicated connection. Those adverse characteristics encouraged the appearing of various methods for efficient data transmission, necessary to minimize the response time, enabling the own publication of vector based geographical information in systems with low transmission rates.

Once all geographic data tends to be very bulky in the majority of geographic data banks, the transfer process may force the user to wait a long time or even it may become prohibitive. To solve these problems, techniques such as progressive transfer and generalization can be rather feasible solutions. Such techniques are being proposed, evaluated and combined nowadays aiming at solving or, at least, soften the problem of latency in the Internet geographic map transmission. The complexity of techniques and algorithms meant for solving such problem is one the major motivations for research in that area, once each approach has different focus and scope.

Several methods have being proposed as to progressively transmit vector spatial data [Cecconi 2004], [Bertolotto 1999], [Buttenfield 2002], [Oh 1999], [Han 2004], [Yang 2004], [Yumin 2004]. Such methods are based on the idea that the user, in general, gradually consumes the map's information, starting from low detail levels scaling to higher ones. Based on that evidence, it is possible to generate maps in a certain way, dividing them and gradually transmiting them.

In the progressive transmission, the maps server divides the maps into a low resolution version and a set of incremental versions that, when incorporated into a certain maps' version, generates a more detailed version of that map. The client is in charge of receiving the maps' detail increments in such a level n, and integrating them into the actual version of the map, generating a map version at the level n+1.

Other approaches to that problem exploit the device's constraints, such as the visualization resolution. This can determine the level of details that can be visualized in a specific device. In general, the device can present fewer details than the ones sent by the maps server. The detail level that can not be visualized increases the transmission cost and does not enhance the map's quality being viewed by the client. This way, the removal of this redundant information minimizes the client's response time without barring the map's quality [Liang 2001].

Our work proposes a Web Service Oriented architecture for a geographic map server with support to progressive transmission, using combined techniques of generalization, content adaptation and progressive transmission, in order to efficiently enhance the map transmission process from the server to the client with minimum impact in the data consistency, increasing the application's usability and minimizing the response time experienced by the user by means of modular inter-operational and reusable components, distributed between client and server.

The remaining of the article is organized as follows: Section 2 presents some of the main work related with Web Services and GIS Services. Section 3 presents the used GIS's framework architecture and the transmission maps architecture handled for implementing the service. It also presents the proposed Maps Service, detailing its architecture, usage scenario, service interface and an implemented client application. Section 4 presents the performance data relative to the results and measurements taken with the new proposed architecture and, finally, Section 5 discusses the conclusions of this work, suggesting some feasible future enhancements.

2. SUPPORT ARCHITECTURES

This work aims is to present an architecture based on GIS Services for a map service that also have progressive map transmission capabillities. Before the proposed architecture presentation, we will present two other architectures that used as the basis for our work. The first one is the iGIS architecture. iGis is a GIS framework for Web map publishing. It is used for the map generation that are made available by the GIS Service. Following, we also describe the progressive transmission architecture used bye the proposed GIS Service.

2.1 iGIS Architecture

The iGIS [Baptista 2004] is a framework with a three layer architecture, aiming at implementing a Geographic Information System based upon the Web following the OpenGeoSpatial standard. As a framework, the iGIS allows the rapid applications' development of Web based geographical information systems.

The iGIS architecture was designed according to the MVC (Model-View-Controller) architecture standard in three layers as depicted in Figure 2. In the presentation layer, the Java Server Pages (JSP) technology is used as to implement dynamic pages. Besides that, SVG and Javascript are used for exhibiting the map, visualization tools and the map processing.

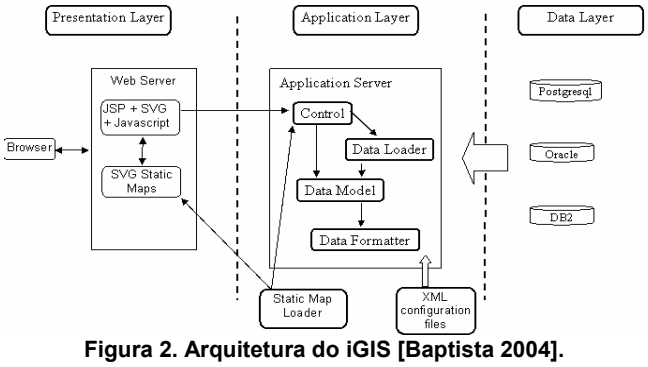


Figura 2. Arquitetura do iGIS [Baptista 2004].

The data layer is represented by a special instance of a database named DataSource. The iGIS can communicate with different DataSources, for example, Oracle, Postgresql, IBM DB2 shapefiles from ESRI. That layer, among others, is responsible for the database independence when handling geographic data. The application layer is responsible for the business logic. That layer manipulates the clients' requests turning them into the OGC model and, finally, gives them a format in an instance of the introduction/presentation layer. This layer is compounded by the following modules: Control, Data Loader and Data Formatter. The data loader is responsible for the data loading from different data sources, which are configured by means of XML files. The control module defines the necessary action sequence for meeting the clients' request, and returns the properly formatted data back to the presentation layer.

The Data Formatter is the module that assures the independence between the data manipulation and presentation. Currently, the iGIS supports formatters for vector and raster data.

As a framework, the iGIS has some extension points. One of them is the Map Formatter, which is useful for implementing a progressive transmission architecture by reusing all the functionalities reminiscent in the framework. Thus, the progressive transmission architecture was implemented within the Formatter package, generating a new sub-pack called "Progressive".

2.2 Progressive Transmission Architecture

As a means of providing the internet with an efficient vector maps transmission and visualization process, is necessary to render the maps into different detail levels. Our approach is based on the progressive map transmission, considering the characteristics of the visualization device, especially its resolution.

The progressive transmission architecture used in this work is detailed in [Costa 2006], and is based on the idea that it is possible to transmit only what can be visualized, progressively, in a similar way as proposed by Liang (2001). However, the used solution is based on a more generic approach that considers the map's low level information, the point's coordinates and the visualization resolution in the device.

By using the maps' progressive transmission, the user must initially work in a low resolution version of the map in order to localize the area that most interest him/her. Then, as soon as a visualization operation is done, e.g., zoom, the user views the interest area with better resolution in an incremental way.

The main idea of the progressive algorithm is to take into account the device's target-resolution. Hence, only the strictly necessary map information to generate a correct visualization in that resolution is sent. That approach removes non visible points and lines.

In the map's rendering process, the points change from map's coordinates to device's coordinates. In this process, some points are transformed and mapped into the same pixel. The used algorithm anticipates that process, removing from the polygon or from a line, consecutive points that would be mapped into the same pixel whenever the map would become rendered.

3. Service-Based Progressive Map Transmission Server

3.1 General Architecture

The progressive transmission GIS Service uses the iGIS framework and its progressive maps generation process starting from spatial database. Firstly, map data is loaded from the database to be published. The published content is defined through XML files that specify the database, the tables and layers of the map to be published. That loading originates several files for the same map. One of them is concerned with the lower resolution map, the others contain the increments and details able to transform a map from a n level to a n + 1 level by means of data integration operations.

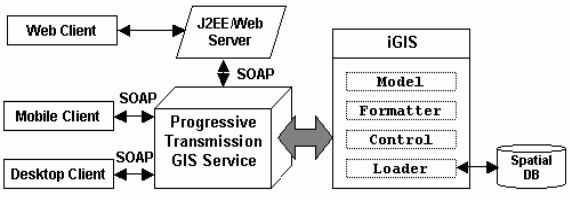


Figure 3. The Proposed GIS Service Architecture

Once the files are generated, the map can be accessed by the GIS Service through their methods, as shown in Figure 3. Initially, the Web Service enables the acquisition of a map in a low version, and later permits the acquisition of a more detailed map for a given resolution level, enhancing its resolution. Yet, there is the possibility for the client to request details of only determined layers of the map, for example request more details about highways; but not so many details of the municipality limits. Interactions between client and map server is done through the SOAP protocol. Thus, the client can be mobile, desktop or even a dynamic Web page. Therein, the iGIS architecture is condensed into just one block, and the implemented GIS Service uses its infrastructure as to provide its services to progressive map clients.

The client must be able of receiving those data and integrate them into the maps they already have in its cache. When receiving data from the increments he/she conducts DOM calls to the SVG map, generating more detailed versions of the related SVG paths, those ones that must be updated while passing from the n level to the n + 1 one.

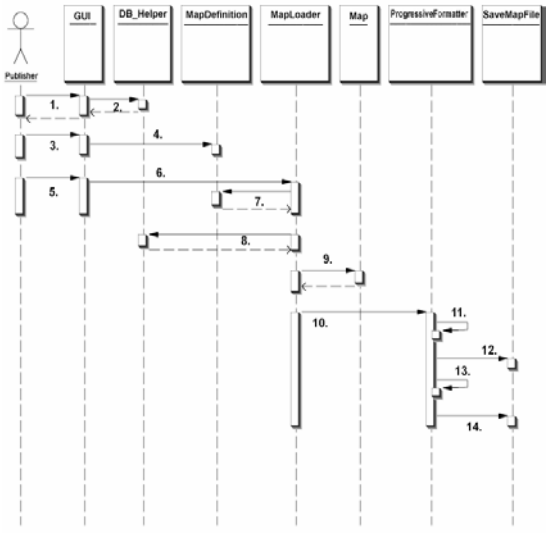
GIS Services proposes to turn available the maps delivery service supporting progressive transmission, independent of platform or programming language. The only restriction to the client is that it must be able of rendering the map into SVG and update it via DOM operations, which is possible for Web clients, Java implemented Desktop clients, .NET or, for instance, even mobile clients. The service is the same and its use can be adapted according to the application. One advantage of its use by Desktop clients is the performance gain got with respect to the Web client, which is up to 25 times faster in terms of processing and 5 times quicker in terms of total response, results got in our tests. As we will see in the results Section, that happens because, in the case of the Web client, DOM/Javascript operations make the map integration process a little bit slower [8] compared with the desktop client, but not until the extent of not justifying its application also to Web clients.

The GIS Service guarantees the inter-operability, platform and implementation language independency, assuring to the clients higher flexibility. A Java client

application may even be in need of more performance as to guarantee higher service quality, for example in the case of critical applications. The implementation of a new client, native for the target platform, can already be sufficient to promote higher performance. Yet, even the data integration algorithm [8] can be modified as to optimize resources usage, in a transparent way for the server, who continues making available its data through its standardized methods with a given format. Furthermore, the progressive transmission service may be integrated in othe Gis systems or can be added other modules to it.

3.2 Maps Generation and client/server interaction Scenario

Initially, maps generation from spatial database must be conducted. This is done through the iGIS Framework, whose task flow is briefly illustrated in Figure 4:



1.Load database maps.
2.Recover metadata from the database.
3.Select map and map layers to be published.
4.Save selected definitions in XML format.
5.Initiate map publishing process.
6.Initiate Map Loader.
7.Recover map definitions.
8.Load spatial data from the database.
9.Create a map's data structure in main memory.
10.Give a format to the map in progressive fashion.
11.Calculate low resolution map.
12.Save low resolution map.
13.Calculate map increments (details).
14.Save map increments into a file.

Figure 4. Progressive maps generation scenario

Once the maps have been generated and their details properly decomposed, the Progressive maps transmission GIS Service starts to work.

3.3 GIS Service Interface

Following, the progressive maps server methods, together with a brief explanation on the behavior of the performed operation, are presented.

String getMapNames() and *String getMapNames(int xMin, int xMax, int yMin, int yMax)*: Obtains the maps' names with support in the progressive transmission stored in the server. The map name will be used by the other methods identifying just a single map. The difference between both methods is that to the second one is assigned as parameter a given rectangular area (box), and only the maps contained within that area will be returned.

String getMapDescription(String mapName): Obtains information on the map with/through the given/defined name. That information is only used by the end user and is not meant to be processed by other services.

String getLowResMap(String mapName): Obtains the map in its lower resolution and less detailed version in the SVG format. Initially, this version is exhibited to the client and, as long as he/she requests more details, the SVG map will be updated via DOM operations, by applying the client's side integration algorithm.

String getLayerNames(String mapName) and String getLayerNames(String mapName, int xMin, int xMax, int yMin, int yMax): Gets the layer names of a given map that will be used by other service's methods. To the second version is assigned as parameter a given rectangular area (box), and only the maps contained within that area will be returned.

String getLayerDescription(String mapName, String layerName): Obtains information about a given legend of a given map. Similar to the information method about the map, those info will only be used by the end user.

String getNextLayerIncr(String mapName, String layerName, int level) and String getNextLayerIncr(String mapName, int level): Gets the details of a given map layer for a given resolution level. The second method is similar to the former, but it gets details on all the layers of a map for a given level. Those details will be processed and integrated to the map already held in the client's cache through DOM operations of the map in SVG.

int getLevels(String mapName): Gets the number of levels (versions) of a given map.

String getMinResolution(String mapName): Obtains the map's lower resolution, allowing the client to adjust his/her viewing window to the less detailed map sent at the beginning of the progressive transmission process.

String getLevelRes(String mapName, int level): Obtains the resolution of a given map level. Similar to the prior method, it helps the client to adjust his/her window when a map is received and integrated, increasing its detail level.

String mapToScreen(double x, double y) and String screenToMap(double x, double y) : Turns a point into real coordinates for a map point in the monitor and in the monitor into a point in real coordinates.

String getDetails(int x, int y): Gets information about a given map point, i.e., information on the objects to which that point belongs to, as for example, rivers, highways and cities existing in that point.

3.4 Reference Application

A desktop client and a Web client were developed as to test the implemented services' functionalities. Additionally, we also conducted a performance evaluation between the Web-based approach, described in [Costa 2006], and the new architecture, based on Web Services. The Axis [Apache 2007] was used to generate the Web Services, together with the Tomcat 5.5 server.

Most of the client application functionalities were previously described, as the application is just a prototype of the implemented services.

Nevertheless, we must highlight the way in which the details request is done. It can be conducted in two ways: synchronously and asynchronously.

In the asynchronous approach, the client is free to perform other tasks while waiting for the increments arrival of a given layer. For example, that is practical in the case the client is interested in the details of the all the map layers. In this scenario, several asynchronous requests of the map's layers details are done via Web services, and as soon as they are received, they are integrated into the map, while details of other layers continue to be obtained, allowing the client to perform other tasks besides the processing and details integration. In addition, the rest of the application consists basically of calls for services previously detailed. The Web client was implemented aiming at comparing its performance against a Desktop client, as will be detailed in the following section.

4. Performance Evaluation

The tests presented herein aim at comparing two progressive transmission approaches, using the same maps and the same technique. The first approach is currently used by the iGIS, described in Section 3.1 and detailed in [Costa 2006], which operates independent ly of the implemented GIS services. The second approach uses the GIS Services architecture proposed in this work. The aim of this evaluation is to compare these two approaches performance and observe the gain obtained with the GIS Services technique, which guarantees flexibility and platform and programming language independence to the client.

Table 1. Maps size and respective transmission times

		Brasil1	Brasil2
Original size (KB)		19139	35847
Quantized Size (KB)		6728	11223
Transm. Time Original Map (s)	**5KB/s (s)**	3827,8	7169,4
	12KB/s (s)	1594,9	2987,3
	25KB/s (s)	765,6	1433,9
Transm. Time Quant. Map (s)	**5KB/s (s)**	1345,6	2244,6
	12KB/s (s)	560,7	935,3
	25KB/s (s)	269,1	448,9

Two different maps were used (Brazil1, and Brazil2, with information on the Brazilian Political Division, whose real and quantized sizes are arranged in Table 1. The real size does not use any technique for efficient map transmission, and the quantized size concerns the map generated by the quantization technique for the chosen maximum resolution. Besides that, the maps transmission times are shown in the Table 1, in order to compare it to other tables that show data relative to those maps using the progressive transmission technique in both approaches.

We intended to measure the extent of the gain via desktop client architecture as compared with the Web client using Javascript processing. That can be verified in Tables 2, 3 and 4. In Table 2, column Incr. represents the increments size for a given map level. Column TJS is the Javascript approach, while column TGS represents the desktop client processing time using GIS Services. The last column represents the Gis Service approach gain with respect to Javascript, meaning how faster the former is than the later. The processing time for the map at level 0 is zero, since the server sends a low resolution map to the client, avoiding any integration processing for map rendering.

Notice that the maps' rendering time was neglected once it did not present much variation in the two approaches.

Table 2. Processing times comparison between clients for Brazil 2 map

Brasil2	Resolution	Incr. (KB)	TJS (s)	TGS (s)	TJS /TGS (gain)
level 0	75x75	1476	0	0	0
level 1	150x150	2876	527,86	20,51	25,74
level 2	300x300	3076	542,28	24,10	22,50
level 3	600x600	3012	555,98	22,88	24,30
level 4	1200x1200	3103	561,13	24,57	22,84
level 5	2400x2400	2917	557,16	24,52	22,72
level 6	4800x4800	2094	540,93	25,53	21,19

In addition, we obtained some additional information concerning the GIS service architecture's real gain compared to the use by Web Javascript clients, weighing the response time (increments transmission time plus their processing time to integrate them into the map). To do that, the results were simulated in low bandwidth networks (hypothetical rates, ideally constant), 5, 12 and 25 KB/s. Values for each network and for each map are shown in Tables 3 and 4.

Table 3. Response times comparison between clients for Brazil 1 map

Brasil1	5 KB/s		12 KB/s		25 KB/s	
	JS	GSD	JS	GSD	JS	GSD
level 0	285,4	285,4	118,9	118,9	57,1	57,1
level 1	506,9	444,4	253,2	190,7	158,9	96,4
level 2	514,4	456,5	254,1	196,2	157,4	99,6
level 3	482,3	419,6	244,3	181,6	155,9	93,2
level 4	376,3	309,5	202,4	135,6	137,7	71,0
level 5	221,9	155,8	138,0	71,9	106,8	40,7
level 6	144,9	78,7	105,6	39,4	91,0	24,8

When comparing the times of Tables 3 and 4 with the times of Table 1, the transmission times of the original and the quantized maps, we observe the usability and efficiency gain of the proposed architecture. Within it, a large step (complete map transmission, in a huge size) is divided in various additional steps (increments transmission), meeting the user's necessities. That can be viewed in Figure 7. The desktop client gain with respect to the Web client is observed comparing both clients' times with a given rate, in Tables 3 and 4. For example, in Brazil 2 map, the desktop client has a total average gain of about 4 to 5 times over the Web client. In the case of the real map transmission, even though it does not demand further details and processing, the initial real map transmission (the only one) takes too much time, being this the main transmission problem of large maps in the Internet. Comparing the progressive transmission techniques with the quantization one, it can be verified the more details are necessary the more advantageous is the use of the progressive approach.

In Tables 3 and 4 we can see that the desktop client gain becomes relatively more evident for higher bandwidths, since the increments transmission time, which is constant for both progressive transmission approaches, has stronger impact in the application than the increments processing time for small transmission rates. Thus, for low bandwidths, a client with large processing capacity can not compensate the problem

of efficient data transmission, which is fairly obvious. On the other hand, the use of the progressive transmission technique (irrespectively of the client) for low bandwidths generates great efficiency in the system, for example, when comparing the progressive transmission method with the quantization technique.

Table 4. Response times comparison between clients for Brazil 2 map

Brasil2	Transmission + Processing Times					
	5 KB/s		12 KB/s		25 KB/s	
	JS	GSD	JS	GSD	JS	GSD
level 0	295,2	295,2	123,0	123,0	59,0	59,0
level 1	1103,1	595,7	767,5	260,2	642,9	135,5
level 2	1157,5	639,3	798,6	280,4	665,3	147,1
level 3	1158,4	625,3	807	273,9	676,5	143,4
level 4	1181,7	645,2	819,7	283,1	685,2	148,7
level 5	1140,6	607,9	800,2	267,6	673,8	141,2
level 6	959,7	444,3	715,4	200	624,7	109,3

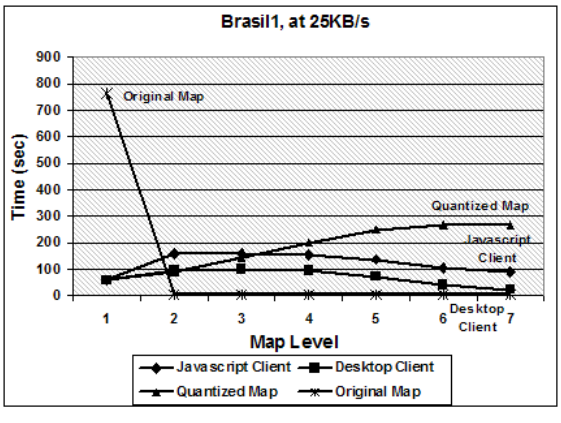


Figure 4. Transmission time comparison for Brazil 1 mp , at 25KB/s, using different approaches

From the tests and the proposed architecture, it was possible to observe some characteristics and advantages derived from the use of the GIS Services when implementing a Map Server with progressive transmission. A higher flexibility in the clients' implementation was reached in different platforms, irrespective of the used programming language. In addition, the asynchrony characteristic fits very well to the problem of progressive transmission: while the client waits for the images' details transmission, the client can perform other tasks in parallel, such as reception and processing of the previously received details. Another benefit derived from the use of Web Services is to provide the client with larger processing capacity by using desktop applications, compared to the scripts running in web browsers.

5. Conclusions and Future Work

In this work was presented a GIS Services architecture for a map server that supports progressive transmission. It was illustrated the use of the Web Services in Geographical Information Systems so as to supply those systems with the advantages of Web Services, according to the Gis' specific characteristics. The proposed architecture of the

progressive transmission GIS Service and its operation were illustrated and the Web service's methods were detailed.

The proposed architecture allows different clients to use vector maps, independently of platform or programming languages. Moreover, the characteristic of the progressive transmission provided better usability and better response time as the application client visualizes and interacts with the maps, mainly when low bandwidth is available.

The GIS Services' architecture allowed the implementation of a desktop client, and it was possible to compare the proposed architecture performance (based on GIS Services, desktop), with a Web Gis, which did not make so evident the gains got through progressive transmission due to the many DOM calls that updated the SVG map.

As future work, we expect to further extend the proposed GIS Services, including other techniques in the architecture, besides the Progressive Transmission, such as multi-resolution and buffering, in order to enhance the system's efficiency and usability.

References

Alameh, N.: Chaining Geographic Information Web Services. IEEE Internet Computing, Vol. 07 (September 2003), pp. 22-29.

Apache Axis. Available online at:http://ws.apache.org/axis/java/index.html. Last accessed on: July, 2007.

Baptista, C. S., Silva, E. R., Leite Jr., F. L., Paiva, A. C.: iGIS: A framework for Web Mapping, Eighth East-European Conference on Advances in Databases and Information Systems. Budapest, Hungary,2004.

Bertolotto, M., Egenhofer, M.: Progressive Vector Transmission, ACMGIS'99, p.152-157.

Bertolotto, M., Egenhofer, M.: Progressive Transmission of Vector Data over the World Wide Web. GeoInformatica 5(4):345-373, 2001

Buttenfield, B.: Transmitting Vector Geospatial Data across the Internet. In: M. J. Egenhofer and D. M. Mark (editors), GIScience, 2002, LNCS, pp. 51-64, Berlin.

Cecconi, A.: Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping, Doctor Thesis, Zurich University, 2003.

Costa, D. C., Paiva, A. C., Teixeira, M. M., Baptista, C. S., Silva, E. R.: A Progressive Transmission Scheme for Vector maps in low-bandwidth environments based on device rendering. 3rd International Workshop on Conceptual Modeling for Geographic Information Systems (CoMoGIS'06), ER2006, 2006

Ferris, C., Farrell, J.: What Are Web Services, Communications of the ACM 46(6), 2003, p. 31

Han, Q., Bertolotto, M.: A Multi-level Data Structure for Vector Maps, GIS'04, November 12-13, 2004, Washington, DC, USA, ACM 2004.

Jeong, C. W., Yu, S. D., Kim, M. S., Chung, Y. J., Lee, J. W.: Development of LBS Application using GML. Research Center for Advanced LBS Technology of

Chonbuk University, Korea, 2004, Available at http://www.gmldays.com/gml2004/papers/GML_LBS-Chonbuk.pdf.

Liang, C., Lee, C. H., Lee, J. D., Bae, H. Y.: Scale-Dependent transmission of spatial vector data on the Internet. The 3rd International Conference on In-formation Integration and Web-based Applications & Services, Austria, 2001.

Luo Y., Liu, X., Wang, W., Wang, X., Xu, Z.: Web Service and Geographical Information Integration. COMPSAC Workshops 2004, p.130-133

Oh, Y.: Advanced Progressive Transmission for Spatial Data in Web based GIS, The Second AEARU Workshop on Web Technology, Oct. 1999, pp.41-46.

Panatkool, A., Laoveerakul, S.: Decentralized GIS Web Services on Grid, Proceedings of the Open source GIS - GRASS users conference 2002, Italy, September 2002.

Papazoglou, M. P.: Service -Oriented Computing: Concepts, Characteristics and Directions. wise, p. 3, Fourth International Conference on Web Information Systems Engineering (WISE'03), 2003

Sliwinski, A. Toward Perceived Value-based Pricing of Geographic Information Services. Proceedings of the 7th AGILE Conference on Geographic Information Science, pp.541-549, 2004.

Tao, V.: Online GIServices. Journal of Geospatial Engineering, Vol. 3, No. 2, pp. 135-143, December, 2001.

Tsou, M.H., Buttenfield, B.P.: Client/Server Components and Metadata Objects for Distributed Geographic Information Services. In Proceedings of the GIS/LIS'98, pp. 590-599, 1998

TU, H.: Pattern Recognition and Geographical Data Standardization. The Proceedings of Geoinformatics'99 Conference, Ann Arbor, 19-21 June, 1999, pp. 1-7

World Wide Web Consortium (W3C),Web Services, http://www.w3.org/2002/ws

World Wide Web Consortium (W3C), Web Services Architecture, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211

Yang, B.S., Purves, R.S., Weibel, R., 2004: Implementation of progressive transmission algorithms for vector map data in web-based visualization, International Archives of Photogrammetry and Remote Sensing, Istanbul, Turkey, July 12-23, 2004, 6 p.

Yumin, T., Tianhe, C.: Web-based GIS services in participatory forest management in China. Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International Volume 7, p.4795 - 4798, 2004.

An Instance-based Approach for Matching Export Schemas of Geographical Database Web Services

Daniela F. Brauner, Chantal Intrator, João Carlos Freitas, Marco A. Casanova

Departamento de Informática – PUC-Rio
Rua Marquês de São Vicente, 225 – 22.453-900 – Rio de Janeiro – RJ – Brazil
{dani, cintrator, jcsfreitas, casanova}@inf.puc-rio.br

Abstract. *This paper describes a semantic approach for matching export schemas of geographical database Web services, based on the use of a small set of typical instances. The paper also contains an extensive experiment, in the context of two gazetteers, Geonames and the ADL gazetteer, to illustrate the approach.*

1. Introduction

A *database Web service* consists of a Web service interface with operations that provide access to a backend database. When a client sends a query to a database Web service, the backend engine submits the query to the backend database, collects the results and delivers them to the client. The *export schema* describes the subset of the backend database schema that the database Web service makes visible to the clients [Sheth and Larson, 1990]. Usually, the export schema consists of a flat table, which does not have complex dependencies with other elements of the backend database schema. In addition, a Web service typically announces its interfaces using Web Service Definition language – WSDL, a W3C standard.

The goal of this paper is to present a semantic approach for matching export schemas of geographical database Web services, based on the use of a small set of typical instances. The paper illustrates the approach with an extensive experiment that uses two gazetteers, Geonames and the ADL gazetteer, an ISO-complaint, pre-defined geographical global schema, and a set of typical geographical locations.

The paper is organized as follows. Section 2 discusses related work. Section 3 introduces the proposed semantic schema matching approach. Section 4 describes the experiment and discusses open issues. Finally, Section 5 contains the final considerations and suggestions for future work.

2. Related work

According to Rahm and Bernstein (2001), schema matching is a basic problem in many database application domains, such as Web-oriented data integration. The *match* operation takes two schemas as input and produces a mapping between elements of the two schemas that correspond to each other. Many techniques for schema and ontology matching have been proposed to automate the match operation. Rahm and Bernstein (2001) present a survey on several schema matching approaches.

Hess et al. (2006) proposes G-Match, an algorithm for geographic ontology matching. G-Match takes two different geographic ontologies as input, measures the similarities of their concepts by considering class and attribute names (string similarity), and hierarchical and topological relationships, producing as output a list of similarity measures between the concepts from the two ontologies. For class name and attribute name matching, they use WordNet [Wordnet, 2006] to feed the algorithm with synonyms. This approach therefore assumes that syntactical and structural similarity implies semantic proximity, which is often not warranted. Natural language dictionaries may be useful, perhaps even multi-language dictionaries (e.g., English-Japanese) to deal with schemas using terms in different languages. In addition, domain- or enterprise-specific dictionaries may sometimes be essential to deal with organizational standards, such as abbreviations for schema element names.

Wang et al. (2004), propose a unified solution to the problem of database schema matching. Their approach is based on an instance-based schema matching technique by domain-specific query probing, applied to Web databases. A Web database is a backend database available on the Web and accessible through a query interface. In particular, a Web database has two different schemas, the interface schema (**IS**) and the result schema (**RS**). The interface schema of an individual Web database consists of data attributes over which users can query, while the result schema consists of data attributes that organizes the query results that users receive.

This approach is based on three observations about Web databases:

1. Improper queries often cause search failure, that is, return no results. For the authors, improperness means that the query keywords submitted to a particular interface schema element are not applicable values of the database attribute to which the element is associated. For instance, if you submit a string to query an attribute that is originally defined as an integer, you get an error. As an example, if you submit a *latitude* value to the search element *place name*.
2. The keywords of proper queries that return results very likely reappear in the returned result pages.
3. There is a global schema (**GS**) for Web databases of the same domain [He and Chang, 2003]. The global schema consists of the representative attributes of the data objects in a specific domain.

The query probing technique consists of exhaustively sending keyword queries to the query interface of different Web databases, and collecting their results for further analysis. Based on the third observation, they assume, for a specific domain, the existence of a pre-defined global schema, and a number of sample data objects under the global schema, called *global instances*. For Web databases, they deal with two kinds of schema matching: *intra-site schema matching* (that is, matching global with interface schemas, global with result schemas, and interface with result schemas) and *inter-site schema matching* (that is, matching two interface schemas or two result schemas).

The data analysis is based on the second observation. Given a proper query, the results will probably contain the re-occurrence of the submitted value (referring to the values of the attributes of the global instances). The results will be organized using the HTML sent to Web browser. Thus, the re-occurrence of the query keywords in the

returned results can be used as an indicator of which query submission is appropriate (i.e., to discover associated elements in the interface schema). In addition, the position of the submitted query keywords in the result pages can be used to identify the associated attributes in the result schema.

The query probing process is based on the following workflow. Given a Web database with its query interface, an element identification component first locates qualified input elements. Then, a query submission component exhaustively submits the attribute values of the global instances into those identified input elements. After collecting the returned results for all submitted queries, a wrapper induction component induces a regular-expression wrapper composed of HTML-tags. Next, a data extraction component employs the induced wrapper to extract structured data objects from query result pages and arrange them into a data table. Finally, the re-occurrences of submitted queries in the columns of this table are counted and stored into a query occurrence cube. Then, using a projection function, say sum, the 3-dimensional cube is projected onto three Query Occurrence Matrices (front, top and left), which exactly reflect the relationship between pairs of the three schemas (i.e., GS and IS, IS and RS, and GS and RS). The main research issue now becomes how to find the correspondence between a pair of schemas in the projection matrices. In this context, to discover *intra-site schema matching* they applied the concept of mutual information. Moreover, to discover *inter-site schema matching*, they applied the idea of vector similarity used in the Vector Space Model from information retrieval [Salton, 1989].

In our paper, we will focus only on the query probing process applied to match export schemas (as result schema in [Wang et al., 2004]), as explained on the next section.

3. Instance-based Schema Matching

Based on the query probing process of Wang et al. (2004), we propose an instance-based approach for schema matching, in the context of geographical database Web services.

A database Web service is a well-specified service that provides Web access to a database. By well-specified, we mean that the service has a XML document (preferably, but not necessarily, a WSDL document) that describes the input attributes (interface schema) and the output attributes (export schema). Note that, by using an XML description, we do not require the definition of an HTML wrapper to locate qualified input (query interface attributes) and output elements (attributes of the result set).

Our first prototype of the schema matching process (Figure 1) starts with the XML descriptions of a set of database Web service, a previously defined global schema, and a set of global instances. For each global instance, the query formulator component creates queries based on the global instances and the Web service input attributes. The query submission component is responsible for submitting the queries to the Web service engine. After collecting the returned results for all submitted queries and storing them in local tables, the result analyzer component analyzes the global instances and the result set looking for re-occurred values, and creates the occurrence matrix.

The occurrence matrix is created with the number of re-occurrences of the global instance value in the result set. For each re-occurred value, the re-occurrence is

attributed to the correspondent export schema attribute (occurrence matrix rows) and the correspondent global schema attribute (occurrence matrix columns). An individual cell corresponds to the re-occurrence frequency of matching the global schema attribute with the export schema attribute.

Given an occurrence matrix, we define that an attribute of the export schema matches an attribute of the global schema as follows. We first normalize the matrix elements (the re-occurrence values) by dividing them by the overall number of returned entries. Then, we define that a pair of attributes *match* iff the normalized value is greater than a given threshold, namely, 0,2 (that is, 20%) in this case, based on our experiments observation.

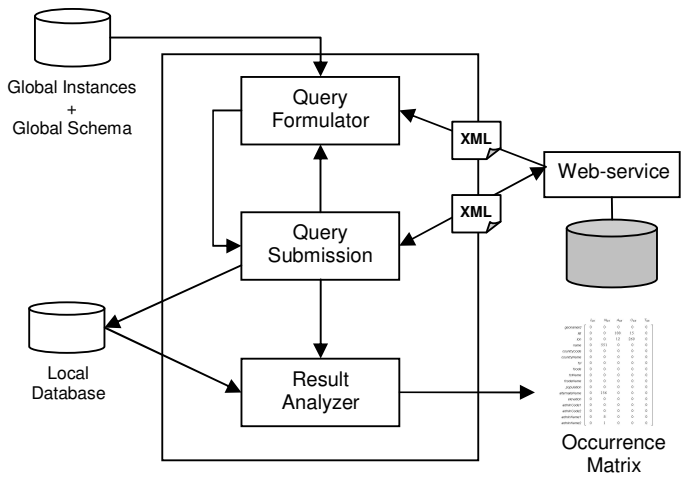


Figure 1. Instance-based Schema Matching Process

4. Experimental Approach

4.1. Global Schema and Global Instances

We designed a set of experiments using two gazetteers, available as database Web services. The experiments adopt a global schema capturing the essential characteristics of a gazetteer, and depend on a set of global instances, describing popular geographic place.

The global schema (see Figure 2) is based on the ISO 19112:2003, the recommended model for spatial referencing using geographic identifiers [ISO/TC211-ISO19112, 2003]. In detail, the global schema contains two classes, GeoInstance and GeoType, based on the ISO recommended classes, SI_LocationInstance and SI_Location Type, respectively. Table 1 and Table 2 show the attributes of classes GeoInstance and GeoType.

The global instances represent the data that will be submitted as queries to the Web services. The global instance set contains a set of geographic place names carefully chosen to cover a number of representative geographic locations. Firstly, we manually

compile a list of 36 popular geographic names that would form the basic reference database. Then, we submitted these 36 distinct names to the Geonames.org Web service. As expected, each of the name-queries returned several results, and we ended-up with thousands of entries for merely 36 initial names. The cleaning-up process of the instances was accomplished by taking the response of each query and manually locating the "most famous" place. All entries, except the "most famous" places, were discarded. The remaining entries were stored in a local database, following the global schema specified on Table 1 and Table 2. As an example, Table 3 shows a fragment of the global instances set.

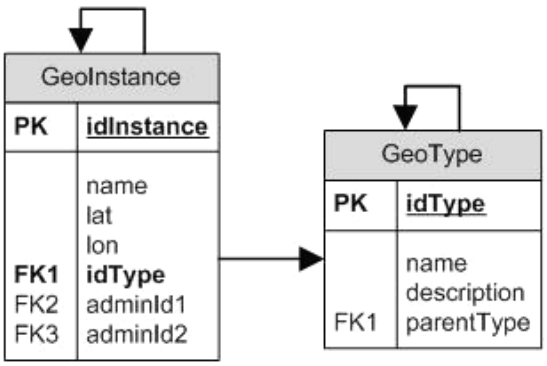


Figure 2. E-R Model of the proposed Geographical Global Schema

Table 1. Attributes of the *GeoInstance* Global Schema element

Atribute name	Description	Data Type
idInstance (I_{GS})	The entry identifier	Integer
name (N_{GS})	The entry name	String
lat (A_{GS})	The entry latitude	Double
lon (O_{GS})	The entry longitude	Double
idType (T_{GS})	GeoType code - Foreign Key (FK) for GeoType.idType	Integer
adminId1 ($A1_{GS}$)	First-order division - FK for GeoInstance.idInstance	Integer
adminId2 ($A2_{GS}$)	Second-order division - FK for GeoInstance.idInstance	Integer

Table 2. Attributes of the *GeoType* Global Schema element

Atribute name	Description	Data Type
idType	The entry identifier	Integer
name	The entry name	String
description	The entry description	String
parentType	The entry parent (broader term) - FK for GeoType.idType	Integer

Table 3. Global Instances fragment

idInstance	name	lat	lon	idType	adminId1	adminId2
175	Galapagos Islands	0.0	-90.5	4	73	-
52	Alps	46.4166667	10.0	15	165	-
149	Atlantic Ocean	10.0	-25.0	9	-	-
90	Niagara Falls	43.083416155	-79.06627052	21	123	-
16	Pão de Açúcar	-22.9472	-43.1561	14	101	-
34	Mississippi River	29.1510582	-89.2533842	19	109	-

4.2. Experimental Geographical Databases Web Services

The set of experiments uses two gazetteers, available as database Web services, Geonames[1] and the Alexandria Digital Library (ADL) Gazetteer[2]. In our experiments, we accessed both gazetteers through their *search-by-place-name* Web services.

Geonames is a gazetteer that contains over six million features categorized into one of nine classes and further subcategorized into one out of 645 feature codes. Geonames was created using data from the National Geospatial Intelligence Agency (NGA) and the U.S Geological Survey Geographic Names Information System (GNIS). Geonames services are available through the Web services. Table 4 presents the Geonames export schema. Figure 3 shows a fragment of the XML response of this service.

The ADL Gazetteer comprises both US and non-US geographic place names. The ADL Gazetteer, and can be accessed through XML- and HTTP-based requests [Janée and Hill, 2004]. Table 5 presents the ADL export schema. Figure 4 shows a fragment of the XML response of this service.

4.3. Experimental Results

Our experiments were executed using the instance-based schema matching process described in Section 3. We used the set of global instances (Section 4.1) and the Web services provided by the ADL Gazetteer and the Geonames (Section 4.2). From the 36 global instances submitted to the gazetteers, the ADL Gazetteer returned 459 registries and the Geonames, 703 registries.

The re-occurrence detection method was created as follows: for the *name* attributes, we used the standard string comparison operator to detect the occurrence of a string in another. For the *latitude* and *longitude* attributes, we first truncated the value to four digits before comparing the values.

[1] Geonames - http://www.geonames.org
[2] ADL Gazetteer - http://www.alexandria.ucsb.edu/gazetteer

Table 4. Geonames Search Web Service Export Schema

Attribute name	Description	Data Type
geonameId	The entry identifier	String
name	The entry primary name	String
alternateNames	Comprises the set of alternative names	String
countryCode	The entry country code (ISO-3166 2-letter code)	String
countryName	The entry country name	String
population	The population of the instance	Number
lat	The entry latitude	Number
lng	The entry longitude	Number
fcl	The feature type super class code	String
fclName	The feature type super class name	String
fcode	The feature type classification code	String
fcodeName	The feature type classification name	String
elevation	The entry elevation, in meters	Number
admCode1	Code for first administrative division	String
admName1	Name for first administrative division	String
admCode2	Code for second administrative division	String
admName2	Name for second administrative division	String
timezone	Timezone description	String

Table 5. ADL Gazetteer Search Web Service Export Schema

Attribute name	Description	Data Type
identifier	The entry identifier	String
placeStatus	The entry place-status (current or former)	String
name	The entry primary name	String
displayName	The entry primary name as it is displayed	String
footprintX	The entry longitude	Number
footprintY	The entry latitude	Number
class	The entry class	String
thesaurus	The thesaurus of the entry class	String
names	Comprises the set of alternative names	names
relationships	The entry “partOf” relationships	String

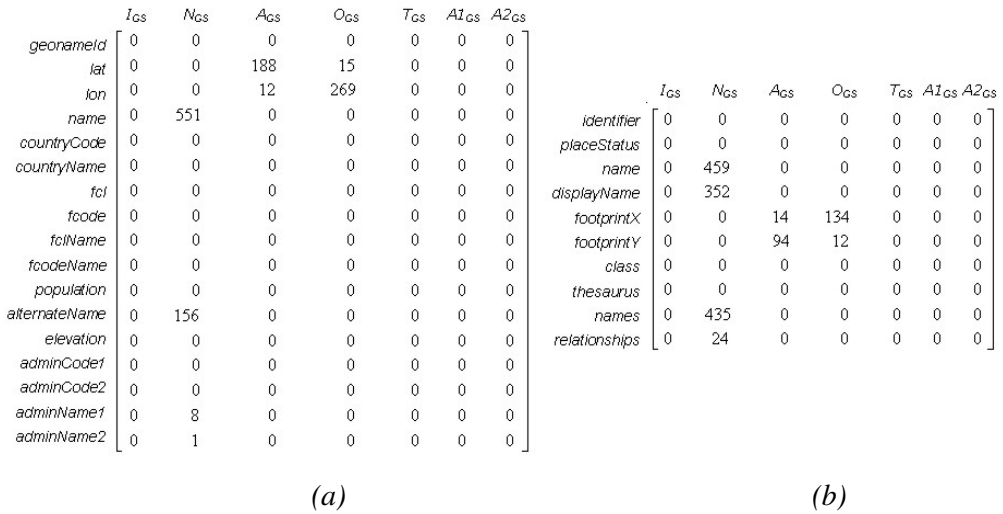
```
<?xml version="1.0" encoding="UTF-8" ?>
- <geonames style="FULL">
    <totalResultsCount>1</totalResultsCount>
  - <geoname>
      <name>Amazon River</name>
      <lat>-0.1666667</lat>
      <lng>-49.0</lng>
      <geonameId>3407729</geonameId>
      <countryCode>BR</countryCode>
      <countryName>Brazil</countryName>
      <fcl>H</fcl>
      <fcode>STM</fcode>
      <fclName>stream, lake, ...</fclName>
      <fcodeName>stream</fcodeName>
      <population />
      <alternateNames>Orellana,Rio Amazonas,Rio Maranon,Rio Solimoes,Rio
        Solimões,Rio el Amazonas,Río Amazonas,Río Marañón,Río el
        Amazonas,Salimoes River,Solimoens</alternateNames>
      <elevation />
      <adminCode1>00</adminCode1>
      <adminName1 />
      <adminCode2 />
      <adminName2 />
      <timezone dstOffset="-3.0" gmtOffset="-3.0">America/Belem</timezone>
    </geoname>
  </geonames>
```

Figure 3. XML response fragment of Geonames.org Search Web Service

```
<?xml version="1.0" encoding="UTF-8" ?>
- <gazetteer-service xmlns="http://www.alexandria.ucsb.edu/gazetteer"
    xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.alexandria.ucsb.edu/gazetteer
    http://www.alexandria.ucsb.edu/gazetteer/protocol/gazetteer-service.xsd" version="1.2">
  - <query-response>
    - <standard-reports>
      - <gazetteer-standard-report>
          <identifier>adlgaz-1-1410143-3a</identifier>
          <place-status>current</place-status>
          <display-name>Amazon River - Brazil</display-name>
        - <names>
            <name primary="true" status="current">Amazon River</name>
            <name primary="false" status="current">Solimoens</name>
            <name primary="false" status="current">Salimoes River</name>
            <name primary="false" status="current">Orellana</name>
            <name primary="false" status="current">Maranon, Rio</name>
            <name primary="false" status="current">Amazonas, Rio</name>
            <name primary="false" status="current">Amazonas, Rio el</name>
            <name primary="false" status="current">Solimoes, Rio</name>
          </names>
        - <bounding-box>
          - <gml:coord>
              <gml:X>-49.0</gml:X>
              <gml:Y>-0.1667</gml:Y>
            </gml:coord>
          - <gml:coord>
              <gml:X>-49.0</gml:X>
              <gml:Y>-0.1667</gml:Y>
            </gml:coord>
          </bounding-box>
        - <footprints>
          - <footprint primary="true">
            - <gml:Point>
              - <gml:coord>
                  <gml:X>-49.0</gml:X>
                  <gml:Y>-0.1667</gml:Y>
                </gml:coord>
              </gml:Point>
            </footprint>
          </footprints>
        - <classes>
            <class thesaurus="ADL Feature Type Thesaurus" primary="true">streams</class>
            <class thesaurus="NIMA Feature Designation" primary="false">STM (stream)</class>
          </classes>
        - <relationships>
            <relationship relation="part of" target-name="UTM grid GE28" />
            <relationship relation="part of" target-name="JOG Sheet Number SA22-0" />
            <relationship relation="part of" target-name="Brazil" target-identifier="adlgaz-1-19-19" />
          </relationships>
        </gazetteer-standard-report>
      </standard-reports>
    </query-response>
</gazetteer-service>
```

Figure 4. XML response fragment of ADL Gazetteer Search Web Service

As a result, we obtain two occurrence matrices (Figure 5). Figure 5 (a) and (b) show, respectively, the occurrence matrix between the global schema and the Geonames export schema, and the occurrence matrix between the global schema and the ADL Gazetteer export schema. As an example, Figure 5 shows that *name* from Geonames had 551 re-occurrences of the values of the attribute N_{GS} from the global schema (N_{GS} represents the attribute *name* of the global schema, see Table 1). For instance, when a global instance *name* value (N_{GS}) as "Mount Everest" was submitted to the Geonames search Web service, the value "Mount Everest" reappeared six times as the value of the attribute *name* from Geonames (Table 6). The final re-occurrence value between the attribute *name* from Geonames and the attribute N_{GS} from the global schema is the sum of the reoccurrence of all 36 names of the submitted global instances to the Geonames service.



(a) *(b)*

Figure 5. Occurrences matrices between (a) Geonames.org Export Schema and GS, and (b) ADL Gazetteer Export Schema and GS

Table 6. Reoccurrence of "Mount Everest" in a fragment of the results of Geonames.org Search Web Service

geonameId	lat	lng	name	country Code	fcode
1283416	27.9833	86.9333	Mount Everest	NP	MT
1004850	-28.15	29.16667	Mount Everest	ZA	MT
4122419	33.78733	-93.3804	Mount Everest Church	US	CH
4334114	29.94326	-90.0904	Mount Everest Baptist Church	US	CH
4341122	29.94104	-90.089	Second Mount Everest Baptist Church	US	CH
4694788	32.70374	-96.7881	Greater Mount Everest Baptist Church	US	CH

Given an occurrence matrix, we define that an attribute of the export schema matches an attribute of the global schema iff the normalized value is greater than 0,2 (as explained in Section 3).

For instance, Figure 5 (a) shows that *name* and *alternateName* from Geonames matches with N_{GS} from the global schema (N_{GS} represents the attribute *name* of the global schema; see Table 1). More precisely, the attribute N_{GS} had 551 reoccurred values on the attribute *name* of the Geonames export schema, what means approximately 78% of the overall of 703 entries returned by the Geonames service. The attribute *alternateName* had 156 reoccurred values, what means approximately 22%. The attributes *lat* and *lon* from Geonames correctly match with A_{GS} and O_{GS} from the global schema, respectively, with approximately 27% and 38%. By contrast, the attribute O_{GS} had 15 re-occurred values on the attribute *lat* from Geonames, which means approximately 2% of the overall reoccurred values. This value indicates that O_{GS} does not match *lat*.

Using the same procedure for the ADL gazetteer, the occurrence matrix in Figure 5 (b) shows that attributes *name*, *displayName* and *names* from ADL all align with N_{GS} from the global schema, with approximately 100%, 77% and 95%, respectively, relative to a total of 459 returned entries. Other correct matches are *footprintX* and *footprintY* from ADL with O_{GS} and A_{GS} from the global schema, respectively.

4.4. Further considerations on global instances

In our experiments, we observed some important issues that need further consideration.

First, the design of the global schema obviously influences the matching process. In our experiments, we observed that some attributes of the export schemas have no direct correspondence with any of the attributes of the global schema, such as the attribute *population* of the Geonames export schema. To overcome this problem, we suggest that the global schema be extended automatically. The idea is to add to the global schema, on demand, new attributes found on export schemas. When a new attribute appears in an export schema, the system must add this new attribute to the global schema and populate the global instances set with its values. The new global schema attribute should be labeled as "recommended" and, after it receives a sufficiently large number of recommendations (evidences coming from other export schemas), it becames an "active" attribute. However, this issue brings new challenges to this approach: update the old global instances with the correct values of the new attribute; and, define the threshold value for the number of recommendations above which the recommended attribute becomes active.

Another issue related to the design of the global schema refers to attributes with temporal aspects. For example, suppose that the global instance set holds data from 2007, but a specific Web service provides data from 1970. In this case, the values of attribute *population*, say, would never re-occur on the returned data.

Second, as already observed in [Wang et al., 2004], the performance of the instance-based matching approach depends on the selection of the global instances. We must carefully select the global instance set in such way that:

1. global instances are representative of the overall application domain to maximize the chance that the global instances are indeed found in the database Web services to be considered;
2. global instances have attribute values that do not match with too many attribute values of an export schema.

Consider again the geographic names domain. Then, to achieve (1), the global instance set must cover, as much as possible, the variety of types of geographic features, and it must contain "famous" places (w.r.t. the region considered) .

Condition (2) is a difficult point, however. For example, if data about the country "Brazil" as a global instance, then "Brazil" will occur several times as *countryName* of several instances returned from the Geonames service. Indeed, an attribute that indicates an administrative area should not be analyzed alone. Instead, it must be analyzed in conjunction with other attributes to eliminate the risk of matching a global instance name that occurs as an administrative name of other global instances. If we have an expressive number of administrative areas as global instances, we will probably generate false matchings between the global attribute *name* and other attributes of the export schema. This problem indeed generalizes to geographic features used as aggregates of other geographic features, such as a mountain range.

As a second example where Condition (2) fails, in our experiments, we noticed that city, state and country names frequently occur inside the character string that defines a geographic feature name. This is the case, for example, with the values of the attribute *displayName* of the ADL Gazetteer, which is used to store the place name as it must to be displayed in the interface of an ADL Gazetteer client. For example, the display name of "Niagara Falls" is "Niagara Falls – Niagara County – New York – United States".

Finally, errors in the attribute values (or in the interpretation of the attribute values) generate another issue that may create false matchings. For instance, in Geonames, we noticed that "Niagara Falls" occurs as an alternate name for a hotel named "Glengate Hotel", located in the state of "Ontario" in "Canada", and that "American Canyon" occurs as an alternate name for a hotel called "Gaia Napa Valley Hotel", located in the state of "California" in the "United States".

5. Conclusion

In this paper, we proposed a semantic approach, using instances, for matching export schemas of geographical database available through Web services. We also described experiments using two real Web gazetteers services. Based on the experiments, we listed some important issues that must be considered when designing the global schema and when selecting the global instances set.

As future work, we intend to improve the instance-based schema matching process in several directions. We plan to improve the re-occurrence detection method; execute a validation step to define formally a threshold to the proportion between reoccurrence values; and prototype a Web databases services mediator as a proof of concept. In addition, we intend to analyze how to improve the performance of the method by including, for instance, the automatic updating of the global schema.

References

He, B. and Chang, C. C. (2003), Statistical schema matching across Web query interfaces. In: Proc. ACM SIGMOD Conference.

Hess, G., Iochpe, C. and Castano, S. (2006), An Algorithm and Implementation for GeoOntologies Integration. GEOINFO 2006, Campos do Jordão, Brazil.

ISO/TC211 (2003). "ISO 19112:2003 Geographic information — Spatial referencing by geographic identifiers". International Standard 19112. Technical report of Technical Committee ISO/TC 211.

Janée, G. and Hill, L. L. (2004), ADL Gazetteer Protocol. Alexandria Digital Library Project. Available at http://www.alexandria.ucsb.edu/gazetteer/protocol/

Rahm, E. and Bernstein, P. A. (2001), A Survey of Approaches to Automatic Schema Matching, The VDLB Journal, vol. 10, pp. 334–350.

Salton, G. (1989), Automatic text processing: the transformation, analysis, and retrieval of information by computer, Addison-Wesley, Reading, MA.

Sheth, A. and Larson, J. (1990), Federated database systems for managing distributed, heterogeneous and autonomous databases. ACM Computing Surveys, v.22 (3), set. 1990. pp.183-236. Available at: http://portal.acm.org/citation.cfm?id=96602.96604

Wang, J., Wen, J. Lochovsky, F.H. and Ma, W. (2004). Instance-based schema matching for web databases by domain-specific query probing, In Proceedings of 30th Intl. Conference on Very Large Data Bases, pp. 408-419, 2004.

Wordnet (2006), Wordnet 3.0 - a lexical database for the English language. Cognitive Science Laboratory, Princeton University, Princeton, NJ – USA. Available at: http://wordnet.princeton.edu

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 121-132.

Model selection for a class of spatio-temporal models for areal data

Juan C. Vivar[1], **Marco A. R. Ferreira**[2]

[1]Departamento de Métodos Estatísticos
Universidade Federal do Rio de Janeiro (UFRJ) – RJ – Brazil

[2]University of Missouri
Columbia – U.S.A.

jcvivar@dme.ufrj.br, ferreiram@missouri.edu

Abstract. *We present a method to perform model selection based on predictive density in a class of spatio-temporal dynamic generalized linear models for areal data. These models assume a latent random field process that evolves through time with random field convolutions; the convolving fields follow proper Gaussian Markov random field processes. Parameter and latent process estimation based on Markov Chain Monte Carlo and the forward information filter backward sampler, respectively, is showed. Finally, an application using several specifications of the general model on homicide data in the State of Espírito Santo is presented showing the results of model selection.*

1. Introduction

The last decade has seen an upsurge of research on spatio-temporal modelling. Massive amounts of spatio-temporal data have become available and the increasing power of computers has made possible the analysis of these datasets with progressively realistic models. Several spatio-temporal models have been proposed in the literature for specific applications for point-referenced data. Examples include meteorology [Ghil et al. 1981], ozone analysis [Guttorp et al. 1994], prediction of snow water [Huang and Cressie 1996], calibration of radar rainfall data [Brown et al. 2001], and analysis of pollutant levels [Huerta et al. 2004]. For areal data have been developed in the disease mapping literature [Bernardinelli et al. 1995, Waller et al. 1997, Knorr-Held 2000, Knorr-Held and Richardson 2003, Schmid and Held 2004].

In [Vivar and Ferreira 2007] we proposed a linear Gaussian spatio-temporal models for areal data that uses *proper Markov random fields*. These models can be cast within a state-space formulation [West and Harrison 1997]. More specifically, we considered a latent random field process that evolves through time with random field convolutions; the convolving fields follow proper Gaussian Markov random field (PGMRF) processes. At each time, the latent random field process is linearly related to observations through an observation equation with errors that also follow a PGMRF.

The spatio-temporal model is

$$\mathbf{y}_t = \mathbf{F}_t'\beta_t + \epsilon_t, \qquad \epsilon_t \sim PGMRF(\mathbf{0}_S, \mathbf{V}_t^{-1}), \tag{1}$$

$$\beta_t - \mu_\beta = \mathbf{G}_t(\beta_{t-1} - \mu_\beta) + \omega_t, \qquad \omega_t \sim PGMRF(\mathbf{0}_S, \mathbf{W}_t^{-1}), \tag{2}$$

where $\mathbf{0}_S$ is the S-dimensional null vector and the errors $\epsilon_1, \ldots, \epsilon_T$ and the system innovations $\omega_1, \ldots, \omega_T$ are independent. The matrix $\mathbf{F}_t$ connects the latent random field process to the field observations, the matrix $\mathbf{G}_t$ describes the spatio-temporal evolution of the process, ω_t is the state innovation field, and the covariance matrices $\mathbf{V}_t$ and $\mathbf{W}_t$ describe the covariance structure of the observational and system errors, respectively. The mean level field μ_β describes the temporal stationary expected behavior of the latent process. It is only defined if the process is temporally stationary, otherwise, it is omitted from the model.

Following the notation of [Ferreira and De Oliveira 2007], $\mathbf{Z} \sim PGMRF(\mu_z, \mathbf{P})$ means that the variable $\mathbf{Z}$ follows a PGMRF process with mean vector μ_z and precision matrix $\mathbf{P}$, that is, the density function of $\mathbf{Z}$ is proportional to $\exp\left(-\frac{1}{2}(\mathbf{z}-\mu_z)'\mathbf{P}(\mathbf{z}-\mu_z)\right)$, where $\mathbf{P} = \tau(\mathbf{I}_S + \phi\mathbf{M})$ where $\mathbf{M}$ is called the neighborhood matrix, τ is a scale parameter, $\mathbf{I}_S$ is the $S \times S$ identity matrix and $\phi \geq 0$ controls the degree of spatial correlation. When $\phi = 0$, the regions are independent and the spatial dependence increases when ϕ increases. See [Vivar and Ferreira 2007, Vivar 2004] for more details, special cases and properties of this class of models.

Because this is a class of linear models it has a good performance with Gaussian or approximately Gaussian data. A natural extension is a model that deals with observations that are distributed in the exponential family. In the next Section we present a specific model for count data. Section 3 is developed Bayesian inference for the parameters. Section 4 describes several specifications of the general model and the method to perform model selection. An application using homicide data in the State of Espírito Santo is shown in Section 5.

2. Spatio-temporal model for count data

Our data consists on the annual number of homicides per county in the State of Espírito Santo, Brazil, from 1979 to 1998. During the 80s and 90s new counties were created in the State of Espírito Santo by fusion or division of older counties. For compatibility purposes, we use here the political map of 1979 in a total of 52 counties.

For each year t and county s, $t = 1, \ldots, T$, $s = 1, \ldots, S$, let n_{ts} denote the population size and y_{ts} the observed number of homicides. As it is typical for count data such as these, we assume that y_{ts} follows a Poisson distribution. More specifically, we assume that $y_{ts}|\lambda_{ts} \sim Po(n_{ts}\lambda_{ts})$, where λ_{ts} is the underlying risk at time t in county s. The Poisson distribution belongs to the exponential family distributions with canonical parameter $\eta_{ts} = \log(\mu)$. Specifically, we identify the following components

- Mean $\mu_{ts} = \exp(\eta_{ts})$ and variance $\Sigma_{ts} = \exp(\eta_{ts})$.
- Linear predictor: $\theta_{ts} = \log(\lambda_{ts})$.
- Link function: $g(\mu_{ts}) = \log(\mu_{ts}/n_{ts}) = \theta_{ts}$.
- Response function: $f(\theta_{ts}) = n_{ts}\exp(\theta_{ts}) = \mu_{ts}$.
- The function $\gamma(\theta_{ts})$ that transform directly the canonical parameter into the linear predictor:

$$\begin{aligned} \eta_{ts} &= \log(\lambda_{ts} n_{ts}) \\ &= \log(\lambda_{ts}) + \log(n_{ts}) \\ &= \theta_{ts} + \log(n_{ts}) \\ &= \gamma(\theta_{ts}) \end{aligned}$$

Then, our general spatio-temporal model for count data results as follows:

$$\begin{aligned} p(y_{ts}|\eta_{ts}) &\propto \exp\{y_{ts}\eta_{ts} - \exp(\eta_{ts}) + \log(y_{ts}!)\}, \\ \eta_{ts} &= \theta_{ts} + \log(n_{ts}), \\ \theta_t &= \mathbf{F}_t'\beta_t, \qquad \theta_t = (\theta_{t1}, \ldots, \theta_{tS})' \\ \beta_t &= \mathbf{G}_t\beta_{t-1} + \omega_t, \qquad \omega_t \sim PGMRF(\mathbf{0}, \mathbf{W}_t^{-1}), \end{aligned} \tag{3}$$

Our main interest is to make inference for $\beta_{(1:T)} = (\beta_1', \ldots, \beta_T')'$.

3. Bayesian inference

When the matrices $\mathbf{F}_t, \mathbf{G}_t$ and $\mathbf{W}_t$ are completely known, the extended Kalman filter can be used to perform inference about the latent process β_t. But in practice these matrices are known only up to the parameter vector ψ and numerical integration methods are required for the Bayesian statistical analysis. Here we favor Markov chain Monte Carlo (MCMC) methods [Gamerman and Lopes 2006, Robert and Casella 1999] that are quite powerful and applicable to general highly structured models [Green et al. 2003] such as our spatio-temporal models for areal data.

It is critically important to design Markov chains with good properties such as fast convergence and small autocorrelation between realizations. With that objective in mind, the Markov chain has to be tailored to the specific spatio-temporal model at hand and will depend on how $\mathbf{F}_t, \mathbf{G}_t$ and $\mathbf{W}_t$ depend on ψ. Nevertheless, the Markov chain may be partitioned in two blocks: simulation of ψ and simulation of $(\beta_0, \ldots, \beta_T)$. The simulation of β is model specific and is briefly discussed in the application of Section 5. For the simulation of the latent process, we use the forward information filter backward sampler (FIFBS) that combines the forward filter backward sampler [Carter and Kohn 1994, Frühwirth-Schnatter 1994] with the information filter and thus benefits from the sparsity of $\mathbf{V}_t^{-1}$ and $\mathbf{W}_t^{-1}$ to accelerate computations. For details of FIFBS, see [Vivar and Ferreira 2007].

4. Model selection

4.1. Proposed models

This subsection presents several special cases of our spatio-temporal model (3). Looking at the data, it is clear that this is not a stationary process, the we need non-stationary models. The first candidate is a model that smooth the data (I). Another alternative is a contamination model (II) since some clusters of counties can be detected through the period under study. Other models considered are related to the notorious increasing mean level of many counties through time. They are the second-order temporal trend model and some variants (III - VIII).

Model I: First-order temporal trend

- $\mathbf{F}_t' = \mathbf{I}_S$ and $\mathbf{G}_t = \mathbf{I}_S$,
- $\mathbf{W}_t^{-1} = \tau(\mathbf{I}_S + \phi\mathbf{M})$.

Model II: Contamination

- $\mathbf{F}'_t = \mathbf{I}_S$,
- $\mathbf{G}_t = \frac{1}{1+\kappa h}\mathbf{H} \longrightarrow \{\mathbf{H}\}_{kl} = \begin{cases} 1, & k = l, \\ \kappa, & k \in N_l, \\ 0, & o.c. \end{cases}$ Contamination matrix
- $\mathbf{W}_t^{-1} = \tau(\mathbf{I}_S + \phi\mathbf{M})$.

Models III e IV: Second-order temporal trend

- $\mathbf{F}'_t = (\mathbf{I}_S, \mathbf{0}_S)$,
- $\mathbf{G}_t = \begin{pmatrix} \mathbf{G}_{1t} & \mathbf{G}_{1t} \\ \mathbf{0}_S & \mathbf{G}_{2t} \end{pmatrix}$, $\mathbf{G}_{it} = \mathbf{I}_S, i = 1, 2.$
- $\mathbf{W}_t^{-1} = \begin{pmatrix} \mathbf{W}_{1t}^{-1} & \mathbf{0}_S \\ \mathbf{0}_S & \mathbf{W}_{2t}^{-1} \end{pmatrix}$, $\mathbf{W}_{it}^{-1} = \tau_i(\mathbf{I}_S + \phi_i\mathbf{M}), i = 1, 2.$

  Model III considers $\phi_2 = 0$.

Model V: Second-order model with velocity equation including contamination and $\phi_2 = 0$

- $\mathbf{F}'_t = (\mathbf{I}_S, \mathbf{0}_S)$,
- $\mathbf{G}_t = \begin{pmatrix} \mathbf{G}_{1t} & \mathbf{G}_{1t} \\ \mathbf{0}_S & \mathbf{G}_{2t} \end{pmatrix}$,

$$\mathbf{G}_{1t} = \mathbf{I}_S \text{ and } \mathbf{G}_{2t} = \frac{1}{1+\kappa_2 h}\mathbf{H} \longrightarrow \{\mathbf{H}\}_{kl} = \begin{cases} 1, & k = l, \\ \kappa_2, & k \in N_l, \\ 0, & o.c. \end{cases}$$

- $\mathbf{W}_t^{-1} = \begin{pmatrix} \mathbf{W}_{1t}^{-1} & \mathbf{0}_S \\ \mathbf{0}_S & \mathbf{W}_{2t}^{-1} \end{pmatrix}$, $\mathbf{W}_{it}^{-1} = \tau_i(\mathbf{I}_S + \phi_i\mathbf{M}), i = 1, 2.$

Model VI: Second-order model with level equation including contamination

- $\mathbf{F}'_t = (\mathbf{I}_S, \mathbf{0}_S)$,
- $\mathbf{G}_t = \begin{pmatrix} \mathbf{G}_{1t} & \mathbf{G}_{1t} \\ \mathbf{0}_S & \mathbf{G}_{2t} \end{pmatrix}$,

$$\mathbf{G}_{2t} = \mathbf{I}_S \text{ and } \mathbf{G}_{1t} = \frac{1}{1+\kappa_1 h}\mathbf{H} \longrightarrow \{\mathbf{H}\}_{kl} = \begin{cases} 1, & k = l, \\ \kappa_1, & k \in N_l, \\ 0, & o.c. \end{cases}$$

- $\mathbf{W}_t^{-1} = \begin{pmatrix} \mathbf{W}_{1t}^{-1} & \mathbf{0}_S \\ \mathbf{0}_S & \mathbf{W}_{2t}^{-1} \end{pmatrix}$, $\mathbf{W}_{it}^{-1} = \tau_i(\mathbf{I}_S + \phi_i\mathbf{M}), i = 1, 2.$

Model VII: Second-order model including contamination on both equations and $\phi_1 = 0$

- $\mathbf{F}'_t = (\mathbf{I}_S, \mathbf{0}_S)$,
- $\mathbf{G}_t = \begin{pmatrix} \mathbf{G}_{1t} & \mathbf{G}_{1t} \\ \mathbf{0}_S & \mathbf{G}_{2t} \end{pmatrix}$, $\mathbf{G}_{it} = \frac{1}{1+\kappa_i h}\mathbf{H}_i \longrightarrow \{\mathbf{H}_i\}_{kl} = \begin{cases} 1, & k = l, \\ \kappa_i, & k \in N_l, \\ 0, & o.c. \end{cases}, i = 1, 2,$
- $\mathbf{W}_t^{-1} = \begin{pmatrix} \mathbf{W}_{1t}^{-1} & \mathbf{0}_S \\ \mathbf{0}_S & \mathbf{W}_{2t}^{-1} \end{pmatrix}$, $\mathbf{W}_{it}^{-1} = \tau_i(\mathbf{I}_S + \phi_i \mathbf{M}), i = 1, 2.$

Model VIII: Second-order model with same acceleration for all sites

- $\mathbf{F}'_t = (\mathbf{I}_S, \mathbf{0}_S)$,
- $\mathbf{G}_t = \begin{pmatrix} \mathbf{G}_{1t} & \mathbf{1} \\ \mathbf{0} & G_{2t} \end{pmatrix}$, $\mathbf{G}_{1t} = \mathbf{I}_S, G_{2t} = 1.$
- $\mathbf{W}_t^{-1} = \begin{pmatrix} \mathbf{W}_{1t}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{2t}^{-1} \end{pmatrix}$, $\mathbf{W}_{1t}^{-1} = \tau_1(\mathbf{I}_S + \phi_1 \mathbf{M})$, $\mathbf{W}_{2t}^{-1} = \tau_2$

4.2. Predictive density

With the ability to fit more complex models comes the necessity to compare those models. There are many criteria to select models in the literature. For example, criteria based on a predictive distribution include those of [Geisser and Eddy 1979, San Martini and Spezzaferri 1984] and [Gelfand et al. 1992] and the references therein. We prefer the predictive density as our criterion of model selection because it naturally penalizes complex models, differently from other ones, like the deviance information criterion (DIC), [Spiegelhalter et al. 2002] that favors overfitting and tends to select very complex models.

Bayesian model selection is usually performed by comparing the posterior probabilities of the competing models. When the competing models have equal prior probabilities, their posterior probabilities are proportional to the respective predictive densities. These densities will depend on the prior distribution of the parameter vector ψ. In order to overcome this difficulty, we use here a training sample approach [Frühwirth-Schnatter 1995] of the first p time observations; this results in calibrated priors for the parameters of each model. Then, Monte Carlo integration is used to compute the predictive distribution under each model for the remaining $T - p$ time observations.

Suppose that there are Q competing spatio-temporal models $M_1, \ldots, M_Q$. The qth model has observational density $p_q(\mathbf{y}_t|\eta_t(\beta_t))$ and evolution density $p_q(\beta_t|\beta_{t-1}, \psi)$. Note that the definitions of β_t and ψ may be (and in general will be) different under each model, but this distinction is omitted in order to keep the notation simple.

Let $p_q(\beta_{1:t-1}, \psi|\mathbf{D}_{t-1})$ denote the joint posterior distribution of ψ and $\beta_1, \ldots, \beta_{t-1}$ under model q up to time $t-1$. Then, the predictive distribution of $\mathbf{y}_t$ under model q given the information up to time $t-1$ will be

$$
\begin{aligned}
p_q(\mathbf{y}_t|\mathbf{D}_{t-1}) &= \int p_q(\mathbf{y}_t|\eta_t(\beta_t))p_q(\beta_t|\beta_{t-1},\psi)p_q(\beta_{1:t-1},\psi|\mathbf{D}_{t-1})d\beta_{1:t-1}d\beta_t d\psi \\
&= \int p_q(\mathbf{y}_t|\beta_{t-1},\psi)p_q(\beta_{1:t-1},\psi|\mathbf{D}_{t-1})d\beta_{1:t-1}d\psi \qquad (4)
\end{aligned}
$$

since

$$
p_q(\mathbf{y}_t|\beta_{t-1},\psi) = \int p_q(\mathbf{y}_t|\eta_t(\beta_t))p_q(\beta_t|\beta_{t-1},\psi)d\beta_t.
$$

The simulation scheme outlined in Section 3 can be used to simulate a sample $(\beta_{t-1}^{(1)},\psi^{(1)}),\ldots,(\beta_{t-1}^{(L)},\psi^{(L)})$ from the joint posterior distribution $p_q(\beta_{1:t-1},\psi|\mathbf{D}_{t-1})$. Then, a Rao-Blackwellized estimate of the predictive density of $\mathbf{y}_t$ given the information up to time $t-1$ is

$$
\hat{p}_q(\mathbf{y}_t|\mathbf{D}_{t-1}) = \frac{1}{L}\sum_{l=1}^{L} p_q(\mathbf{y}_t|\beta_{t-1}^{(l)},\psi^{(l)}) \qquad (5)
$$

Thus, we adjust an MCMC scheme for each time point t and make the prediction for the subsequent time $t+1$. Using the fact that the joint predictive density of $\mathbf{y}_{t^*},\mathbf{y}_{t^*+1},\ldots,\mathbf{y}_T$ can be written as $p_q(\mathbf{y}_{t^*},\mathbf{y}_{t^*+1},\ldots,\mathbf{y}_T|\mathbf{D}_{t^*-1}) = \prod_{t=t^*+1}^{T} p_q(\mathbf{y}_t|\mathbf{D}_{t-1})$, an estimate of the joint predictive density under model q is

$$
\hat{p}_q(\mathbf{y}_{t^*},\mathbf{y}_{t^*+1},\ldots,\mathbf{y}_T|\mathbf{D}_{t^*-1}) = \prod_{t=t^*+1}^{T} \hat{p}_q(\mathbf{y}_t|\mathbf{D}_{t-1}),
$$

where t^* is such that $p_q(\psi|D_{t^*})$ is proper for all $q = 1,\ldots,Q$.

As a result, one selects the model with the highest predictive density. Thus, the selected model will not only be the model with highest posterior probability but will also be the model with the best predictive performance. When one model has posterior probability close to one, that model is clearly the winner. But very often several models will have similar posterior probabilities. In that case, those several models should be reported and predictions should be computed by model averaging [Clyde and George 2004].

5. Application

Within the Bayesian paradigm, the models are complete with the specification of prior distributions for β_0, τ_i, ϕ_i and $\kappa_i, i = 1,2$; the usual assumption of independent priors is used here. The prior for β_0 is a multivariate normal with certain mean vector and diagonal precision matrix with elements close to 0, corresponding to vague information. The prior for τ_i is a gamma distribution $Ga(4,4)$ leading to a gamma full conditional distribution. The prior for ϕ_i proportional to 1 if $0 < \phi_i < 1$ and proportional to ϕ_i^{-5} if $\phi_i \geq 1$. The prior for κ_i is a uniform distribution in the interval $(0,1)$. Usually is not an easy task the estimation of ϕ_i and τ_i, hence their priors are somehow semi-informative.

For each model, the MCMC scheme discussed in Section 3 was used with the simulation of the latent process by FIFBS. Moreover, the updating for τ_i was performed with independent Gibbs steps and the updating for ϕ_i and κ_i was performed with Metropolis steps for $\log\phi_i$ and for κ_i. This MCMC scheme was implemented using Ox [Doornik 2002]. For each model, 90000 iterations were run and the first 10000 iterations

were discarded as burn-in. Then, we saved every 20th iteration yielding a final sample of size 4000 for each parameter.

For model selection, the first ten time points were used as training sample; that is, $t^* = 10$. Table 1 shows the logarithm of the predictive density for the several models and the mean squared error of the one-step ahead prediction (as a simpler comparison method, since it doesn't take account of all uncertainty). From the table, the best model is the contamination model (Model II). In order to understand the difference in performance between the models, Figure 1 shows the one-step-ahead predictive densities for all the models. Model II is the winner in almost all the time points.

Table 1. Logarithm of the predictive density and mean squared error of the prediction for all the considered models.

Model	Log p.d.	MSE
I	-1881.40	1766.06
II	**-1873.36**	**1749.60**
III	-4254.69	13164.13
IV	-2364.12	3033.33
V	-5815.18	13857.93
VI	-3365.69	8094.99
VII	-2227.64	2337.85
VIII	-2008.27	2228.14

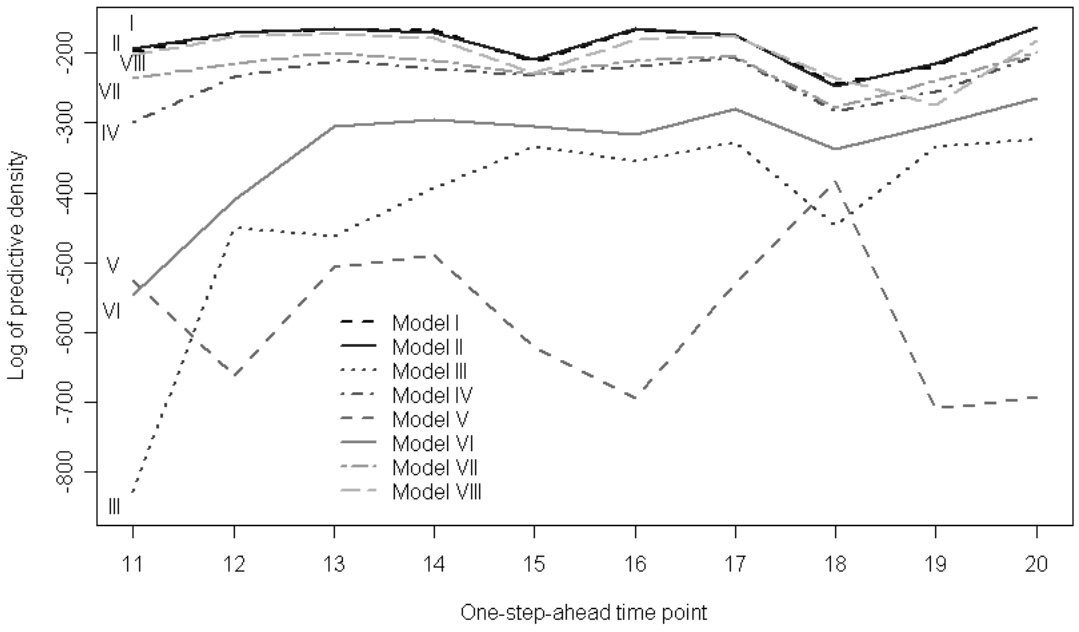


Figure 1. One-step ahead logarithm of the predictive density.

Table 2 shows point estimates of the parameters of the contamination model using all time observations and the respective quantiles in order to elaborate confidence intervals

of 95%. These results show that the innovations have a strong spatial correlation and a moderate precision indicating that the innovations have low magnitude. The value of the contamination index κ indicates very low but (statistically) significant interaction between sites in subsequent times.

Table 2. Posterior results using all data for the best model selected: Contamination model (II).

		Quantiles		
Parameter	Mean (s.d.)	2.5%	50%	97.5%
τ	1.518 (0.441)	0.795	1.459	2.514
ϕ	6.469 (2.376)	3.081	6.108	7.778
κ	0.0003 (0.0002)	0.00002	0.0003	0.0008

Figure 2 shows the posterior means of both the latent level (first row of each panel) and the innovation process (second row of each panel) for the years 1981, 1984, 1987, 1990, 1993 and 1996. The maps on the first rows of each panel represent the posterior mean of the risk per 100 thousand inhabitants in a scale from white (low risk) to black (high risk). In the beginning, the risk level was almost uniform for all counties, including metropolitan area (east-center region). Moreover, throughout the years the violence increases at the center of the State and some northern counties. The maps on the second rows of each panel represent the posterior mean of the innovations in a scale from white (high reduction in the risk) to black (high increase in the risk). These innovation maps are very informative, as they represent estimates of year specific spatially structured effects. For example, after accounting for the contamination effect, in 1993 there was a year-specific increase of the risk level in the center and southern regions and a decrease of the risk level in the northern part of Espírito Santo State.

6. Conclusions

We have presented a method to perform model selection of spatio-temporal models for areal observations in the exponential family. This method is based on predictive densities and in this paper we present a specific general spatio-temporal model for count data. Several different specifications of this model are presented and applied to the annual number of homicides in the State of Espírito Santo in the 1979-1998 period.

The proposed Bayesian analysis using MCMC with embedded FIFBS allows for full account of the uncertainty and the predictive-density-based model selection pointed to the contamination model as the best model among the proposed ones. A possible explanation is that neighbor counties may have similar security policies causing an increasing or decreasing violence process. This behavior was reflected on the maps representing the innovations, showing estimates of spatially structured effects.Further research will include different models for the homicides maybe considering some covariates; and model selection for other kind of data, like binomial data.

References

Bernardinelli, L., Clayton, D., Pascutto, C., Montomoli, C., Ghislandi, M., and Songini, M. (1995). Bayesian analysis of space-time variation in disease risk. *Statistics in Medicine*, 14:2433–2443.

Brown, P. E., Diggle, P. J., Lord, M. E., and Young, P. C. (2001). Space-time calibration of radar-rainfall data. *Applied Statistics*, 50:221–242.

Carter, C. K. and Kohn, R. (1994). On Gibbs sampling for state space models. *Biometrika*, 81:541–553.

Clyde, M. and George, E. I. (2004). Model uncertainty. *Statistical Science*, 19:81–94.

Doornik, J. A. (2002). *Object-Oriented Matrix Programming Using Ox*. Timberlake Consultants Press and Oxford, London, 3rd. edition. URL: http://www.nuff.ox.ac.uk/Users/Doornik.

Ferreira, M. A. R. and De Oliveira, V. (2007). Bayesian reference analysis for Gaussian Markov Random Fields. *Journal of Multivariate Analysis*, 98:789–812.

Frühwirth-Schnatter, S. (1994). Data augmentation and dynamic linear models. *Journal of Time Series*, 15(2):183–202.

Frühwirth-Schnatter, S. (1995). Bayesian model discrimination and bayes factors for linear gaussian state-space models. *Journal of the Royal Sttistics Society, Series B*, 57:237–246.

Gamerman, D. and Lopes, H. (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Chapman and Hall, 2nd edition.

Geisser, S. and Eddy, W. F. (1979). A predictive approach to model selection. *Journal of the American Statistical Association*, 74:153–160.

Gelfand, A. E., Dey, D. K., and Chang, H. (1992). Model determination using predictive distributions with implementation via sampling-based methods (with discussion). In Bernardo, J., Berger, J., Dawid, A., and Smith, A., editors, *Bayesian Statistics 4*. London: Oxford University Press.

Ghil, M., Cohn, S., Tavantzis, J., Bube, K., and Isaacson, E. (1981). Applications of estimation theory to numerical weather prediction. In Bengtsson, L., Ghil, M., and Källén, E., editors, *Dynamic Meteorology: Data Assimilation Methods*, pages 139–224. New York: Springer-Verlag.

Green, P. J., Hjort, N. L., and Richardson, S. (2003). *Highly Structured Stochastic Systems*. Oxford University Press.

Guttorp, P., Meiring, W., and Sampson, P. D. (1994). A space-time analysis of ground-level ozone data. *Environmetrics*, 5:241–254.

Huang, H. C. and Cressie, N. A. C. (1996). Spatio-temporal prediction of snow water equivalent using the Kalman filter. *Computational Statistics and Data Analysis*, 22:159–175.

Huerta, G., Sansó, B., and Stroud, J. R. (2004). A spatio-temporal model for Mexico City ozone levels. *Journal of the Royal Statistical Society, Ser. C*, 53:231–248.

Knorr-Held, L. (2000). Bayesian modelling of inseparable space-time variation in disease risk. *Statistics in Medicine*, 19:2555–2567.

Knorr-Held, L. and Richardson, S. (2003). A hierarchical model for space-time surveillance data on meningococcal disease incidence. *Applied Statistics*, 52:169–183.

Robert, C. P. and Casella, G. (1999). *Monte Carlo Statistical Methods*. Springer-Verlag.

San Martini, A. and Spezzaferri, F. (1984). A predictive model selection criterion. *Journal of the Royal Statistical Society, Series B*, 46:296–303.

Schmid, V. and Held, L. (2004). Bayesian extrapolation of space-time trends in cancer registry data. *Biometrics*, 60:1034–1042.

Spiegelhalter, D., Best, N., Carlin, B., and van der Linde, A. (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B*, 64:583–640.

Vivar, J. C. (2004). Uma nova classe de modelos espaço-temporais para dados de área. Master's thesis, Departamento de Métodos Estatísticos, IM – UFRJ, Rio de Janeiro, Brasil.

Vivar, J. C. and Ferreira, M. A. R. (2007). Spatio-temporal models for gaussian areal data. *(Submitted to Journal of Computational and Graphical Statistics)*.

Waller, L. A., Carlin, B. P., Xia, H., and E., G. A. (1997). Hierarchical spatio-temporal mapping of disease rates. *Journal of the American Statistical Association*, 92:607–617.

West, M. and Harrison, P. J. (1997). *Bayesian Forecasting and Dynamic Models*. Springer, New York, 2nd. edition.

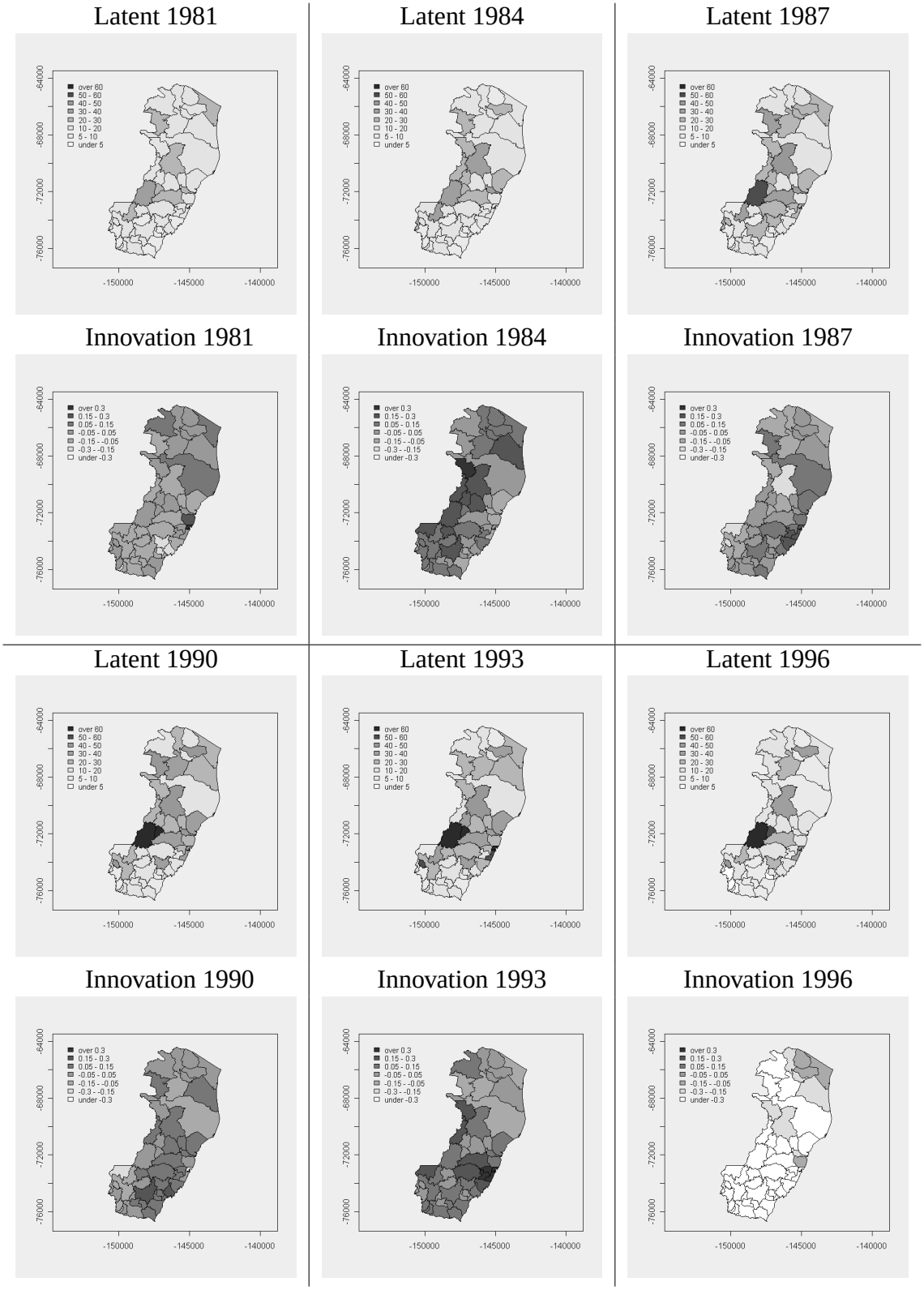


Figure 2. Homicides data - Spatio-temporal contamination model. Posterior means of latent and innovations fields for years 1981, 1984, 1987, 1990, 1993 and 1996.

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 133-148.

Rule-based evolution of typed spatiotemporal objects

Olga Oliveira Bittencourt[1], Gilberto Câmara[1], Lúbia Vinhas[1], Joice Seleme Mota[1]

[1]Image Processing Division – National Institute for Space Research (INPE)
Avenida dos Astronautas 1758 – 12227-001 – São José dos Campos – SP – Brazil
{olga,gilberto,lubia,joice}@dpi.inpe.br

Abstract. *This paper describes a model for spatiotemporal objects whose location is fixed, but its boundaries and properties change. We refer to these as evolving objects. We consider cases where the evolution of an object is dependent of its type and propose a rule-based approach for description of spatiotemporal object's evolution. By developing semantics of type-based evolution, we can keep a detailed history of change. We present an example where the model is able to represent type conversions and recover the evolution history of a set of objects. The model allows answers to questions about causes of change and thus deals with cases not supported by models based on objects of a unique type.*

1. General Information

A major research topic in GIScience is modelling and representation of geographical objects whose properties change. We distinguish two broad categories of spatiotemporal objects. The first case concerns those objects that change their position and extent continuously. We refer to those as *moving objects*. Moving objects arise, for example, in location-based systems that deal with spatial and temporal position of planes, storms or cars. The second type concerns objects that do not move, but whose geometry, topology and properties change. We refer to those objects as *evolving objects*. Evolving objects arise, for example, in urban cadastre and in land change patterns.

The two categories of spatiotemporal objects need different ways of data modelling, representation and algorithms. Handling moving objects demands notions such as *trajectory* (Güting et al., 2000), plus specialized query methods (Sistla et al., 1997) and indexing techniques (Šaltenis et al., 2000). The widespread availability of location-based systems motivated advances in moving objects databases (Erwig and Schneider, 2002) (Güting and Schneider, 2005). By contrast, handling evolving objects requires tracking the changes that occurred during an object's lifetime, such as creation, splitting and merging (Hornsby and Egenhofer, 2000; Medak, 2001).

One recent technique for handling spatiotemporal objects is event-based calculus (Worboys, 2005; Worboys and Hornsby 2004; Vidal and Rodriguez 2005). Event-based calculus captures the semantics of spatiotemporal objects by specialized events that are external to the objects themselves. Each application is associated to a specialized set of events. For example, the semantics of traffic objects uses events such as *departure*, *arrival*, or *unexpected destination* (Hornsby and Cole, 2007). Event-based techniques have proven useful in applications such as traffic models.

In this paper, we deal with evolving objects. We deal with cases where the simple rules of merging and splitting are not enough to describe their evolution. These

situations arise when objects are defined not only by their shape and properties, but also by their *type*. Consider the case of riverbanks. Definition of what is 'the river' and what is 'the land' changes over the seasons. When a river expands during the wet season, the part of the land that is flooded will be split and merged with the river. The object that matches the flooded area will change its type and properties. In the dry season, this object may become land once again. In this evolution, expansions and contractions produce junctions and splitting which are type-dependent. In this and similar cases, recording the history of changes needs keeping track of type-dependent cases. This requires a higher-level of semantics above that of the basic operations of creation, splitting and merging. We shall refer to those objects as *typed evolving objects*. This raises the question we explore in this paper: "*How can we deal with spatiotemporal objects whose evolution is type-dependent?*"

This paper proposes a rule-based approach for description of object's evolution. The rules arise from knowledge about the application domain. They provide a higher-level semantics layer that uses low-level operations and deals with type dependency. By developing semantics of type-based evolution, we can keep a detailed history of changes. Then, we can recover the evolution history of a set of objects and answer important questions about the causes of changes. Our proposal involves defining a set of object types and a set of functions applicable on those types. This leads to an algebraic formulation, which can be implemented easily in a functional language or translated into an imperative language or a specialized query processor.

The rule-based evolution proposed in this paper has some similarities with the event-based calculus. In both cases, we describe changes in objects by a set of occurrences. The main difference is that rule-based evolution uses functions, which are more general than events. An event can be modelled as a function, but there are some functions that are needed to describe an object's evolution which are not occurrents. For example, the function *history* provides a description of the changes in an object. Therefore, the function-based approach can be seen as a generalization of the event-based calculus to include occurrents (modelled as events) and other types of operations.

This paper is organized as follows. In section 2, we present the idea of rule-based algebras for describing evolution of spatiotemporal objects, and review previous work on the subject. In section 3, we present the operations defined for evolving objects. In section 4, we present a case study on using rule-based evolution. In section 5, we discuss how to implement the proposal. In section 6 we present some conclusions and future work.

2. Rule-based Evolution of Spatiotemporal Objects and its Relation to Previous Work

Pelekis et al. (2004) and Roddick et al. (2004) review the different types of spatiotemporal data models proposed in the literature. They point out the differences between models that describe moving objects and those focused on object lifelines. Moving objects are the ones whose position and extent change continuously. Güting et al. (2003) and Güting and Schneider (2005) propose an algebra for moving objects, composed by a set of spatiotemporal data types, axioms and their operations. Algebraic data types provide a conceptually clean foundation for representation and querying of moving objects (Güting et al., 2003). The specific data types defined to handle moving

objects are *moving points* (objects where only the position in space is relevant) and *moving regions* (objects where the position and the time-dependent extent are relevant). This algebra was implemented using SECONDO (Güting et al., 2004), an extensible and modular DBMS environment created to support development of algebras. The set of data types and operations can answer questions such as: "*Given the trajectories of snowstorms and aeroplane flights, which flights went through a snowstorm?*".

A second area of research concerns object lifelines. These works focus on describing the history of incremental changes in the life of an object. This happens in applications such as urban cadastre, where parcels are created, merged, split and wiped out. To keep track of an object's history, all changes need to be modelled and recorded. Hornsby and Egenhofer (2000) stress the need to preserve an object's identity as it changes its geometry, topology, and attributes in time. This view is also supported by Grenon and Smith (2003). Medak (2001) developed an algebra to model object lifelines. Medak's algebra provides a set of basic operations, which are a foundation for more specific applications. The literature on object lifelines uses three basic ideas: identity, life, and genealogy. Identity is the characteristic that distinguishes each object during all its life. Life is the time period from the object's creation until its elimination. Genealogy implies managing the changes that an object has during its life. Medak (2001) and Hornsby and Egenhofer (2000) propose the use of ideas such as *creation*, *destruction*, *merging*, *splitting*, *death* and *reincarnation* to record the history of the objects, following the changes and keeping its identity.

Our paper focuses on object lifelines. The current work on object lifelines focuses on objects of the same type and the usual examples involve parcels and counties. For more complex cases, we need to consider operations involving objects of different types, which arise in many real-life applications. Consider the question "*when does a tropical storm become a hurricane?"* To answer this question, we need to consider more than the trajectory of the storm. There are many conditions that determine how tropical storm becomes a hurricane. They include the storm's wind-force, the sea surface temperature and the trajectory. These conditions need to be fed into a set of rules that will eventually convert an object of type '*tropical storm'* to an object of type '*hurricane*'.

As a second example, consider the history of Bangladesh during the last 800 years. Islam was introduced to Bengal in the XII century. By the XVI century, the Mughal Empire controlled the area around the Bay of Bengal. The British gained control of Bengal in 1757. When India was partitioned in 1947, Bengal was split along religious lines, with the western part going to India and the eastern part joining Pakistan as a province called '*East Pakistan'*. The people from Bangladesh gained their independence from Pakistan in 1971 (Wikipedia, 2007). Thus, the region changed their status many times during the last 800 years. A history of the region that would consider the Islam culture as its building block would need to account for the various status changes. Consider the Islamic area around the Bay of Bengal as a spatiotemporal object. Its status changed from '*part of an empire'* to '*part of a province of an empire*' and then to ' *province of an independent country*' and finally to '*independent country*'.

These examples lead us to consider how to enrich our models of spatiotemporal objects to capture the complexity of such changes. They lead us to propose the idea of *typed spatiotemporal objects*. Our view of types comes from Computer Science, where

types are tools for expressing abstractions in a computer language (Cardelli and Wegner, 1985). On a theoretical level, a type is a set of elements in a mathematical domain that satisfy certain restrictions. A *typed spatiotemporal object* is an object whose evolution is subject to constraints that are specific to its type. Thus, objects of type '*independent country*' will have different rules for evolution than those of type '*province*'. Using types to model the evolution, we can gather richer models for capturing the semantics of evolving objects.

3. Operations for Evolving Objects

To carry out the evolution in a computer model we defined specific operations to evolving objects. This section informally presents these operations with some definitions and conventions we adopted:

1. Data types, functions and instances use `monospaced` font. Types are in **`boldface,`** their instances and reserved words are shown in `normal` font. For lists associated to types, we use **`list_typename`**.
2. Type definitions follow usual conventions for abstract data types. Types have an externally viewable set of functions and a set of axioms that are applicable to these functions.
3. Each function has a signature, given a
   `function:` **`typeA × typeB → out_type`**.
   `typeA` and **`typeB`** are the types of the input parameters and **`out_type`** is the type of the output.
4. We describe each rule in pseudocode. For attribution, we use ':='. '`As`', '`in`' and '`with`' are reserved words. For sets of objects, we use '[]'. For groups, we use '{}' and the separator is ';'.

We defined the functions *create*, *split*, *merge*, *evolve*, *setType*, *getType*, *getInstance* and *remove* in our experiments. Table 1 presents an informal set of signatures and explanations for these evolving operations.

Table 1. Informal definition of evolving operations

Function	*Signature*
`create`	**`timestamp x type → ST_object`** Given a specific type and its timestamp, create an instance of a spatiotemporal object of the same type.
`getInstance`	**`ST_object x timestamp → S_object`** Recover the static instance of the spatiotemporal object, given a timestamp.

setType	**ST_object x type → ST_object** Set the type of the spatiotemporal object.
getType	**ST_object x timestamp → type** Given a timestamp, retrieve the type of the spatiotemporal object.
merge	**ST_object x ST_object x timestamp → ST_object** Given two evolving objects, merge them produce an object based on the evolution rules.
split	**ST_object x ST_object x timestamp → ST_object x ST_object** Given two evolving objects, split them produce two objects based on the evolution rules.
remove	**ST_object → null** Removes the spatiotemporal object of the model.

We created the *evolve* function to allows grouping evolutions according with similarity ideas or specific actions significant to user. These evolutions can be further recognized during the history recovery. Its signature is:

`evolve` **`identifier timestamp { functions }`**

Further, we defined a parametric function to recover the *history* of objects. This function allows defining richer and relevant forms of recovering information inside the history of objects. It is more than just recovering the static operations. Our distinct signatures give us kinds of looking the history by different points of view, recovering distinct information and combining it with relevant information related to spatiotemporal object's evolution. The basic signature of *history* is:

`history` **`ident_option form_option time_option → [ST_object]`**

Each parameter option and their combinations, presented in Table 2, allow recovering different aspects of evolving object *history*.

Table 2. Parametric function *history*

Parameter	*Signature*
ident_option	**ST_object → [ST_object]** Recover the evolution of a spatiotemporal object using its identifier. This evolution consists in a set of evolving objects with their respective timestamps.

	Alias → [ST_object] Recover the evolution of a spatiotemporal object using the alias defined to the spatiotemporal object.
form_ option	complete **→ [ST_object]** Recover the complete history of the spatiotemporal object, including all evolutions that happened with any split or merge of this object. The idea is recovering the tree with the original object on the root and all subtrees with its evolutions.
	reverse **→ [ST_object]** Recover the reverse history of the object with the focus on the last evolution to its beginning.
time_ option	from **timestamp → [ST_object]** Fix the early timestamp to recover the history. The result will contain information until the last evolution of the spatiotemporal object.
	until **timestamp → [ST_object]** Fix the end timestamp to recover the history. The result will contain information from the spatiotemporal object creation until the evolution before or equal the mentioned timestamp.
	from **timestamp** until **timestamp → [ST_object]** Fix the early and end timestamp to recover the history of the spatiotemporal object.

4. A Simple Example of Rule-based evolution

To evolve our objects, we used a rule-based evolution approach. The idea is to derive a set of rules from domain knowledge. These rules will act as the constraints on the specific types of spatiotemporal objects. In a general case, we propose a series of steps for using the concepts of typed spatiotemporal objects and rule-based evolution for modelling a real-life case, as outlined below:

1. Use a domain expert to elicit the different types of objects needed to model the problem.
2. Use a domain expert to set up the rules that govern the evolution of the different types of objects.
3. Express the object types and the evolution rules in a computer model. Preferably, develop a set of algebraic data types and operations on them.
4. Use the computer model to capture the history of the study area.

In the next subsection, we present a case study where the idea of rule-based evolution was applied to model the evolution of Bangladesh case.

4.1. The Bangladesh case

This section exemplifies the rule-based evolution of spatiotemporal objects. We present a case study focusing on the history of Bangladesh (illustrated in Figure 1) during the last 800 years. The history of the region that would consider the Islam culture as its building block would need to account for the various status changes. It is on that viewpoint that our example focuses on.

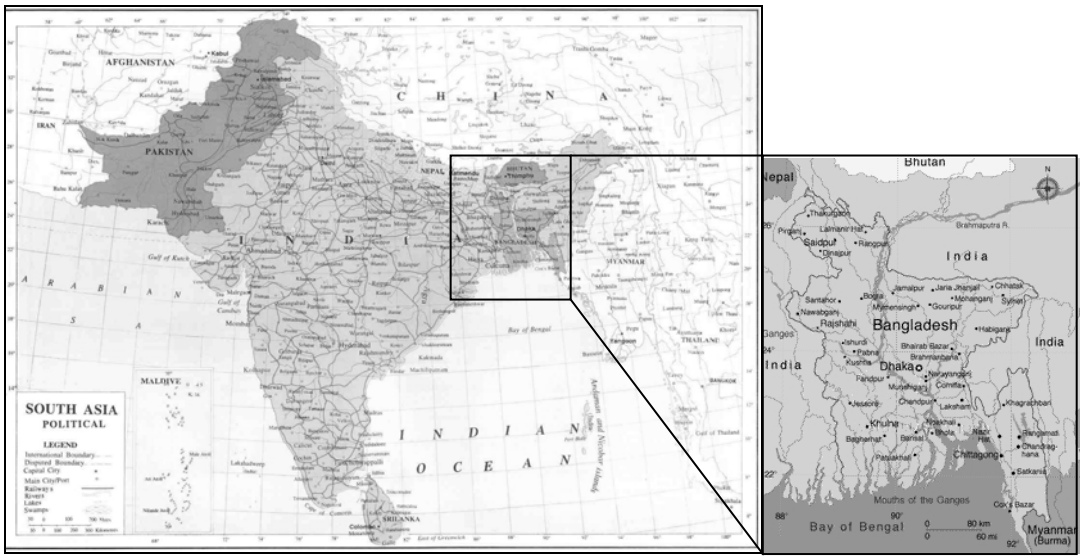


Figure 1. Illustration of Bangladesh in South Asia.

Consider the Islamic area around the Bay of Bengal as a spatiotemporal object. Bangladesh status changed from '*part of an empire'* to '*part of a province of an empire*' and then to '*province of an independent country*' and finally to an '*independent country*'. A model to account for its evolution would have rules, rather defined by a domain expert, such as the following:

R1. *Splitting a province produces two provinces.*

R2. *Splitting a province of one Empire produces one province and one Empire without that province.*

R3. *Splitting a province of one Country produces one province and one Country without that province.*

R4. *Merging two provinces produces one province.*

R5. *Merging a province and a country produces a Country with a new region and one region that is part f the Country.*

R6. *Merging a province and an Empire produces an Empire with a new region and one region that is part f the Empire.*

Table 3 summarizes the main information captured on these rules. It presents rules, functions, input types of evolving objects and output types of resulting evolving objects.

Table 3. Summary of Bangladesh case rules

Rule	Function	Input Types		Output Types	
		Status	Status	Status	Status
Province_ split	split	Province		Province	Province
Province_ independence	split	Province part of an Empire	Empire	Province	Empire
Province_ independence	split	Province part of a country	Country	Province	Country
Province_ conquest	merge	Province	Province	Province	
Country_ conquest	merge	Country	Province	Country	Province part of a Country
Empire_ conquest	merge	Empire	Province	Empire	Province part of an Empire

4.2. Modelling the evolution

Our spatiotemporal objects evolve as *merging* and *splitting* as the result of conquests and independence events. The set of rules defined by the domain expert govern the evolution of these events. The rules are expressed in a computer model through the combination of input types of evolving objects and the possible functions defining the output evolving objects. The evolution of rules is expressed by the operations on them.

To exemplify the evolutions and to be able of querying the history by different viewpoints, we defined some operations that describe the main events occurred. Consider the indications of area_x, for example area_1, the spatial object. The model allows answer questions such as: *"How regions evolve? What happens when two objects merge?"* and understand how changes occurred in Bangladesh.

1. Islam was introduced to Bengal in the XII century.

```
create area_1 as "Ancient Asia" with status := "Empire";
create area_2 as "Bay_of_Bengal";
create area_3 as "Bengal" in "XII century";
setType Bengal with religion := "Islam";
merge "Ancient Asia" "Bengal";
```

The evolution on merge operation will follow the rules for merge and split operations. It will detect that the applied rule is 'Empire_conquest' of 'Ancient Asia' Empire that evolves Bengal to 'Province part of an Empire'.

2. By the XVI century, the Mughal Empire controlled the area around the Bay of Bengal.

```
create area_4 as “Mughal Empire” with status := “Empire”;
evolve Mughal_Empire_control_Bengal in “XVI century”
{
    split “Ancient Asia” Bengal;
    merge “Mughal Empire” Bengal;
};
```

This step contains two different evolutions. The first is a ‘Province_independence’ where Bengal is a Province that is not anymore part of ‘Ancient Asia’ and ‘Ancient Asia’ is an Empire without Bengal Province. The second evolution is an ‘Empire_conquest’ where Bengal is ‘Province part of an Empire’, ‘Mughal Empire’ in this case.

This step also presents the option of grouping evolutions in a meaningful event defined by ‘Mughal_Empire_control_Bengal that favours the understanding of evolutions. Other steps in this example continue representing changes on the area.

3. The British gained control of Bengal in 1757.

```
create area_5 as “United Kingdom” with status := “Empire”;
evolve British_control_of_Bengal in 1757
{
    split “Mughal Empire” Bengal;
    merge “United Kingdom” Bengal;
};
```

4. When India was split in 1947, Bengal was split along religious lines, with the western part going to India and the eastern part joining Pakistan as a province called ‘*East Pakistan*’.

```
create area_6 as “Pakistan” with status := “Country”;
create area_7 as “India” with status := “Country”;
evolve divisao_provincia_Bengala in 1947
{
    split “Britain” Bengala;
    split Bengala area_4 as West_Bengal and East_Pakistan;
    merge East_Pakistan Pakistan;
    merge West_Bengal India;
};
```

5. The people from Bangladesh gained their independence from Pakistan in 1971.

```
evolve Bangladesh_independence in 1971
{
    split Pakistan East_Pakistan;
    create East_Pakistan as Bangladesh with status :=
    “country”;
};
```

4.3. Recovering the history

To exemplify some uses of history function, tables 4 and 5 show the result of two history queries. These tables include the most relevant information about the model: (a) the timestamp; (b) operations applied on the spatiotemporal objects; (c) the optional alias on that timestamp; (d) the spatiotemporal object's type; (e) the typed evolution of the spatiotemporal object on that timestamp.

Table 4 shows the answer to the query '`history Bangladesh from 1971`'. This query focuses on changes in the object since its creation in 1971. Then, the only result that this query will recover is the creation of the object in 1971.

Table 4. Result of 'history Bangladesh from 1971'

Timestamp	*Operation*	*Alias*	*Status Type*	*Evolution*
1971	Creation	Bangladesh	Country	Bangladesh_independence

Table 5 shows a summary of the answer to the query '`history Bangladesh reverse`'. This query specifies the interest in all changes that occurred with the object Bangladesh and its predecessors. The query specifies the reverse order, then, the user focuses on looking from the last Bangladesh evolutions until the beginning of the generation.

Table 5. Answers to 'history Bangladesh reverse'

Timestamp	*Operation*	*Alias / Status Type*	*Evolution*
1971	Split Pakistan; creation;	Bangladesh / Country	Bangladesh_independence
1947	Split Britain; merge Pakistan;	East_Pakistan / Part of a Country	divisao_provincia_Bengala
1715	Split Mughal Empire; merge with Britain;	Bengal / Province part of a Province of an Empire	British_control_of_Bengal
XVI century	Split Ancient Asia; merge Mughal Empire;	Bengal / Province part of an Empire	Mughal_Empire_control_Bengal
XII century	Creation; Type religion: = Islam; Merge Ancient Asia;	Bengal / Province part of an Empire	Nondefined

With this approach and the operations we are able of easily answering questions such as *"How do the Bengal region of XVI century evolved?", "What happened when the Bengal region was split?" or "What happened when East_Pakistan merged with Pakistan?"*. These questions show some new possibilities that our approach allows to model and query data.

5. Implementation Road Map

This section presents an implementation road map to the rule based evolution approach. The first step is to create evolution rules composed by: 1) detecting the objects of interest; 2) analysis of evolution cases by the specialist and; 3) development of a set of evolution rules. Through the set of rules it is possible to define the second step: proposing and implementing a group of operations to characterize and evolve the spatiotemporal objects.

The set of evolution rules leads to an algebraic formulation, which can be easily implemented in a functional language or translated into an imperative language or a specialized query processor. From an implementation point of view, the rule-based evolution of typed spatiotemporal objects is possible and simple.

Currently, we are in the second step on the development of a model to land use changes in the Amazon region (Escada et al., 1997, INPE, 2005). This is a complex scenario and future work will present this complete model based on the idea of evolution rules. The experiments are being performed in a prototype developed in the environment TerraHS (Costa et al., 2006). TerraHS integrates the functional programming and the spatial databases for GIS application development. We are using TerraHS because it allows prototyping algebras and we are implementing our evolving objects and functions in an easy and complete manner.

6. Conclusions

In this paper, we deal with *evolving objects*. We are interested in cases where the simple rules of merging and splitting are not enough to describe their evolution and the evolution of objects is dependent of their *types*.

This paper proposes the concept of *typed spatiotemporal objects* and the use of rule-based evolution approach to capture a detailed history of changes of spatiotemporal objects. Rule-based evolution works best when the domain knowledge is well known, and we are able to assign a meaningful type system to the objects. Our proposal involves defining a set of object types and a set of functions applicable on those types. Then, we can recover the evolution history of a set of objects, answer important questions about causes of change and thus deals with cases not supported by models based on objects of a unique type.

Future works will be realized on other areas and scenarios. Currently, we are interested in studying the evolution of land use changes in the Amazon region. For next steps, an algebra of evolving objects will be developed as well as new operations to advance our model and to characterize other problems.

Acknowlegment

Gilberto Camara's work is partly funded by CNPq (grants PQ – 300557/19996-5 and 550250/2005-0) and FAPESP (grant 04/11012-0). Olga Oliveira Bittencourt's work is funded by CAPES.

References

CARDELLI, L. and WEGNER, P., 1985, On Understanding Type, Data Abstraction, and Polymorphism. *ACM Computing Surveys,* **17,** 471-552.

COSTA, S. S., CÂMARA, G., and PALOMO, D., 2006, TerraHS: Integration of Functional Programming and Spatial Databases for GIS Application Development. In: *Advances in Geoinformatics, VII Brazilian Symposium in Geoinformatics, GeoInfo2006*, Davis, C. A. Jr. and Monteiro, A. M., eds. (Campos do Jordão, Brazil: Springer), **24**, 127-148.

ERWIG, M. and SCHNEIDER, M., 2002, Spatio-Temporal Predicates. *IEEE Transactions on Knowledge and Data Engineering,* **14,** 881-901.

ESCADA, M. I. S., MONTEIRO, A. M., AGUIAR, A. P., CARNEIRO, T. and CAMARA, G., 2005, Análise de padrões e processos de ocupação para a construção de modelos na Amazônia (Analysis of land use patterns and processes for the construction of models in Amazonia: Experiments in Rondonia). In *XII Brazilian Symposium on Remote Sensing*, (Goiania, Brazil: SELPER), 2973-2983.

GRENON, P. and SMITH, B., 2003, SNAP and SPAN: Towards Dynamic Spatial Ontology. *Spatial Cognition & Computation,* **4,** 69-104.

GÜTING, R. H., BEHR, T. and ALMEIDA, V., 2004, SECONDO: An Extensible DBMS Architecture and Prototype. Report (Hagen: Fernuniversität Hagen).

GÜTING, R. H., BOHLEN, M. H., ERWIG, M., JENSEN, C. S., LORENTZOS, N., NARDELLI, E., SCHNEIDER, M. and VIQUEIRA, J. R. R., 2003, Spatio-temporal Models and Languages: An Approach Based on Data Types. In *Spatio-Temporal Databases*, Koubarakis, M., ed., (Berlin: Springer).

GÜTING, R. H., BÖHLEN, M. H., ERWIG, M., JENSEN, C. S., LORENTZOS, N. A., SCHNEIDER, M. and VAZIRGIANNIS, M., 2000, A Foundation for Representing and Querying Moving Objects. *ACM Transactions of Database Systems,* **25**.

GÜTING, R. H. and SCHNEIDER, M., 2005, *Moving Objects Databases.* (New York: Morgan Kaufmann).

HORNSBY, K. S. and EGENHOFER, M., 2000, Identity-Based Change: A Foundation for Spatio-Temporal Knowledge Representation. *International Journal of Geographical Information Science,* **14,** 207-224.

HORNSBY, K. S. and COLE, S., 2007, Modeling Moving Geospatial Objects from an Event-based Perspective. *Transactions in GIS*, **11(4)**, 555-573.

INPE, 2005, Monitoramento da Floresta Amazônica Brasileira por Satélite (Monitoring the Brazilian Amazon Forest by Satellite). Report (São José dos Campos: INPE).

MEDAK, D., 2001, Lifestyles. In *Life and Motion of Socio-Economic Units. ESF Series*, A. U. Frank, Raper, J., & Cheylan, J.-P., ed., (London: Taylor & Francis).

PELEKIS, N., THEODOULIDIS, B., KOPANAKIS, I. and THEODORIDIS, Y., 2004, Literature review of spatio-temporal database models. *The Knowledge Engineering Review*, **19,** 235-274.

RODDICK, J., EGENHOFER, M. and HOEL, E., 2004, Spatial, Temporal and Spatiotemporal Databases Hot Issues and Directions for PhD Research. *SIGMOD '93, SIGMOD Record,* **33,** 126-131.

ŠALTENIS, S., JENSEN, C. S., LEUTENEGGER, S. T. and LOPEZ, M. A., 2000, Indexing the positions of continuously moving objects. *ACM SIGMOD Record*, **29,** 331-342.

SISTLA, A. P., WOLFSON, O., CHAMBERLAIN, S. and DAO, S., 1997, Modeling and Querying Moving Objects. *Proceedings of the Thirteenth International Conference on Data Engineering,* 422-432.

VIDAL, C. and RODRIGUEZ, A., 2005, A Logical Approach for Modeling Spatio-temporal Objects. *2nd International Workshop on Conceptual Modeling for Geographic Information Systems, CoMoGIS 2005*, 218-227.

WIKIPEDIA, 2007. Wikipedia, The Free Encyclopedia. Retrieved in October, from: http://en.wikipedia.org/wiki/Bangladesh.

WORBOYS, M. F., 2005, Event-oriented approaches to geographic phenomena. *International Journal of Geographical Information Science,* **19(1)**, 1-28.

WORBOYS, M. F. and HORNSBY, K. S., 2004, From Objects to Events: GEM, the Geospatial Event Model. In: *Third International Conference, GIScience 2004*, Egenhofer, M. J., Freksa, C. and Miller, H. J. eds., (Adelphi, USA: Springer), 327-343.

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 147-158.

An Architecture Based on Multi-Agent Systems and Geographic Databases for the Development of Georeferenced Ecological and Social Simulations

Pablo Souza Grigoletti[1,*], **Antônio Carlos da Rocha Costa**[1,2]

[1]Instituto de Informática/PPGC – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

[2]Escola de Informática/PPGINF – Universidade Católica de Pelotas (UCPel)
Rua Félix da Cunha, 412 – 96.010-000 – Pelotas – RS – Brazil

grigoletti@gmail.com, rocha@atlas.ucpel.tche.br

Abstract. *This work is situated in the intersection of 4 different areas: Social Simulations, Ecological Simulations, Multi-Agent Systems and Geocomputing. Its main objective is to propose a multi-agent architecture designed for the development and execution of geographically referenced social and ecological simulations, that is, social and ecological simulations that use geographically referenced data taken from a Geographic Database, supporting the dynamic modeling of social and/or ecological systems. First of all, is introduced the motivations and objectives of the development of this new architecture, detailing the advantages of join these 4 areas, and the needs of the existent architectures. The features of this architecture and a complete study case are presented too. Finally, a comparison with others architectures and the conclusions are described.*

1. Introduction

Studies about the relationships between the mankind and its environment are each time more present in the scientific community. In this way, the Computer Science has been giving important contributions, providing simulation tools for making the analysis of this kind of relationship. A great number of computer models have been developed to study the complex systems that compose these relationships.

A model is a simplification – smaller, with less details and less complexity – of a structure or system [Gilbert and Troitzsch 1999]. In general, the process of creating a model is one of the most important pieces of the simulation development.

Simulations are very helpful when it is necessary to understand complex systems. In many cases, researchers do not have success when they model systems using pure analytical mathematical methods. Simulations of social and ecological phenomena may then be used as tools for analyzing and understanding the complexity of the phenomena.

For instance, when studying the consequences that human actions produce on the environment, those simulations focus on modeling the set of effects of the population of agents on the common environment, by the study of aspects such as the complexity,

[*]CNPq scholarship holder - Brazil.

emergence, self-organization and dynamics created by the agents' actions on the environment [Castelfranchi 1998][Ferber 1999].

The use of Multi-Agent Systems (MAS) as a tool for the development of social and ecological simulations strengthens the scientific studies in those areas. The main reason for this is that it is possible to directly apply the concepts of agents and agents' society on the modeling of natural societies. In this way, nowadays MAS are frequently used for this kind of simulation [Gilbert and Troitzsch 1999].

On the other hand, the study of spatial phenomena resulted, in the last decades, in the development of specific computational systems for the modeling and analysis of the geographic space: the Geographic Information Systems (GIS). A GIS is an information system that allows the capture, modeling, manipulation, recovery, analysis and representation of geographically referenced data [Worboys 1995]. In a GIS, the storage component generally is called Geographic Database (GDB).

The use of georeferenced social and ecological data (stored in a GDB, for example) helps a lot to increase the realism of the simulation of anthropic actions, that is, the men's actions on the environment. It allows to show clearly the dependencies that may exist between the social process and the physical environment where it happens. This kind of social and ecological simulation is frequently called geographically referenced social simulation [Boero 2006].

This work presents an architecture based on MAS that supports the development and execution of geographically referenced social and ecological simulations. Simulations performed using this architecture may access spatial information available in a GDB.

The paper is organized as follows. In Section 2, the main motivations and objectives of the work are introduced. A review of related works is presented in Section 3. Section 4 details the MAS architecture for georeferenced social and ecological simulations. A complete study case is presented in Section 5. Section 6 brings the conclusion.

2. Motivations and Objectives

The four main motivations for this work are:

- according to [Bordini et al. 2005], in the MAS area, and consequently in social simulations based on MAS, the modeling and representation of the environment where the agents are situated is an extremely important aspect, however not so much explored. In reactive agent systems, the agents do not have memory and, in this case, the environment has an important function, since it is only by perceiving the environment that agents can take decisions. In cognitive agent systems, the agents have an internal representation of the environment in which they are situated. In this case, these agents take their decisions based on the changes that their perceptions cause in their internal representation. In both cases, the modeling of the environment is an extremely important activity in the of the simulations.
- for some time now, Cellular Automata (CA) have been used in simulation models to represent social and ecological environments. However, the traditional CA theoretical base [Neumann 1966] does not allow that the automata move in the environment, the only changes allowed are the changes in the information in the

automata cells. In a MAS based model both are possible: information concerning locations in the environment can be changed, and the automata (in this case, agents) can move in the environment;

- Traditionally, Geocomputing emphasized the representation of spatial phenomena in a static way, so that the main abstraction used is the map. However, there are a lot of spatial phenomena that are dynamic, and this static representation can not capture it well. In this way, abstractions that allow an appropriate representation of spatio-temporal phenomena are a need in this area [Pedrosa and Câmara 2004], and recent work has been done to supply this need (e.g. [Rocha et al. 2001]);
- strengthening the need to develop spatio-temporal models, [Santos 1996] has described the space as "indivisible of the human beings that inhabit and modify it all the time".

Based on these motivations, the main objective of this work is to provide an architecture, based on MAS and GDB concepts, for the development and execution of georeferenced social and ecological simulations. We intend to analyze the advantages of joining these two areas in the simulation context, focusing on the spatio-temporal representation of the entities that exist in a physical environment.

3. Related Works

With the aim of identifying the important features for a new simulation architecture, an analysis of the features of some agent-based simulation platforms was done: **(i)** *Swarm* [Minar et al. 1996], **(ii)** *Repast* [North et al. 2006], **(iii)** *SeSam* [Klügl et al. 2006], **(iv)** *NetLogo* [Tisue and Wilensky 2004], **(v)** *OBEUS* [Torrens and Benenson 2005] e **(vi)** *SMA-SIG* [Gonçalves 2003]. The main results of this analysis are:

- few platforms provide abstract functions to create movement behaviours in the modeled entities; however, this kind of behaviour is very important to obtain more realistic simulations.
- a dynamic connection with the GDB allows a better use of the geographic data, providing an easy way to do complex spatial queries; however, few platforms provide such feature.
- it is not enough to have simulation platforms that allow the modeling of the environment and of the geographic attributes of entities; it is necessary to create perceptions and behaviours for the agents that use these features.
- it is important to use programming structures that allow users with little programming expertise do create simulation models without having to expend large efforts to learn programming techniques;
- it is necessary that the platform provides some well acepted standard for the communication between the active entities of the simulation model. These communication means are useful when it is necessary to create simulations (situations) that the entities should cooperate;
- the use of a discrete-time scheduler makes easy the simulations analysis, allowing the user to check simulation information at each time step.

In this way, the proposed architecture intend that puts together the positive features of the available simulation platforms, and overcome the shortcomings that they present. This architecture and its features are presented in the next section.

4. Proposed Architecture

In the proposed architecture there are two basic kinds of agents: **mobile agents** (agents that can change their position on the environment) and **fixed agents** (agents that can not change their position on the environment). Both mobile agents and fixed agents occupy space in the environment. These agents are specialized according to their geometric shape: *point agent, line agent, polygon agent*. These agents' shapes have a direct relationship with the ones used in GDB to represent objects from the real world. There is also a special kind of agent that does not have shape and does not occupy in the environment, and may be used to gather information about the simulation in a way programmed by the user (in addition to the automatic gathering of information performed by the simulator).

In this architecture the agents are organized using the concept of *layer of agents*. This organization mode makes easy the development of a model and does not interfere in the ways the agents interact. In an abstract vision of the architecture (presented in Figure 1), it is possible to identify three main layers:

- **auxiliar layer:** it is an optional layer. It contains auxiliar agents. The use of these agents aims in help on the development of the model. They are used to collect some information to ease the analysis of the simulations and do not represent agents that exist in the system being simulated;
- **social layer:** it contains the agents of the society that is being modelled;
- **spatial layer:** it contains the agents of the environment that is being represented.

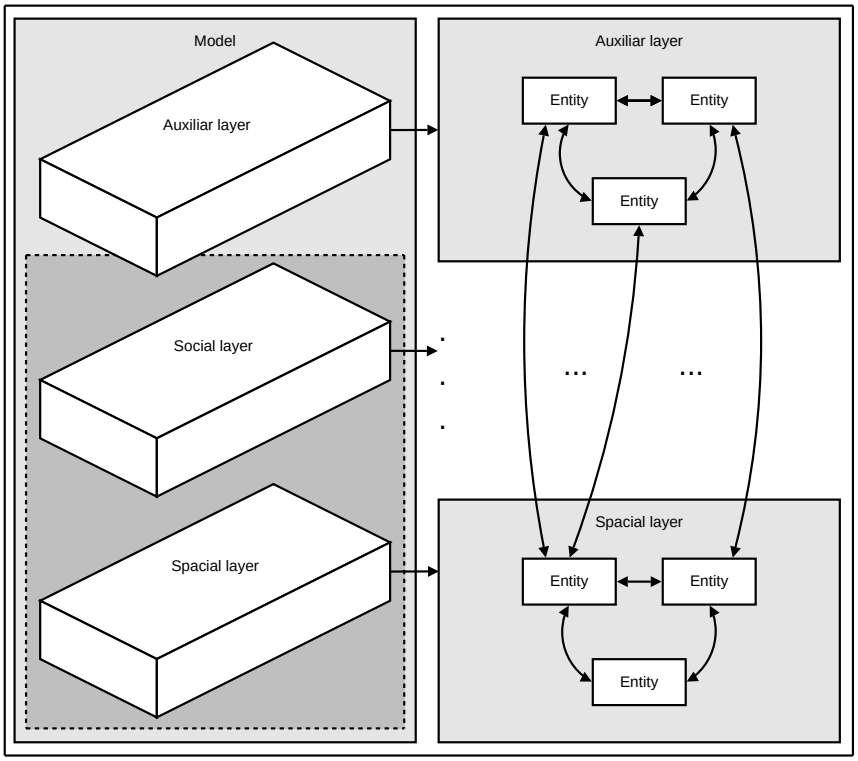


Figure 1. Abstract model of the proposed architecture.

In some cases, the social and spatial layers can be coupled and represented as just one layer. This unique representation strengthens the ideas presented in [Santos 1996],

where the geographic space is described as an indivisible of the human beings that inhabit and modify it all the time. In such cases, the aim is to join the elements that compose the environment (the geographic objects that represent the real world) and the events that change the structure of this environment (the human actions and physical processes).

When the agents are created, their shape, location, attributes, perception and behaviour is defined. In general, the location and some other attributes are data from the GDB. The shape, behaviour, perception and another attributes are defined by the user (possibly in a configuration file), as is illustrated in Figure 2. In this way, *the agents in the simulation model represent, in a form subject to dynamic evolution, the static data gathered from the GDB*.

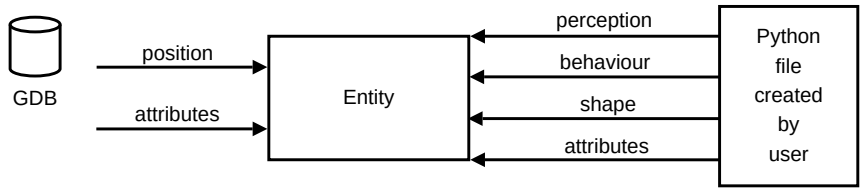


Figure 2. Definition of an agent.

The scheduling of the agents' behaviours is based is a discrete event system scheduler. It operates according to two possible policies: sequential or random. Given the problems that the scheduling policies may introduce [Michel et al. 2001], both policies are necessary to make sure that the scheduler will not influence the simulations results.

The agents' behaviours are defined using the Python language syntax, which is a "Very High Level Language" (VHLL) and has a very simple syntax.

Mobile agents may change their position in the environment and then can explore different places. In this way, mobile agents can perform different actions, according to the places in which they are. To achieve more realistic simulations in an easier way, the architecture provides a set of pre-defined high-level abstraction movement behaviours, such as the ones presented in [Reynolds 1999], which the user/programmer may easily incorporate in his agents.

The integration with the GDB happens in a dynamic way. The access to the geographic data is realized during all the simulation period. In this way, all the GDB functions and operators can be used by the entities. In this context, some important tools, techniques and structures of Geocomputing may be directly used in the simulations, allowing the use of more detailed geographic information and consequently the development of better spatial models for simulations.

Two or more agents can be declared to be adjacent. When this happens and one of the agents changes its shape or position, the shape and position of the adjacent agents are modified too. The main motivation to introduce this feature is that it makes possible to use agents with adjacent borders, just like territorial divisions of quarters and cities, the modeling of edges of lakes, the division of countryside areas etc., that behave so that when border is changed the adjacent borders are changed too.

Also, a number of perception processes for the agents (based on distance, kind of perceived agent and kind of layer) are pre-defined in the architecture. Moreover, the com-

munication between the agents can occur either in a direct form (by message exchanges), or in an indirect form (using a shared data structure, called a *blackboard*).

A prototype of the architecture, called **G**eo**R**eferenced **S**imulations **P**latform (**GRSP**), was developed. The main objetive of this development was to provide a better way to evaluate all the features and functionalities of this architecture. The main computational tools used for that were: **(i)** *Python programming language*, **(ii)** *PostgreSQL database system*, **(iii)** *PostGIS extension and GEOS library* (used to allow that PostgreSQL uses vectorial geographic data, according to the OGC-SFS standards [Open Geospatial Consortium 2007]). The organization of the prototype system is illustrated in Figure 3.

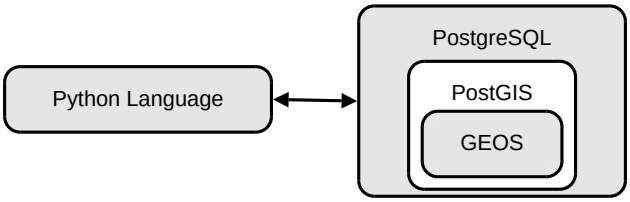


Figure 3. Organization of the main used computational tools.

Figure 4 presents the graphical user interface of the prototype system. A number of social and ecological simulations were developed on this prototype. One of them is presented as a case study, in Section 5.

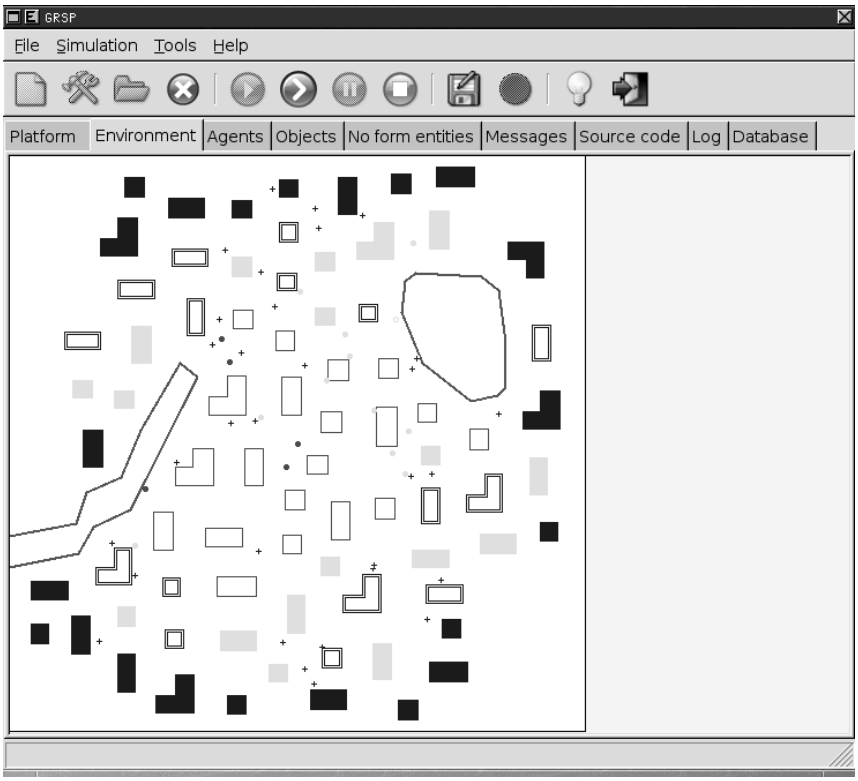


Figure 4. The GRSP graphical interface allows the control of the simulations, the visualization of entities information, their positions, as well as the Python file, log file and GDB information.

5. Study Case

This study case is inspired on the fusion of **some ideas** of two study cases, both presented previously in [Grigoletti 2007]. The first one is about the peripherisation process [Barros and Sobreira 2002], which is characteristic of Third World cities, more specifically of Latin American cities. Peripherisation can be defined as a kind of growth process characterised by the expansion of borders of the city through the formation of peripheral settlements, which are, in most cases, low-income residential areas. The second one is about the influence of policemen actions on crime number, in a certain urban area [Vasconcelos and Furtado 2005]. These two ideias are just used as motivation, in order to create the following social simulation study case.

5.1. The Scene

In this scene, the population is divided in three distinct economic groups according to the pyramidal model of distribution of income in Third World countries. All population have the same locational preferences, that is, they all want to inhabit close to the areas that are served by infrastructure, with nearby commerce, job opportunities, security (police stations), and so on. As in Third World cities these facilities are found mostly close to the high-income residential areas, the preference of location is to be close to a high-income group residential area.

What differentiates the behaviour of the three income groups is the restrictions imposed by their economic power. Thus, (**i**) the high-income group is able to inhabit in any place of its preference; (**ii**) the medium-income group can inhabit everywhere except where the high-income group is already inhabiting and, in turn, (*iii*) the low-income group can inhabit only in the vacant space.

On the other hand, in this scene there are policeman and criminal groups. When a policeman is near a criminal, usually he will capture it and put it in the jail (penitentiary). If an area usually has more criminals than policemen, this area is considered an area with a high crime rate. High-income groups only inhabit areas with a low crime rate. Medium-income groups inhabit areas with low or medium tax of crime. Low-income groups do not mind about that.

The main objective of this study case is to discover the relationship between the amount of crimes in a certain urban area and the economic profile of the group that locates in this area.

5.2. The Model

To create the spatial model was used the urban area of Porto Alegre city (Brazil), represented by four maps, shown in Figure 5 (all the maps were stored as vectorial data in a GDB):

- **urban area map:** this map contains information about the streets and blocks of the studied area;
- **real estate map:** this map contain information about the real estate that can be acquired by people to build/buy/lend the buildings;
- **police stations map:** this map contain information about the police stations;
- **penitentiary map:** this map contain information about the penitentiary.

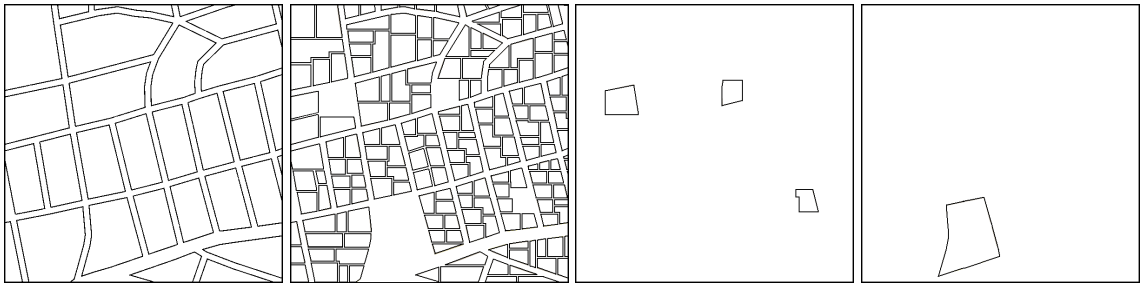


Figure 5. In order, from left to right, the urban area map, the real estate map, the police stations map, the penitentiary map.

In this model were created ten kinds of agents:

- **policeman agents (*mobile point agents*):** these agents are created in some police station. They move around the urban area (avoiding obstacles) looking for criminals. When they find some criminal agent they capture it and put it in the jail (penitentiary);
- **criminal agents (*mobile point agents*):** these agents are created in random places in the urban area. They move around the urban area (avoiding obstacles);
- **high-income, medium-income, low-income agents (*mobile point agents*):** these agents are created in random places in the urban area. They move around (avoiding obstacles) looking for a real estate (to build/buy/rent a building);
- **block agents (*fixed polygon agents*):** these agents contain the real estates of the urban area;
- **real estate agents (*fixed polygon agents*):** these agents represent the real estates of the urban area (that will be bought/etc.);
- **police station agents (*fixed polygon agents*):** theses agents represent the police stations of the urban area;
- **penitentiary agent (*fixed polygon agent*):** this agent represents the penitentiary;
- ***auxiliary agents*:** these agents are used to get information about the simulation.

5.3. Results

The policeman agents are represented by black dots (•) and the criminal agents are represented by white dots (○). The real estates are bought by high-income agents are in light gray, the ones bought by medium-income agents are in dark gray and the ones bought by low-income agents are in black. Figures 6, 7, 8 represent some steps of this simulation. High-income, medium-income, low-income agents were not pictured, to allow for clearer images.

6. Conclusion

The proposed architecture seems to bring the following contributions:

- **to the Social Simulation area:** the architecture of a simulation system (and a prototype) that allows for a detailed spatial modeling of (social) environmental entities, based on Geocomputing data structures, more specifically vectorial data from a GDB;

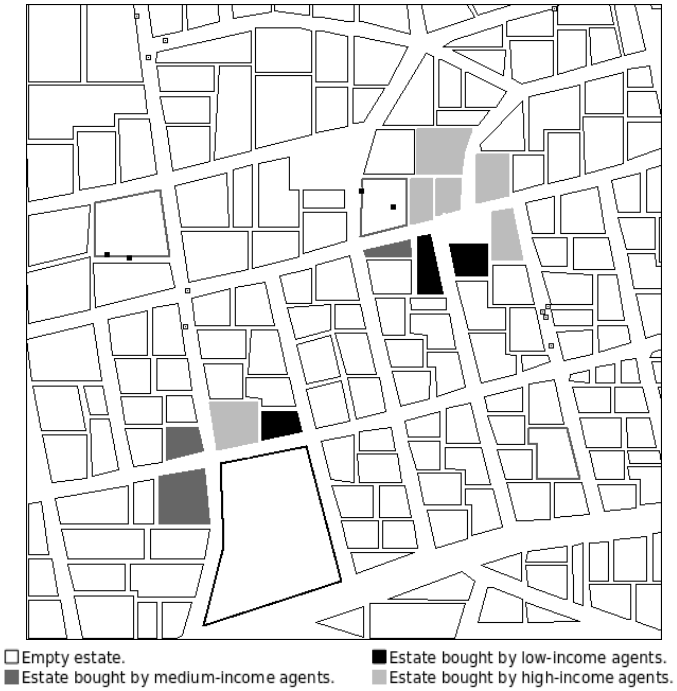


Figure 6. The simulation in its 13th step. The simulation is in the beginning. Few agents have bought a real estate. Few policeman and criminal agents are on the streets.

Figure 7. The simulation in its 80th step. It is possible to see that the highest concentration of real estates bought by high-income agents is near a police station, because they are the agents that have higher chances to succeed in buying such areas.

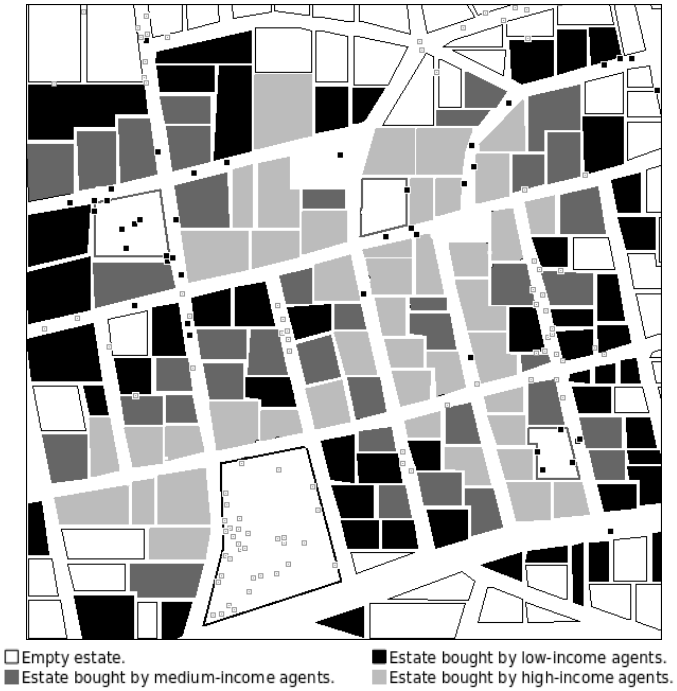


Figure 8. The simulation in its 155th step. It is clear the peripherisation process in this simulation. In the center of the area, high-income real estates (light gray), around it medium-income real estates (dark gray) and near the borders of this area are the low-income real estates (black). There are some center points with low-income real estates because in these areas there are a lot criminals and few policeman agents.

- **to the Georeferenced Simulation area:** the architecture of a simulation system (and a prototype) that allows for the development of detailed modeling of anthropic processes, modeling not just actions of isolated individuals, but also the effects of social interactions on those actions, given that the simulation of direct interactions between agents is possible;
- **to the Geocomputing area:** an abstraction tool to represent dynamic spatio-temporal processes and events, using MAS based simulations.

The computational tool GRSP can be considered another contribution of this work. The study case was presented just to show the use of some developed features. Beyond it, the architecture contribute in the way it joins some needs of others architectures.

The final contribution can be described as: *to provide the structure that are needed to allow for the use of MAS in social and ecological simulations, using geographic models created by GDB and so that MAS can generate spatial data to be used by GDB*. In other words, the work seems to have shown that both Geocomputing and MAS based simulations can benefit from the MAS-GDB coupling.

A comparison of the features found in the architectures studied in Section 3 with the features of the proposed architecture (and created on GRSP software) is presented in Table 1.

Table 1. Comparative table of the MAS based simulations architectures. Legend: $\surd$: *feature implemented*, $\star$: *feature partially implemented*, blank: *feature not found or not implemented*.

Feature	Proposed Architecture	Swarm	Repast	SeSAm	NetLogo	OBEUS
High level movement behaviours	$\surd$					
Use of raster data		$\star$	$\surd$		$\surd$	
Dynamic connection with GDB	$\surd$					
Use of vector data	$\surd$					$\surd$
Easy way to develop	$\star$		$\star$	$\surd$	$\star$	
Communication based on FIPA standard	$\surd$			$\surd$		
Discrete time scheduler	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
Auxiliar entities	$\surd$	$\surd$				
Adjacent entities	$\surd$				$\surd$	

References

Barros, J. and Sobreira, F. (2002). City of Slums: self-organisation across scales. *CASA Working Paper Series*, (55).

Boero, R. (2006). The spatial dimension and social simulations: A review of three books. *Journal of Artificial Societies and Social Simulation*, 9(4).

Bordini, R. H., Costa, A. C. d. R., Hübner, J. F., Moreira, Á. F., Okuyama, F. Y., and Vieira, R. (2005). MAS-SOC: a social simulation platform based on agent-oriented programming. *Journal of Artificial Societies and Social Simulation*, 8(3). Available at http://jasss.soc.surrey.ac.uk/8/3/7.html.

Castelfranchi, C. (1998). Simulating with cognitive agents: The importance of cognitive emergence. In Sichman, J. S., Conte, R., and Gilbert, N., editors, *Proceedings of the First International Workshop on Multi-Agent Systems and Agent-Based Simulation (MABS '98)*, volume 1534 of *Lecture Notes in Computer Science*, pages 26–44, Berlin, Germany. Springer-Verlag.

Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley Longman Publishing Co., Inc., Boston, USA.

Gilbert, N. and Troitzsch, K. G. (1999). *Simulation for the Social Scientist*. Taylor & Francis, Inc., Bristol, PA, USA.

Gonçalves, A. (2003). Multi-Agentes para Simulação em Sistemas de Informação Geográfica. Master's thesis, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Lisboa, Portugal.

Grigoletti, P. S. (2007). Uma Arquitetura Baseada em Sistemas Multiagentes para Simulações em Geoprocessamento. Master's thesis, UFRGS, Porto Alegre, RS, Brasil.

Klügl, F., Herrler, R., and Fehler, M. (2006). SeSAm: implementation of agent-based simulation using visual programming. In *Proceedings of the fifth international joint*

conference on Autonomous agents and multiagent systems (AAMAS '06), pages 1439–1440, New York, NY, USA. ACM Press.

Michel, F., Ferber, J., and Gutknecht, O. (2001). Generic Simulation Tools Based on MAS Organization. In *Proceedings of the 10 European Workshop on Modelling Autonomous Agents in a Multi Agent World (MAMAAW'2001)*.

Minar, N., Burkhart, R., Langton, C., and Askenazi, M. (1996). The swarm simulation system: a toolkit for building multi-agent simulations. Working Paper 96-06-042, Santa Fe Institute, Santa Fe, New Mexico, USA.

Neumann, J. V. (1966). *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA.

North, M. J., Collier, N. T., and Vos, J. R. (2006). Experiences creating three implementations of the repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation*, 16(1):1–25.

Open Geospatial Consortium (2007). OpenGIS Simple Features Specification For SQL Revision 1.1.

Pedrosa, B. and Câmara, G. (2004). *Modelagem Dinâmica e Geoprocessamento*. EMBRAPA, Brasília, DF, Brasil.

Reynolds, C. W. (1999). Steering Behaviors For Autonomous Characters. In *Proceedings of Game Developers Conference*, pages 763–782, San Francisco, CA, USA. Miller Freeman Game Group.

Rocha, L. V., Edelweiss, N., and Iochpe, C. (2001). Geoframe-T: a temporal conceptual framework for data modeling. In *Proceedings of 9th ACM International Symposium on Advances in Geographical Information Systems*, pages 47–52. ACM Press.

Santos, M. (1996). *A Natureza do espaço: técnica e tempo, razão e emoção*. Hucitec, São Paulo, SP, Brasil.

Tisue, S. and Wilensky, U. (2004). Netlogo: A simple environment for modeling complexity. In *Proceedings of the International Conference on Complex Systems (ICCS2004)*, pages 16–21.

Torrens, P. M. and Benenson, I. (2005). Geographic Automata Systems. *International Journal of Geographical Information Science (IJGIS2005)*, 19(4):385–412.

Vasconcelos, E. and Furtado, V. (2005). Um Simulador Tutorial Multi-Agente para Treinamento da Alocação de Equipes Policiais. In *Anais do XVIII Encontro Nacional de Inteligência Artificial*, pages 892–901, Porto Alegre, RS, Brasil.

Worboys, M. F. (1995). *GIS: A Computing Perspective*. Taylor & Francis, London, England.

Polygon Clipping and Polygon Reconstruction

Leonardo Guerreiro Azevedo[1], Ralf Hartmut Güting[2]

[1] Computer Science Department, Graduate School of Engineering, Federal University of Rio de Janeiro, PO Box 68511, ZIP code: 21945-970, Rio de Janeiro, RJ, Brazil

[2]LG Datenbanksysteme für neue Anwendungen, FB Informatik, Fernuniversität Hagen, D-58084 Hagen, Germany

leogazevedo@gmail.com, rhg@fernuni-hagen.de

Abstract. *Polygon clipping is an important operation that computers execute all the time. An algorithm that clips a polygon is rather complex. Each edge of the polygon must be tested against each edge of the clipping window, usually a rectangle. As a result, new edges may be added, and existing edges may be discarded, retained, or divided. Multiple polygons may result from clipping a single polygon. After clipping, we may have a set of segments, which must be handled to generate the clipped polygon. This work proposes two new algorithms: clipping polygon against a rectangle window, and polygon reconstruction from a set of segments. The algorithms were implemented in Secondo, a platform for implementing and experimenting with various kinds of data models.*

1. Introduction

Polygon clipping is one of those humble tasks computers do all the time. It's a basic operation in creating graphic output of all kinds. Polygon clipping is defined by Liang and Barsky (1983) as the process of removing those parts of a polygon that lie outside a clipping window. A polygon clipping algorithm receives a polygon and a clipping window as input. The algorithm must evaluate each edge of the polygon against each edge of the clipping window, usually a rectangle. As a result, new edges may be added, and existing edges may be discarded, retained, or divided. Multiple polygons may result from clipping a single polygon. Two examples of a polygon clipping are presented in Figure 1.

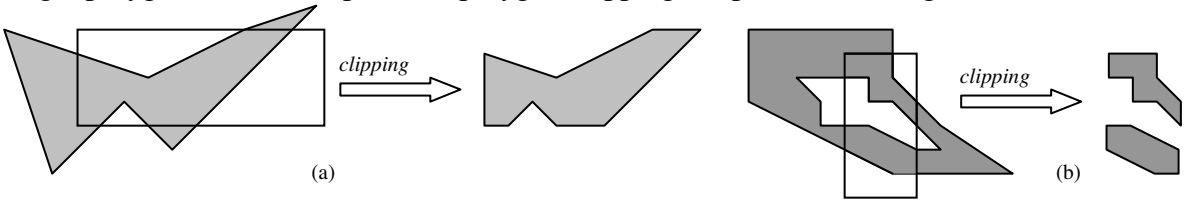


Figure 1 – Examples of polygon clipping by a rectangle window: (a) clipping a polygon that does not have hole; (b) clipping a polygon that has a hole.

There are several well-known polygon clipping algorithms, each having its strengths and weaknesses. The oldest one (from 1974) is called the Sutherland-Hodgman algorithm, as presented by Newman and Sproull (1979). In its basic form, it is relatively

simple. It is also very efficient in two important cases, one being when the polygon is completely inside the boundaries and the other when it's completely outside.

The Liang-Barsky algorithm (1983) is a good deal more complicated, but in certain cases fewer intersections need to be calculated than for Sutherland-Hodgman. Therefore, it may be somewhat faster when many polygon lines intersect with the clipping boundaries. The Weiler and Atherton (1977) algorithm is even more complicated. It allows clipping of a subject polygon by an arbitrarily shaped clip polygon. It is generally applicable only in 2D. Even more ways to clip a polygon exist. None of them is totally perfect. Often, it is possible to feed a weird polygon to an algorithm and retrieve an incorrect result. One of the vertices will disappear, or a ghost vertex will be created. Therefore, the hunt for the perfect clipping algorithm is still open.

Usually polygon's points are stored as an ordered list of points. This structure is employed by many applications, and it is simple to read the polygon and draw it on a Computer Graphics Interface. However, there are some cases where it is not possible to store segments as a simple connected list of points. For instance, when the polygon has holes, it is required an extra information to define which points belong to the external cycle and which ones belong to the internal cycle (or which ones belong to the hole). There are many approaches in the literature to store polygons. For example, Scholl and Voisard (1989) defined general polygons, and Voisard (1992) extended this to general types for points and lines, while Gargano *et al*. (1991) gave only a single type for all kinds of geometric objects; a value is essentially a set of sets of pixels. Güting and Scheneider (1995) proposed the introduction into the DBMS the concept of a *realm,* a finite, user-defined structure that is used as a basis for one or more system data types. Realms are somewhat similar to enumeration types in programming languages. A realm used as a basis for spatial data types is essentially a finite set of points and non-intersecting line segments. All points, lines, and polygons associated with objects *(*spatial attribute values*)* can be defined in terms of points and line segments present in the realm. In fact, spatial attribute values are created only by selecting some realm objects. The polygon structure employed in this work was proposed by Güting *et al*. (1995) and Güting and Schneider (1995), it is presented in section 2.1 (Definition 4).

Polygon reconstruction is the process of reconstructing a polygon from a set of segments those are not in any specific order. For instance, the segments may be stored in a way that a segment that follows another segment does not has a common point. One example of application where this algorithm may be used is polygon clipping. After clipping, the output segments may not be ordered, and the reconstruction algorithm could be used to compute the polygon.

In this work, we propose two new algorithms: an algorithm for polygon clipping by a rectangle window; and an algorithm for polygon reconstruction from a set of segments. The algorithms do not assume any specific orientation of polygon's segments, they do not rely on the computation of parity or wrap numbers of a reference point. Besides, each segment can be processed independent from the others, since all information needed to evaluate one segment is stored within it. The algorithms handle polygons that have multiple boundaries (a polygon that is composed by more than one part) as well as polygons with

holes. The algorithms were implemented in Secondo system (Dieker and Güting, 2000; Güting *et al.*, 2005), and according to the data structure described in the ROSE algebra (Güting, 1993; Güting *et al*, 1995; Güting and Schneider, 1995).

This work is divided in sections, as follows: Section 1 is this introduction; Section 2 presents important definitions for our proposals; Section 3 presents the polygon clipping algorithm; Section 4 presents the polygon reconstruction algorithm; Section 5 presents the implementations we have done in Secondo; finally, in Section 6, we present our conclusions.

2. Preliminary Definitions

In order to present the details of our algorithms proposals it is needed first to define some concepts.

Definition 1) Cycle Direction

The cycle direction defines where is located the enclosed part of a polygon related to its segments. A cycle having the enclosed part on the left is called counterclockwise. On the other hand, when the enclosed part is on the right the cycle is clockwise.

Definition 2) (x,y)-lexicographic Order of two Points

Let $p_1=(x_1,y_1)$ and $p_2=(x_2,y_2)$ be two points in 2-d, the (x,y)-lexicographic order is defined as $p_1 < p_2 \Leftrightarrow x_1 < x_2 \vee (x_1 = x_2 \wedge y_1 < y_2)$ (Güting *et al*, 1995).

Definition 3) Halfsegment

A crucial idea for the representation of the relatively complex polygons values is to regard them as ordered sequences of *halfsegments* (Güting *et al*., 1995; Güting and Schneider, 1995). Let $S = \{(p, q) \mid p, q \in X \times Y, p < q \}$ denote the set of line segments in the $X \times Y$ plane, where p and q are end points. The equality of two segments $s_1 = (p_1, q_1)$ and $s_2 = (p_2, q_2)$ is defined as $s_1 = s_2 \Leftrightarrow (p_1 = p_2 \wedge q_1 = q_2) \vee (p_1 = q_2 \wedge p_2 = q_1)$. Without loss of generality, we normalize S by the assumption that $\forall s \in S : s = (p, q) \Rightarrow p < q$ which enables us to speak of a *left* and a *right end point* of a segment. Let further $H = \{(s, d) \mid s \in S, d \in \{left, right\}\}$ be the set of *halfsegments*. A halfsegment $h = (s, d)$ consists of an segment s and a flag d emphasizing one of the segment's end points which is called the *dominating point* of h. If $d = left$ then the left (smaller) end point of s is the dominating point of h, and h is called *left halfsegment*. Otherwise, the right end point of s is the dominating point of h, and h is called *right halfsegment*. Hence, each segment s is mapped to two halfsegments (s, *left*) and (s, *right*), as presented in Figure 2.

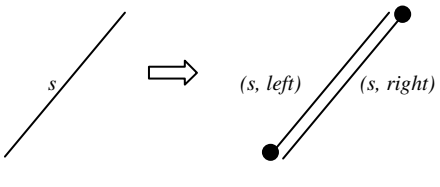


Figure 2 – The mapping of a segment in two halfsegments (*s*, *left*) and (*s*,*right*)

Let *dp* be the function which yields the dominating point of a halfsegment. For two distinct *halfsegments* h_1 and h_2 with a common end point *p*, let α be the enclosed angle such that $0 < \alpha \leq 180°$. Let a predicate *rot* be defined as follows: $rot(h_1, h_2)$ is true iff h_1 can be rotated around *p* through α to overlap h_2 in counter-clockwise direction. We can now define a complete order on *halfsegments* which is basically the (*x*, *y*)-lexicographic order by dominating points. For two *halfsegments* $h_1 = (s_1, d_1)$ and $h_2 = (s_2, d_2)$ it is:

$$h_1 < h_2 \Leftrightarrow dp(h_1) < dp(h_2) \vee (dp(h_1) = dp(h_2) \wedge ((d_1 = right \wedge d_2 = left) \vee (d_1 = d_2 \wedge rot(h_1, h_2)))) \quad (1)$$

Definition 4) Polygon

The polygon structure employed in this work was proposed by Güting *et al*. (1995) and Güting and Schneider (1995). In order to define a polygon, first it is need to define the concepts of cycle, hole and face. A cycle and a hole are sets of connected halfsegments. A face is a pair (*c*, *H*) where *c* is a cycle and $H = \{h_1, \ldots, h_m\}$, where each h_i is a hole (*H* can possibly be empty). Figure 3 presents an example of a polygon composed by three faces (*f* , *f'*, and *f''*). The face *f* is composed by the cycle *c* and the hole *h*. The face *f'* is composed by the cycle *c'* and the holes h_1' and h_2'. Finally, the face *f''* is composed by the cycle *c''* and it has no hole.

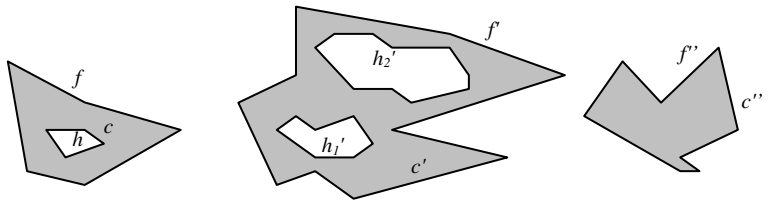
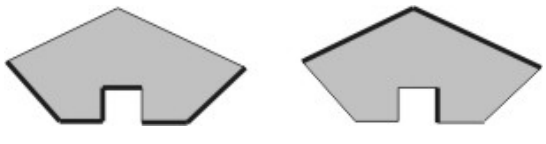


Figure 3 – Example of a polygon

In practice, a polygon is represented essentially as an ordered list (array) of halfsegments. The order used is the one suitable to support plane-sweep algorithms, basically lexicographic order on dominating points, presented in Definition 3. Each halfsegment has a set of attributes storing the cycle (or hole) and the face that it belongs. Besides, each halfsegment has an attribute named *edge number* that specifies the position of the halfsegment in the cycle that it belongs.

Definition 5) InsideAbove Flag of a Halfsegment



a) InsideAbove = true b) InsideAbove = false

Figure 4 – Examples of InsideAbove value.

The *InsideAbove* flag of a segment is true when the area inside the polygon lies above the segment; or, if the segment is a vertical line, it means that the area inside the polygon is on the left of its segment. Figure 4 presents examples of InsideAbove values.

Definition 6) PartnerNumber of a Halfsegment

As presented in Definition 4, a polygon is represented essentially as an ordered list (array) of halfsegments. The PartnerNumber attribute of a halfsegment stores the position of its partner in that array. In other words, the PartnerNumber attribute of a right dominating halfsegment is the position of its corresponding left dominating halfsegment in the array of halfsegments of the polygon, and vice-versa.

Definition 7) Turning Point

The turning point term has been introduced by Liang and Barsky (1983). A turning point is a point at the intersection of two clipping polygon edges that must be added to the clipped polygon in order to keep the connectivity of the original polygon. Figure 5.a presents an example of clipping a polygon by a window. In order to keep the connectivity of the original polygon it is needed to consider the edges corresponding to the turning points highlighted in Figure 5.b. In our work we extended the definition of turning point. We add a flag to the turning point, named *Direction* that stores the direction of the polygon's area on the edge that the turning point lies (*Left*, *Right*, *Up*, or *Down*), as presented in Figure 5.c. The resulting clipped polygon is presented in Figure 5.d.

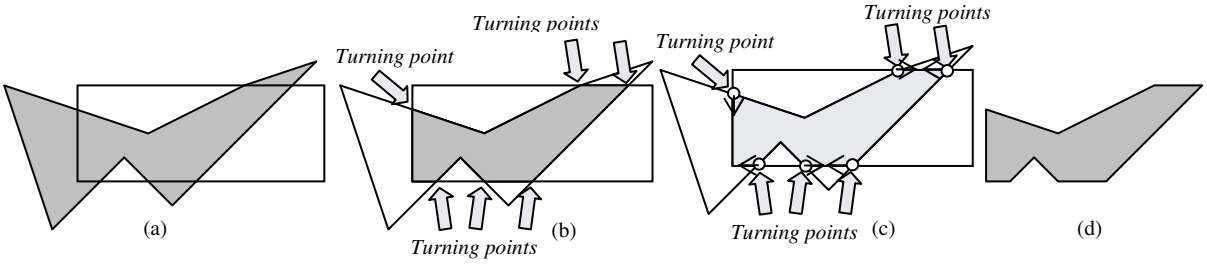


Figure 5 – Example of turning point.

Definition 8) Coverage Number of a Halfsegment

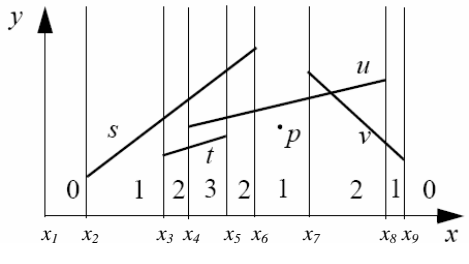


Figure 6 – Example of halfsegments including the coverage numbers of the vertical strips (Güting and Ding, 2004).

The coverage number of a halfsegment was defined by Güting and Ding (2004). Coverage number represents the number of segments that cross each vertical stripe of the plane between two x-coordinates. Figure 6 presents the coverage numbers for a set of halfsegments. In this example, two halfsegments cross the stripe between x_3 and x_4 coordinates. Güting and Ding (2004) present a simple algorithm to compute the coverage number of halfsegments in a single scan through an array of halfsegments.

3 Polygon Clipping

The polygon clipping algorithm has as input a set of halfsegments of a polygon, and produces a set of halfsegments corresponding to the portion of the polygon's halfsegments that are inside the window. In addition, new halfsegments corresponding to the turning points (Definition 7) are also returned. In other words, new halfsegments may be added, and existing halfsegments may be discarded, retained, or divided. Multiple polygons may also result from clipping a single polygon. It is important to emphasize that the polygon clipping algorithm, with few changes, can be used to return the portion of the polygon that is outside the window, instead of the portion that is inside the window. We have implemented both algorithms in Secondo; however, because of the space for this paper, we will present only the polygon clipping algorithm that returns the portion of the polygon inside the window.

In our proposal we use Sutherland-Cohen line clipping algorithm (Newman and Sproull, 1979) to clip the halfsegments against the window. We choose that algorithm because it is probably the most efficient method for trivial acceptance and rejection cases, which are both the most frequently encountered cases in window clipping. This algorithm can be implemented using either integer or floating point arithmetic; thus covering a wider set of applications (Maillot, 1992).

```
algorithm ClippingPolygonSegments
INPUT : HSA=<h1,h2,...hn> (Halfsegment Array)
        w = Rectangle
OUTPUT: cHSA = clipped halfsegments and the halfsegments
               resulting from the evaluation of turning points
cHSA = Ø;
turningPointSets = Ø;
FOR i=1 TO n DO
  IF (hi has left dominating point) THEN
    IF (SutherlandCohenLineClipping(hi, w, clippedhs, intersectionPoint,
        isIntersectionPoint)) THEN
      IF (isIntersectionPoint) THEN
        EvaluateTurningPoint(w, intersectionPoint, turningPointSets, hi);
      ELSE
        cHSA.Add(clippedhs);
        EvaluateTurningPoint(w, clippedH.leftPoint,  turningPointSets, hi);
        EvaluateTurningPoint(w, clippedH.rightPoint, turningPointSets, hi);
      END-IF;
    END-IF;
  END-IF;
END-FOR;
cHSA.Add(getTurningPointHalfSegments(turningPointSets));
```

Figure 7 – Algorithm for clipping polygon's segments by a window.

It is important to emphasize that the clipping of one halfsegment is completely independent of the clipping of any other halfsegment. Thus, it is possible to employ a parallel implementation. Besides, it is not needed to clip both left and right halfsegments. We can clip one type of halfsegment, and discard the other one. The only prerequisite is that the halfsegments must have the *InsideAbove* flag set. This flag is used to handle turning points. The fact that a fast algorithm for trivial rejection and trivial acceptance cases is used has oriented the method to spend most of the computing time evaluating the cases when a segment of the polygon boundaries is not completely rejected and not trivially accepted.

The algorithm for clipping the polygon's halfsegments by a window is presented in Figure 7. The *SutherlandCohenLineClipping* algorithm is used to compute the clipping. It has as input a halfsegment (h_i) and the window (w). The output may be a halfsegment (*clippedhs*) or a point (*intersectionPoint* is the point resulting from the intersection and *isIntersectionPoint* stores this information). If the intersection is a point, it is needed to evaluate this point as a turning points. In the case of a halfsegment intersection, the clipped halfsegment is added to the list (array) of the output halfsegments, and the evaluation of turning points is executed over the end points of the clipped halfsegment. Of course, if the halfsegment is completely inside the window and has no intersection point with the window (trivial acceptance), its end points do not need to be evaluated as turning points. This test is accomplished by the *EvaluateTurningPoint* sub-algorithm. This algorithm is presented in Sub-Section 3.1. The last step of the algorithm to clip halfsegments of a polygon is responsible for creating new halfsegments from the points that were considered as turning points. This algorithm is described in Sub-Section 3.2.

```
algorithm EvaluateTurningPoint
INPUT : w = a rectangle described by the coordinates (x_min, y_min) and (x_max, y_max)
        p = Point
        turningPointSets = for each window egde there is a set recording the
                           turning point of the edge
        h = halfsegment that the point p belongs to
OUTPUT: If the point is evaluated as a turning point it is added to the Turning
        Point Set of the edge
  tp = p;
  IF (p.x = w.x_min) THEN //left edge
    IF (h.insideAbove) THEN
      tp.direction = UP;
    ELSE
      Tp. direction = DOWN;
    END-IF;
    turningPointSets[LEFT].add(tp);
  ELSE //right edge
    IF (h.insideAbove) THEN
      tp. direction = UP;
    ELSE
      tp. direction = DOWN;
    END-IF;
    turningPointSets[RIGHT].add(tp);
  END-IF;
  IF (p.y = w.y_min) THEN //bottom edge
    IF (h.leftPoint > w.y_min) THEN
      tp.direction = GetDirection(p, h.leftPoint,  x_min, y_min, h.insideAbove);
    ELSE
      tp.direction = GetDirection(p, h.rightPoint, x_min, y_min, h.insideAbove);
    END-IF;
    turningPointSets[BOTTOM].add(tp);
  ELSE
    IF (p.y = w.y_max) THEN  //top edge
      IF (h.leftPoint > w.y_min) THEN
        tp.direction = GetDirection(p, h.leftPoint, x_min, y_min, h.insideAbove);
      ELSE
        tp.direction = GetDirection(p, h.rightPoint, x_min, y_min, h.insideAbove);
      END-IF;
      turningPointSets[BOTTOM].add(tp);
    END-IF;
  END-IF;
```

Figure 8 – Algorithm to evaluate turning points.

3.1 Evaluating Turning Points

The turning point evaluation algorithm uses the *InsideAbove* flag (Definition 5) to define how a point must be handled for creating new edges. Only the points that lie on window's

edges are considered as turning points. Figure 8 presents the algorithm used to evaluate turning points.

According to the turning point evaluation algorithm (Figure 8), it is easy to define the direction of the turning points that lie on the vertical window's edges (left and right edges). That is because when the *InsideAbove* flag of the halfsegment that the turning point belongs is true, then the polygon is above the turning point, and the direction is *UP*. Otherwise, if the *InsideAbove* flag has value equal to false, the polygon is under the halfsegment, and the direction of the turning point is *DOWN*. On the other hand, the same reasoning does not apply when handling turning points that lie on the horizontal edges (bottom and top edges). The *InsideAbove* flag's value is not enough to define the turning point direction. An additional test must be executed. This test has just to determine whether the area of the polygon is to the right or to the left of the turning point. Figure 9 presents the algorithm that returns the direction of turning points that lie on top/bottom window's edges.

```
algorithm GetDirection
INPUT: tp = turning point
        p = point of the same half segment that the turning point tp belongs
            and is above tp
       (x, y) = the left coordinate of the vertex of the window edge
        insideAbove = insideAbove flag's value
  IF (insideAbove) THEN
    IF (tp.x > p.x) THEN
      return RIGHT;
    ELSE
      return LEFT;
    END-IF;
  ELSE
    IF (tp.x > p.x) THEN
      return LEFT;
    ELSE
      return RIGHT;
    END-IF;
  END-IF;
```

Figure 9 – Algorithm to compute the direction of turning points that lie on the top/bottom window's edges.

3.2 Creating New Segments from Turning Points

The algorithm that creates new segments from turning points basically sort the turning points accordingly with the *x* and *y*-axis and point's direction. Afterwards, it connects properly the turning points to produce the new segments. The algorithm that creates new halfsegments from turning points is presented in Figure 10.

```
algorithm CreateNewSegments
INPUT: edge = indicates which edge is been handled(LEFT, RIGHT, TOP or
                                                   BOTTOM)
        bPoint,ePoint = end points of the edge
        turningPointSet = a set of the turning points of the edge
        cHSA = set of half segments in which the new half segments will be added

OUTPUT: cHSA with the new half segments
 IF edge == TOP or edge == LEFT THEN
   InsideAbove = false;
 ELSE /*RIGHT or BOTTOM edges*/
   InsideAbove = true;
 END-IF;
 begin = 0;
 end = turningPointSet.size();
 tp = turningPointSet[begin];
 IF (tp.Direction== LEFT or tp.Direction == DOWN) and not tp.Rejected THEN
     cHSA.addHalfSegments(tp,bPoint,InsideAbove);
     DiscardTurningPoints(turningPointSet, tp, ASCENDIN_GORDER, begin);
 END-IF;
 tp = turningPointSet[end];
 IF (tp.Direction== RIGHT or tp.Direction == UP) and not tp.Rejected
    and there is no rejected turning point equals to tp THEN
     cHSA.addHalfSegments(tp,ePoint,InsideAbove);
     DiscardTurningPoints(turningPointSet, DESCENDING_ORDER, end);
 END-IF;
 WHILE (begin<end) DO
  tp1 = GetNotRejectedTurningPoint(turningPointSet, ASCENDING_ORDER, begin);
  IF tp1 == NULL THEN
    return;
  END-IF;
  tp2 = GetNotRejectedTurningPoint(turningPointSet, DESCENDING_ORDER, end);
  IF tp2 == NULL THEN
    return;
  END-IF;
  cHSA.addHalfSegments(tp1,tp2,InsideAbove);
END-WHILE;
```

Figure 10 – Algorithm to create new halfsegments from turning points.

4 Polygon Reconstruction Algorithm

The polygon reconstruction algorithm has as input a set of halfsegments. The halfsegments do not have any information about which polygon's part they belong to (face, cycle or cycle's edge). The algorithm cross the halfsegments, and adjusts properly the face number, cycle number, and edge number (which we named as polygon attributes of a halfsegment), according to the definition of polygon presented in Definition 4. This algorithm can be used to reconstruct any kind of polygon from its halfsegments. For example, it can be used to reconstruct a polygon from a set of halfsegments resulting from clipping a polygon against another polygon. The algorithm is presented in Figure 11. Two sub-algorithms are called by the reconstruction polygon algorithm. They are *ComputeCycle* and *GetFaceNumber*. The algorithm *ComputeCycle* sets the face number, cycle number and edge number of halfsegments that belong to a particular cycle. The algorithm to get face numbers

(*GetFaceNumber*) returns the face number that a halfsegment belongs, or it returns -1, indicating that the halfsegment does not belong to any face that was processed yet.

```
algorithm PolygonReconstruction
INPUT: HSA=<h1,h2,...hn> (Halfsegment Array)
OUTPUT: HSA = each halfsegment has the face, cycle, and edge numbers set

VARIABLES: face = array that stores in position i the last cycle number of the
                  face i
           hsSet = array that stores in the position i if the half segment hi
                   had already the face number, the cycle number and the edge
                   number set. This array is initialized with values false.
  IF HSA is not sorted in halfsegment order THEN
    Sort HSA in halfsegment order;
  END-IF;
  IF the halfsegments of HSA do not have the partner number set THEN
    Set partner number of the halfsegments of HSA;
  END-IF;
  face[0] = 0; /*0 is assigned to the first cycle of the first face */
  lastFaceNumber = 0;
  isFirstHS = true;
  FOR i=1 TO n DO
    IF hi has left dominating point and not hsSet[i] THEN
      IF isFirstHS THEN
        isFirstHS = false;
        hi.faceNumber = 0;
        hi.cycleNumber = 0;
      ELSE
        existingFaceNumber = GetFaceNumber(HSA, hi, hsSet, i);
        IF existingFaceNumber is equal to -1 THEN
          lastFaceNumber++;
          hi.faceNumber = lastFaceNumber;
          hi.cycleNumber = 0;
          /*to store the first cycle number of the face lastFace*/
          face[faceNumber-1]=0;
        ELSE
          hi.faceNumber = existingFaceNumber;
          face[faceNumber]++;
          hi.cycleNumber = face[faceNumber];
        END-IF;
      END-IF;
      hi.edgeNumber = 0;
      ComputeCycle(HSA, hi, hsSet);
    END-IF;
  END-FOR;
```

Figure 11 – Algorithm for polygon reconstruction

5 Implementations in Secondo

Secondo (Diker and Güting, 2000; Güting *et al.*, 2005) is a new generic environment supporting the implementation of database systems for a wide range of data models and query languages. It is developed as a research prototype at the Fernuniversität in Hagen. The implementation of each algebra in Secondo is based on the concept of second-order signature (Güting, 1993) with the first signature describing type constructors and the second signature describing operations on these type constructors. An algebra can be plugged into Secondo with the central part of the Secondo code unchanged. After recompiling Secondo, we can use the newly added algebra.

In Secondo, the operators *windowclippingin* and *windowclippingout* were implemented in SpatialAlgebra (Figure 12). The operators can handle line and polygon (which are named as region), returning the portion inside or outside a rectangle window, respectively. If it receives as input a line and a rectangle window, it returns a clipped line. If it receives as input a region and a rectangle window, a clipped region is returned. The polygon reconstrucion algorithm was implemented as methods in the Region class of Spatial Algebra, and it is used by the clipping algorithms to compute the clipped polygons.

```
Name: windowclippingin
Signature: (line x rect) -> line, (region x rect) --> region
Syntax:   windowclippingin( _, _ )
Meaning:  computes the part of the object that is inside the window.
Example:  query Flaechen feed extend[InWindow: windowclippingin(
                  .geoData, bbox(thecenter))] project[InWindow]
                  filter[not(isempty(.InWindow))] consume
Name: windowclippingout
Signature: (line x rect) -> line, (region x rect) --> region
Syntax:   windowclippingout( _, _ )
Meaning:  computes the part of the object that is outside the window.
Example:  query windowclippingout(trajectory(train7), bbox(thecenter))
```

Figure 12 - Clipping operators implemented in Spatial Algebra in Secondo

6 Conclusion

Polygon clipping is an important operation that computers execute all the time. Often, it is possible to feed a weird polygon to an algorithm and retrieve an incorrect result. One of the vertices may disappear, or a ghost vertex may be created. There are many clipping polygon proposals in the literature. However, none of them is totally perfect. Therefore, the hunt for the perfect clipping algorithm is still open. In this work, we proposed two new algorithms: polygon clipping by a rectangle window; and polygon reconstruction.

Polygon reconstruction is a consequence of polygon clipping. In other words, clipping a polygon produces a set of unordered segments. These segments must be handled in order to adjust segments attributes to produce the polygon. So, we proposed a polygon reconstruction algorithm that may be used in any case where it is needed to compute a polygon from an unordered set of segments. For instance, compute a polygon from a set of segments produced by clipping a polygon by a rectangle window or clipping a polygon by another polygon.

In the case of the polygon clipping algorithm, we proposed and implemented two possibilities, an algorithm to return the portion of the polygon that are inside a rectangle window, and an algorithm to return the portion of the polygon that are outside the window. The algorithms were implemented in Secondo system (Diker and Güting, 2000; Güting *et al.*, 2005), a platform for implementing and experimenting with various kinds of data models. Hence we employed the data structures described in the ROSE algebra (Güting, 1993; Güting *et al*, 1995; Güting and Schneider, 1995). Althoug our algorithm

implementations in Secondo has a good performance, we do not execute an experimental evaluation against other similar algorithms, which we plan to do as future work.

7 References

Dieker, S. and Güting, R. H. (2000) "Plug and Play with Query Algebras: SECONDO A Generic DBMS Development Environment", In: Proceedings of the International Database Engineering and Applications Symposyum (IDEAS), Japan, September.

Gargano, M., Nardelli, E., and Talamo, M. (1991) "Abstract data types for the logical modeling of complex data", Information Systems, 16(5):565-584.

Güting, R. H. (1993) "Second-order signature: A tool for specifying data models, query processing, and optimization", In: ACM SIGMOD Record, vol. 22 , issue 2 (June), pp. 277 - 286.

Güting, R.H. and Schneider, M. (1995) "Realm-Based Spatial Data Types: The ROSE Algebra", VLDB Journal 4, 100-143.

Güting, R.H., and Ding, Z. (2004) "A Simple But Effective Improvement to the Plumbline Algorithm", Information Processing Letters 91 (2004), 251-257.

Güting, R.H., Almeida, V., Ansorge, D., Behr, T., Ding, Z., Höse, T., Hoffmann, F., Spiekermann, M. (2005) "SECONDO: An Extensible DBMS Platform for Research Prototyping and Teaching", In: 21st Intl. Conf. on Data Engineering (ICDE, Tokyo, Japan), 2005, 1115-1116.

Güting, R.H., de Ridder, Th., Schneider, M. (1995) "Implementation of the ROSE Algebra: Efficient Algorithms for Realm-Based Spatial Data Types", In: Proceedings of the 4th International Symposium on Large Spatial Databases, Portland, August.

Liang, Y. and Barsky, B. (1983) "An analysis and algorithm for polygon clipping", Commun. ACM 26, 11 (Nov), 868-877.

Mayllot, P.-G. (1992) "A new, fast method for 2D polygon clipping: analysis and software implementation", In: ACM Transactions on Graphics (TOG), v.11,issue 3, p.276-290, July.

Newman, W. M., and Sproull, R. F. (1979) "Principles of Intemctiue Computer Graphics", McGraw-Hill Book Company.

Scholl, M. and Voisard, A. (1989) "Thematic map modeling", In: Proceedings of the First International Symposium on Large Spatial Databases, Santa Barbara, CA, 1989.

Voisard, A. (1992) "Bases de donn6es g6ographiques: du module de donn6es fi l'interface utilisateur". Ph.D. Thesis, University of Paris-Sud (Centre d'Orsay).

Weiler, K. and Atherton, P. (1977) "Hidden surface removal using polygon area sorting", In: Proceedings of the 4th annual conference on Computer graphics and interactive techniques, San Jose, California, pp. 214 - 222.

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 171-182.

Weighted Overlay, Fuzzy Logic and Neural Networks for Estimating Vegetation Vulnerability within the Ecological Economical Zoning of Minas Gerais, Brazil

Luis M. T. Carvalho[1], Moisés S. Ribeiro[2], Luciano T. de Oliveira[1], Thomaz C. A. Oliveira[1], Julio N. Louzada[3], José R. S. Scolforo[1], Antonio D. Oliveira[1]

[1]Departamento de Ciências Florestais
[2]Departamento de Engenharia
[3]Departamento de Biologia

Universidade Federal de Lavras (UFLA)
Caixa Postal 3.037 – 37.200-000 – Lavras – MG – Brazil

{passarinho,jlouzada,jscolforo,donizete}@ufla.br,
moisessantiago@hotmail.com, oliveiralt@yahoo.com.br,
thomaz@vialavras.com.br

Abstract. *This paper describes the research carried out within the framework of the Ecological Economical Zoning of Minas Gerais (ZEE-MG) to model vegetation vulnerability derived by a number of spatial inference methods. Methods based on weighted overlay, fuzzy logic, and neural networks were compared in terms of visual similarity between maps, the degree of restrictiveness concerning vulnerability, and the easiness of implementation. It was concluded that weighted overlay is the best approach to be used within the ZEE-MG.*

Resumo. *Este artigo descreve os estudos realizados durante os trabalhos do Zoneamento Ecológico Econômico de Minas Gerais para modelar a vulnerabilidade da vegetação derivada de vários métodos de inferência espacial. Métodos baseados em sobreposição ponderada, lógica fuzzy e redes neurais foram comparados em função da similaridade visual entre os mapas, do grau de restrição em relação à vulnerabilidade e da facilidade de operacionalização. Foi concluído que o método de sobreposição ponderada é a melhor alternativa para ser usada no ZEE-MG.*

1. Introduction

The growing concern with natural resources brought a number of mechanisms able to guide human activities and reduce environmental impacts. Environmental studies have been used as the basis for the definition of laws that regulate land use practices. Ecological Economical Zonings are examples of such mechanisms based on the proposal of zones, which are subject to a certain model of use according to degrees of natural vulnerability and social potentiality (MMA, 2005).

Among the various actions to be implemented by the Government of Minas Gerais State within the framework of its Structural Project PE 17, the Action nº P322 (Zoneamento Ecológico-Econômico do Estado de Minas Gerais – ZEE-MG) aims at

supporting policy making related to environmental management by means of a statewide diagnosis of economical, social, ecological and biophysical sustainability.

Indicators of natural vulnerability are used within the ZEE-MG to determine the susceptibility of natural systems to human impacts. Natural vulnerability is defined in the present study as the capacity that a certain land unit has to resist and or to recover from impacts caused by human activities considered normal, i.e. not subject to environmental licensing. It is assumed that if the land unit presents a certain level of vulnerability to human activities considered normal, it will also present the same or a higher level of vulnerability to an activity subject to licensing. The concept takes into account the present condition of biotic and physical aspects of the land unit, where already disturbed areas are less vulnerable than well preserved ones.

Due to the great importance of modeling vulnerable areas for the ZEE-MG, research on alternative methods for integrating the various indicators has been carried out. In this study, the approach implemented to combine the main factors driving vegetation vulnerability will be presented and compared to other methods. Vulnerabilities of flora and fauna form the biotic component of natural vulnerability within the ZEE-MG. Weighted overlay was initially selected by the ZEE-MG team and used for most map combinations in previous studies of the biotic (Carvalho and Louzada, 2007) and abiotic components. As a further development, our main objective in the present paper was to investigate alternative methods of spatial inference, viz. fuzzy logic and neural networks, to generate maps of vegetation vulnerability for the State of Minas Gerais, Brazil, and evaluate their suitability to be used instead of weighted overlay.

2. Study site and data sets

The study area comprises the whole State of Minas Gerais. The data compiled and included in the ZEE-MG were structured in a GIS using the raster data model (Burrough & Mcdonell, 1998). Spatial resolution, determined by the pixel size, was defined for the ZEE-MG as 270x270 m, representing about 7 ha on the ground.

A set of specific indicators derived from variables that represent environmental aspects might determine different levels of vegetation vulnerability. In a higher hierarchical level, vegetation vulnerability of a certain region is one of the factors determining the natural vulnerability of this area. The variables used to derive indicators of vegetation vulnerability comprised a 30x30 m resolution land cover map for the State (Scolforo & Carvalho, 2006) and a map relative to areas of special ecological interest devised by a number of vegetation specialist from Minas Gerais (Drummond et al., 2005).

The following indicators of vegetation vulnerability were used in the present study: regional relevance of physiognomies, conservation degree, spatial heterogeneity of physiognomies and conservation priority.

2.1. Indicators 1 to 9: Regional relevance of physiognomies

Regional relevance of physiognomies (Figure 1) was defined for a pixel as the ratio between the actual area of a certain physiognomy (e.g. forest) in that pixel and the total

area of the same physiognomy in a certain region. The physiognomy area for a ZEE-MG pixel (270x270 m) is determined by simply counting the number of 30x30 m pixels within the ZEE-MG pixel. The regions considered in this study correspond to the administrative boundaries of the Regional Councils of Environment (COPAM). In this case, high values of regional relevance were obtained for areas of vegetation remnants in regions with very little vegetation representing that physiognomy.

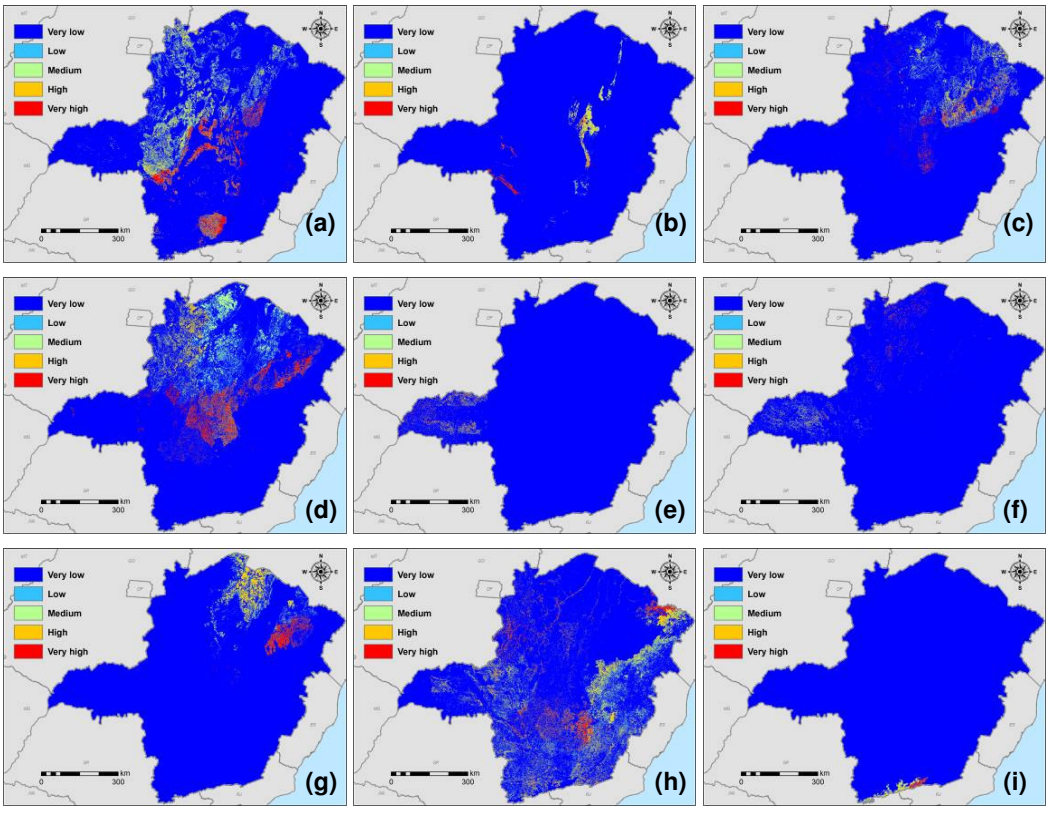


Figura 1. Regional relevance of (a) grass land, (b) rocky grass land, (c) open savanna, (d) savanna *stricto sensu*, (e) savanna woodland, (f) savanna palm land, (g) deciduous forest, (h) semi deciduous forest, and (i) evergreen forest.

2.2. Indicator 10: Degree of conservation

Following the same idea of the previous indicator, the degree of conservation (Figure 2a) was determined by counting the number of 30x30 m pixels covered by natural vegetation within each 270x270 m pixel of the ZEE-MG. Hence, well preserved areas are considered highly vulnerable to human impacts.

2.3. Indicator 11: Spatial heterogeneity

Again, this indicator (Figure 2b) was calculated by counting the number of different physiognomies that occur within each ZEE-MG pixel. This indicator captures transition areas between different physiognomies, which are thought to be highly important and, consequently, vulnerable as well.

2.4. Indicator 12: Conservation priorities

The last indicator of vegetation vulnerability (Figure 2c) was obtained by reclassifying and rasterizing vector maps relative to priority areas defined by expert knowledge. In the work of Drummond et al. (2005), vegetation specialists from all over Minas Gerais have indicated areas of relevant species endemism, the occurrence of rare or threatened species, areas of high biodiversity, ecological corridors, unique combinations of biotic and abiotic associations, and areas that lack floristic studies.

The reclassification scheme is presented in Table 1. Conservation priority classes were adjusted to fit the legend used within all outputs of the ZEE-MG project.

Table 1. Class correspondence between classification systems.

Conservation priority classes (Drummond et al., 2005)	**ZEE-MG vulnerability classes**
None	Very low
Corridor	Low
Potential	Medium
High	High
Very high	Very high
Extreme	Very high
Special	Very high

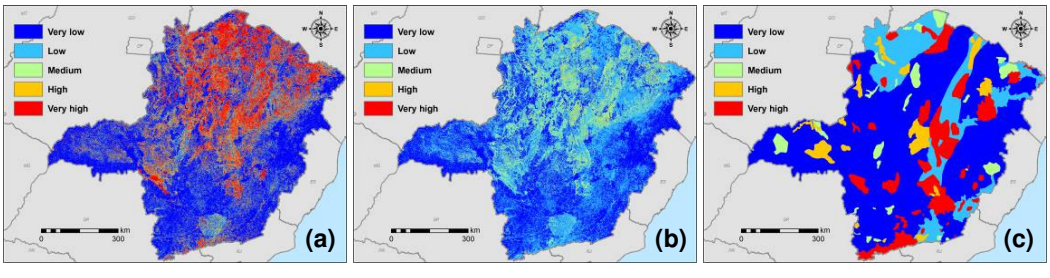


Figura 2. (a) Degree of conservation, (b) spatial heterogeneity, and (c) conservation priorities.

3. Methodology

All input data sets as well as the model outputs were registered to the Albers Conic Equal Area Projection (datum SAD-69) and resampled to 270x270 m using a nearest neighbor algorithm when necessary.

Spatial inference to come up with final maps of vegetation vulnerability was carried out by using weighted overlay, fuzzy logic, and neural networks, as detailed in the next sections. Vulnerability represented by the models outputs were classified as (1) Very low, (2) Low, (3) Medium, (4) High, and (5) Very high.

3.1. Weighted Overlay

Models based on overlay operations using weights allow a more flexible map combination when compared to operations based on Boolean logic. Modeling via Boolean logic, which has been for long used to analyze physical variables, involves the combination of binary maps generated by the application of operators (AND for

intersection, OR for union, and NOT for inclusion) that indicate two distinct conditions. Nevertheless, due to its crisp classification nature, the involved variables are considered to have the same importance to the problem at hand and the combination output will be described by a simple binary number, 1, vulnerable, or 0, not vulnerable, when vulnerability is the characteristic to be modeled.

Weighted overlay was used in this study because it is a simple and straightforward technique for spatial inference using multiple multi-class maps (ESRI, 2002). Furthermore, weights in the model represent the relative importance of each variable included in the analysis, as well as the relative importance of the classes of each variable according to a given objective. Meirelles et al. (2007a) state that the use of weighted overlay allows the inclusion of expert knowledge and the adjustment of intrinsic characteristics of each variable in the model. However, it must be highlighted here that the weights are considered constant for each variable and or class over the whole study area, which is not the case for most real world phenomena.

Following the framework of the ZEE-MG, all 12 indicators were input to the weighted overlay model with weights defined according to Table 2.

Table 2. Weights defined for each indicator of vegetation vulnerability.

Indicator	Indicator weight	Class	Class weight
Regional relevance	8	Very low	1
		Low	6
		Medium	10
		High	12
		Very high	12
Degree of conservation	12	Very low	1
		Low	6
		Medium	10
		High	12
		Very high	12
Spatial heterogeneity	4	Very low	1
		Low	6
		Medium	10
		High	12
		Very high	12
Conservation priority	12	Very low	1
		Low	2
		Medium	6
		High	12
		Very high	12

3.2. Fuzzy Logic

Methods based on fuzzy logic are very similar to weighted overlay, with the advantage that the combination rules are more flexible, thus promoting an improvement in the linear nature of the latter technique.

Instead of classifying geographical information in classes defined by exact boundaries and thereafter attributing weights to each class, one might reproduce the input data in a continuous scale using the assumption of continuous membership values. For each x value of the indicator variable, a $\mu(x)$ value is generated by a membership function, the so called fuzzyfication process, where the pair $(x, \mu(x))$ is known as the

fuzzy set. Membership functions are not necessarily linear; they can assume any analytical or arbitrary form that is appropriate to the problem under consideration.

Semantic models can be represented through various types of membership functions. The models of fuzzy classification used for environmental data are extensions of the functions that were generated by Kandel (1986). In the present paper, the symmetric fuzzy models were defined as followed:

$$FP_x = \mu A(x) = 1/\left\{1 + d\left(x - b\right)^2\right\} \qquad \text{for } 0 \le x \le N$$

Where,
FP_x = Fuzzy membership function;

$\mu A(x)$ = Fuzzy membership level;

d = Parameter that is responsible for the function type;

b = Parameter that defines the domain of X according to the central concept.

After fuzzyfication of each variable through the membership functions, fuzzy operators were applied in order to combine the different layers. These operators allow distinct ways of manipulating simultaneously a set of layers containing fuzzy values through a process of fuzzy overlay.

Operator Fuzzy Gamma:

$$\mu_{combinação} = \left(SAF\right)^{y} * \left(PAF\right)^{1-y}$$

$$SAF = \mu_{combinação} = 1 - \Pi\left(1 - \mu_i\right)$$

$$PAF = \mu_{combinação} = \Pi\mu_i$$

Where,
SAF = Fuzzy Algebraic Sum;

PAF = Fuzzy algebraic product;

$\mu_{combination}$ = Membership value of the themes combination

μ_I = Fuzzy membership value for the map that stands in that order;

$\prod$ = Theme maps (indicators) considered in the analyses of the phenomenon.

y = chosen parameter in the interval [0,1]

When y equals 1, the resulting map is identical to the result of fuzzy algebraic sum, and when y equals 0, the resulting map is identical to the result of fuzzy algebraic product. By varying the value of y, it is possible to obtain output values that assure certain flexibility between the tendency of growth of the fuzzy algebraic sum and the tendency of decrease of the fuzzy algebraic product. According to Meirelles et al. (2007b), modeling via fuzzy algebraic sum considers that if two evidences (e.g.

indicators of vulnerability) point to the same researched hypothesis, one will reinforce the other, and the resulting combination will have more support than the input evidences. Hence, the result of this operation is always a value greater or equal to the largest input value of fuzzy membership. On the other hand, the combination using the fuzzy algebraic product produces results that are always smaller or equal to the smallest input fuzzy membership value. As the goal of the present study is the maximization of vegetation vulnerability, two values of the parameter *y* were used for the operator fuzzy gamma: $y = 0.75$ and $y = 0.95$.

Operator Fuzzy Convex Sum:

If $A_1,.....,A_k$ are subsets of X, and $w_1,.....,w_k$ are non negative weights then the convex combination of $A_1,....,A_k$ is:

$$\mu_A \Delta \sum w_j \mu_{Aj}$$

Where, $\sum w_j = 1$

This procedure of defining weights is very similar to modeling via weighted overlay, but the class values are continuous in the interval [0,1]. The convex sum is generally used when the effects of the indicators are not equal. In the present study, weights were defined for the operator convex sum according to Table 2 as well.

3.3. Neural Networks

Neural networks are problem solving algorithms of the machine learning field. They use methods and techniques inspired on historical facts and models of biological neurons and networks. These biologically inspired models are extremely efficient when the pattern of classification is not a simple and trivial one (Barreto, 2002). Neural networks have shown to be helpful in the solution of practical problems as well as capable of classifying highly complex data (Kanellopolous et al., 1997).

Self Organizing Maps

For the present work three different types of networks were used. Unsupervised neural networks called Self Organizing Maps (SOM) (Kohonen, 1990), were used to create vulnerability maps. SOM was implemented in two configurations: coupled and uncoupled with a *k*-means clustering algorithm. Unsupervised learning does not need input samples for pattern recognition. This perfectly fits the scope of this work since there was no collection of training data representing vulnerability classes.

The parameters presented in Table 3 were chosen after a number of trials and following empirical knowledge.

Table 3. SOM neural network parameters.

Parameter	SOM (without *k*-means)	SOM (with *k*-means)
Input layer neurons	12	12
Output layer neurons	9	36
Initial neighborhood radius	5.24	9.49
Minimum learning rate	0.5	0.5
Maximum learning rate	1	1
Iterations	874,080	628,992
Quantization Error	0.0241	0.0187

Fuzzy ArtMap

For Mather (1999) the use of soft classification paradigms with neural networks is adequate when we want to avoid errors of classification due to ambiguity of the generated classes. The third type of network considered in the present study was based on the ART (Adaptative Resonance Theory) (Carpenter et al., 1991), which exhibits a high degree of stability in order to preserve significant past learning, but remain enough adaptable to incorporate new information whenever it might appear (Carpenter, 1989). Fuzzy ArtMap is a clustering algorithm that operates on vectors with fuzzy analog input patterns (real numbers between 0 and 1) and incorporates an incremental learning approach which allows it to learn continuously without forgetting previous learned patterns. The Fuzzy ArtMap parameters used to estimate vegetation vulnerability in this study are summarized in Table 4. Again, the parameters were defined after a number of tests and according to previous expert knowledge.

Table 4. Fuzzy ArtMap neural network parameters.

Parameter	Fuzzy ArtMap
F1 layer neurons	24
F2 layer neurons	6
Choice parameter	0.01
Learning rate	1
Vigilance parameter	0.95
Iterations	48,923

4. Results and Discussions

The process of combining indicators of vegetation vulnerability generated a number of raster maps according to different inference models. For comparison purposes, the results for an example area within the State of Minas Gerais are illustrated in Figures 3 and 4.

4.1. Weighted overlay x Fuzzy logic

In Figure 3, models based on fuzzy logic are compared to the model generated via weighted overlay.

When comparing the maps of vegetation vulnerability generated by weighted overlay (Figure 3a) and the operator Fuzzy Gamma (y=0,95) (Figure 3b), it is observed that vulnerability classes are different for some regions. Some areas classified as medium and high vulnerability by the operator Fuzzy Gamma (y=0,95) were classified as low and very high vulnerability when using weighted overlay. Hence, the former seems to be less restrictive then the latter. This pattern might be due to the fact that weighted overlay uses a constant weight for the entire map extent. By decreasing the value of y, the output also shows a decrease in the values of fuzzy memberships probably due to the tendency of minimization that is characteristic when y approaches zero. Among all other fuzzy operators, the Fuzzy Gamma (y=0,75) is the most similar to the output generated by weighted overlay.

On the other hand, the Fuzzy Algebraic Sum (Figure 3c), maximized membership values when compared to the other operators used in the analysis. More

classes of very high vulnerability were obtained in this case showing that it is the most restrictive model.

Fuzzy Convex Sum (Figure 3d) showed results that are very close to the ones obtained with Fuzzy Gamma (y=0,95), except for some areas that were classified as having medium vulnerability by Fuzzy convex sum and low vulnerability by Fuzzy Gamma. The other classes remained constant between the two operators. Weighting performed by Fuzzy Convex Sum probably caused this difference. This evidence shows the flexibility of this operator while using weights during the classification process.

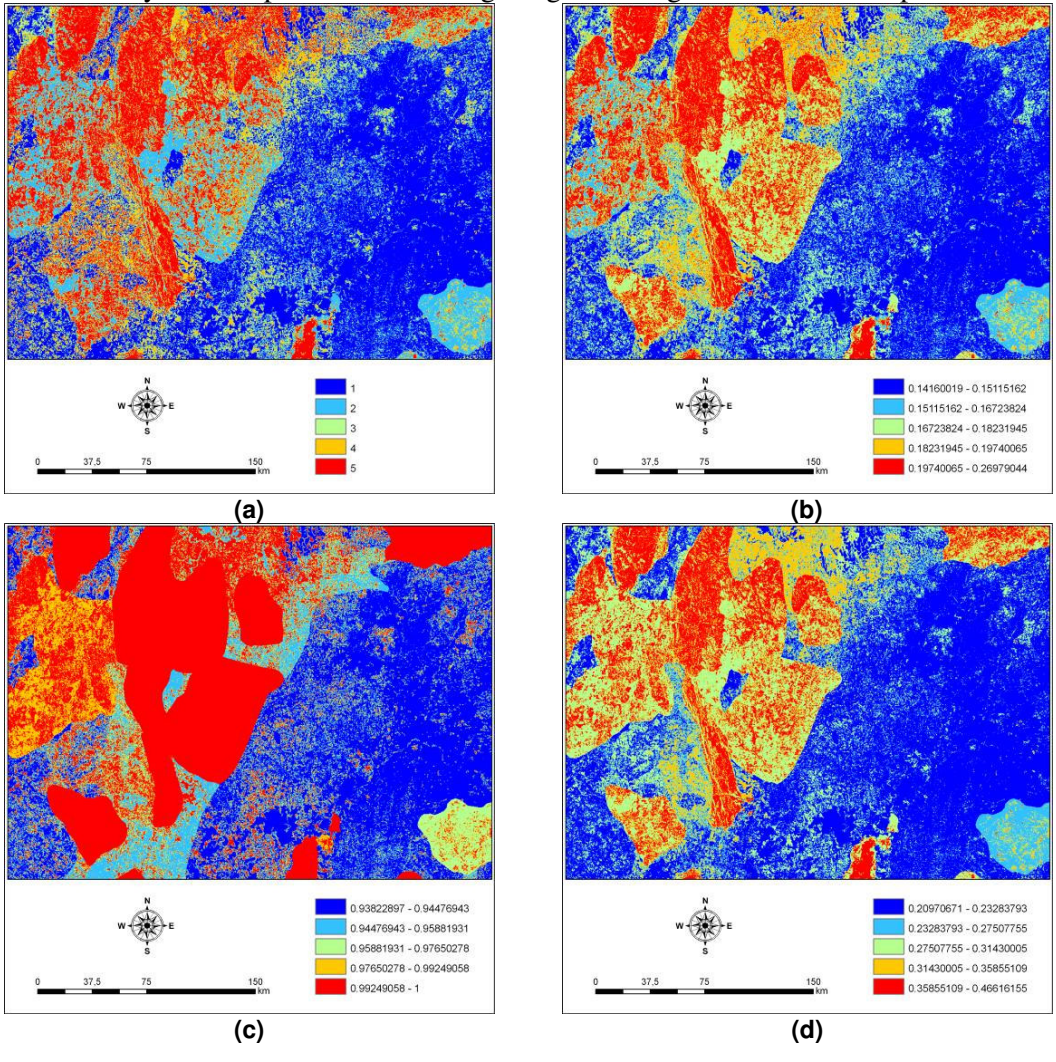


Figura 3. Vegetation vulnerability maps generated by the following models: (a) Weighted overlay, (b) Fuzzy Gamma (y = 0,95), (c) Fuzzy Algebraic Sum, and (d) Fuzzy Convex Sum.

4.2. Weighted overlay x Neural networks

In Figure 4, models based on neural networks are compared to the reference model generated via weighted overlay (Figure 4a).

The map produced using Fuzzy ArtMap (Figure 4b) showed patterns similar to the results of Fuzzy Algebraic Sum maximizing most of the membership values and leading to a more restrictive scenario characterized by homogeneous zones. These patterns show a strong influence of the indicator related to conservation priorities. On the other hand, the map produced using SOM without *k*-means was not influenced by this indicator at all. It is noticed that the degree of conservation was the most important factor driving vegetation vulnerability when this model was implemented. Vulnerability was either very low or very high, with a few areas showing intermediate values.

Finally, the map produced using SOM with *k*-means clustering (Figure 4c) was similar to the reference map produced using weighted overlay (Figure 4a). It showed a better balance while representing the influence of each indicator. SOM with *k*-means also presented smoother transitions between classes. Nevertheless, neural networks have been criticized because of its "black box" nature. In fact, it is difficult for a non-expert to understand and set the network parameters, leading to an arbitrary result of vulnerability classes and less control of the indicators influences.

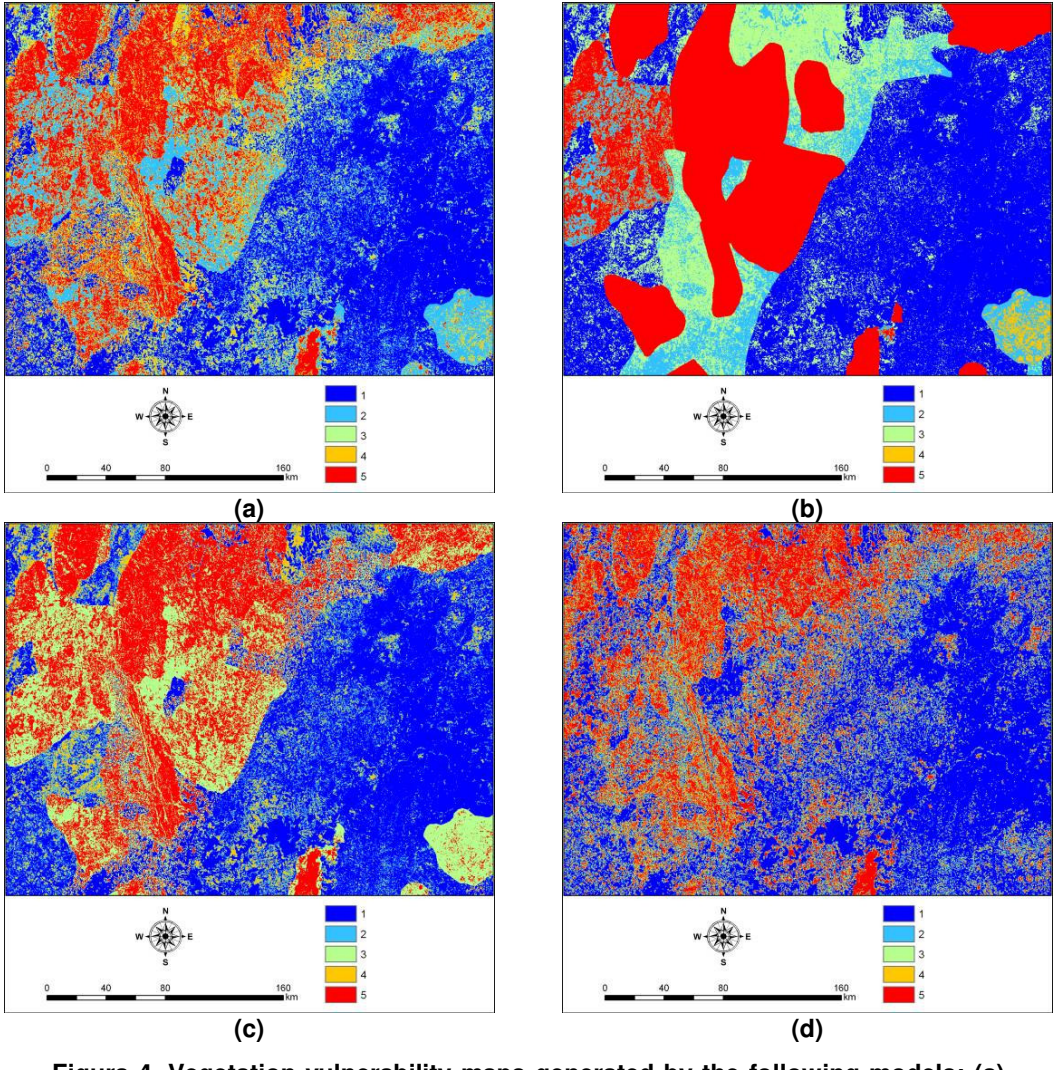


Figura 4. Vegetation vulnerability maps generated by the following models: (a) Weighted overlay, (b) Fuzzy Artmap, (c) SOM *k*-means, e (d) SOM.

4.3. Comparison criteria

As no reference data was available for estimating the accuracy of vegetation vulnerability maps, comparison criteria were empirically defined based on the visual similarity between maps, the degree of restrictiveness concerning vulnerability, and the easiness of implementation. Evaluation of these criteria led to the choice of weighted overlay as the most robust inference method considered in the present study, and thus kept within the framework of the ZEE-MG. SOM *k*-means, Fuzzy Gamma, and Fuzzy Convex Sum, showed similar results to the weighted overlay procedure, approaching patterns of vulnerability thought to be closer to reality according to the knowledge of experts involved in the project. Even so, weighted overlay was chosen because it is easier to be implemented with complete control over the involved indicators.

In Figure 5, the vegetation vulnerability map produced using weighted overlay is presented for the entire State of Minas Gerais.

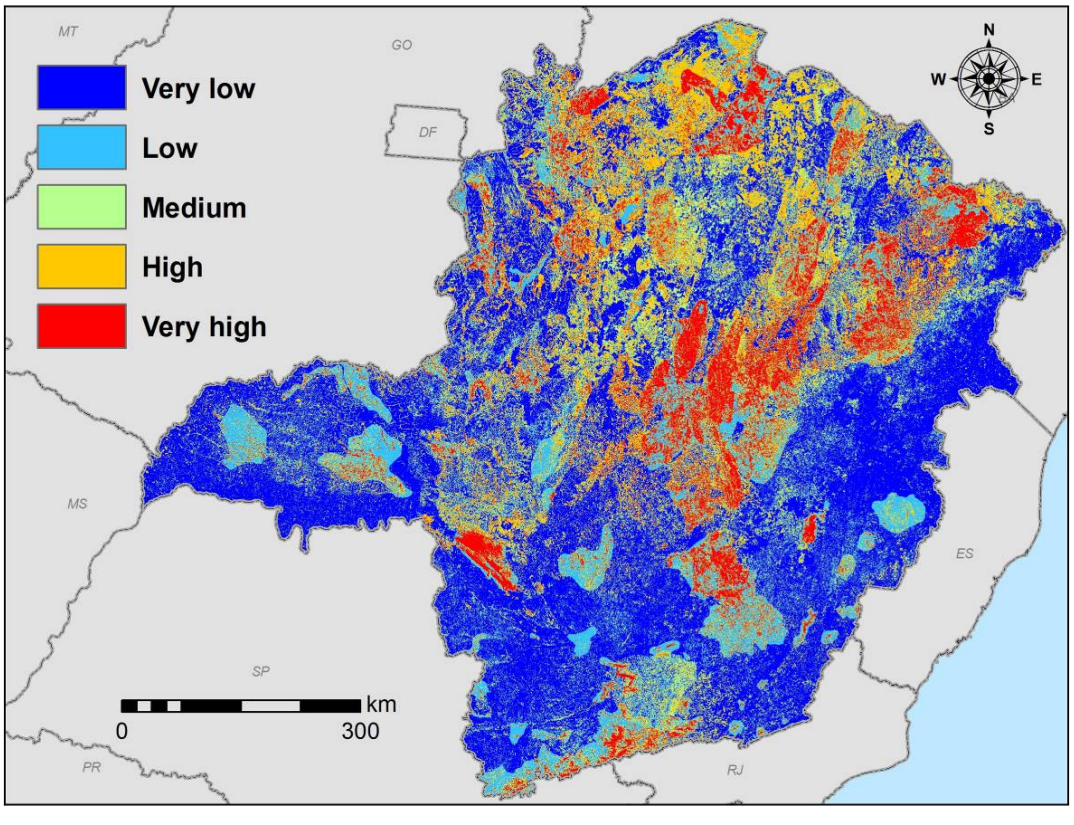


Figura 5. Vegetation vulnerability map obtained via weighted overlay and chosen according to the defined comparison criteria.

5. Conclusions

In this paper, a number of inference methods were implemented to produce maps of vegetation vulnerability. Methods based on fuzzy logic and neural networks were compared to weighted overlay, which was considered to be the reference map because it was already implemented during previous phases of the ZEE-MG.

We concluded that weighted overlay will not be replaced by any of the other tested methods. They are less intuitive, dependent on a number of arbitrary parameters, demand more computational power, and do not provide significant improvements when compared to the map produced using weighted overlay.

Nevertheless, fuzzy logic seems to be a promising approach and further research will be carried out in order to test different fuzzification methods, as well as different fuzzy operators to produce maps of vegetation vulnerability and to combine them with other biotic and physical components of natural vulnerability.

Neural networks provided interesting results, but due to the difficulties in setting the network parameters the method will be disregarded within the ZEE-MG.

Finally, a framework to collect field data concerning vegetation vulnerability classes will be developed to provide a robust base to carry out vulnerability map comparisons.

References

Burrough, P. A.; Mcdonell, R.A. (1998). "Principles of geographical information systems." Oxford: University Press, 333 p.

ESRI. (2002). "Using ArcGIS Spatial Analyst." Redlands: ESRI Press. 232 p.

Scolforo, J. R. S; Carvalho, L. M. T. (2006). "Mapeamento e inventário da flora nativa e dos reflorestamentos de Minas Gerais." Lavras: Editora UFLA. 288 p.

Kandel, A. (1986). "Fuzzy mathematical techniques with applications." Massachusetts: Addison-Wesley. 148p.

Meirelles, M. S. P.; Moreira, F. R.; Camara, G. (2007a). "Técnicas de inferência espacial", Geomática: modelos e aplicações ambientais, Meirelles, M. S. P.; Camara, G.; Almeida, C. M. Brasília: EMBRAPA. p.107-189.

Meirelles, M. S. P.; Moreira, F. R.; Camara, G.; Coelho Netto, A. L.; Carneiro, T. A. de À. (2007b). "Métodos de inferência geográfica: aplicação no planejamento regional na avaliação ambiental e na pesquisa mineral" Geomática: modelos e aplicações ambientais, Meirelles, M. S. P.; Camara, G.; Almeida, C. M. Brasília: EMBRAPA. p. 283-357.

MMA. (2005). "Diretrizes Metodológicas para o Zoneamento Ecológico-Econômico do Brasil." Brasília: Ministério do Meio Ambiente. 104p.

Drummond, G. M.; Martins, C. S.; Machado, A. B. M.; Sebaio, F. A.; Antonini, Y. (2005). "Biodiversidade em Minas Gerais: um atlas para sua conservação." Belo Horizonte: Fundação Biodiversitas. 222p.

Carvalho, L. M. T.; Louzada, J. N. "Zoneamento Ecológico-Econômico do estado de Minas Gerais: abordagem metodológica para caracterização do componente flora." Anais do XIII Simpósio Brasileiro de Sensoriamento Remoto (SBSR). São José dos Campos: INPE, 2007. p. 3789-3796.

Kohonen, T. (2001). "The Self-Organizing Map." Springer Series in Information Sciences.

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 183-194.

An efficient algorithm to compute the viewshed on DEM terrains stored in the external memory

Mirella A. Magalhães[1], Salles V. G. Magalhães[1], Marcus V. A. Andrade[1], Jugurta Lisboa Filho[1]

[1]Departamento de Informática – Universidade Federal de Viçosa (UFV)
CEP 36.570.000 – Viçosa – MG – Brazil

{mirella, smagalhaes, marcus, jugurta}@dpi.ufv.br

Abstract. *Nowadays, there is a huge volume of data about terrains available and generally, these data do not fit in the internal memory. So, many GIS applications require efficient algorithms to manipulate the data externally. One of these applications is the viewshed computation that consists in obtain the visible points from a given point p. In this paper, we present an efficient algorithm to compute the viewshed on terrains stored in the external memory. The algorithm complexity is $O(scan(N))$ where N is the number of points in a DEM and $scan(N)$ is the minimum number of I/O operations required to read N contiguous items stored in the external memory. Also, as shown in the results, our algorithm outperforms the known algorithms described in the literature.*

1. Introduction

Terrain modeling is an important area in GIS applications and in general, a terrain can be represented by a *triangulated irregular network (TIN)* or a *Raster Digital Elevation Model (DEM)* [Li et al. 2005, Felgueiras 2001]. A TIN is a vector based representation of a surface made up of irregularly distributed nodes with three dimensional coordinates (x, y, and z) that are connected and arranged in a network of non overlapping triangles. Thus, the surface is approximated by triangle patches and the elevation (the z coordinate) of any point can be interpolated from the vertices of the planar triangle containing the (x, y) coordinates of the point. A DEM is a digital file or a matrix consisting of terrain elevations for ground positions at regularly spaced horizontal intervals.

There is no consensus about which of these representations is the best and there are many discussion about this theme [Kumler 1994, Floriani et al. 1999, Felgueiras 2001]. Anyway, we can say that DEM requires a simple data structure, it is easier to analyze and has high accuracy at high resolution, but it requires high memory space and it is time-consuming processing. On the other hand, TIN has a restricted accuracy, requires more complex algorithms, but it is less memory-consuming and more time-efficient processing. Given its simplicity, in this work, we consider a terrain represented by a DEM.

The recent technological advances in data collection (such as LiDAR) have produced a huge volume of data about Earth's surface [USGS 2007]. For example, a $100km \times 100km$ terrain sampled at $1m$ resolution results in 10^{10} points. And, regardless of the representation used, most of the computational systems can not store/process this huge volume of data internally and thus, they need to be processed in the external memory, generally disks. Since the required time to access and transfer data from and to the exter-

nal memory is much longer than time for internal processing, the algorithms performing external processing must minimize data access [Arge 1997, Goodrich et al. 1993].

More specifically, these algorithms should be designed and analyzed considering a computational model that evaluates the algorithm complexity based on data transfer operations instead of cpu processing operations. One of these models was proposed by Aggarwal and Vitter [Aggarwal and Vitter 1988] where the algorithm complexity is measured considering the number of I/O (input/output) operations executed.

An important GIS problem related to terrain modeling is the computation of all points that can be viewed by a given point (the observer); the region formed by the visible points is named *viewshed* [Floriani and Magillo 2003, Franklin and Ray 1994]. This problem has been widely studied in many applications such as to determine the minimum number of cellular phone towers to cover a region [Ben-Moshe et al. 2007a, Camp et al. 1995, Bespamyatnikh et al. 2001], to optimize the number and position of guards to cover a region [Franklin and Vogt 2006, Eidenbenz 2002], to analyze the influences on property prices in an urban environment [Lake et al. 1998], to optimize path planning on DEM [Lee and Stucky 1998], etc.

In this work, we present an I/O efficient algorithm to compute the viewshed of a point on terrains represented by DEM stored in the external memory. Our algorithm is an adaptation of Franklin and Ray's method [Franklin and Ray 1994, Franklin 2002] to allow an efficient manipulation of huge terrains (5GB or more). The large number of disk accesses is optimized using the library STXXL [Dementiev et al. 2005]. Comparing our algorithm with the original one (adapted to perform external processing) and with the algorithm proposed by Haverkort et al. [Haverkort et al. 2007], the tests showed that our algorithm is about 3.5 times faster than both algorithms and also, it is much simpler and easier to implement than the latter.

The paper is organized as follow: the section 2 gives a brief description about works on viewshed computation and also, on I/O-efficient algorithms for general problems and for viewshed computation too; in the section 3, the viewshed concepts are formally presented; in section 4, the I/O-efficient computational model is shortly described; in section 5, the algorithm is described in details and its complexity is presented in section 6; the tests results are given in section 7 and the conclusions in section 8.

2. Related Works

The visibility on terrains has been widely studied in many different areas. For example, Stewart [Stewart 1998] shows how the viewshed can be efficiently computed for every point of a DEM and his interest involves radio transmission towers positioning. Kreveld [van Kreveld 1996] proposes a sweep-line approach to compute viewshed in $O(nlogn)$ time on a $\sqrt{n} \times \sqrt{n}$ grid. In [Franklin 2002, Franklin and Ray 1994], Franklin and Ray describe experimental studies for fast implementations of visibility computation and present several programs that explore various trade-offs between speed and accuracy. Kim, Rana and Wise in [Young-Hoom et al. 2004] analyze two strategies to use viewshed for optimization problems. Ben-Moshe et al. [Ben-Moshe et al. 2004b, Ben-Moshe et al. 2004a, Ben-Moshe et al. 2007b] have worked on visibility for terrain simplification and for facilities positioning. For a survey on visibility algorithms, see [Floriani and Magillo 2003].

Some problems related to external memory processing are discussed by Aggarwal and Vitter [Aggarwal and Vitter 1988]. They proposed a computational model to evaluate the algorithm complexity considering the number of input/output operations executed. In [Goodrich et al. 1993], Goodrich et al. presented some variants for the sweep plane paradigm considering external processing and Arge et al. [Arge et al. 1995] described a solution for the external processing of line segments in the context of GIS. This technique was also used to solve problems in hydrology such as the computation of the water flow and watershed [Arge et al. 2003] on huge terrains.

Recently, Haverkort et al. [Haverkort et al. 2007] presented an adaption of the Kreveld's method to compute the viewshed on terrains stored in the external memory. The (I/O) complexity of this algorithm is $O(sort(n))$, where n is the number of points in the terrain. It is worth to say that our algorithm described in this paper is faster and easier to implement than that one.

3. Viewshed Problem

Most of GIS problems related to visibility involve the viewshed computation and in general, they are optimization problems such as the optimal positioning of facilities, the siting guards minimization, path planing, etc.

The visibility problems can be classified into two major categories: visibility queries and visibility structures computation. The visibility queries consist in checking if a given point is visible or not from an observer (another point) on the terrain. This query can be answered considering that a point q is visible from another point p if and only if the segment connecting the two points, named *the line of sight*, is strictly above the terrain (except on the ending points p and q). See figure 1

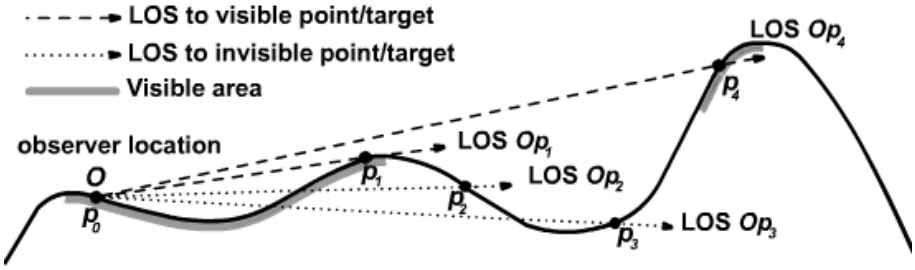


Figure 1. Points Visibility: p_1 and p_4 are visible from p_0; p_2 and p_3 are not visible from p_0.

The visibility structures computation consists in determining some terrain features such as the horizon, the viewshed, etc. The viewshed of a point p on a terrain T can be defined as:

$$viewshed(p) = \{q \in T |\ q \text{ is visible from } p\}$$

Usually, it is convenient to restrict the viewshed to a smaller region, for example, to consider only the points inside a circle centered at p with radius r, the *radius of interest*, that is,

$$viewshed(p, r) = \{q \in T |\ \text{distance}(p, q) \leq r \text{ and } q \text{ is visible from } p\}$$

Unless explicitly said otherwise, when the radius of interest has been defined, we will use $viewshed(p)$ to refer $viewshed(p, r)$.

Usually, the viewshed (in a DEM) is represented by a grid whose size is defined by the radius of interest and each cell stores 1 or 0 to indicate if that cell (point) is visible or not, respectively.

4. I/O efficient Algorithms

As mentioned before, when processing a huge amount of data, the data transfer between fast internal memory and slow external storage (such as disks) often becomes the computation bottleneck. Usually, many GIS software packages implement algorithms for terrain manipulation that were designed assuming internal processing whose aim is to minimize the internal computation time; consequently they often do not scale to large datasets.

In recent years, much research has been done on this topic including the definition of computational models for design and analysis of algorithms that manipulate data in external memory. A model largely accepted was proposed by Aggarwal and Vitter [Aggarwal and Vitter 1988]. Shortly, assuming that M is the internal memory size, B is the disk block size and N is the problem size, this model defines each I/O operation as the transfer of one (disk) block from the external to the internal memory or vice-versa. Then, the measure of performance is determined by the number of such I/O operations executed. The internal computation time is assumed to be free.

Also, the complexity of an algorithm is given based on the complexity of some fundamental problems such as scan or sort N contiguous elements stored in the external memory whose complexities related to I/O operations are:

$$\begin{aligned} scan(N) &= \Theta\left(\frac{N}{B}\right) \\ sort(N) &= \Theta\left(\frac{N}{B}\log_{\left(\frac{M}{B}\right)}\left(\frac{N}{B}\right)\right) \end{aligned}$$

It is important to notice that, usually $scan(N) < sort(N) << N$ and so, in many practical situations, it is significantly better to have an algorithm doing $sort(N)$ instead of N I/O operations. Therefore, many algorithms try to reorganize the data in the external memory to decrease the number of I/O operations executed.

5. External Memory Viewshed Computation (EMVS)

Our algorithm, named External Memory Viewshed (EMVS), is based on the method proposed by Franklin and Ray [Franklin and Ray 1994] that computes the viewshed of a point on a terrain represented as a internal memory matrix. A short description of this method is given below.

5.1. Franklin and Ray's Method

Given a terrain represented by a $n \times n$ elevation matrix T and given a point p on T, the algorithm computes the viewshed of p considering a circle of radius r (the radius of interest) centered at p. The algorithm performs a radial sweep of this circle using a ray, a line of sight (*LOS*), starting at p and thus, it walks along each *LOS* to determine if each

terrain position on the *LOS* is visible from p or not. A terrain position q is visible from p if the *LOS* does not intersect any position whose height is higher than q.

To simplify the circle sweeping, the algorithm uses a square bounding box centered at p with side $2r$ and the lines of sight are defined connecting p to each cell in the square border. Initially, all cells inside the bounding box are set as not visible and for each line of sight l, the algorithm starts at p setting the height of l as $-\infty$ (i.e., a big negative number). So, this height is updated (increased) whenever a higher cell is reached, that is, supposing the current l's height is h and the next cell height is h', if $h < h'$ then the cell is marked as visible and the l's height is updated to h'; on the other hand, if $h \geq h'$, the cell status and the l's height are preserved. The viewshed is stored as a $2r \times 2r$ bit matrix where the visible positions are indicated by 1 and the not visible by 0 (the positions inside the square but outside the circle are set as not visible).

At first glance, the algorithm could be easily adapted to access the terrain points stored in the external memory in the sequence determined by the radial sweep. But, since the terrain matrix is, as usual, stored row by row (in a file), the radial sweeping order would require a "random" access to the file and the execution time would be unacceptably long. Therefore, we adapted this algorithm to avoid the random access order.

5.2. The EMVS algorithm

The basic idea is to generate a list containing the terrain positions (points) sorted by the processing order, that is, the points will appear in the list in the sequence that they will be processed. Thus, instead of accessing the file in a random sequence to process the points as they are reached along the line of sight, the algorithm will access (and process) the points in a sequential order.

It is important to say that the list is also stored in the external memory, but it is managed by a special library STXXL (*Standard Template Library for Extra Large Data Sets*) [Dementiev et al. 2005] that implements containers and algorithms to process huge volumes of data. This library allows an efficient manipulation of data stored externally and, as stated by the authors, "it can save more than half the number of I/Os performed by many applications".

More specifically, the algorithm creates a list L of pairs (c, i) where c is a matrix cell (a terrain point) and i is an index that indicates "when" the cell c should be processed. That is, if a cell c has the index $i = k$ then c will be the kth cell to be processed.

To compute the indices, the lines of sight (originating at the observer p) are numbered in the counterclockwise order starting in the horizontal left to right line of sight that receives the number 0 - see figure 2. Thus, the cells are numbered increasingly along each line of sight; when a line of sight ends, the enumeration continues from the observer (again numbered) following the next line of sight. Of course, a same cell (point) can receive multiple indices since it can be intercepted by many lines of sight. It means that a same point can appear in multiple pairs in the list L, but each pair will have a different index. Also, if the observer is near to the terrain border, that is, if the distance between the observer and the terrain border is smaller than the radius of interest r, some "cells" in the line of sight can be outside the terrain. In this case, those "cells" still will be numbered but they will be ignored and will not be inserted in the list L. This is done to avoid additional tests during the indices computation.

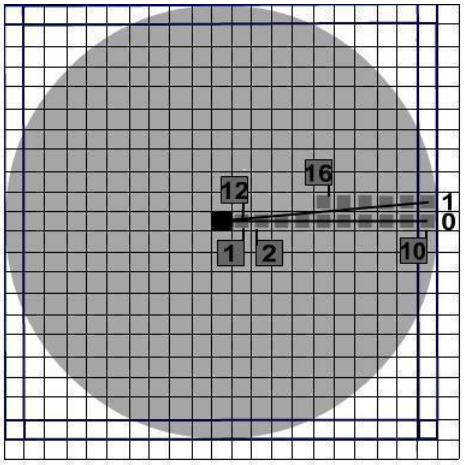


Figure 2. Line of sight numeration

It is important to notice that if the cells indices were computed following the lines of sight as described above, the cells still would be randomly accessed as in the original algorithm. So, to build the list L, the algorithm reads the terrain cells sequentially from the external file and for each cell c, it determines (the number of) all lines of sight that intercept the cell.

Since a cell is not "undimensional", we can determine the cells intercepted by a line of sight using a process similar to the line rasterization [Bresenham 1965]. That is, let s be the side of each (square) cell and suppose the cell is referenced by its center. Also, let a be a line of sight whose slope is α and suppose that $0 < \alpha \leq 45°$ [1]. So, given a cell $c = (c_x, c_y)$, see figure 3, the line of sight a "intersects" the cell c if and only if the intersection point between a and the vertical line c_x is between the points $(c_x, c_y - 0.5s)$ and $(c_x, c_y + 0.5s)$; more precisely, given $(q_x, q_y) = a \cap c_x$, a intersects c if and only if $c_y - 0.5s \leq q_y < c_y + 0.5s$.

Then, as it is easy to see, all lines of sight intersecting the cell c are those between the two lines passing through the points $(c_x, c_y - 0.5s)$ and $(c_x, c_y + 0.5s)$ - figure 3. Let k_1 and k_2 be the numbers of these two lines respectively. Thus, considering the line enumeration sequence, the number of all intersecting lines are $k : k_1 \leq k \leq k_2$.

Now, given a cell c, let r be the number of a line of sight intercepting c. Then, the index i of the cell c associated to r is given by the formula $i = r * n + d$, where n is the number of cells in each ray (this number is constant for all rays in the bounding square box) and d is the (horizontal or vertical) distance between the points c and p - see figure 4. Notice that the distance d is defined as the maximum between the number of rows and columns from p to c.

Next, the list L is sorted by the elements index and then, the cells are processed in the sequence given by the sorted list. Notice that, when a cell c is processed, all the "previous" cells that could block its visibility were already processed. So, the visibility of c can be computed, as described above, just checking the height of the cells along the line of sight. When a cell located on the square border is processed, it means that the

[1] For $45° < \alpha \leq 90°$, use a similar idea interchanging x and y.

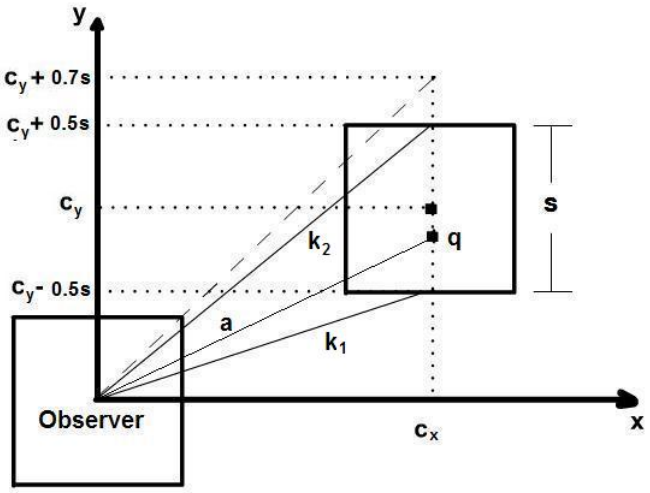


Figure 3. Lines of sight intersecting a cell

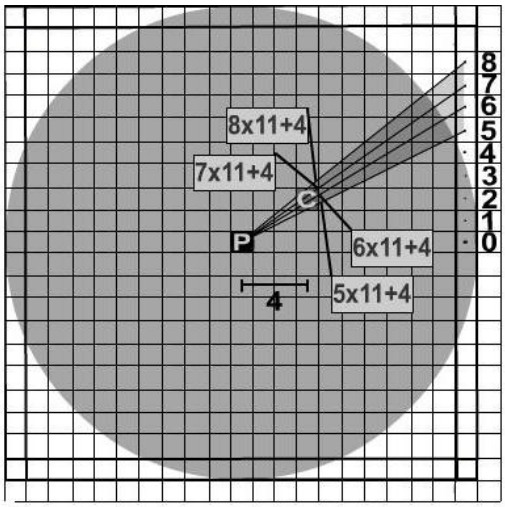


Figure 4. The index cell computation.

processing of a line of sight has finished and the next cell in the list will be the observer's cell indicating that the processing of a new line of sight will start.

For the sake of efficiency, the algorithm uses another list L' (also stored externally and managed by *STXXL*) to keep only the visible cells. More precisely, when the algorithm determines that a cell c is visible, this cell is inserted in the list L'. This list is, in general, much smaller than L since it does not keep the indices and also, usually many points are not visible.

Given the visible cells in L', the algorithm saves the viewshed as a matrix of 0s and 1s such that a visible position is indicated by 1 and a not visible position by 0. This matrix is also stored externally and to generate it (avoiding "random access") the list L' is sorted lexicographically by x and y and each element of the sorted list is stored in the corresponding matrix position.

Finally, it is worth to say that an efficiency improvement is achieved storing a piece of the terrain matrix in the internal memory and so, a lot of I/O operations are avoided. The idea is to store in the internal memory the cells around the observer since those cells are processed more times than the farthest cells. Then, the algorithm selects all cells inside a square centered at the observer position, stores those cells in the internal memory and they are not inserted in the list L. In this way, when a cell needs to be processed, the algorithm checks if it is in the internal memory. If yes, the cell is processed normally; otherwise, it is read from the list L.

6. Algorithm complexity

Let T be a terrain represented by a $n \times n$ elevation matrix. So, T has n^2 cells (points). Also, let p be the observer's point and let r be the radius of interest. As described in section 5.2, the algorithm analyzes the cells that are inside the $2r \times 2r$ square centered at p. Assuming that each cell's side is s then there are, at most[2], $\frac{2r}{s}$ cells in each square's side which implies there are $\frac{8r}{s}$ cells on the square's perimeter. Let $K = \frac{r}{s}$. Thus, the algorithm shots $8K$ lines of sight and since each line of sight has K cells, the list L has, in the worst case, $O(K^2)$ elements.

In the first step, the algorithm does $\frac{n^2}{B}$ I/O operations to read the cells and to build the list L. Next, the list with $O(K^2)$ elements is sorted and then it is swept to compute the cell's visibility. Thus, the total number of I/O operations is:

$$O\left(\frac{n^2}{B}\right) + O\left(\frac{K^2}{B}\log_{\left(\frac{M}{B}\right)}\left(\frac{K^2}{B}\right)\right) + O\left(\frac{K^2}{B}\right)$$

Since the radius of interest r is (much) smaller than n (the terrain matrix side) then K is smaller than n and so, the number of I/O operations is given by $O(\frac{n^2}{B}) = O(scan(n^2))$.

The algorithm also uses an additional external list L' to keep the visible cells and this list needs to be sorted. But, since the list size is (much) smaller than the size of L, the number of I/O operations executed in this step does not change the algorithm complexity. Thus, we can conclude that the algorithm complexity is $O(scan(n^2))$.

7. Results

The algorithm EMVS was implemented in C++, using *g++ 4.1.1*, and the tests were executed in a PC Pentium with 2.8 GHZ, 1 GB of RAM, 80 GB 7200 RPM serial ATA HD running Mandriva Linux.

The algorithm execution time was compared with the Franklin and Ray's algorithm (WRF_VS) which was adapted to manipulate huge terrains stored externally. The adapted algorithm also maintains part of the terrain in internal memory in a similar way that EMVS does. The table 1 and the charts in the figure 5 show the execution time (in seconds) to compute the viewshed considering different radii of interest (ROI) on terrains of different sizes. The 1.45GB (27596 $\times$ 27596 points) and 5.7GB (55192 $\times$ 55192 points) terrains were generated by the concatenation of 4 and 16 instances of a 363MB

[2]If the observer is close to the terrain border, the square might not be completely contained in the terrain.

(13798 $\times$ 13798) matrix representing the Hawaii Big Island. The 2GB (32427 $\times$ 32427 points) terrain was generated by the concatenation of many instances of a 1201×1201 matrix representing the Lake Champlain West (USA-Canada border). These datasets are interesting because they have large height variations since they include lake, ocean and mountains.

Each table entry was obtained by the average of three execution time using different observer positions randomly selected on each terrain.

	1.45GB		2GB		5.7GB
ROI	EMVS	WRF_VS	EMVS	WRF_VS	EMVS
100	21	77	26	121	78
500	26	81	30	122	85
1000	33	97	36	128	99
5000	137	438	73	316	248
10000	478	1313	219	833	643
15000	836	2977	446	1855	1663

Table 1. Execution time (in seconds)

.

Based on these results, it is possible to conclude that the EMVS algorithm is about 3.5 times faster than WRF_VS and also, the former can process much larger terrain (5.7GB or more) while the latter is limited to 2 GB. On the other hand, it is important to say that when the terrain is "small" (i.e. when it fits in the internal memory) the WRF_VS algorithm is a little faster than EMVS, mainly because the lists management and sorting add a time overhead that is not amortized when the terrain size is small.

Furthermore, comparing the EMVS execution time with those reported by Haverkort et al. [Haverkort et al. 2007], we can conclude that our algorithm, besides of being much simpler and easier to implement, is also more than 3.5 times faster than that one. Additionally, it is worth to say that, in their tests, they used a Power Macintosh G5 dual 2.5 GHz, 1GB RAM and 80 GB 7200 RPM that is considerable faster than the machine used in our tests. Thus, it is correct to suppose that our algorithm is still faster than that one.

8. Conclusions and future works

We presented an I/O-efficient algorithm to compute the viewshed of a point in huge terrains represented by a raster DEM stored in the external memory. As tests showed, our algorithm is more than 3.5 times faster than the other ones described in the literature and also, it can process very huge terrains (we used it in 5.7GB terrain). Furthermore, the algorithm is quite simple to understand and to implement. The algorithm implementation is available at http://www.dpi.ufv.br/ marcus/TerrainModeling/EMViewshed/EMVS.tgz as an open source code distributed under Creative Common GNU GPL license [Creative Commons 2007].

As a next step, we started to work on the NP-hard optimization problem to site observers in huge terrains stored in the external memory. Our aim is to develop an approximation algorithm to place the "almost" minimum number of observers necessary to

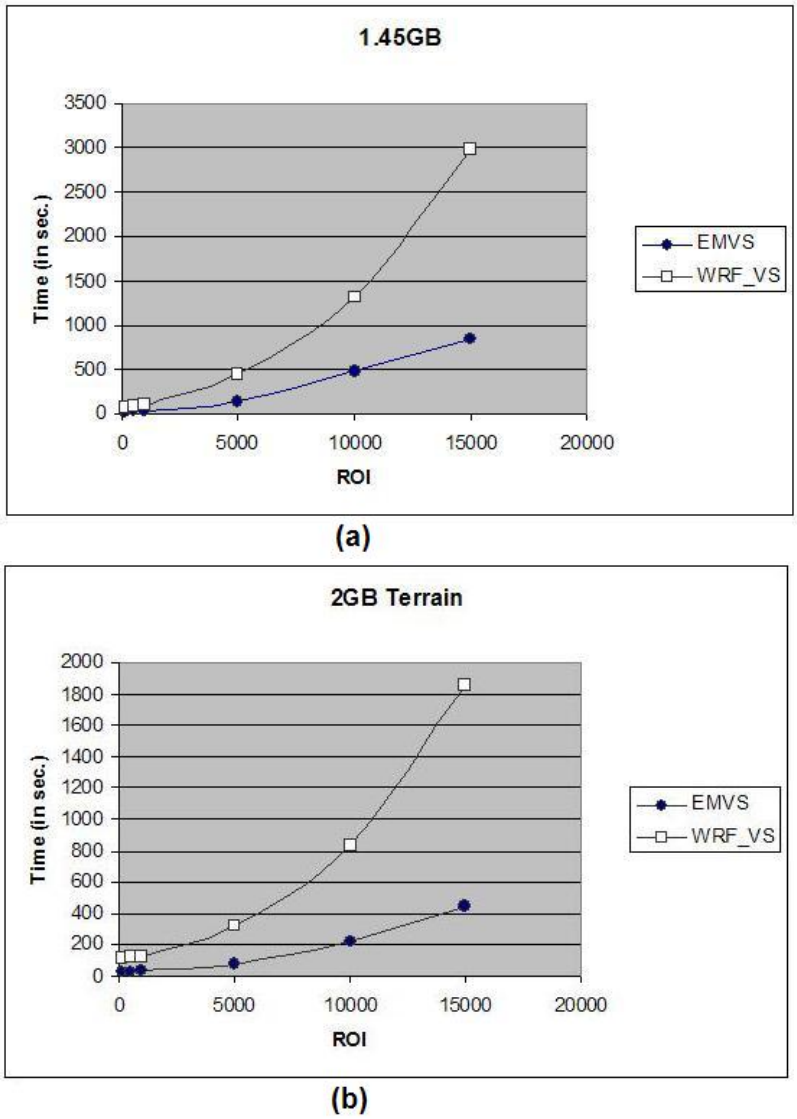


Figure 5. Execution time charts: (a) in 1.45GB terrain; (b) in 2GB terrain.

cover "almost" the whole terrain. More precisely, given a terrain T, we want to determine where to site the almost minimum number of observers to cover visually an user defined percentage of the terrain. This problem has a wide range of applications such as: telecommunications (cellular towers), military (guards), agriculture (irrigation), etc.

Acknowledgment

This work was partially supported by CNPq and FAPEMIG.

References

Aggarwal, A. and Vitter, J. S. (1988). The input/output complexity of sorting and related problems. *Communications of the ACM*, 9:1116–1127.

Arge, L. (1997). External-memory algorithms with applications in geographic information systems. In In M. van Kreveld, J. Nievergelt, T. R. e. P. W., editor, *Algorithmic Foundations of GIS*. Springer-Verlag.

Arge, L., Chase, J. S., Halpin, P., Toma, L., Vitter, J. S., Urban, D., and Wickremesinghe, R. (2003). Efficient flow computation on massive grid terrains. *GeoInformatica*, 7:283–313.

Arge, L., Vengroff, D. E., and Vitter, J. S. (1995). External memory algorithms for processing line segments in geographic information systems. In *In Proc. European Symposium on Algorithms, LNCS 979*, pages 295–310.

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

Ben-Moshe, B., Carmi, P., and Katz, M. (2004a). Approximating the visible region of a point on a terrain. In *Proc. Algorithm Engineering and Experiments (ALENEX'04)*, pages 120–128.

Ben-Moshe, B., Katz, M., Mitchell, J., and Nir, Y. (2004b). Visibility preserving terrain simplification - an experimental study. *Comp. Geom., Theory and Applications*, 28:175–190.

Ben-Moshe, B., Katz, M. J., and Mitchell, J. S. B. (2007b). A constant-factor approximation algorithm for optimal 1.5d terrain guarding. *SIAM J. Comput*, 36(6):1631–1647.

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

Bresenham, J. (1965). An incremental algorithm for digital plotting. *IBM Systems Journal*.

Camp, R. J., Sinton, D. T., and Knight, R. L. (1995). Viewsheds: A complementary management approach to buffer zones. *Wildlife Society Bulletin*, 25:612–615.

Creative Commons (2007). http://creativecommons.org/license/cc-gpl (accessed august 2007).

Dementiev, R., Kettner, L., and Sanders, P. (2005). Stxxl : Standard template library for xxl data sets. Technical report, Fakultat fur Informatik, Universitat Karlsruhe. http://stxxl.sourceforge.net/ (acessed on July 2007).

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

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

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

Floriani, L. D., Puppo, E., and Magillo, P. (1999). Applications of computational geometry to geographic information systems. In J. R. Sack, J. U., editor, *Handbook of Computational Geometry*, pages 303–311. Elsevier Science.

Franklin, W. R. (2002). Siting observers on terrain. In Springer-Verlag, editor, *In D. Richardson and P. van Oosterom editors, Advances in Spatial Data Handling: 10th International Symposium on Spatial Data Handling*, pages 109–120.

Franklin, W. R. and Ray, C. (1994). Higher isn't necessarily better - visibility algorithms and experiments. In *6th Symposium on Spatial Data Handling*, Edinburgh, Scotland.

Franklin, W. R. and Vogt, C. (2006). Tradeoffs when multiple observer siting on large terrain cells. In *12th International Symposium on Spatial Data Handling*.

Goodrich, M. T., Tsay, J. J., Vangroff, D. E., and Vitter, J. S. (1993). External-memory computational geometry. In *IEEE Symp. on Foundations of Computer Science*, pages 714–723.

Haverkort, H., Toma, L., and Zhuang, Y. (2007). Computing visibility on terrains in external memory. In *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments / Workshop on Analytic Algorithms and Combinatorics (ALENEX/ANALCO)*.

Kumler, M. P. (1994). An intensive comparison of triangulated irregular network (tins) and digital elevation models (dems). *Cartographica*, 31(2).

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

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

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

Stewart, A. J. (1998). Fast horizon computation at all points of a terrain with visibility and shading applications. In *IEEE Trans. Visualization Computer Graphics*, pages 82 – 93.

USGS (2007). The USGS Center for LIDAR Information Coordination and Knowledge. http://lidar.cr.usgs.gov/ (accessed October 2007).

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

Young-Hoom, K., Rana, S., and Wise, S. (2004). Exploring multiple viewshed analysis using terrain features and optimization techniques. *Computers and Geosciences*, 30:1019–10323.

Comparison of Machine Learning Algorithms for Mapping the Phytophysiognomies of the Brazilian Cerrado

Luciano T. de Oliveira[1], Thomaz C. de A. Oliveira[2], Luis M. T. de Carvalho[3], Wilian Soares Lacerda[4], Samuel R. de S. Campos[5], Adriana Z. Martinhago[6]

[1]Departamento de Ciências Florestais – Universidade Federal de Lavras (UFLA)
Caixa Postal 3.037 – 37.200-000 – Lavras – MG – Brazil

oliveiralt@yahoo.com.br, thomaz@vialavras.com.br, passarinho@ufla.br, lacerda@ufla.br, samuelcampos@ufla.br, dricazn@gmail.com

Abstract. *This present work describes the classification of the Phytophysiognomies present in the Brazilian Cerrado biome through the means Artificial Intelligence; data from remote sensing images and other sources served as input for these algorithms to generate the vegetation maps. The data acquired was of many types so that it fully described the various Phytophysiognomies present in biome and served as training data for the machine learning algorithms. Various statistical and neuro-computation based algorithms were used for pattern recognition in the data so that we could build a good generalization model for the biome. A vegetation map was successfully generated with each algorithm. Finally a comparison among these algorithms was made so that we could find the best algorithm that fitted the problem of mapping this biome.*

keywords: Image classification, decision trees, maximum likelihood, neural network, Cerrado Phytophysiognomies

1. Introduction

The cerrado biome of tropical South America covers about 2 million km^{2,} an area approximately the same as that of Western Europe, representing ca. 22% of the land surface of Brazil. The biome was named after the vernacular term of its predominant vegetation type a fairly dense woody savanna of shrubs and smalls trees. The term Cerrado (Portuguese for "half-closed," "closed" or "dense") was probably applied to this vegetation originally because of the difficulty of traversing it on horseback. (Oliveira-Filho et al 2002) The constant threat to the Brazilian Cerrado has lead to the necessity of strategies and measures to promote the monitoring and mapping of this biome. The Cerrado has a large biodiversity but it's fragmentation throughout the years has lead to the losses of exemplars from this biome. This process can be noticed in the red books of fauna (Machado et al., 1998) and flora (Mendonça & Lins, 2000) of the Minas Gerais.

The absence of a precise mapping occurs not only of this biome but to the others present in the state of Minas Gerais too. This leads to difficulties in the environmental

management due to database deficiencies reflecting in other aresas management of the state.

This can be noticed in the northern part of the Minas Gerais state, where the biome of Cerrado occurs very intensively. In this region we can observe areas with big social problems that intensify by the lack of social and forest management that lead to a clandestine exploration of vegetable coal which is intensified by the product's high market value.

This work's main objective is to develop an efficient methodology to generate the mapping of the various phytophysiognomies present in the Savanna biome region and to promote a comparison among the classification algorithms used to generate theses vegetation maps. Precise vegetation mapping can help the monitoring and the environmental administration of such areas. The main objective of this work is to propose a methodology based on machine learning algorithms that can help achieve this objective of precise mapping of the phytophysiognomies present in the Cerrado biome.

2. Methods

2.1. Field sampling of the phytophysiognomies.

Initially the classification proposed by Ribeiro & Walter (1998) was used to promote the characterization and for the choice of division level of the phytophysiognomies in the Cerrado biome.

Throughout qualitative analyses of images EMT+, during a low humidity period and a high humidity period, some areas were identified as representative areas of forest fragments and also components of the agricultural landscape (Louzada, 2000).

Some field expeditions by air and land occurred for identification and analyses of the distribution of the remaining phytophysiognomies of the region; these were latterly used as samples of earth observation trueness.

The phytophysiognomical levels were adapted from the original classification from Ribeiro & Walter (1998) because of the necessity to adequate the characteristics of the analyzing sensor that is not capable of resolving small forest fragments that are representative of one phytophysiognomy. Valey-side marshy grasslands fragments are very small and tend to be found mixed among riverine forests. Open grassland and grassland with scattered shrubs fragments are very associated to each other. By the characteristics of the fragments just described, some of the fragments individually would not be captured because of their small size and the object's size must be at least three times the size of the sensor's resolution.

Analyzing an individual date of remote sensing data to extract meaningful vegetation biophysical information is often of value. However timing is a very important when attempting to identify different vegetation types or to extract useful vegetation biophysical information (e.g. biomassa, chlorophyll characteristics) from remotely sensed data (Jensen 2000). The group of samples was initially established from the EMT image from a dry spell period/high humidity period. From the entire group of samples, 30% were separated

randomly from each class of phytophysiognomies. These samples were used as training data for the algorithms, accuracy data of the classification and latter as test data of the accuracy (**Table 1**). This last one being only used on the decision tree algorithms. By this way, was a compromise between stratified sampling and randomly chosen sampling was set.

TABLE 1 - Total number of sampled pixels

	Forests	**Savannic**	**grassland**	**Eucalipte plantation**	**pasture**	**cropland**	**Bair soil**	**Water**	**Shades**
Samples	8237	67251	53956	18710	42396	1814	38487	5635	2104

2.2. Image Processing

With the goal of reducing the noise resulting form image fusion, and due to the atmospheric influence in the panchromatic image, the Lee filter with a 3x3 window was used so that it would reduce the texture resulting from the noise but not the image detail.

Vegetation indices are dimensionless, radiometric measures that function as indicators of relative abundance and activity of green vegetation. A vegetation index should: maximize sensitivity to plant biophysical parameters, normalize external effects such as Sun angle, normalize internal effects such as canopy background variations. There are more than 20 vegetation indices in use (Jensen 2000). A NDVI (Normalized Difference Vegetation Index) was calculated in all of the images, making a relationship in the band that represents the red and infrared wavelengths bands, those bands correspond to the bands 3 and 4 if the EMT+ images.

The Tasseled Cap transformation is a global vegetation index. Theoretically, it may be used anywhere in the world to disaggregate the amount of soil brightness, vegetation, and moisture content in individual pixels in a Landsat MSS or Thematic Mapper image (Jensen 2000). The coefficients necessary for the Tasseled Cap (**Table 2**) were only applied on the ETM+ images. (Crist & Cicone 1984) calculated these coefficients so that they can be applied on digital images.

TABLE 2 - Coefficients of Tasseled Cap applied on Landsat images

Indexes	**ETM+ 1**	**ETM+ 2**	**ETM+ 3**	**ETM+ 4**	**ETM+ 5**	**ETM+ 7**
Brightness	0,3037	0,2793	0,4743	0,5585	0,5082	0,1863
Greenness	-0,2848	-0,2435	-0,5436	0,7243	0,0840	-0,1800
Wetness	0,1509	0,1973	0,3279	0,3406	-0,7112	-0,4572

The mixture fractions were obtained taking into account the simplex theory (Correia, 1983, Aguiar, 1991; Mather, 1999; Tso & Matter 2001, Schowengerdt, 1997), thus obtaining the pure pixels from the extremes of the distribution from the sampling space domain (Red X infrared).

According to (INPE 2002), there is no necessity to convert the digital pixels into values of reflectance when the values were obtained from the image itself. Thus it was decided to leave the values in digital numbers.

The model was applied with some restrictions; the fractions of shades, vegetation and soil could not overcome 100% of total mixture found in a pixel. One image of 25 classes was generated by the ISODATA unsupervised classification method, applying 10 iterations with a minimum value of 10 pixels per class and gathering a number of six isolated pairs in the class. This image was created with the intention of simplifying the information of the image, helping in this case the performance of classifying algorithms and the distinction of the each pixel in the classification. This is helpful, once the classifiers work on every pixel individually.

With altitude curves in the forms of vectors from IBGE institute, it was generated one image of the classes of altitude. In this image the altitude was rearranged into a 0 to 255 range, the lowest altitude being 0 and the highest altitudes of the region 255.

For the river buffer it was necessary the extraction of the whole of the hydrography through a visual analyses of high resolution images. The buffer ranges from 0 up 255, zero being the value where river stands and it gradually increases up to 255 as the distance increases from the river. Values of 255 correspond to locations distant from a river pixel.

The images of classes of altitude and hydrography are very important to this work since they establish import features and relationships that characterize the vegetation of the Cerrado biome.

2.3. Image classification

As commented earlier, the main objective of this work is to compare image classification algorithms among themselves. For Moreira (2003) an automatic image identification and classification can be sought as the analyses and the manipulation of images through computational techniques, with the goal of extracting information regarding an object of the real world. For this research, maps of the phytophysiognomies of the Cerrado biome were generated with the following algorithms: Decision trees, Maximum likelihood, Kohonen's self Organizing maps with supervised learning, Multi layer perceptrons and Fuzzy ART maps Neural networks.

2.3.1. Maximum Likelihood

The Maximum likelihood is a statistical algorithm that necessitates some previous sampling before its operation (learning stage of the classifier), where it can be established a previous indication of the number and a specific pattern of a certain class. (Lillesland & Kiefer, 2000).

This classifier is based in the Bayesian theory of probability; it uses an array of patterns and a covariance matrix from a Gaussian distribution sample set. (Lillesland & Kiefer, 2000, Gonzáles e Woods, 2000 ; Tso & Mather, 2001). The classification is therefore

defined by the smallest number of standard deviation from sample set. Thus each pixel is classified according to an average array and covariance matrix. The maximum likelihood distinguishes from the other classifiers by having good overall performance for classifying Earth's surface. (Carvalho, 2001 ; Marcelino et al., 2003 ; Oliveira et al., 2002). 30% from the total number of samples of earth trueness were used as training data set of the algorithms; these values are present in **Table 1**. In the maximum likelihood it was considered that all the pixels had the same probability of belonging to each one of the present classes.

2.3.2. Decision Tree

The decision tree is a non parametrical classifier that is based in the inductive learning of a human being, where one can learn to separate the classes throughout training data (Quilan, 1986). From the training data, that can be describe as a set of attributes (e.g. altitude, reflectance, NDVI, etc), a binary rule can be established so that the samples set can be divided into two more homogenous data sets than the original set. This procedure will occur until the divisions lead up to each desired class of attributes. The decision rules are obtained by the definition the best discriminative function based on linear combinations of the certain attributes (Breinman et al., 1984).

With the generation of all the described images it was obtained a set of attributes that were extracted from the training and testing data sets. These sets were used to generate and choose the best decision trees, using the Gini algorithm.

2.3.3. Neural Networks

The neural networks are problem solving algorithms of the artificial intelligence that use methods and techniques inspired on historical facts and models of biological neurons and networks. These biological inspired models are extremely efficient when the pattern of classification is not a simple and trivial one (Barreto 2002). Theses networks have shown to be helpful in the resolution of problems of practical scope. Problems such as voice recognition, optical character recognition, medical diagnosis and other practical scope problems are by no means complex problems to the human brain and sensor as they are for a computer to resolve. Theses problems however can be resolved computationally through an artificial network of neurons.

Even though some researchers do not recognize the neural networks as being the general natural solution surrounding the problems of recognizing patterns on processed signals, it can be noticed that a well trained network is capable of classifying highly complex data (Kanellopolous et all 1997).

According to (Wilkinson 1997) the use of neural networks in pattern recognition and classification has grown in the last years in the field of remote sensing. A neural network needs to be capable of transforming spectral radiations into thematic maps that represent the reality.

For the interest of this work we used different types of networks. A SOM (self organizing maps) Kohonen (1990), network was used for classifying the vegetation with

supervised learning. The following described parameters used in this research on neural networks were reached through experiments and tests and limited computer power. The networks were trained and re-trained several times. Various tests were done with different network parameters aiming to reach the networks that best classified our problem of generating vegetation map. The supervised SOM had the following parameters: 31 layers with 31 neurons per layer, with variable learning rate that went from 0.5 to 1.0. The number of epochs required was of 20776 with a final quantification error of 0.1672.

A multi layer perceptron was also used for this work with the following parameters: only one hidden layer, sigmoid activation function, initial neighborhood radius of 46.25, learning rate of 0.1 and momentum of 0.5. The network was trained with 10000 iterations that lead out 95.11% of correct classifications in the training data set.

2.3.4. Soft classifiers

For (Mather 1999) the use of Fuzzy, or soft classifiers, is adequate when we want to avoid errors of classification due to ambiguity of the classes generated during the classification. When a pixel has characteristics that can include it in two or more classes, future errors of classification will occur due to this ambiguity. Fuzzy maps allow a determined pixel to be in different classes at the same time depending on the pertinence level of the pixel to each class. A fuzzy ArtMap can be generated based on the ART (Adaptavide Resonance Theory) (Carpenter et al, 1991) which is a theory that describes the biological cognitive learning of the living creatures. The ART networks were specially developed to resolve the stability-plasticity dilemma and exhibit a high degree of stability in order to preserve significant past learning, but remains adaptable enough to incorporate new information whenever it might appear (Carpenter, 1989). Fuzzy ART is a clustering algorithm that operates on vectors with fuzzy analog input patterns (real numbers between 0.0 and 1.0) and incorporates an incremental learning approach which allows it to learn continuously without forgetting previous learned states.

2.4 Training and Classification

The classification preceded as described earlier, using training samples which correspond to approximately 30 % of the total number as seen in **Table 1**. The training phase must happen to each algorithm before it can be used for classification. In the maximum likelihood algorithm training, it was considered that each pixel had the same probably of being in each class.

It was used the same training data set for the maximum likelihood, the decision tree and the also in all the kinds neural networks and fuzzy ArtMaps. For the test of theses decision trees a new set data was extracted from the original data set with values described in **Table 1,** for which the set had the same number of pixels as the training data set.

2.5 Accuracy and comparison of the generated images

As commented earlier, the main objective of this work is to verify the accuracy of these classifiers comparing them. To accomplish this, a set of accuracy samples were used as

seen in **Table 1**. With the accuracy samples a confusion matrix was generated, by which the Kappa coefficient (Colganton & Green, 1999; Tso & Mather, 2001) was extracted from. By doing this, we can compare statistically the quality of each algorithm to resolve this problem with its dataset.

3. Results

3.1. Data mining, image classification, analyses of the matrixes.

After the training phase a multivariate decision tree with the lesser possible relative cost was chosen, that is, one that has smallest possible mixture of classes on the terminal leaves (Breiman et al., 1984).

With the previous selection of the tree and its respective confusion matrix was generated using the accuracy samples. These matrixes were also generated using the maximum likelihood and for each of the type of neural classifier.

The set of temporal images obtained high Kappa coefficient values (**table 3**). This can be explained by the fact that a temporal set of images captures the phonological cycle of the vegetation.

TABLE 3 - Results of Kappa coefficient from the set of images Temporal Landsat

classification	**Max likelihood**	**Decision tree**	**MLP**	**Supervised SOM**	**Fuzzy ArtMap**
	Kappa	Kappa	Kappa	Kappa	Kappa
Values	0,9190	0,9574	0,9465	0,8043	0,9635

The classification of the Cerrado biome followed the previous steps which included classification with: MPL, Fuzzy ArtMap neural network, decision tree, and maximum likelihood from the set temporal EMT+ (**Figures 1a, 1b e 1c).** A confusion matrix was generated for the evaluation of the best Kappa coefficient **Table 4**

After applying the Landis & Koch (1977) evaluation all the classifications were all defined as excellent.

It was noticed that the Fuzzy ArtMap neural network obtained a better efficiency than all the others algorithms analyzed, thus assuring the better quality for this algorithm to classify the phytophysiognomies of the Cerrado biome.

TABLE 4 - Confusion matrix for Temporal Landsat with 96,98% of accuracy and 0,9635 of Kappa Coenfficient

Class	Water	Cropland	Bare soil	Shades	Pasture	Eucalipte plantation	Savannic	Grassland	Forests	Total
Water	1480	29	85	13	9	12	12	4	9	**1653**
Cropland	0	386	0	0	0	0	0	0	0	**386**
Bare soil	3	2	2953	0	5	11	0	1	24	**2999**
shades	2	0	0	469	0	9	11	0	0	**491**
pasture	2	1	1	0	4264	0	14	18	0	**4300**
Eucalipte plantation	3	0	23	8	0	2499	3	59	1	**2596**
Savannic	8	0	0	10	15	16	2184	72	1	**2306**
Grassland	2	0	7	0	26	83	96	7731	18	**7963**
Forests	0	2	41	0	1	0	0	15	3297	**3356**
Total	**1500**	**420**	**3110**	**500**	**4320**	**2630**	**2320**	**7900**	**3350**	**26050**

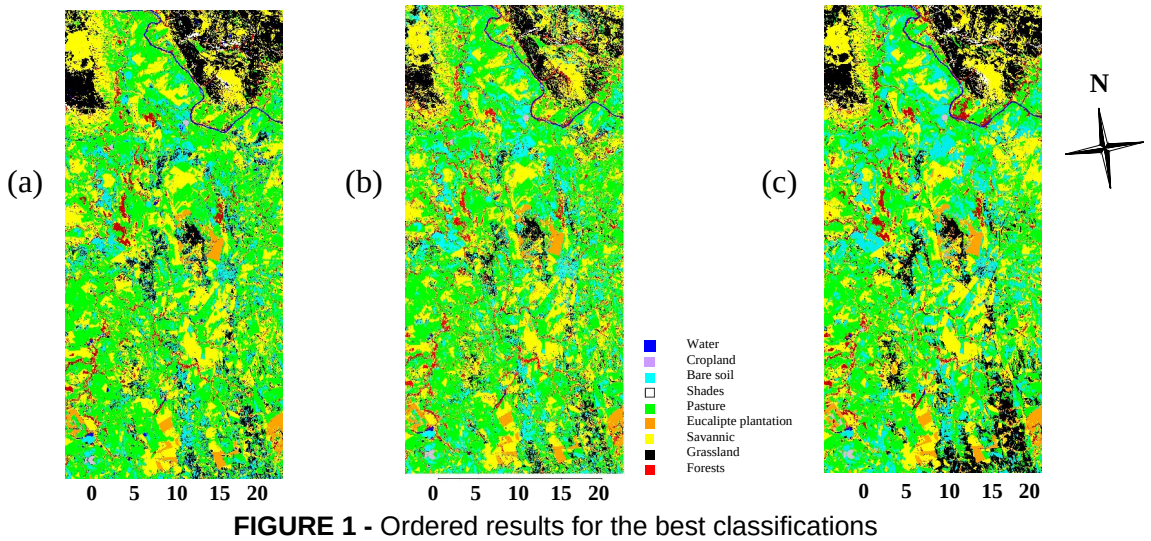


FIGURE 1 - Ordered results for the best classifications
(a) Fuzzy ArtMap neural network, (b) Decision tree (c) MLP

4. Conclusions

There are several artificial intelligence algorithms that can be used in remote sensed data to classify images and generate theme maps. All these algorithms depend in some way to the operators experience in setting up parameters of the algorithms to reach their optimal performance. When these parameters are set wisely all the algorithms work efficiently showing good overall performance, thus remembering that all these parameters should be readjusted to different data sets. The algorithms: Max likelihood, Decision tree, Decision tree, Multi layer perceptron, Fuzy ART maps showed efficiency in classifying the phytophysiognomies in the Cerrado biome. The supervised neural network using Fuzzy ARTmaps was the most efficient of the algorithms, followed by the decision tree, multy-layer perceptron and maximum likelihood.

The results show that machine learning algorithms are highly capable of mapping the Phytophysiognomies of the Brazilian Cerrado and should be highlighted that these techniques could be improved in future work so that influence of the operator should be diminished on the results.

5. References

Aguiar, A. P. D. (1991) Utilização de atributos derivados de proporções de classses dentro de um elemento de resolução de imagem (“pixel”) na classificação multiespectral de imagens de sensoriamento remoto. São José dos Campos: INPE,.

Breiman, L., Friedman, J. H., Olshen, R. A. (1984) Classication and regression tress. Belmont: Chapman & Hall,. 358 p.

Carpenter G.A, (1989) Neural Network Models for Pattern Recognition and Associative Memory. Neural Networks, 2, 243-257,.

Carpenter G. A., (1991) Crossberg, S., and Reynolds, J.H, ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network. Neural Networks, 4, 565-588,.

Carvalho, L. M. T. (2001) Mapping and monitoring forest remnants: a multi-scale analysis of spatio-temporal data. 2001. 140 p. Thesis (Doctor) - Wageningen University, Wageningen..

Colgaton, R. G., Green, K. (1999) Assenssing the accuracy of remotely sensed data: principles and practices. New York: Lewis Publishers, 137 p.

Correia, V. R. M. (1983) Estudo das medidas de qualidade para estimação de proporções de classes em elementos de resolução de imagens.. Dissertação (Mestrado) - INPE, São José dos Campos.

Crist, E. P., Cicone, R. C. (1984) A physically – based transformation of thematic mapper data – the TM Tasseled Cap. IEEE Transactions on Geocience and Remote Sensing, Los Alamitos, v. 22. n. 3, p. 256-262,.

Gaboardi, C. (2002) Utilização de imagem coerência SAR para classificação do uso da terra: Floresta Nacional do Tapajós. 137 p. Dissertação (Mestrado) - INPE, São José dos Campos.

Winkinson G. (1997) Open Questions in Neurocomputing for Earth Observation.

Instituto Nacional de Pesquisas Eespeciais. (2002) Divisão de Processamento de Imagens (INPE-DPI). SPRING, Manual do usuário [on line]. São José dos Campos.

Kouokoulas, S.; Blackbum, G. A. (2001) Introducing new indices for accuracy evaluation of classified images representing semi-natural woodland environments. Photogrammetric Engineering & Remote Sensing, Los Alamitos, v. 67, n. 4, p. 499 -510.

Landis, J. R., Koch, C. H. (1977) The measurement of observer agreement for categorical data. Biometrics, Washington, v. 33, n. 3, p. 159-174.

Lillessand, T. M., Kiefer, R. W. (1999) Remote sensing and image interpretation. 4. ed. USA: John Wiley. 724 p.

Louzada, J. N. C. (2000) Efeitos da fragmentação florestal sobre a estrutura da comunidade scarabaeidae (Insecta, coleoptera). 87 p. Tese (Doutorado) – Universidade Federal de Viçosa, Viçosa, MG.

Machado, A. B. M., Fonseca, G. A. B., Machado, R. B., Aguiar, L. M. S., Lins, L. V. (1998)Livro vermelho das espécies ameaçadas de extinção da fauna de Minas Gerais. Belo Horizonte: Fundação Biodiversitas. 608 p.

Mather, P. M. (1999) Computer processing of remotely-sensed images: an introduction. 2. ed. Nottingham, UK: Johb Wiley,. 292 p.

Mendonça, M. P., Lins, L. V. (2000) Lista vermelha das espécies ameaçadas de extinção da flora de Minas Gerais. Belo Horizonte: Fundação Biodiversitas/Fundação Zôo-Botânica de Belo Horizonte. 160 p.

Molenaar, M. (1998) An introduction to theory of Spatial object modelling for GIS. Enschede, The Netherlands: Taylor & Francis,. 246 p.

Moreira M. A., (2003) Fundamentos de Sensoriamento Remoto e Metodologias de Aplicação, 2ª Edição Revista e Ampliada Eidtora UFV 295p.

Kanellopoulos G.G., Wilkinson F.Roli, J.Austin, (1997) Neuro-computation in Remote Sensing Data Analysis,.

Kohonen, T., 1990, The Self-Organizing Map. Proceedings of the IEEE, 78: 1464-80.

Oliveira, L.T.; (2004) Fusão de imagens de sensoriamento remoto e mineraçao de dados geográficos para mapear as fitofisionomias do Bioma Cerrado.. 131p. (CDD – 621.3678 – 526.982) Dissertação (Mestrado em Manejo Ambiental)– UFLA. Lavras. 2004.

Ribeiro, J. F., Walter, B. M. T. (1998) Fitofisionomias do bioma Cerrado. In: Sano, S., Almeida, S. P. (Ed.) Cerrado: ambiente e flora. Planaltina: EMBRAPA-CPAC,. p. 89-169.

Schowengerdt, R. A. (1997) Models and metods for image processing. 2. ed. New York: Academy Press,. 522 p.

Shimabukuro, Y. E. (1987) Shade images derived from linear mixing models of multispectral measurements of forested areas.. Dissertation (Doctor of Phylosophy) - Colorado State University, Fort Collins.

Tso, B., Mather, P. M. (2001) Classification Methods for remotely sensed data. New York: Taylor & Francis, 332 p.

Oliveira-Filho, A. T. , Ratter J. A., (2002) The Cerrados of Brazil: ecology and natural history of a neotropical savannah, Columbia University Press Publishers, New York Chischester, West Sussex p. 91-121

Jensen J. R. 2000, Remote Sensing of the enviroment: An Earth resource perspective, Pretince Hall Series in geographic information science, Upper SaddleRiver, New Jersey 07458

SHORT PAPERS

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 209-214.

Construção de Mosaicos Georreferenciados Usando Imagens Aéreas de Pequeno Formato para SIG

Natal Henrique Cordeiro[1], **Bruno Motta de Carvalho**[1], **Luiz Marcos Garcia Gonçalves**[1]

[1]Univerdidade Federal do Rio Grande do Norte(UFRN)
Campus Universitário – Lagoa Nova – Natal – RN – Brasil

natal@ppgsc.ufrn.br, motta@dimap.ufrn.br, lmarcos@dca.ufrn.br

Abstract. *We propose to use small format aerial images (SFAI) that are considered as not controled and stereo-photogrammetry techniques for construction of georreferenced mosaics. The images are obtained using a simple digital camera coupled to a small radio controled helicopter. Techniques for removing common distortions are applied and the relative orientation of the models are performed based on perspective geometry. Then, ground truth points are used to get absolut orientation, plus a definition of scale and a coordinate system which relates image measures to the ground. The mosaic is to be read into a GIS system, which is also being developed based on AJAX, providing useful information to users. Preliminary results have shown the aplicability of the system.*

Resumo. *Propomos construir mosaicos georreferenciados com técnicas da estéreo-fotogrametria usando imagens aéreas de pequeno formato(SFAI) não controladas. As imagens são obtidas com uma câmera digital comum acoplada a um helicóptero aeromodelo. Técnicas para corrigir distorções é aplicada e a orientação relativa dos modelos é realizada baseada na geometria perpectiva. Pontos reais do terreno são usados para executar a orientação absoluta, definindo uma escala e um sistema de coordenadas que relacione medidas da imagem com o terreno. O mosaico será manipulado em um SIG, que esta sendo desenvolvido baseado em AJAX, fornecendo informação útil aos usuários.*

1. Introdução

A digitalização dos processos analógicos propiciou a geração de mosaicos georreferenciados, o que antes não era possível com qualidade sem um processo fotográfico complexo. Um mosaico nada mais é que uma colagem de várias imagens adjacentes, visando ter uma visão mais ampla (e na mesma escala) de uma determinada cena. No caso, um mosaico georreferenciado pode ser obtido após uma série de etapas, que envolvem a retirada de erros e distorções causados pelo processo ou pelo sistema de aquisição, com aplicação de transformações visando reconstruir as posições e orientações relativa, das imagens umas às outras, e absoluta, em relação à cena, e finalmente a definição e adoção de uma escala e sistema de representação.No processo cartográfico, imagens adquiridas por técnicas controladas são utilizadas, sendo este processo, caro, pois envolve uso de câmeras especiais e o emprego de aviões específicos para aquisição. O uso de imagens de satélite é uma alternativa, porém menos precisa em relação a profundidade e em relevos acentuados, comparado ao modelo utilizado neste trabalho, caso esta seja necessário. Com o uso do helicóptero aeromodelo podemos obter imagens com escalas bem próximas da superfície, além de realizar mapeamento de quaisquer regiões, como exemplo, a título de inovação, neste projeto, trabalhamos com regiões marinhas, carentes de mapeamento detalhado.

2. Estado da Arte

Como visto, o processo de aquisição usando uma câmera digital comum acoplada no helicóptero aeromodelo torna-se mais barato se comparado aos processos de aquisição via satélite ou de câmeras aerofotogramétricas acopladas em aviões. No entanto, como são disponibilizadas SFAI sem um maior controle, torna-se indispensável readaptar e/ou criar metodologias adequadas para este formato de imagem. Sistemas como o ArcView da ESRI, ERDAS da Leica Geosystem, Regeemy do INPE entre outros como [Grandi et al. 2000, Lhuillier et al. 2001, Hsu 2001], permitem gerar mosaicos de qualidade usando técnicas de registro em imagens aéreas de grande formato (BFAI). Convém ressaltar que, por definição, imagens do tipo BFAI são controladas, sendo em alguns casos consideradas com distorções mínimas e até mesmo já georreferenciadas em algumas aplicações. Isso facilita substancialmente o processo, o que não ocorre no tipo de problema que estamos tratando. No presente projeto, estaremos empregando SFAI com vários tipos de distorções e que presenciam pouquíssimos pontos de controle. Os método apresentados por [Albrecht and Michaelis 1998, Nogueira], expõem áreas relacionadas com este projeto, como técnicas da estéreo-fotogrametria a fim de reduzir erros em regiões com relevo acentuado e geração de mapas de disparidades.

3. Geração de mosaicos georreferenciados com SFAI's

O processo de reconstrução de mosaicos georreferenciados é dificultado pelo uso de SFAI. Além disso, ocorrem variações de posição e orientação do helicóptero durante o vôo, que podem gerar erros imprevisíveis e ainda, os parâmetros intrínsecos da câmera digital usada podem causar distorções radial e radiométrica. Para a obtenção de resultados interessantes e de qualidade, vários procedimentos ou técnicas devem ser aplicados, sendo elas, basicamente, técnicas de calibração de câmera, correção das distorções radial e radiométrica, reconstrução a partir de estéreo-fotogrametria e a geração do mosaico georreferenciado propriamente dita. Determinamos os parâmetros internos e externos da câmera no módulo de calibração com modelo Tsai. Seguida da calibração de câmera realizamos a correção das distorções radial, com a Equação 1 e radiométrica, com a Equação 2 que tem por objetivo corrigir as distorções ou degradações oriundas do processo de aquisição da imagem, tanto geométrica como de iluminação respectivamente.

$$\begin{aligned} x &= x_d(1 + k_1 r^2 + k_2 r^4) \\ y &= y_d(1 + k_1 r^2 + k_2 r^4) \\ r &= \sqrt{x_d^2 + y_d^2} \end{aligned} \tag{1}$$

Na Equação 1, x_d e y_d são os pontos na imagem distorcida, r é a distância do centro da imagem até o pixel e k_1 e k_2 são os coeficientes de distorção. Em seguida, é realizada a correção da distorção radiométrica, usando a Equação 2, [Trucco and Verri 1998].

$$E(p) = L(P)\left[\frac{\pi}{4}\left(\frac{d}{\hat{z}}\right)^2 cos^4\alpha\right] \tag{2}$$

Convém ressaltar que a iluminação na imagem P decresce o mesmo que a quarta potência do cosseno do ângulo formado pelo raio principal que chega em P com o eixo ótico[Trucco and Verri 1998].

3.1. Estereofotogrametria

Na aplicação de monitoramento ambiental, foco deste trabalho, temos recobrimento tanto longitudinal (cerca de 70%) quanto lateral (30%) entre as imagens que farão parte do mosaico e cada imagem é adquirida de uma posição diferente. Isso propicia que técnicas de reconstrução estéreo sejam empregadas visando melhorar ainda mais a qualidade do mosaico final. O principal problema das técnicas de reconstrução a partir de imagens estéreo é descobrir quais pontos em cada imagem correspondem às projeções de um mesmo ponto da cena. Este problema é mais conhecido como $matching$ [Marr and Poggio 1979], sendo ele a etapa mais demorada e uma das mais estudadas em reconstrução estéreo. Determinadas as correspondências de todos os pixels das imagens, esta informação pode ser utilizada na construção do mosaico. A profundidade de cada pixel pode ser determinada em relação a um referencial fixo, por triangulação, em relação às câmeras. Esta profundidade pode ajudar a distinguir as características ou atributos de um dado pixel que aparece em mais de uma imagem. Note que, no pior caso, uma média entre os atributos pode ajudar a minimizar problemas de erros das imagens. A correspondência entre as imagens pode ser feita por área ou atributo [Marr and Poggio 1979]. Neste trabalho, a correspondência por área pode ser usada, com algumas simplificações observadas adiante. Este tipo de operação é realizado com aplicação de operadores de correlação cruzada normalizada (ou simplesmente correlação) ou então pela soma do quadrado das diferenças (SSD) [Ballard and Brown 1982]. A SSD é mais rápida de ser calculada do que a correlação, mas não é imune a variações de contraste e brilho nas imagens, problemas que não afetam a correlação cruzada normalizada, dada abaixo:

$$r_{x,y} = \frac{n\sum(x_i y_i) - \sum(x_i)\sum(y_i)}{\sqrt{n\sum(x_i^2) - (\sum x_i)^2}\sqrt{n\sum(y_i^2) - (\sum y_i)^2}}.$$

onde n é o número de amostras em cada sinal. Para o $matching$, a correlação é restrita a uma região (janela de comparação) de cada imagem, sendo n a área desta janela.

3.2. Orientação relativa

Aguns princípios de estereofotogrametria são empregadas na fase de orientação relativa dos modelos produzidos por cada par de imagens consecutivas, visando determinar as relações espaciais que o helicóptero possuia no momento de tomada de cada imagem, dada aproximadamente pelo GPS de bordo. O problema de orientação relativa é atualmente bem determinado dentro da área de fotogrametria e encontra-se formalizado em livros e artigos [Wolf 1983]. Com as simplificações, com apenas 6 pares de pontos conhecidos em cada modelo (entre cada par de imagens), uma boa precisão pode ser obtida na determinação de coeficientes de transformação que deverão retirar as distorções causadas pelo posicionamento e orientação (desconhecidos) do helicóptero.

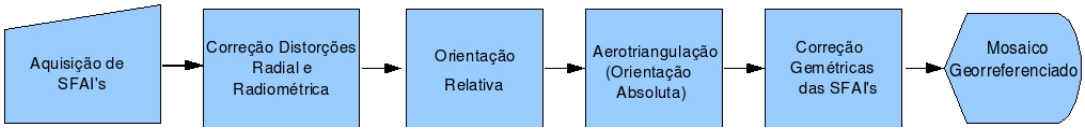
3.3. Orientação absoluta (escala e georreferenciamento)

Para o georreferenciamento em si (determinação de escala e referenciamento a um sistema de coordenadas), são determinados a priori, na região a ser imageada, pontos de controle, cujas coordenadas são determinadas por GPS. Então, usando técnicas de aerotriangulação [Wolf 1983], estas coordenadas conhecidas são estendidas para os pontos determinados no processo de orientação relativa. A partir destes, em caso de não assunção de um modelo de relevo plano, poderia-se estender a todos os outros pontos de todas as imagens,

gerando assim coordenadas de terreno, referenciadas em relação a um sistema de coordenadas, para todo o mosaico [Gonçalves]. Note que cada modelo (par de imagens) pode ser ligado ao posterior via uma das imagens que é comum a dois modelos adjacentes. Estender as coordenadas dos pontos de controle significa usar este recobrimento para extrapolar as coordenadas de uma imagem a outra. Note que um erro inerente ao processo de digitalização aparece aqui. A técnica de aerotriangulação adotada neste trabalho usa o modelo dos mínimos quadrados para minimizar estes erros no processo de determinação de coordenadas para os pontos de cada modelo. Ao final, obtém-se por um ajuste de bloco as coordenadas georreferenciadas de todos os pontos. Usando estas, pode-se determinar quais os coeficientes das transformações necessárias a serem aplicadas em cada imagem para geração do mosaico final.

4. Implementações parciais

A linguagem de desenvolvimento utilizada é C/C++, com bibliotecas do QT Designer.



Após a etapa de calibração, foi calculada a equação de mapeamento radial e radiométrica com seguinte interpolação de pixels, resultando na imagem corrigida. Para as correções geométricas oriundas da vista perspectiva, inicialmente, foi implementado um módulo, onde, a partir de duas imagens, aplicamos transformações para coincidir as medidas geométricas, preparando para fase de mosaico. Para o uso deste módulo, é preciso definir os pontos de controle nas imagens e a equação de mapeamento. Assim obteremos os coeficientes que determinarão as transformações. Estes coeficientes são determinados pelo processo de aerotriangulação, em fase de implementação. Foram implementados neste módulo a transformação afim, com a Equação 3 e transformação projetiva, com a Equação 4 com o método de interpolação bilinear. Para a geração de mosaicos, basta inserir pontos correspondentes em ambas as imagens que será realizado o mosaico.

$$\begin{bmatrix} X^* \\ Y^* \end{bmatrix} = \begin{bmatrix} a_{11}\ a_{12} \\ a_{21}\ a_{22} \end{bmatrix} * \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} \tag{3}$$

Onde a_{ij} são parâmetros correspondentes a dois fatores de escala, um de rotação, e um de não ortogonalidade(cisalhamento), com i e j= 1 ou 2; X e Y são coordenadas a ser transformadas no espaço; X^* e Y^* são coordenadas no espaço transformado; X_0 e Y_0 são parâmetros de translação na direção X e Y [Nogueira].

$$X^* = \left[\frac{a_{11}X+a_{12}Y+a_{13}}{a_{31}X+a_{32}Y+1}\right] Y^* = \left[\frac{a_{21}X+a_{22}Y+a_{23}}{a_{31}X+a_{32}Y+1}\right] \tag{4}$$

Onde a_{ij} são os parâmetros das tranformações geométricas, com i e j= 1, 2 ou 3; X e Y são os valores medidos no sistema de referência; X* e Y* são os valores calculados para o sistema de ajuste [Nogueira].

5. Resultados Parciais

Experimentos parciais foram realizados, visando testar os módulos implementados até o momento. A Figura 1 mostra o resultado do processo de calibração de Tsai implementado com correções radial e radiométrica. A Figura 2 mostra a aplicação de transformações afim e projetiva, às imagens. Estas transformações serão aplicadas na construção do mosaico, com coeficientes fornecidos pelo processo de orientação relativa e absoluta, descritos no texto, não implementados ainda. As duas transformações são importantes e tem em comum modificar a posição, escala e forma, no entanto a transformação projetiva se mostrou mais eficaz em algumas ocasiões comparada a tranformação afim por afetar também o paralelismo. As Figuras 3 e 4 expõem os mosaicos gerados sem e com correção de iluminação nas áreas de recobrimento, ainda sem a determinação precisa dos coeficientes de transformação, usando a metodologia a ser implementada aplicando orientação relativa e absoluta. Nesta etapa foram obtidas imagens com a câmera do helicoptero, adquiridas em terra, por ora e se mostrou ideal após corrigir as distorções radial, radiométrica, geométrica e de iluminação nas áreas de recobrimento do mosaico.

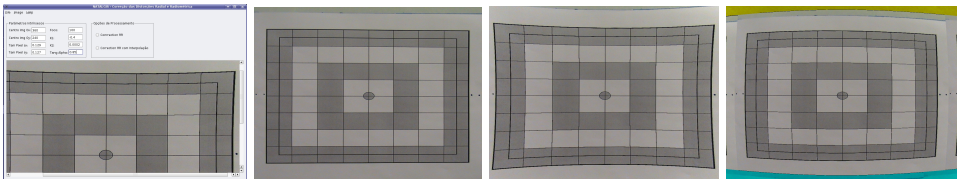


Figure 1. Módulo de Correção Radial e Radiométrica / Imagem Original / Imagem Distorcida "Pincushion" K1=-3.5 / Imagem Distorcida "Barrel" K1=+4.5

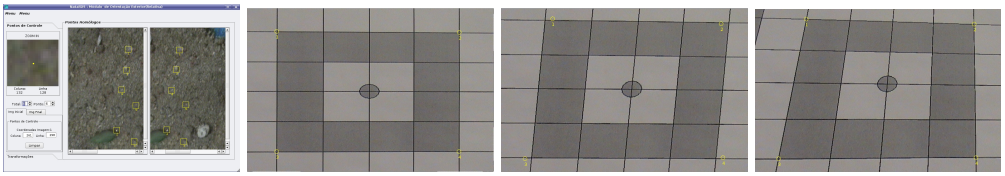


Figure 2. Módulo de Correção Geométrica e Mosaico / Imagem Inicial / Imagem após transfomação Afim/ Imagem após Transformação Projetiva

Figure 3. Mosaico de 9 imagens sem correção de iluminação

Figure 4. Mosaico de 10 imagens com correção de iluminação em algumas áreas

6. Conclusão

Propomos neste trabalho o desenvolvimento de um sistema completo, que visa gerar mosaicos georreferenciados por meio das técnicas de estéreo-fotogrametria, usando imagens aéreas de pequeno formato obtidas por uma filmadora digital comum acoplada a um helicóptero aeromodelo. A contribuição principal do presente trabalho reside no fato deste tipo de imagens ter sido ainda muito pouco explorada na geração de mosaicos georreferenciados, talvez em função do uso de imagens de grande formato e controladas ser geralmente a técnica adotada nos projetos de cartografia. Note que o uso deste último tipo de imagem torna o projeto de monitoramento constante caro. Projetos como o nosso são essenciais a regiões costeiras, implicando em sobrevôos rotineiros visando checar determinadas características. Com a implementação parcial destas técnicas, mostramos ser possível desenvolver uma metodologia a baixo custo. Estas técnicas se mostram essenciais na busca de dados mensuráveis confiáveis de modo que possam gerar um monitoramento de áreas com maiores detalhes, no qual, satélites não operam ao mesmo nível de escala próxima ao terreno e nas quais o uso de fotogrametria aérea profissional se torna caro. Já foram estudadas e principalmente formalizadas todas as técnicas necessárias, estando na fase final de implementação. Os próximos passos são o sobrevôo da região dos Parrachos de Maracajaú, com o helicóptero já adquirido (vôos experimentais já estão sendo realizados em terra). Depois de adquiridos os dados, estes servirão de base para a construção do mosaico e consequente alimentação do SIG de monitoramento ambiental.

References

Albrecht, P. and Michaelis, B. (1998). Stereo photogrammetry with improved spatial resolution. In *ICPR '98: PROC-Volume 1*, page 845, Washington, DC, USA. IEEE Computer Society.

Ballard, D. and Brown, C. (1982). *Computer Vision*. Pratice-Hall, Englewood Cliffs, New Jersey.

Gonçalves, L. *Reconstrução a partir de estéreo fotogrametria - UFRJ - 1995*. Rio de Janeiro - BRASIL.

Grandi, G., Mayaux, P., Rauste, Y., Rosenqvist, A., Simard, M., , and Saatchi, S. (2000). The global rain forest mapping project jers-1 radar mosaic of tropical africa. *IEEE Transactions On Geoscience and Remote Sensing*.

Hsu, S. (2001). Geocoded terrestrial mosaics using pose sensors and videos registrations. In *ICCV. PROC*, pages 834–841. IEEE Computer Society.

Lhuillier, M., Quan, L., Shum, H., and Tsui, H. (2001). Relief mosaics by joint view triangulation. In *ICCV - PROC*, pages 785–790. IEEE Computer Society, USA.

Marr, D. and Poggio, T. (1979). A computational theory of human stereo vision. In *PROC*, volume 204, pages 301–328. Royal Society Publishing.

Nogueira, F. Geração automática de mapas de disparidade em visão estéreo - UNICAMP - 1998. Master's thesis.

Trucco, E. and Verri, A. (1998). *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall, Upper Saddle River, New Jersey.

Wolf, P. (1983). *Elements of Photogrammetry*. McGraw-Hill Book Company, Singapure.

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 215-220.

Análise da complexidade de texturas em imagens urbanas utilizando dimensão fractal

André R. Backes[1], Adriana B. Bruno[1], Mauro N. Barros Filho[2], Odemir M. Bruno[1]

[1]Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
Caixa Postal 667 – 13.560-970 – São Carlos – SP – Brazil

[2]Departamento de Arquitetura e Urbanismo
Faculdade de Ciências Humanas ESUDA
Rua Bispo Cardoso Ayres, S/N – 50.050-090 – Recife – PE – Brazil

backes@icmc.usp.br, adriana@arstechnica.com.br,

bruno@icmc.usp.br, mbarrosfilho@gmail.com

Abstract. *This paper presents a study of the correlation between Fractal Dimension and urban morphological patterns. In remote sensing images, the urban morphological patterns are represented by complex interations of different surfaces, and each surface corresponds to a kind of texture. In this work, Fractal Dimension measures are applied to estimate the texture in remote sensing images from some urban areas, in order to permit a classification of their morphological patterns.*

Resumo. *Este artigo apresenta um estudo sobre a correlação entre Dimensão Fractal e os padrões morfológicos urbanos. Em imagens de sensoriamento remoto, os padrões morfológicos urbanos são representados por complexas interações de diferentes superfícies, e cada superfície corresponde a um tipo de textura. Neste trabalho é realizada a comparação de duas metodologias para a estimativa da Dimensão Fractal em texturas de imagens de sensoriamento remoto, de modo a verificar qual abordagem melhor se adequa à análise urbana.*

1. Introdução

A morfologia urbana é resultante de um complexo arranjo espacial de edificações, lotes, quadras e vias. Esse arranjo varia em função das características físico-ambientais e sócio-econômicas existentes na cidade, e da dinâmica do processo de uso e ocupação do solo urbano. Diante desta complexidade, a análise da morfologia urbana tem sido, predominantemente, conduzida de modo subjetivo, sendo incapaz de oferecer medidas quantitativas que possam descrevê-la, de modo mais preciso.

Com os recentes avanços tecnológicos, imagens de sensoriamento remoto têm sido cada vez mais utilizadas em mapeamentos e estudos urbanos. Imagens de satélite apresentam ampla cobertura, atualizações freqüentes e baixo custo, sendo uma rica fonte de informações para a análise da morfologia urbana. Porém, imagens de áreas urbanas são resultantes de uma complexa interação entre diferentes superfícies, dificultando a análise de padrões morfológicos urbanos com valores diversos de reflectância espectral.

Assim, o uso de informações texturiais pode auxiliar no aprimoramento da classificação de imagens.

Texturas são padrões visuais caracterizados pela repetição, seja exata ou com pequenas variações, de entidades ou sub-padrões, que representam características físicas, como brilho e cor, da superfície de um objeto [Ebert et al. 1994, Rosenfeld and Kak 1982]. Texturas descrevem uma grande quantidade de informações sobre uma imagem [Tuceryan and Jain 1993, Julesz 1975], sendo muito utilizadas no reconhecimento e classificação de padrões espaciais [Tuceryan and Jain 1993]. No entanto, apesar de seu amplo uso e importância, a textura é um termo intuitivo, e que carece de uma definição mais precisa ou formal [Ebert et al. 1994, Emerson et al. 1999].

Em processamento de imagens e visão computacional, existem diversas abordagens para a análise de texturas, como: Campo Aleatório de Markov [Li 1995, Giordana and Pieczynski 1997], Redes Neurais [Teke and Atalay 2006], Wavelet [Unser 1995], momentos Invariantes [Gonzales and Woods 1992] e a Dimensão Fractal [Chen and Bi 1999, Kaplan 1999, Chaudhuri and Sarkar 1995].

Neste trabalho, é utilizada a Dimensão Fractal [Chaudhuri and Sarkar 1995, Coelho and COSTA 1995] para estimar o grau de complexidade da textura de imagens de satélite obtidas pelo Google Earth de diferentes áreas urbanas da cidade de São Carlos (SP). Será realizada uma análise comparativa entre duas abordagens para estimar a dimensão fractal: uma aplicada em imagens binárias e outra em níveis de cinza, verificando as vantagens e desvantagens dos métodos. A seguir, o método e sua implementação são discutidos.

2. Metodologia

A metodologia proposta está dividida em 2 partes. A primeira consiste na seleção de imagens. A segunda envolve a aplicação de dois algoritmos para estimar a dimensão fractal. Essas partes são detalhadas a seguir.

2.1. Seleção das imagens

As imagens das áreas urbanas de São Carlos foram selecionadas em função de dois critérios: (i) distância ao centro da cidade; e (ii) altitude de observação. O primeiro critério foi definido devido à forte correlação entre padrões morfológicos urbanos e a densidade construtiva. As áreas periféricas estão sujeitas a um processo de ocupação mais rarefeito que as áreas centrais da cidade. O segundo critério foi estabelecido em razão das imagens obtidas em diferentes altitudes apresentarem distintos níveis de detalhamento e, conseqüentemente, de complexidade. Neste trabalho foi utilizada a altitude de 15.000 pés, considerando a comparação de diferentes altitudes realizada por Backes et al. [Backes et al. 2007].

Inicialmente, a cidade de São Carlos (SP) foi dividida em 5 anéis concêntricos, delimitando regiões a partir do marco central da cidade de São Carlos (Praça Praça Dom José Marcondes Homem de Mello). Em seguida foram selecionadas duas amostras de imagens de 200x200 píxeis contidas em cada anel. As regiões utilizadas para extrair as imagens está apresentada na Figura 1.

2.2. Estimativa da Dimensão Fractal

A Dimensão Fractal pode ser definida como uma medida da complexidade de objetos. Aplicada a texturas, ela permite quantificar a complexidade da organização de seus pixels, onde este nível de complexidade está diretamente relacionado com o aspecto visual, bem como, com a homogeneidade da textura. Assim, a Dimensão Fractal permite quantificar uma textura em termos de homogeneidade, possibilitando sua comparação com outras texturas [Chaudhuri and Sarkar 1995].

Neste trabalho foram utilizados duas abordagens para aferir a dimensão fractal de imagens: dimensão fractal de bouligand-minkowiski aplicada em imagens binárias [Plotze et al. 2005, Tricot 1995] e BoxCounting volumétrico, aplicado em imagens de nível de cinza [Backes and Bruno 2006, Backes et al. 2007]. A diferença fundamental entre as duas abordagens é considerar a profundidade das imagens, ou seja considerar as informação das imagens binárias ou as imagens de níveis de cinza. A estimativa da dimensão fractal em imagens binárias é uma abordagem clássica da literatura, apresentando uma série de métodos. Na primeira abordagem, foi adotado o bouligand-minkowiski (considerado por Tricot como o mais preciso [Tricot 1995]) na sua versão multiescala [Plotze et al. 2005]. E na segunda abordagem foi utilizado a versão multiescala do algoritmo de Boxcounting adaptado para níveis de cinza, sendo o seguinte conjunto de caixas considerado: $\{1, 14, 27, 40, 53\}$.

3. Resultados

Conforme já comentado, a diferença fundamental entre os métodos analisados é a profundidade das imagens. A Figura 2 apresenta as imagens em níveis de cinza e binárias, com um exemplo de cada classe utilizada no experimento, note que as letras são correspondentes as regiões da Figura 1. As imagens binárias foram obtidas por meio do método de binarização ou limiarização [Gonzales and Woods 1992], que consiste em discretizar os níveis de cinza em duas classes. Foi utilizado o método de binarização automático de Otsu [Otsu 1979].

Observando as imagens de níveis de cinza e binárias, pode ser notado que as imagens binárias não contemplam toda a riqueza de detalhes presentes nos níveis de cinza.

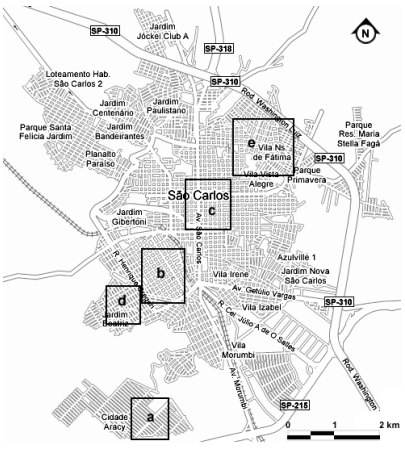


Figura 1. Mapa da cidade e respectivas localizações das imagens das áreas usadas no experimento.

Em algumas regiões (Figura 2(a)) ocorre a presença de regiões homogêneas na imagem binária, na qual o seu correspondente nas imagens de níveis de cinza são heterogeneidade e complexidade. Deste modo, como é de se esperar, uma técnica que explore diretamente as imagens em níveis de cinza deverá obter uma melhor performance.

Os resultados preliminares obtidos com os dois algoritmos utilizados para estimar a dimensão fractal das imagens selecionadas são apresentados nas Figura 3(a) e Figura 3(b). Uma vez que o gráfico da figura da esquerda é relacionada a uma imagem binária, ou seja um plano, ele apresenta uma variação da dimensão fractal entre 1 e 2 (Espaço 2D). Já o da direita, apresenta uma variação entre 2 e 3, relativa ao espaço 3D por se tratar de uma técnica volumétrica [Backes and Bruno 2006].

Como pode ser observado, o método de estimativa de minkowski em imagens binárias, não conseguiu separar adequadamente as classes do experimento, conforme exibe o gráfico da Figura 3(a). Note ainda, neste gráfico, que existe apenas uma curva diferenciada, ela é correlacionada a uma imagem que apresenta uma grande região homogênea na imagem binária, não correspondente a imagem original. Como mostra os gráficos da Figura 3(b) o método de BoxCounting, conseguiu separar adequadamente as classes de imagens do experimento. Observe que todos os pares correspondentes apresentam o mesmo tipo de padrão no gráfico.

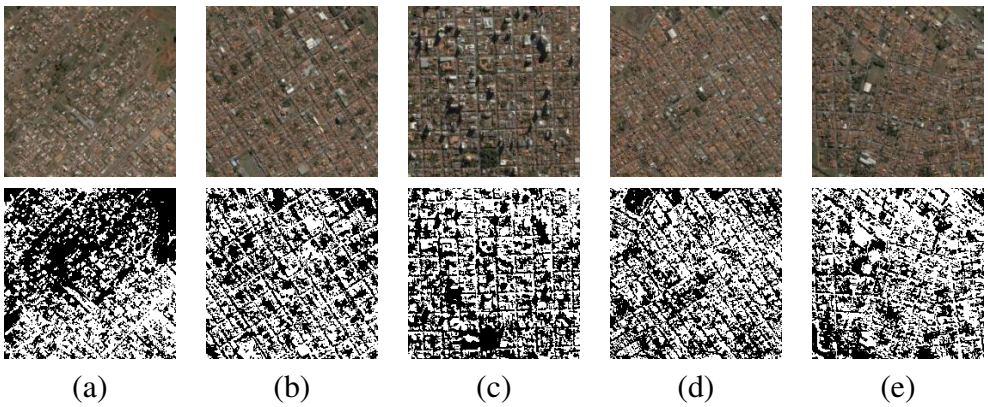


Figura 2. Imagens de satélite de diferentes áreas obtidas a 15000 pés de altitude. Parte superior, imagens em niveis de cinza. Parte inferior, imagens binárias

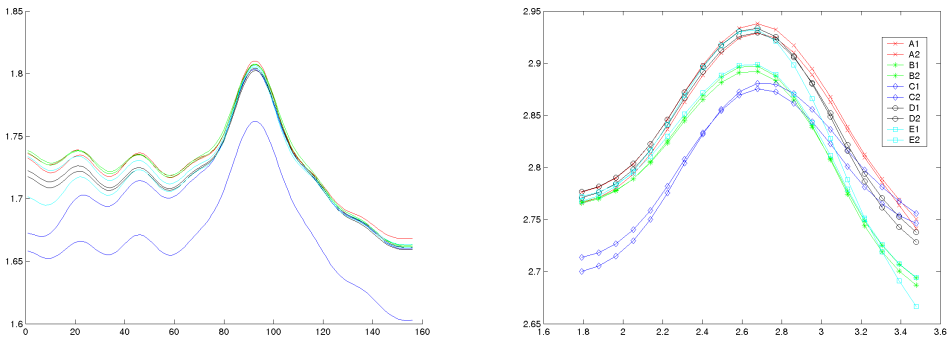


Figura 3. Curvas de dimensão fractal. (a) Minkowiski e (b) Boxcounting volumétrico

A Figura 4 apresenta o mapa da cidade de S. Carlos, com anéis concêntricos e suas respectivas dimensões fractais aferidas pelo método BoxCounting volumétrico. Percebe-se que, à medida que se afasta do centro da cidade, o valor da Dimensão Fractal aumenta. Esse aumento da complexidade indica uma maior heterogeneidade dessas áreas, ou seja, a organização das estruturas morfológicas nessas regiões apresentam um padrão mais caótico, menos regular ou homogêneo. Nota-se também que as áreas vizinhas ou que estejam a uma distância aproximadamente igual do centro da cidade apresentam valores de complexidade parecidos, logo a organização de suas estruturas morfológicas é semelhante. Isso é corroborado pelo fato de áreas centrais das cidades serem alvo de maior número de benfeitorias, portanto melhor estruturadas, e de não sofrerem de processos de ocupação espontâneos ou informal.

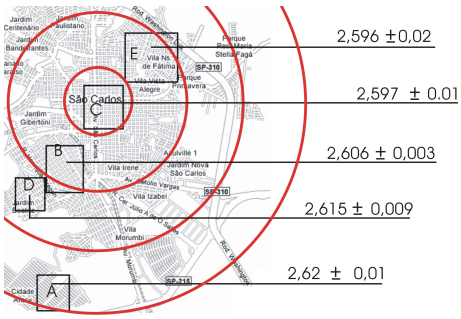


Figura 4. Anéis concêntricos, apresentando as regiões de mesma distância do marco central da cidade e sua Dimensão Fractal.

4. Conclusão

Neste trabalho foi apresentado um estudo sobre a utilização do método de estimativa de Dimensão Fractal na análise de características morfológicas de imagens de satélite de áreas urbanas, cuja interação resulta em padrões de texturas complexos. Por meio da Dimensão Fractal é possível quantificar a complexidade dessa textura e, conseqüentemente, estimar o nível de desenvolvimento urbano de uma determinada área, permitindo a sua comparação com demais regiões de uma mesma cidade. Os resultados demonstraram que existe correlação entre o nível de complexidade e os padrões morfológicos das áreas urbanas analisadas, evidenciando o grande potencial dessa técnica na descrição e classificação de padrões morfológicos urbanos, sendo uma ferramenta valiosa para auxiliar no planejamento e gestão de cidades.

5. Agradecimentos

Odemir M. Bruno agradece ao CNPq (Procs. #303746/2004-1 e #504476/2007-6) e a FAPESP (Proc. #06/54367-9). André R. Backes agradece a FAPESP (Proc. #06/54367-9) pelo apoio financeiro ao doutorado. Adriana B.Bruno agradece ao CNPq (Proc. #110763/2007-6) pelo apoio financeiro à iniciação científica. Mauro Barros Filho agradece à Faculdade de Ciências Humanas ESUDA.

Referências

Backes, A. R., Bruno, A. B., Filho, M. N. B., and Bruno, O. M. (2007). Dimensão fractal aplicada em imagens de satélite de áreas urbanas (to appear). *III Workshop de Visão Computacional*.

Backes, A. R. and Bruno, O. M. (2006). Segmentação de texturas por análise de complexidade. *INFOCOMP Journal of Computer Science*, 5(1):87–95.

Chaudhuri, B. B. and Sarkar, N. (1995). Texture segmentation using fractal dimension. *IEEE Trans. Pattern Anal. Mach. Intell*, 17(1).

Chen, Y. Q. and Bi, G. (1999). On texture classification using fractal dimension. *IJPRAI*, 13(6):929–943.

Coelho, R. C. and COSTA, L. F. (1995). The box-counting fractal. dimension: Does it provide an accurate subsidy for experimental shape characterization? if so, how to use it? In *Anais do Sibgrapi 95*, pages 183–191.

Ebert, D., Musgrave, K., Peachey, D., Perlin, K., and Worley (1994). *Texturing and Modeling: A Procedural Approach*. Academic Press.

Emerson, C. W., Lam, N. N., and Quattrochi, D. A. (1999). Multi-scale fractal analysis of image texture and patterns. *Photogrammetric Engineering and Remote Sensing*, 65(1):51–62.

Giordana, N. and Pieczynski, W. (1997). Estimation of generalized multisensor hidden markov chains and unsupervised image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, (5):465–475.

Gonzales, R. C. and Woods, R. (1992). *Digital Image Processing*. Addison Wesley.

Julesz, B. (1975). Experiments in the visual perception of texture. *Scientific American*, 232(4):34–43.

Kaplan, L. M. (1999). Extended fractal analysis for texture classification and segmentation. *IEEE Transactions on Image Processing*, 8(11):1572–1585.

Li, S. Z. (1995). *Markov Random Field Modeling in Computer Vision*. Springer-Verlag.

Otsu, N. (1979). A threshold selection method from grey-level histograms. *IEEE Trans. on Systems, Man, and Cybernetics*, 9(1):62–66.

Plotze, R. O., Falvo, M., Pádua, J. G., Bernacci, L. C., Vieira, M. L. C., Oliveira, G. C. X., and Bruno, O. M. (2005). Leaf shape analysis using the multiscale minkowski fractal dimension, a new morphometric method: a study with passiflora (passifloraceae). *Canadian Jornal of Botany*, 83(3):287–301.

Rosenfeld, A. and Kak (1982). *Digital Picture Processing Vol. 2*. Academic Press, Orlando.

Teke, A. and Atalay, V. (2006). Texture classification and retrieval using the random neural network model.

Tricot, C. (1995). *Curves and Fractal Dimension*. Springer-Verlag, New York.

Tuceryan, M. and Jain, A. K. (1993). Texture analysis. *Handbook of Pattern Recognition and Computer Vision*, pages 235–276.

Unser, M. (1995). Texture classification and segmentation using wavelet frames. *IEEE Trans. Image Processing*, 4(11):1549–1560.

Utilização de imagens de sensoriamento remoto de alta resolução para realizar a contagem de copas em povoamento de *Eucalyptus spp*.

Frederico Pereira Reis[1], Luciano T. de Oliveira[2], Luis Marcelo T. de Carvalho[3]

[1,2,3]Universidade Federal de Lavras, MG (UFLA)
Campus Universitário - Caixa Postal 3037 – 37200-000 - Lavras MG – Brasil

[1,2,3]LEMAF - Departamento de Ciências Florestais - UFLA

fredreis1@gmail.com, oliveiralt@yahoo.com.br, passarinho@ufla.br

Abstract. *The present work aims at performing automatic tree crown counting in planted Eucalypt forests. High resolution remote sensing imagery and digital image processing using Lee Filters and the unsupervised classification algorithm ISODATA were used. The classification result that best extracted tree crowns was exported to a GIS for tree counting. The resulting number of trees was compared to the number obtained by visual interpretation.*

Resumo. *O presente trabalho objetiva, através da utilização de imagens de sensoriamento remoto de alta resolução e processamento digital de imagens, realizar a contagem automática de copas individuais em um povoamento plantado de Eucalyptus spp., através da técnica que associa o filtro de Lee com o classificador não-supervisionado ISODATA. Após as etapas de processamento, o resultado da classificação que mais representou as copas das árvores foi exportado para ambiente SIG (Sistema de Informações Geográficas), onde foi realizada a contagem automática dos polígonos que representaram melhor as copas. Os resultados foram comparados com o parâmetro, que foi obtido através da interpretação visual da imagem.*

1. Introdução

Imagens obtidas por satélites, aviões, ou sensores digitais aerotransportados fazem cada vez mais, parte de nosso cotidiano, sendo utilizadas por cientistas e profissionais para monitorar o que se passa na superfície de nosso planeta de uma maneira abrangente, periódica e de baixo custo.

O planejamento econômico das florestas plantadas serve de subsidio para tomada de decisões para o abastecimento das empresas consumidoras de madeira, como as siderúrgicas, papel e celulose, dentre outras. Convencionalmente, os dados de inventários florestais têm sido coletados principalmente por pesquisas de campo, que são dispendiosas e demoradas (Hyyppä et al., 2000), e muita das vezes com erros devido a falta de treinamento da equipe de campo.

Diferentes métodos para a delineação de copas individuais têm sido apresentados e diferentes técnicas têm sido usadas, a fim de diminuir o tempo gasto e reduzir custos. Essas técnicas incluem "*template matching*", que são técnicas que

procuram correlacionar pequenas partes da imagem com uma imagem modelo (Pollock, 1996; Larsen & Rudemo, 1998; Olofsson, 2002), métodos baseados em regiões de crescimento (Pinz, 1989; Culvenor, 2002; Pouliot, King, Bell, & Pitt, 2002; Erikson, 2003; Erikson, 2004; Erikson, 2006), métodos probabilísticos (Descombes & Pechersky, 2006; Perrin, Descombes, Zerubia, & Boureau, 2006), e métodos baseados no contorno dos alvos (Gougeon, 1995; Brandtberg & Walter, 1998).

Frente a isso, este estudo tem como objetivo principal, realizar a individualização de árvores em um povoamento florestal plantado, através de técnicas de processamento digital de imagens, a fim de gerar melhores parâmetros dendrométricos.

2. Material e Métodos

O trabalho foi conduzido no projeto FX39, talhão 02, pertencentes a uma fazenda (Figura 01) de área 637 hectares de *Eucalyptus spp*. com 7 anos de idade, plantados no espaçamentos 3x3, denominada "Projeto Cara Preta".

O projeto Cara Preta está situado a 21°35' de Latitude Sul e 47°35' de Longitude Oeste, em altitude variando de 600 a 700 m, no município de Santa Rita do Passa Quatro, SP. O clima da região é do tipo CWa, segundo o sistema de Köppen, com predominância de chuvas no verão e inverno relativamente seco.

A escala da foto é de 1:6000, e o vôo foi realizado no ano de 2004.

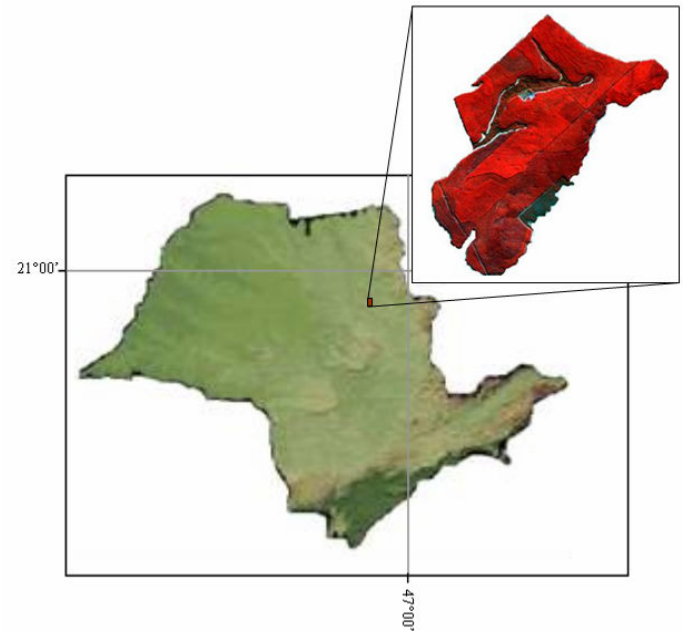


Figura 1. Localização da área de estudo.

Para realização da contagem das árvores individuais, foram seguidos os seguintes passos:

1) Primeiramente a imagem obtida do sobrevôo foi digitalizada e em seguida foi realizada a contagem dos indivíduos do talhão por meio de processamento visual de imagens. Esses dados foram considerados como parâmetro nesse estudo.

2) Os próximos passos envolveram as etapas de pré-processamento de imagens digitais. Inicialmente optamos pela filtragem dos dados, onde utilizamos o Filtro de *Lee* (Lee & Jong-Sen, 1981), que adota um modelo multiplicativo para o ruído e obedece ao critério de "*local linear minimum mean square error*". Local, porque utiliza estatísticas locais do *pixel* a ser filtrado, admitindo a não estacionaridade da média e da variância do sinal. É um filtro linear porque realiza uma linearização por expansão em série de Taylor da multiplicação do sinal e o ruído em torno da média, utilizando apenas os termos lineares. O resultado da linearização transforma o modelo multiplicativo do ruído em aditivo, ou seja, o ruído e o sinal tornam-se independentes; e, finalmente, "*minimum mean square error*", porque minimiza o erro médio quadrático através do filtro de *Wiener* (filtro baseado no critério de mínimo erro médio quadrático).

3) Assim, com os dados melhorados, foi feita uma classificação não-supervisionada *Isodata*. O classificador *Isodata* é um algoritmo de classificação que identifica padrões típicos nos níveis de cinza. Esses padrões são classificados efetuando-se visitas de reconhecimento a alguns exemplos escolhidos para determinar sua interpretação. Os padrões são geralmente referidos como agrupamentos ou nuvens (*clusters*) em decorrência da técnica adotada. Dessa forma, as classes são determinadas pelas análises de agrupamentos (Schowengerdt, 1997; Gonzáles e Woods, 2000). Neste método a imagem é sucessivamente varrida e os agrupamentos dos *pixels* (*clusters*), vão sendo alterados, ocorrendo à agregação de novos *pixels*, divisão ou fusão de *clusters*. Esta classificação consiste na identificação dos níveis de cinza, onde os *pixels* analisados são submetidos a algoritmos de agrupamento, formando assim os agregados de dados.

No processo de classificação o algoritmo agrupa os *pixels* em diferentes classes espectrais de acordo com alguns critérios estatísticos pré-determinados. Uma primeira suposição é assumida para o centro de cada agrupamento. A distância euclidiana entre cada *pixel* e o centro dos agrupamentos é calculada. (Tso & Mather, 2001). Através de um processo iterativo o algoritmo vai mudando o centro dos agrupamentos até que as distâncias mínimas até o centro do agrupamento formam uma classe.

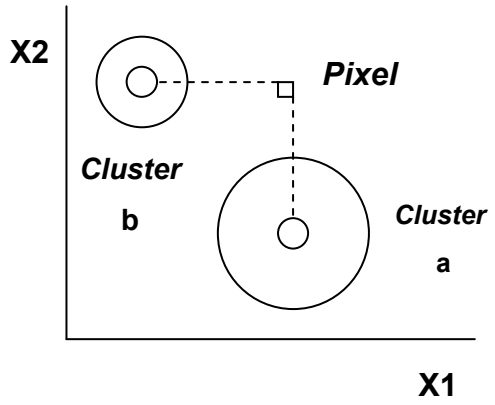


Figura 02. *Cluster* circular usando a medida de distancia euclidiana.

O cálculo da distancia entre um *pixel* e o centro do *cluster* normalmente usa a medida de distância euclidiana $D^2_E = (X_i - M_j)^2$, onde:

X_i: Vetor observado no i-ésimo *pixel*

M_j: Vetor médio do j-ésimo *cluster*

A dimensão do vetor X_i é igual ao número de bandas usadas na imagem.

4) O classificador gerou 30 classes. Daí foi escolhida a classe que melhor representava a copa das árvores, e exportada na forma de polígonos para ambiente SIG (Sistema de Informações Geográficas), para que fosse feita a contagem desses polígonos automaticamente. De posse dos dados da contagem automática, foi feita uma comparação com a contagem manual (parâmetro) e foram gerados os resultados.

(a) (b)

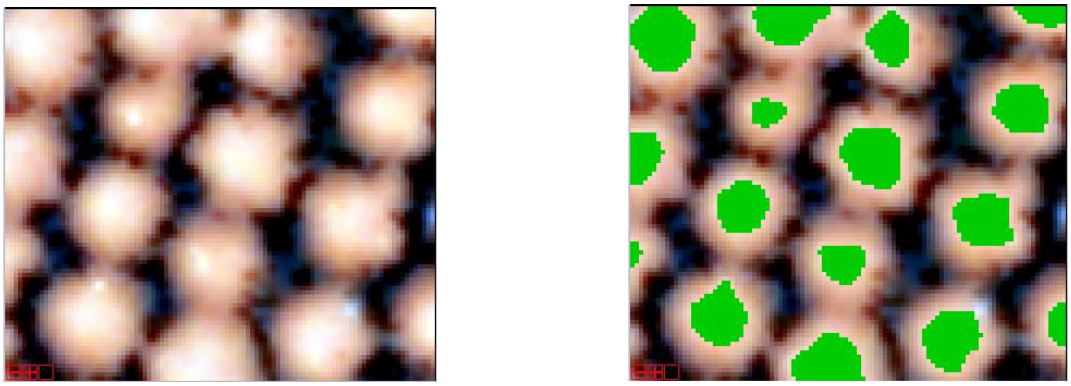


Figura 03. (a) Imagem original, mostrando a copa das árvores e (b) mostrando em verde os polígonos que melhor representaram a copa das árvores.

3. Resultados

Pelo método de processamento visual e contagem individual de árvores, foram encontradas 9.494 árvores.

Os polígonos dessa classe foram contados automaticamente, e chegou-se ao número de 8.884 árvores, atingindo um acerto de 93,58%, comparados com nosso parâmetro.

O número final pode ser considerado muito bom, bastante próximo da verdade de campo, mas alguns erros implícitos devem ser considerados, quando se leva em conta a acurácia do método, como os erros de omissão (árvores que deixaram de ser contadas) e erros de comissão (marcação de árvores não existentes, ou marcadas erroneamente ou de múltiplos picos de radiância na copa individual da árvore).

As próximas figuras ilustram melhor o que são os erros de omissão que foram encontrados.

(a) (b)

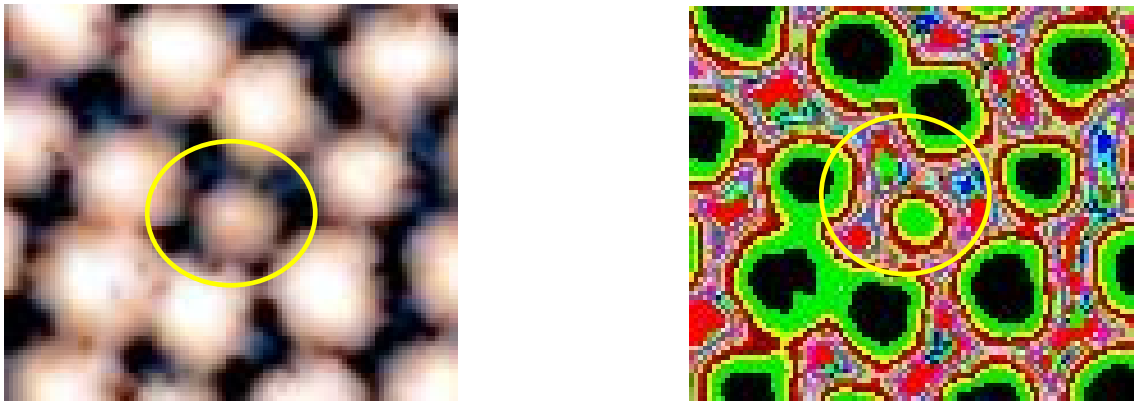


Figura 04. (a) Imagem original, mostrando as copas das árvores e (b) mostrando a imagem classificada e um dos erros de omissão encontrados.

A próxima figura ilustra melhor o que é o erro de comissão.

(a) (b)

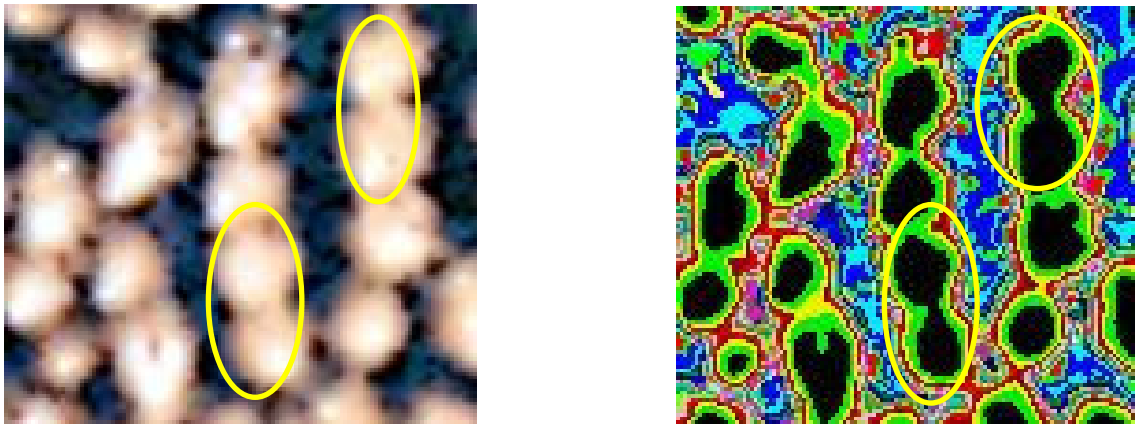


Figura 05. (a) Imagem original, mostrando as copas das árvores e (b) mostrando a imagem classificada com alguns erros de comissão encontrados.

4. Conclusões

Segundo os resultados obtidos, podemos concluir que sem levar em consideração os erros de comissão e erros de omissão, o resultado encontrado foi bastante satisfatório, visto que não foi necessário fazer a quantificação dos erros, pois o intuito do trabalho é realizar a contagem de árvores.

Podemos concluir também que há, em quase todos os trabalhos estudados (Korpela, 2006; Gougeon. 1995; Eriksson 2003; Eriksson, 2004 e Eriksson 2006), certa tendência à subestimação dos resultados.

A utilização do filtro de *Lee*, seguido pela classificação *Isodata* é uma técnica bastante promissora, mas precisa ser testada para plantios com diferentes espaçamentos, diferentes idades e terrenos com diferentes inclinações.

5. Bibliografia

ERIKSON, M. (2003). "*Segmentation of individual tree crowns in colour aerial photographs using region growing supported by fuzzy rules"*. *Canadian Journal of Forest Research* 33(8), 1557–1563.

ERIKSON, M. (2004). "*Species classification of individually segmented tree crowns in high-resolution aerial images using radiometric and morphologic image measures*". *Remote Sensing of Environment* 91(3), 469–477.

ERIKSON, M. (2006). "*Two preprocessing techniques based on grey level and geometric thickness to improve segmentation results"*. *Pattern Recognition Letters* 27(3), 160–166.

GOUGEON F. A. (1995). "*A crown-following approach to the automatic delineation of individual tree crowns in high spatial resolution aerial images"*. *Canadian journal of remote sensing*. Vol 21. pp 274-284. 1995

GONZALEZ, R. C.; WOODS, R. E**.** (2000) "Processamento de Imagens Digitais". São Paulo: Editora *Edgard Blücher* LTDA, 509 p.

HYYPPÄ, J.; HYYPPÄ, H.; INKINEN, M.; ENGDAHL, M.; LINKO, S.; ZHU, Y. (2000). " *Accuracy comparison of various remote sensing data sources in the retrieval of forest stand attributes*". *Canadian journal of remote sensing*, v. 128, p. 109-120.

KORPELA, I.; ANTILLA, P.; PITÄKEN, J. (2006). "*The performance of a local maxima method for detecting individual tree tops in aerial photographs*". *International Journal of Remote Sensing*. Vol 27, N°6, 20, 1159-1175.

LARSEN, M. AND RUDEMO, M., (1998). "*Optimizing templates for finding trees in aerial photographs"*. *Pattern Recognition Letters* 19, 1153–1162.

OLOFSSON, K., (2002). "*Detection of single trees in aerial images using template matching*". *In ForestSat* 2002, *Operational tools in forestry usingremote sensing techniques., Edinburgh, Scotland*.

PINZ, A., (1989). "*Final results of the vision expert system VES: Finding trees in aerial photographs"*, pp. 90–111. *OCG-Schriftenreiche 49, Oldenbourg Verlag*.

POLLOCK, R., (1996). "*The Automatic Recognition of Individual Trees in Aerial Images of Forests Based on a Synthetic Tree Crown Image Model*". *Ph. D. thesis, University of British Colombia, Vancouver, Canada.*

SCHOWENGERDT, R. A. (1997) "*Remote Sensing, Models and Methods for Image Processing*". *San Diego: Academic Press*, 522 p.

TSO, B.; MATHER, P.M. (2001). "*Classification Methods for remotely sensed data*". *New York: Taylor & Francis*, 332 p.

Integração do SGBD Oracle Spatial e do Google Earth para disponibilizar informações relacionadas ao Inventário Florestal de Minas Gerais

Samuel R. de Sales Campos[1], Adriana Z. Martinhago[1], Thomas C. de A. Oliveira[1], Luca Araújo Egas Prieto[1], Ronaldo Aparecido da Silva[1], Aleksander Maduro França[1], Ivayr Dieb Farah Netto[1]

[1]Departamento de Engenharia Florestal – Universidade Federal de Lavras (UFLA)
Caixa Postal 3037 – 37.200-000 – Lavras – MG – Brasil

samuelcampos@ufla.br, dricazm@gmail.com, thomaz@vialavras.com.br,
{luca, ronaldo, alex, netto}@ufla.br

Abstract. *The technologies of database management systems and their related spatial extension are in constant advance and are indispensable tools in the research area of GIS. This work presents the integration of Oracle Spatial with Google Earth in order to create a WWW application that was created to disseminate information provided from the project of the inventory of the forests from the Minas Gerais state of Brazil.*

Resumo. *O assunto Sistemas Gerenciadores de Banco de Dados e suas extensões espaciais estão em constante avanço e estão se tornando indispensáveis na área de Sistemas de Informações Geográficas. Este artigo apresenta o desenvolvimento de um aplicativo WEB para dados geográficos que visa integrar tecnologias como Oracle Spacial e Google Earth para disponibilizar informações relacionadas ao Inventário Florestal do Estado de Minas Gerais.*

1. Introdução

Atualmente, observa-se um grande crescimento da inclusão de técnicas para tratamento de dados geográficos nos sistemas de informação. Estes sistemas são denominados Sistemas de Informação Geográfica (SIG) e segundo [Câmara 1996] são sistemas automatizados usados para armazenar, analisar e manipular dados geográficos, ou seja, dados que representam objetos e fenômenos em que a localização geográfica é uma característica inerente à informação e indispensável para analisá-la.

Estes sistemas começaram a ser desenvolvidos no início das décadas de 80 e 90 como simples sistemas *stand-alone*. Estes não tinham a capacidade de compartilhar ou gerenciar dados de forma eficiente, isto porque foram construídos com centenas de funções e constituídos de pacotes genéricos, dificultando muito sua utilização por pessoas leigas [Ferreira 2003].

Banco de Dados Espaciais ou Geográficos foram incorporados aos SIGs para tratar estas deficiências e com o intuito de armazenar e gerenciar este tipo de informação, fornecendo suporte a consultas e diversas estruturas de índices e manipulações de dados espaciais.

Depois da evolução de Banco de Dados espacias, a Web se tornou uma das mídias mais importantes e preferidas para disseminação de informações geográficas, pois evoluiu de simples páginas estáticas para páginas com conteúdo dinâmico extraídos principalmente de Sistemas Gerenciadores de Banco de Dados – SGBD.

O objetivo deste trabalho é apresentar a integração entre o SGBD Oracle Spatial e o Google Earth no desenvolvimento de uma aplicação Webmapping para disponibilizar na Internet informações referentes ao Inventário Florestal do estado de Minas Gerais, o qual consiste no mapeamento da flora nativa e dos reflorestamentos existentes no estado, etapa efetuada em 2004, e no monitoramento contínuo desta cobertura.

A maior motivação para a realização deste trabalho foi a interatividade das informações mostrando para o usuário o dado espacial no Google Earth e ao mesmo tempo acessando as informações relacionadas ao dado espacial contidas no Banco de Dados. Em um modelo de servidor de dados baseado em arquivos KMZ.

2. Banco de Dados Espaciais

Devido o aumento na utilização de SIG's tem-se buscado cada vez mais uma solução para o gerenciamento dos dados (espaciais, alfanuméricos ou imagens). Para realizar este gerenciamento o estudo da tecnologia de banco de dados é importante, principalmente banco de dados espaciais.

Segundo [Silberschatz 1999], banco de dados espaciais são banco de dados utilizados para armazenar informações geográficas, como mapas.

Os SGBDs atuais possuem extensões espaciais para o melhor aproveitamento do SGBD com dados deste tipo, pois quando se utiliza à extensão espacial pode-se trabalhar com tipos de dados espaciais definidos por elas, tais como ponto, linha e polígono. Estas extensões permitem que tais dados sejam manipulados como qualquer outro tipo de dado de SGBD. Além desta característica, há a extensão da linguagem SQL ofertando operações e funções para consultar relações espaciais.

Entre as extensões espaciais mais utilizadas estão: Oracle spatial [Murray 2003], PostGIS [PostGIS 2005] e a extensão espacial do SGBD MySQL [MySQL 2005], sendo esta última a mais incompleta e menos usada.

3. Inventário Florestal de Minas Gerais

Para o desenvolvimento e aplicação de uma metodologia adequada, foi estabelecido um convênio entre a Universidade Federal de Lavras (UFLA) e a Fundação de Desenvolvimento Científico e Cultural (FUNDECC), entidades que, através do Laboratório de Estudos e Projetos em Manejo Florestal (LEMAF), são as executoras do Mapeamento e Inventário da Flora Nativa e dos Reflorestamentos do Estado de Minas Gerais.

Tais informações serão utilizadas como instrumento de política, planejamento e gestão florestal e ambiental pelo Instituto Estadual de Florestas, pela Secretaria de Estado do Meio Ambiente e Desenvolvimento Sustentável e por outras esferas do governo do estado de Minas Gerais.

4. Metodologia

Para o desenvolvimento da aplicação Webmapping do Inventário Florestal do estado de Minas Gerais foram utilizadas várias tecnologias. Destacamos: o SGBD Oracle e sua extensão espacial Oracle Spatial, árvore hiperbólica, Java, Google Earth e PHP.

4.1. Modelagem do Banco de Dados

O Banco de Dados do Inventário Florestal de Minas Gerais foi todo modelado de acordo com a modelagem OMT-G [Borges 2001]. Esta modelagem foi escolhida por apresentar um grande poder de expressividade para modelar dados geográficos e dar suporte a todas as estruturas necessárias a uma modelagem de qualidade para um banco de dados espaço-temporal.

4.2. Banco de Dados espacial

Para a criação do Banco de Dados do Inventário Florestal do estado de Minas Gerais foi escolhido o Sistema Gerenciador de Banco de Dados Oracle e sua extensão espacial Oracle Spatial [Murray 2003]. Este SGBD foi escolhido no desenvolvimento deste trabalho por ser um SGBD robusto, estável e por possuir uma extensão espacial que atende todos os requisitos necessários para o desenvolvimento deste projeto.

As tabelas do Banco de Dados do Inventário Florestal foram criadas a partir de ESRI Shape (.shp) gerados pelo projeto. Estes shapes foram importados para o Oracle através da ferramenta Arc Catalog via ArcSDE.

4.3. Interface com Árvore Hiperbólica

A árvore hiperbólica é uma ferramenta de navegação formada por uma rede de nós. Ela destina maior espaço para o nó que está em foco e mostra o seu contexto com tamanho progressivamente reduzido à medida que se distancia do foco. O usuário pode alterar o foco clicando em qualquer nó. Quando isso ocorre, o nó clicado é transladado para o centro e todos os outros se rearranjam na periferia.

A Árvore Hiperbólica é indicada para a visualização de grandes hierarquias, pois, mesmo com milhares de nós, é possível observar as informações dos nós no entorno do foco.

Ela foi escolhida devido à facilidade de acesso as informações armazenadas em seus nós, que foram dispostos de maneira a dividir o estado de Minas Gerais hierarquicamente em suas Bacias, Sub-Bacias, Municípios e Fragmentos, abrangidos no projeto.

A árvore hiperbólica foi desenvolvida utilizando a linguagem de programação Java.

4.4. Google Earth

O Google Earth é um aplicativo que oferece ao usuário um globo virtual composto por imagens de satélite ou áreas de todo o planeta. É um programa *stand-alone,* ou seja, precisa ser instalado no computador do usuário.

Além das imagens que fornecem um registro fotográfico do planeta, o Google Earth integra um Sistema de Informações Geográficas (SIG), o que possibilita que sejam visualizadas camadas de informação sobrepostas ao globo virtual [Piliar 2006].

Para utilizar o Google Earth integrado ao SGBD Oracle Spatial foram criados arquivos com extensão KMZ. Estes arquivos são gerados a partir de um aplicativo na linguagem DELPHI que se conecta a um banco de dados Oracle de onde são retiradas as informações espaciais para a construção desses arquivos que é realizado uma única vez, desde que os objetos geográficos não mudem sua forma ou atualização no espaço. O arquivo KMZ nada mais são que arquivos XMLs contendo informações geográficas (pontos, linhas, polígonos).

O link de informações contido em cada objeto desenhado (arquivo kmz) no Google Earth é um hipermapa que faz a requisição de uma página em PHP a qual faz a comunicação com SGBD e apresenta todas as informações relacionadas ao objeto consultado.

5. Resultados

A Figura 1 apresenta a tela inicial da aplicação SIG Web desenvolvida neste trabalho. Esta tela apresenta a interface criada com a tecnologia de árvore hiperbólica. A qual está dividida da seguinte forma: Estado; Bacia; Sub-bacia; Municipio; Fragmento ou Unidade de Conservação.

O usuário pode fazer a busca por uma destas informações, ou simplismente por uma espécie de árvore ou fitofisionomia como cerrado, campo rupestre, floresta semidecidual, entre outras.

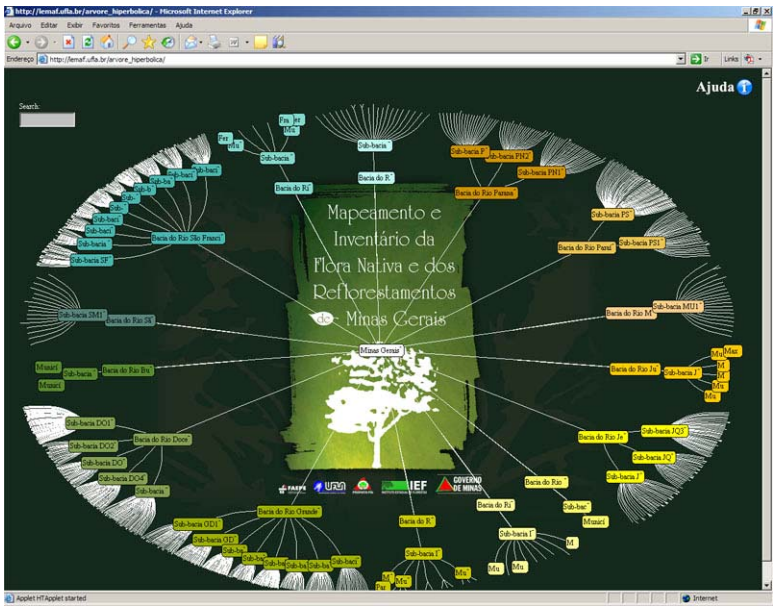


Figura 1. Tela inicial da aplicação SIG Web do Inventário Florestal de MG

A aplicação possui um campo de busca que pode ser visto no canto superior esquerdo da tela inicial. Para fazer uma busca qualquer o usuário digita alguma informação no campo e a árvore hiperbólica vai filtrando as informações corretas. A Figura 2 apresenta a árvore hiperbólica sendo filtrada em uma consulta.

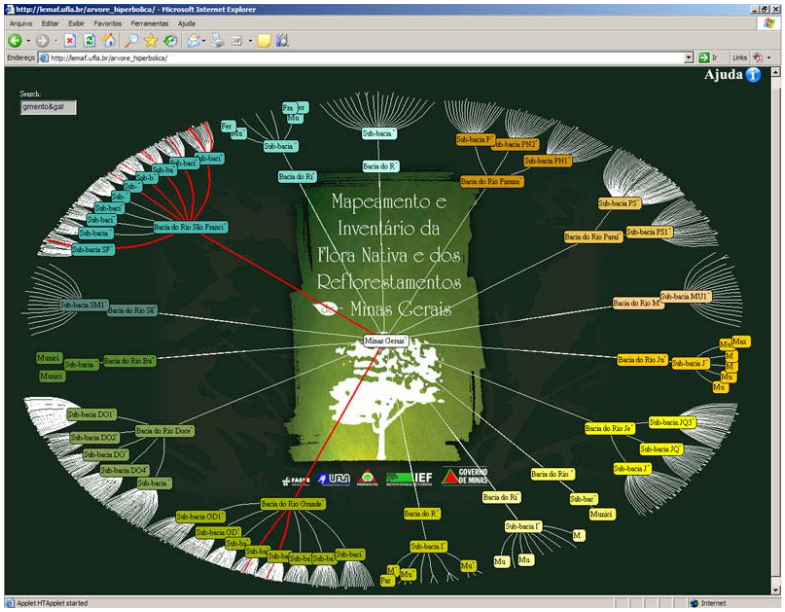


Figura 2. Filtragem dos dados relacionadas a uma consulta.

Após localizar a informação consultada, o usuário clica no objeto desejado. Neste instante nosso servidor de dados transmite ao cliente as informações espaciais em um arquivo KMZ que retrata os objetos espaciais do Banco de Dados, ou seja, ao clicar em um objeto o Google Earth é aberto mostrando as informações espaciais contidas no KMZ (por exemplo: Fragmento Galheiros) e o hipermapa. Neste momento que entra a integração entre os objetos geográficos e as informações alfanuméricas contidas no Banco de Dados. A Figura 3 mostra o Google Earth com a informação buscada pelo usuário.

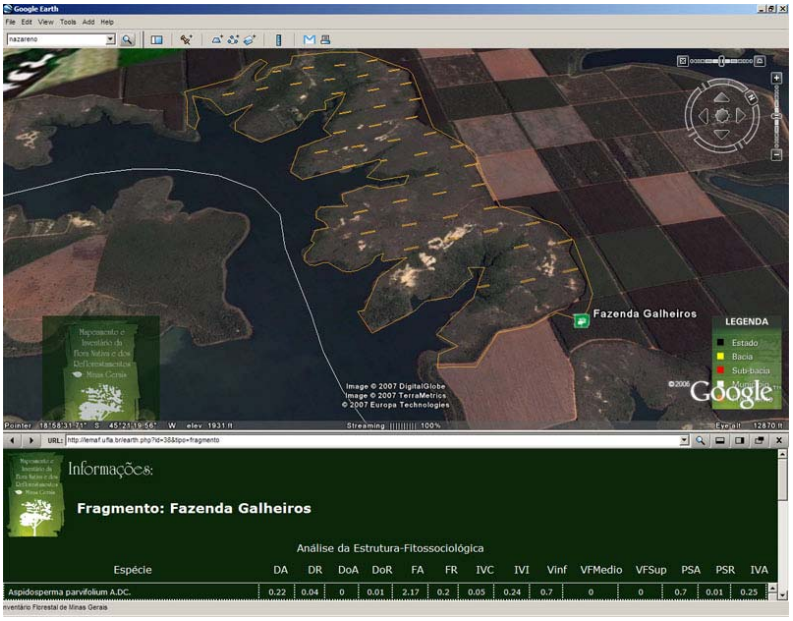


Figura 3. Resultado da busca apresentada no Google Earth.

O Google Earth apresenta a localização espacial do fragmento desejado (poderia ser o estado todo, bacia, sub-bacia, município ou unidade de conservação) e um link

para as informações contidas no banco de dados espacial relacionadas com o objeto apresentado (no nosso exemplo fragmento galheiros).

O usuário também tem a opção de fazer buscas espaciais. Um exemplo seria o cálculo do fragmento mais próximo de um determinado município que contêm a fitofisionomia campo cerrado.

6. Considerações Finais

A utilização de Sistemas de Informações Geográficas tem se tornado a cada dia mais freqüente o que impulsiona os estudos na área de gerenciamento dos dados geográficos, principalmente Banco de Dados geográficos. A disseminação destes dados na Internet esta se tornando cada vez mais comum e essencial. Neste trabalho foi apresentado à integração entre o SGBD Oracle Spatial e o Google Earth para diponibilização de dados geográficos referentes ao Inventário Florestal de Minas Gerais. Pretende-se desta forma, contribuir como instrumento para a formulação de políticas públicas e gestão ambiental voltadas para a prática do desenvolvimento sustentável.

Como trabalhos futuros podemos citar a criação de novas funcionalidades para a aplicação, utilização do Google Maps ao invés do Google Earth, assim o usuário não precisará ter o Google Earth instalado na máquina para usar a aplicação.

7. Referências

Borges K. A. V., Davis C. A., Laender A. H. F., OMT-G: An Object-Oriented Data Model for Geographic Applications, Geoinformatica, v.5 n.3, p.221-260, September 2001

Câmara, G., Casanova, M. A., Hemerly, A. S., Magalhães, G. C. and Medeiros, C. M. B. (1996) “Anatomia de Sistemas de Informação Geográfica”, In: X Escola de Computação, Instituto de Computação, UNICAMP, Campinas.

Ferreira, K. R. (2003) “Interface para Operações Espaciais em Banco de Dados Geográficos”, São José dos campos, SP: Mestrado, Instituto Nacional de Pesquisas Espaiciais (INPE). 100p.

Murray, C., (2003) “Oracle Spatial User's Guide and Reference 10g Release 1 (10.1)”, Redwood City, Oracle Corporation.

Mysql (2005) “Manual de Referência do MySQL 4.1”. Disponível em http://dev.mysql.com/doc/refman/4.1/pt/index.html. Acesso em: 29 de abril de 2007.

Piliar, G. G. (2006) “Cidades Híbridas: Um estudo sobre o Google earth como ferramenta de escrita virtual sobre a cidade”. Porto Alegre. RS.

Postgis. (2005) “PostGIS Manual”. Disponível em: http://postgis.refractions.net/docs/postgis.pdf . Acesso em: 05 de maio de 2007.

Silberschatz, A. (1999), “Sistema de Banco de Dados”. São Paulo: Pearson Education do Brasil. 778p.

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 233-238.

Desenvolvimento de SIG para Web utilizando MDA

Carlos Eduardo R. de Mello, Geraldo Zimbrão da Silva, Jano M. de Souza

Programa de Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro (UFRJ)
Caixa Postal 68.511 – Zip Code: 21945-970 – Rio de Janeiro – RJ – Brazil

{carlosmello,zimbrao,jano}@cos.ufrj.br

Abstract. *This work aims at increasing the development productivity of the Geographic Information Systems through the MDA. We defined an UML extension which adds geographic information to the class diagram. From this extension, we have been implementing a cartridge in the AndroMDA tool which allows the automatic generation of the MapServer configuration files.*

Resumo. *O objetivo deste trabalho é aumentar a produtividade no desenvolvimento de SIG para Web através do uso do padrão MDA. Definimos uma extensão da UML que agrega ao modelo de classes informações geográficas utilizadas nos SIG. A partir desta extensão, estamos implementando um cartucho na ferramenta AndroMDA que permite a geração automática dos arquivos de configuração do MapServer.*

1. Introdução

No desenvolvimento de Sistemas de Informação Geográfica (SIG), ferramentas como servidores de mapas são utilizadas para dar suporte à busca, recuperação e visualização de mapas. Uma ferramenta de código-aberto muito utilizada é o MapServer, um ambiente de desenvolvimento de SIG para Web desenvolvido pela Universidade de Minnesota [Carvalho, 2004]. Para utilizar o MapServer é necessário que o desenvolvedor conheça todas as suas funcionalidades, como arquivos de configuração, etiquetas utilizadas e a arquitetura do ambiente. O desenvolvimento de SIG utilizando esse ambiente pode se tornar um processo longo e repetitivo. Dependendo do tamanho e da complexidade do SIG, os arquivos de configuração e trechos de código-fonte podem ser replicados inúmeras vezes pelo desenvolvedor, gerando retrabalho.

Portanto, propomos a utilização do padrão de Arquitetura Orientada a Modelo (*Model Driven Architecture* – MDA) [Mellor, 2004] no desenvolvimento de SIG para Web, com o objetivo de melhorar a produtividade dos desenvolvedores. Esse padrão apóia todo o ciclo de vida de desenvolvimento de aplicações, através da utilização de modelos. No MDA, a especificação das funcionalidades do sistema é isolada da especificação da implementação das funcionalidades para uma plataforma ou tecnologia específica. Com isso, além da possibilidade de construir modelos de uma maneira formal, o MDA também permite que transformações automáticas sejam realizadas entre modelos, com o objetivo de se obter um produto final de software [Mellor, 2004]. Essas transformações automáticas diminuem o tempo, o esforço e o retrabalho no desenvolvimento, melhorando a produtividade. Existem várias ferramentas que implementam o padrão MDA, uma delas é o AndroMDA [Bohlen, 2003]. Esta é uma ferramenta de código-aberto para geração de código através de transformações

automáticas. Essas transformações são realizadas por cartuchos, capazes de gerar novos modelos ou código.

O objetivo deste trabalho é tentar aumentar a produtividade do desenvolvimento de SIG para Web, através da geração automática de código a partir de modelos de Diagrama de Classes. Como meio para alcançar este objetivo, propomos uma extensão da Linguagem de Modelagem Unificada (*Unified Modeling Langague* – UML) [Booch, 1999], de forma que esta dê suporte ao desenvolvimento de aplicações de SIG para Web. Além disso, estamos desenvolvendo um cartucho para ferramenta AndroMDA, para que a geração automática do código da estrutura básica dos arquivos de configuração do MapServer possa ser realizada.

Na seção 2 deste trabalho, apresentamos uma visão geral de MDA, seus modelos e seu funcionamento. Na seção 3, apresentamos nossa proposta de extensão da UML para apoiar o desenvolvimento de SIG para Web. Em seguida (seção 4), apresentamos as transformações implementadas no cartucho do AndroMDA para gerar a estrutura básica dos arquivos de configuração do MapServer. Finalmente, na seção 5, descrevemos as considerações finais e o andamento deste trabalho.

2. Visão geral do MDA

Nesta seção, apresentamos uma visão geral do MDA e da ferramenta AndroMDA. Mais detalhes sobre o MDA podem ser encontrados em [Mellor, 2004][Kleppe, 2003][Frankel, 2003].

A Arquitetura Orientada a Modelo (*Model Driven Architecture* – MDA) é um padrão do Grupo de Gerenciamento de Objetos (*Object Management Group* – OMG) [Mellor, 2004]. Esse padrão compreende todo o ciclo de vida de desenvolvimento de aplicações, através de modelos de desenvolvimento de software. No MDA, as especificações das funcionalidades do sistema estão isoladas das especificações de implementação para uma plataforma ou tecnologia específica. Portanto, o MDA encoraja a especificação de um Modelo Independente de Plataforma (*Platform Independent Model* – PIM), isto é, que não contém informações de nenhuma plataforma ou tecnologia específicas no modelo. O PIM é transformado em um Modelo Específico de Plataforma (*Platform Specific Model* – PSM), onde informações específicas da tecnologia de implementação são acrescentadas ao modelo. O PSM gerado a partir do PIM pode ser transformado em código da aplicação para ser executado na plataforma tecnológica escolhida.

O modelo de nível mais alto desses modelos descritos anteriormente é o Modelo Independente de Computação (*Computation Independent Model* – CIM). Esse modelo descreve o sistema dentro do seu ambiente (domínio de negócio), apresentando o que é esperado que o sistema faça, sem detalhes de como isto será feito. Portanto, os requisitos do sistema são modelados através de um CIM.

O CIM pode ser modelado utilizando a UML ou outras linguagens, de acordo com os requisitos de análise. O PIM e o PSM podem ser modelados utilizando qualquer linguagem de especificação. Tipicamente, a UML é utilizada, pois é um padrão para especificação de sistemas para domínios genéricos e ainda pode ser estendida para apoiar linguagens para domínios específicos.

Na figura 1, podemos observar como os modelos estão relacionados. Os requisitos do sistema são modelados no CIM. A partir modelo do CIM, são realizados refinamentos dos requisitos no PIM. A transformação do CIM para o PIM é feita manualmente, portanto, até esse ponto não é realizada nenhuma transformação automática. A transformação do PIM para o PSM de uma plataforma específica é feita de forma automática por ferramentas de MDA. O código para a plataforma específica é então gerado a partir das transformações realizadas no PSM.

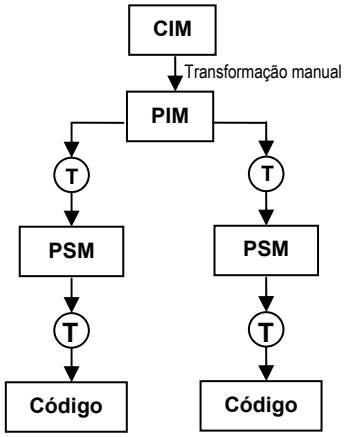


Figura 1 – Arquitetura Orientada a Modelo [Mazón, 2005]

Na ferramenta AndroMDA, as transformações de modelo para modelo e de modelo para código são realizadas pelos cartuchos. O AndroMDA possui cartuchos padrões que permitem o desenvolvimento de sistemas genéricos. Entretanto, esses cartuchos podem ser estendidos, para que sejam suportadas novas transformações para plataformas específicas ou para modelos específicos [Bohlen, 2003]. Também é possível a criação de novos cartuchos, onde novas transformações são definidas com base em novos modelos ou tecnologias.

3. Extensão da UML

Nesta seção, apresentamos uma proposta de extensão da UML, de modo que esta ofereça suporte adequado para a geração dos arquivos de configuração do MapServer.

Os SIG geralmente utilizam o conceito de camadas para a apresentação de mapas [Camara, 1996]. Cada camada é formada por objetos espaciais e essas são sobrepostas formando as imagens dos mapas. O ambiente MapServer também utiliza o conceito de camadas, estas são descritas por um arquivo de configuração chamado *mapfile* [MapServer, 2004]. Além da descrição das camadas, o *mapfile* também possui todas as informações necessárias para que o MapServer reproduza a imagem do mapa referente a esse *mapfile*. Cada imagem reproduzida de um mapa possui um *mapfile* correspondente e cada *mapfile* descreve apenas um mapa. Portanto, a geração correta dos *mapfiles* é fundamental para o desenvolvimento de um SIG.

Entretanto, dependendo do tipo de mapa a ser reproduzido, o seu respectivo *mapfile* pode ficar com um número grande de linhas. Esse excesso de linhas no *mapfile* pode levar o desenvolvedor a esquecer de colocar camadas ou os nomes corretos dessas camadas no *mapfile*. Outro problema é a associação dos objetos espaciais contidos nos

mapas (rios, cidades, estados *etc.*) com as classes de negócio do próprio SIG (rios, cidades, estado *etc.*).

Portanto, para tentar amenizar esses problemas, propomos uma extensão do Diagrama de Classes da UML. Incluímos no Diagrama de Classes dois estereótipos e dois valores etiquetados.

O primeiro estereótipo cujo nome foi definido por *Layer* só pode ser aplicado a uma classe que represente uma camada em um mapa no sistema. Este estereótipo possui um valor etiquetado chamado *geometry* cujos valores possíveis são: *LINE*, *POLYGON*, *POINT*. Esses possíveis valores definem o tipo geométrico do objeto espacial representado pela classe, isto é, linha, polígono ou ponto [Güting, 1994].

O segundo estereótipo que definimos é o *ItemClass*. Este só pode ser associado a um atributo de uma classe que possua o estereótipo *Layer*. O atributo com o estereótipo *ItemClass* define como dividir os objetos espaciais de uma camada em vários grupos. Por exemplo, uma camada onde temos várias rodovias e cada rodovia possui o atributo *principal*, definindo o tipo da rodovia. Se quisermos reproduzir um mapa com as rodovias principais coloridas de azul e as rodovias secundárias de vermelho, precisamos indicar para o MapServer qual o atributo que contém esta informação. O estereótipo *ItemClass* possui o valor etiquetado *NumClasses* que contém o número de grupos formados dentro da camada. No exemplo anterior, podemos ter dois grupos de rodovias, principais e secundárias, portanto o valor etiquetado de *NumClasses* é 2.

Na figura 2, podemos observar um exemplo de uma aplicação modelada na extensão do diagrama de classes proposto. As classes **UnidadeFederacao**, **Municipio**, **Sede** e **Rodovia** são classes de negócio do sistema e estão associadas a objetos espaciais contidos em um mapa. Note que a classe **Concessionaria** é apenas uma classe de negócio, portanto, não possui nenhum estereótipo ou valor etiquetado.

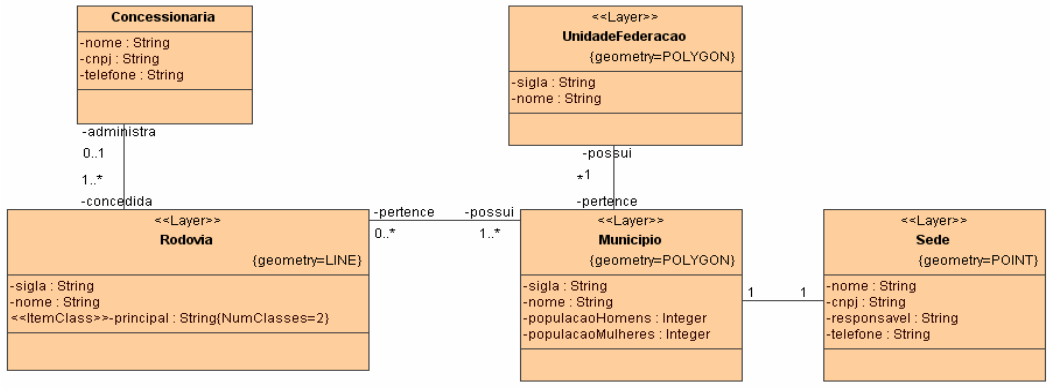


Figura 2 – Aplicação modelada na extensão do modelo de classes

4. Transformações

Nesta seção apresentamos as transformações realizadas pelo cartucho do AndroMDA que estamos implementando para a geração dos arquivos de configuração do MapServer. A seguir descrevemos a seqüência das operações realizadas pelo AndroMDA e pelo cartucho no processo de geração do *mapfile*.

O AndroMDA carrega o modelo do diagrama de classes a partir de um XML padrão exportado por uma ferramenta de modelagem. Em seguida, o AndroMDA gera

um modelo de Metafacades que realiza transformações nos dados de maneira que o cartucho possa trabalhar e tratar esses dados do modelo.

A primeira transformação que o cartucho realiza é geração do esqueleto básico do *mapfile*. Este esqueleto possui o modelo básico e o nome do *mapfile*. O nome do *mapfile* gerado é o mesmo nome do pacote no qual às classes com estereótipo *Layer* estão. Portanto, para a geração de todos os *mapfiles*, o cartucho procura os pacotes que contenham classes com estereótipo *Layer* e gera o esqueleto básico para todos os *mapfiles* definidos de acordo modelo.

A segunda transformação implementada é a geração das camadas. Para todas as classes com o estereótipo *Layer* são geradas camadas com o mesmo nome. Portanto, no exemplo da Figura 2, as classes **Sede**, **Municipio**, **UnidadeFederacao** e **Rodovia** geram as camadas **Sede**, **Municipio**, **UnidadeFederacao** e **Rodovia**, respectivamente, no *mapfile* correspondente ao pacote onde estão essas classes.

A última transformação é a geração dos grupos de objetos espaciais dentro de uma mesma camada. Estes grupos de objetos são chamados de classes dentro do *mapfile*. Toda camada no *mapfile* possui pelo menos uma classe, isto é, pelo menos um grupo de objetos espaciais. Portanto, o cartucho procura nos dados carregados do modelo as classes que possuem o estereótipo *Layer* e que não possuem nenhum atributo com esteriótipo *ItemClass*. Para essas classes é gerada no *mapfile* correspondente apenas uma classe (grupo) dentro da camada com o mesmo nome da classe do modelo. Entretanto, para as classes com estereótipo *Layer* que possuem algum atributo com o estereótipo *ItemClass*, são geradas classes (grupos) dentro da camada com o mesmo nome da classe do modelo. O nome destas classes (grupos) é o mesmo nome da camada concatenada por um número inteiro. O número de classes (grupos) geradas no *mapfile* será igual ao valor do número contido no valor etiquetado *NumClasses* que o atributo possui.

5. Considerações finais

Neste artigo, apresentamos a implementação de um cartucho para o AndroMDA que realiza a geração automática dos arquivos de configuração do MapServer (*mapfiles*).

As transformações realizadas pelo cartucho não geram todo o conteúdo do *mapfile*, mas sim a sua estrutura básica, através das informações que estão disponíveis no modelo. Para a geração completa do *mapfile* é necessário que a extensão da UML utilizada ofereça mais informações sobre os mapas a serem gerados. Entretanto, o objetivo do modelo é ser independente de plataforma, por isso a agregação de mais informações ao modelo deve ser feita com cautela.

Estamos desenvolvendo o cartucho de maneira que novas extensões da UML possam ser implementadas. Atrelado a isso, também estamos trabalhando em propostas de estereótipos e valores etiquetados para diagramas de atividades. Com isso, esperamos que mais informações úteis para o desenvolvimento de SIG possam ser incorporadas ao modelo, sem que ocorra dependência de plataforma.

6. Agradecimentos

Agradecemos ao CNPQ pelo apoio financeiro.

Referências

Bohlen, M. (2003) "AndroMDA", www.andromda.org.

Booch, G.; Rumbaugh, J. and Jacobson, I. (1999) "The Unified Modeling Language User Guide", Massachusetts.

Camara, G. et al. (1996) "Anatomia de Sistemas de Informação Geográfica", Instituto de Computação, UNICAMP, Campinas.

Carvalho, C. A. (2004) "Desenvolvimento de Aplicações WebGIS em MapServer.", EMBRAPA, Campinas.

Frankel, D. S. (2003) "Model Driven Architecture. Applying MDA to Enterprise Computing. Indianapolis", Wiley, Indiana.

Güting, R.H. (1994) "An Introduction to Spatial Database Systems", Special Issue on Spatial Database Systems of the VLDB Journal, Vol.3, No.4, October.

Kleppe, A., Warmer, J. and Bast, W. (2003) "MDA Explained. The Practice and Promise of The Model Driven Architecture.", Addison Wesley.

MapServer, (2004) "MapServer Documentation Project.", http://mapserver.cttmar.univali.br.

Mazón, J. R., Trujillo, J., Serrano, M. and Piattini, M. (2005) "Applying MDA to the development of data warehouses", Proceedings of the 8th ACM international workshop on Data warehousing and OLAP, Bremen, Germany, November.

Mellor, S., Scott, K., Uhl, A. and Weise, D. (2004) "MDA distilled: principles of Model-Driven Architecture", Addison-Wesley.

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 239-250.

O projeto WebMAPS: desafios e resultados

Carla Geovana do N. Macário[1 2], Claudia Bauzer Medeiros[1], Rodrigo Dias Arruda Senra[1]

[1] Instituto de Computação
Universidade Estadual de Campinas
Caixa Postal 6176 – 13084-971 – Campinas/SP – Brasil

[2]Embrapa Informática Agropecuária
Caixa Postal 6041 – 13083-886 – Campinas/SP – Brasil

{carlamac,cmbm}@ic.unicamp.br, rsenra@acm.org

Abstract. *This paper describes challenges and results of WebMAPS, a multi-disciplinary project under development at UNICAMP. Its goal is to develop a platform based on Web Services for agro-environmental planning. It requires state of the art research in specification and implementation of software that relies on several kinds of distributed information - satellite images, data from sensors and from agricultural production and geographic data.*

Resumo. *Este trabalho descreve desafios e resultados do projeto WebMAPS, um esforço multidisciplinar envolvendo ciências agrárias e de computação, em desenvolvimento na UNICAMP. Seu objetivo é desenvolver uma plataforma baseada em serviços Web para o planejamento agro-ambiental. Requer pesquisa de ponta voltada à especificação e à implementação de software com acesso a vários tipos de informação distribuída - imagens de satélite, dados provenientes de sensores, dados de produção agrícola e dados geográficos.*

1. Introdução

A agricultura é uma atividade de destaque na economia brasileira, contribuindo significativamente para o PIB brasileiro. Em 2005, o PIB atingiu R$ 1.929 bilhões, sendo R$ 537 bilhões provenientes de atividades agrícolas, o que corresponde a quase 30% do montante brasileiro. Diversos fatores contribuem para estes números. Um deles é a disponibilidade cada vez maior de sistemas que auxiliam o planejamento e o gerenciamento da produção.

Pesquisas em Ciências Agrárias geram informação essencial para a agricultura brasileira. No entanto, decisões sobre o que plantar (e quando, onde e como) exigem acesso confiável a dados e informação atualizada. Além disso, há necessidades de desenvolvimento de modelos sofisticados, requerendo cooperação entre especialistas do domínio e cientistas da Computação. Portanto, soluções que ajudem a aquisição, o processamento e a disseminação de dados em tempo hábil são cada vez mais necessárias. O WebMAPS (Sistema baseado na WEB Semântica para Monitoramento Agrícola e Previsão de Safras) visa contribuir nesse sentido, dando apoio ao processamento de dados científicos heterogêneos provenientes de diversas fontes. Seu objetivo final é o estabelecimento de uma plataforma base para a formulação, implementação e avaliação

de políticas integradas de planejamento agrícola. O projeto, que iniciou em 2003 apoiado por um edital Universal do CNPq, envolve pesquisadores de três unidades da UNICAMP - do Instituto de Computação, da Faculdade de Engenharia Agrícola e do CEPAGRI [Medeiros et al. 2006]. Este trabalho descreve alguns dos desafios e resultados do projeto obtidos até o momento, estando organizado da seguinte forma: a seção 2 descreve os principais tópicos de pesquisa envolvidos no projeto. A seção 3 aborda a metodologia usada no desenvolvimento do sistema, focando na implementação do protótipo disponível na web. Algumas da iniciativas existentes nas mesmas áreas de pesquisa do projeto encontram-se na seção 4. Por fim, a seção 5 apresenta conclusões e atividades em desenvolvimento.

2. Alguns Aspectos de Pesquisa Envolvidos

A especificação e o desenvolvimento do WebMAPS envolve pesquisa multidisciplinar em Ciências da Computação e Ciências Agrárias. Do lado computacional, estão sendo estudados pontos em aberto nas áreas de grandes bancos de dados e algoritmos para manipulá-los, engenharia de software de grandes sistemas, processamento digital de imagens e interfaces multimodais. Do lado de Ciências Agrárias, os estudos vêm envolvendo aspectos de sensoriamento remoto, desenvolvimento de novas metodologias de previsão de safra, especificação de novos métodos e ferramentas de apoio a decisão no domínio agrícola, dentre outros. A combinação dos resultados destas duas áreas é o principal diferencial do projeto.

Na área de Ciência da Computação vários pontos vêm sendo estudados. Um dos tópicos centrais são sistemas de bancos de dados espaço-temporais, para integrar informação extraída de imagens de satélite a dados climatológicos de modo a permitir fazer associações *ad-hoc* entre eles, para regiões e períodos arbitrários. Outros dois desafios são a especificação e implementação de algoritmos que possibilitem estabelecer correlações entre séries temporais e sua evolução, a partir do uso de novos resultados na área de casamento de padrões, e a especificação e desenvolvimento de algoritmos para processamento das imagens de satélite, a partir de características de conteúdo, visando busca por padrões temporais. Ainda outro tópico, o gerenciamento da rastreabilidade de produtos agrícolas, envolvendo dentre várias, técnicas de *workflow* e interoperabilidade de serviços Web. Finalmente, estão sendo levados em conta aspectos de *design* e implementação de interfaces que permitam diferentes modos de visualização e interação com os dados armazenados, visando facilitar consultas interativas sobre evolução espaço-temporal dos dados.

Em Ciências Agrárias, a ênfase tem sido dada a aspectos de sensoriamento remoto, envolvendo séries de imagens de satélite [Lunetta et al. 2003]. Essas imagens podem ser apresentadas em bandas individuais ou em índices de vegetação, como o NDVI (*Normalized Difference Vegetation Index*), que é calculado pela diferença de valores de reflectância das bandas Infravermelho (IV) e Vermelho (V) e normalizado pela soma dos valores de reflectância destas duas bandas. O índice permite avaliar as condições da biomassa de uma cultura, podendo ser usado para monitoramento e previsão de safra. Uma curva de valores NDVI para uma região em um período pode ser comparada com curvas de outros períodos (para determinar o comportamento da biomassa) e também com outras curvas do mesmo período mas de outras regiões. Estes dados podem também ser comparados com séries históricas de temperatura, de chuva ou de outros dados meteorológicos. Estão

previstas famílias de ferramentas para: (a) manipulação de dados de cadastro e produção; (b) manipulação de dados de sensores; (c) manipulação de imagens de satélite; (d) uso de *workflows* para modelagem de processos. A idéia é disponibilizar produtos que sejam gerados por tais ferramentas, mas também permitir a execução *on-line*, combinada, de um ou mais módulos.

3. Aspectos de Implementação

O WEBMaps vem sendo desenvolvido combinando prototipação rápida com a disponibilização de ferramentas e produtos aos especialistas da área, para obter *feedback* quanto ao uso dos módulos desenvolvidos. Tal *feedback* vem possibilitando aperfeiçoar o levantamento de requisitos e a especificação de módulos adicionais. A análise de requisitos foi guiada pela Semiótica Organizacional [Liu et al. 2007, Schimiguel et al. 2005, Prado et al. 2000] e o desenvolvimento do projeto envolve metodologias de teste de desempenho [Torres-Zenteno et al. 2006]. A maior parte da implementação vem utilizando a tecnologia J2EE e o servidor de aplicações Zope, adotando um modelo de arquitetura multi-camadas. Dados são armazenados no PostGreSQL, prevendo-se sua migração para PostGIS. A variação espaço-temporal dos dados ainda está em discussão, podendo, por exemplo, vir a usar a proposta de [Faria 1998]. Para geração dos perfis de NDVI estão sendo utilizadas imagens do sensor MODIS (*Moderate Resolution Imaging Spectroradiometer*) com resoluções: espacial de 250 metros, temporal diária, radiométrica 16 bits e espectral de 36 bandas.

Algumas destas ferramentas já estão disponíveis [Macário et al. 2007]. Permitem o cadastro de propriedades, talhões e culturas, além de algumas consultas sobre tais entidades (figura 1). A figura 2 ilustra o resultado de uma consulta *on-line* ao sistema, na Web. Ela mostra a evolução, ao longo do tempo, dos valores NDVI de uma região selecionada pelo usuário. Este exemplo, cujos parâmetros foram as coordenadas da região e um intervalo de tempo (jan. 2001 a jan. 2003), usou como base um conjunto de 100 imagens do sensor MODIS das regiões Sul e Sudeste do Brasil.

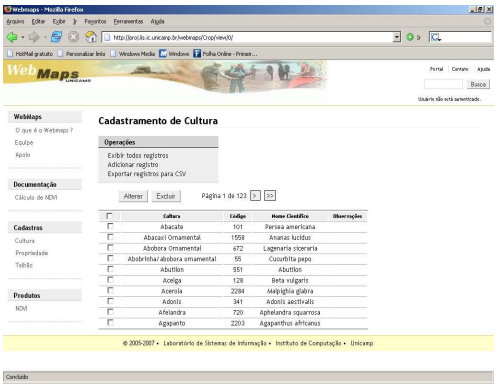


Figura 1. Cadastro de Culturas

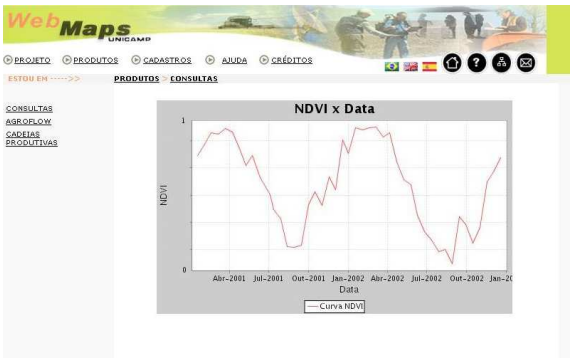


Figura 2. Resultado de Consulta

Para o usuário, a curva indica o comportamento da cultura (no caso, cana) naquela região ao longo daquele período. Este comportamento pode ser relacionado a coletas obtidas para fenômenos associados (chuva, temperatura) e à produtividade da cultura para a mesma região e período. De posse desse gráfico, o usuário pode comparar o comportamento de sua lavoura com anos anteriores que tiveram comportamento semelhante,

permitindo a definição de melhor época para realizar a colheita e a previsão da qualidade da cana produzida e de quanto será sua produção.

Do ponto de vista da Computação, a cópia de tela corresponde à materialização, via código, de vários aspectos de pesquisa de ponta, ressaltando-se: integração de pesquisa em processamento de imagens de satélite e bancos de dados geográficos; implementação de novos métodos de teste de software para a Web, ou projeto de interface de software geográfico para a Web.

Ainda outros resultados do trabalho incluem modelagem de transações em cadeias produtivas [Bacarin et al. 2004] e o desenvolvimento de novos descritores de curvas obtidas a partir de séries históricas de chuva e temperatura, que facilitam a busca por similaridade [Mariote et al. 2007]. Finalmente, o trabalho de [Kondo et al. 2007] especificou e desenvolveu um conjunto de serviços web para rastreabilidade de eventos e produtos em cadeias produtivas agrícolas, usando como base *workflows* da cadeia do leite. Usuários enviam requisições a um serviço que intermedia interações com outros serviços dedicados, responsáveis por encapsular o acesso a diferentes repositórios de registros de eventos. Este conjunto de serviços permite executar uma grande gama de consultas que combinem eventos de uma cadeia agrícola, facilitando gerenciar a qualidade de seus processos e produtos.

4. Trabalhos Correlatos

O acesso confiável a dados na Web é essencial para a geração de informação de apoio a políticas agrícolas. Apesar de iniciativas que estão surgindo nos EUA [Jakubauskas et al. 2001] e na Europa [Lemoine and Kidd 1998], há ainda uma enorme carência de ferramentas que agilizem a produção da informação estratégica. As iniciativas internacionais são centradas em pesquisa em Ciências Agrárias, sendo a Computação usada principalmente como fornecedora de infra-estrutura.

A especificação e o desenvolvimento de sistemas como o WebMAPS são alvo de pesquisa multidisciplinar em nível mundial. Do lado da Computação, há demanda por novos resultados em: bancos de dados e serviços Web [Lockemann et al. 1997] (gerenciamento de fontes de dados volumosos e heterogêneos contendo imagens de satélite, mapas digitais, dados climáticos, econômicos e outros); processamento de imagens [Beucher and Meyer 1993, Soile 1996] (algoritmos específicos para segmentação e recuperação por conteúdo); engenharia de software (especificação, testes e desenvolvimento do software), redes e sistemas distribuídos [Fenstermacher and Ginsburg 2002] (para integrar e processar os dados e ferramentas e disponibilizá-los na Web); e interfaces [Kosch 2002] (contemplando múltiplos tipos e propósitos de interação). Do lado da pesquisa em Ciências Agrárias (domínio alvo), há problemas na identificação dos dados, no procedimento de amostragem, no uso de sensoriamento remoto e no aperfeiçoamento de algoritmos para análise e visualização de informação. Outros problemas a serem abordados consideram, também, processamento de séries temporais [Muthukrishnan et al. 2004, Wu et al. 2005], rastreabilidade de processos e cadeias produtivas [Roder and Tibken 2006] e ontologias [Hochmair 2005].

5. Conclusões e Trabalhos em Andamento

O WebMAPS é motivado por problemas em aberto na geração de informação agrícola estratégica para a tomada de decisões tanto no âmbito governamental (apoio a políticas públicas) quanto no de cooperativas agrícolas e agronegócios. Há vários diferenciais deste projeto em relação a iniciativas internacionais de natureza semelhante: aênfase em pesquisa multidisciplinar em Computação aplicada a Ciências Agrárias - na maioria das iniciativas, a pesquisa é centrada no último domínio e apenas utiliza recursos da Tecnologia da Informação; e a adequação ao contexto geográfico brasileiro, com forte ênfase em dados obtidos via sensores; utilização de novos resultados de pesquisa sobre a Web Semântica; exploração, em tempo real, do conteúdo de imagens, e não apenas de dados textuais; consideração de aspectos de interação humano-computador.

Há vários trabalhos em andamento. Um deles está voltado à disponibilização de ferramentas de projeto e execução de *workflows* para atividades em agricultura. Outras iniciativas envolvem desenvolvimento de consultas por conteúdo a imagens de satélite, modelos baseados em curvas NDVI para identificação de culturas e mecanismos para melhorar a interoperabilidade semântica de dados geoespaciais.

Referências

Bacarin, E., Medeiros, C. B., and Madeira, E. (2004). A collaborative model for agricultural supply chains. In *OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2004*, number 3290 in LNCS, pages 319–336.

Beucher, S. and Meyer, F. (1993). The morphological approach to segmentation: The watershed transformation. In *In Proc. of the International Symposium on Mathematical Morphology*, pages 433–481.

Faria, G. (1998). Um banco de dados espaco-temporal para desenvolvimento de aplicacoes em sistemas de informação geografica. Master's thesis, Instituto de Computação - Universidade Estadual de Campinas.

Fenstermacher, K. and Ginsburg, M. (2002). A lightweight framework for cross-application user monitoring. *IEEE Computer*, 35(3):51–59.

Hochmair, H. H. (2005). Ontology matching for spatial data retrieval from internet portals. In *First International Conference GeoSpatial Semantics, GeoS*, pages 166–182.

Jakubauskas, M., Legates, D., and Kastens., J. (2001). Harmonic analysis of time-series avhrr ndvi data. . *Photogrammetric Engineering and Remote Sensing*, 67(4):461–470.

Kondo, A. A., Medeiros, C. M., Bacarin, E., and Madeira, E. R. M. (2007). Traceability in food for supply chains. In *3rd International Conference on Web Information Systems and Technologies (WEBIST)*, pages 121–127, Barcelona, Spain.

Kosch, H. (2002). MPEG and multimedia database systems. *ACM SIGMOD Record*, 31(2):34–39.

Lemoine, G. and Kidd, R. (1998). *Operational European Cereal Monitoring: Methodological Consideration*. Ispra, Italy. Disponível em <http://conferences.esa.int/98c07/papers/P089.pdf>. Acesso em jul, 2007.

Liu, K., S.Y.Liao, and Chong, S. (2007). *Semiotics for Information Systems Engineering – re-use of high-level artefacts*. Disponível em:

<http://www.scit.wlv.ac.uk/ jphb/cp4040/rolandonotes/CSNDSP2002/Papers/A1/A1.1.pdf>. Acesso em jul, 2007.

Lockemann, P., Kolsch, U., Koschel, A., Kramer, R., Nicolai, R., Wallrath, M., and Walter, H. (1997). The network as a global database: Challenges of interoperability, proactivity, interactiveness, legacy. In *23nd VLDB Conference*, pages 239–250.

Lunetta, R., Johnson, D., Lyon, J., and J., C. (2003). Impacts of imagery temporal frequency on landcover change detection monitoring. *Remote Sensing and Environment*, 89(4):444–454.

Macário, C. G. N., Senra, R. D. A., Medeiros, C. B., Lamparelli, R. A. C., Júnior, J. Z., Rocha, J. V., Madeira, E. R. M., Martins, E., Baranauskas, M. C. C., Leite, N. J., and Torres, R. S. (2007). Monitoramento de safras via web: Um caso de sucesso em pesquisa multidisciplinar. In *6o. Congresso Brasileiro de Agroinformática - SBIAgro 2007*. Aceito para publicação.

Mariote, L. E., Medeiros, C. M. B., and Torres, R. S. (2007). Diagnosing similarity of oscillation trends in time series. In *International Workshop on spatial and spatio-temporal data mining - (SSTDM'07)*, Omaha, USA.

Medeiros et al., C. M. B. (2006). *WebMAPS II - Sistema baseado na WEB Semântica para Monitoramento Agrícola e Previsão de Safras*. Projeto Universal CNPq. Iniciado em 2003 - renovado em 2006.

Muthukrishnan, S., Shah, R., and Vitter., J. S. (2004). Mining deviants in time series data streams. In *6th International Conference on Scientific and Statistical Database Management*.

Prado, A., Baranauskas, M., and Medeiros, C. (2000). Cartography and geographic information system as semiotic systems: A comparative analysis. In *8th ACM Symposium on Advances in Geographic Information System*, pages 161–166.

Roder, A. and Tibken, B. (2006). A methodology for modeling inter-company supply chains and for evaluating a method of integrated product and process documentation. *European Journal of Operational Research*, 169(3):1010–1029.

Schimiguel, J., Baranauskas, M. C. C., and Medeiros., C. B. (2005). Usabilidade de aplicações SIG Web na perspectiva do usuário: um estudo de caso. In *VI Simpósio Brasileiro de Geoinformática - GEOINFO*, Campos do Jordão, SP.

Soile, P. (1996). Morphological partitioning of multispectral images. *Journal of Electronic Imaging*, 5(3):252–265.

Torres-Zenteno, A., Martins, E., Torres, R. S., and Cuaresma, M. J. E. (2006). Teste de desempenho em aplicacoes sig web. In *IX Workshop Iberoamericano de Ingeneria de Requisitos y Ambientes de Software- IDEAS*, La Plata, Argentina.

Wu, H., Salzberg, B., Sharp, G., Jiang, S. B., Shirato, H., and Kaeli., D. (2005). Subsequence matching on structured time series data. In *ACM SIGMOD Conference on Management of data.*, pages 682–693.

Coverage representation in TerraLib

Vitor Dantas, Marcelo G. Metello, Melissa Lemos, Marco A. Casanova

Tecgraf – Computer Graphics Technology Group
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rua Marquês de São Vicente, 225 – CEP 22.453-900 – Rio de Janeiro – RJ – Brazil

{vitorcd,metello,melissa,casanova}@tecgraf.puc-rio.br

Abstract. *Coverage representations, as defined by the OGC specifications, are useful for representing a wide range of geographic phenomena. However, in most GIS projects, coverage representations do not show a common interface and the line between coverages and features with geometry is unclear. We propose an extension to the TerraLib library, using a unifying approach to manage coverage representations, including performance considerations as a major issue.*

1. Introduction

The OGC specification introduces two basic ways of modeling geospatial features, namely *features with geometry* and *coverages* [OGC 1999]. While the former emphasizes the identity of each single feature, the later emphasizes the whole picture, evidencing relationships and the spatial distribution of earth phenomena. The decision of which one is best depends on the application, since moving from one concept to the other is possible.

A coverage defines a function that maps some *spatial domain* into a *value set*. In general, the specialization of coverages reduces to the specialization of the spatial domain. For instance, a Discrete Point Coverage is one whose spatial domain is a set of discrete points. Figure 1 shows an example of a Discrete Point Coverage whose value set has two dimensions, so that each point is associated with a temperature value and a wind speed value, i.e., to a value vector (t_i, s_i), $1 \leq i \leq 6$.

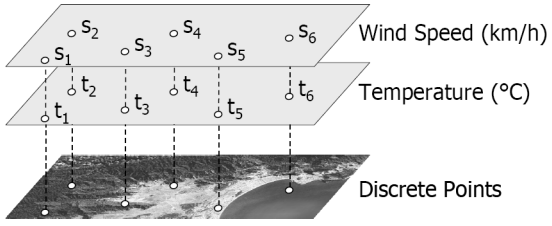


Figure 1. Discrete Point Coverage with two dimensions.

In this work, we aim at developing representations for several coverage subtypes – such as Discrete Point Coverage, Surface Coverage and Line String Coverage – and at defining a common abstract representation for them, as an extension of the TerraLib library [Câmara et al. 2000]. TerraLib is an open-source project that provides tools for the development of geographic information systems [Vinhas and Ferreira 2005].

One often finds in the literature other designations for the concept of coverage. The relation between features with geometry and coverages is similar to what has been discussed about *discrete objects* and *continuous fields* [Couclelis 1992], or *geo-objects* and *geo-fields* [Vinhas 2006].

Although most GIS projects address the problem of coverage representation, few establish a clear distinction between the concepts of features with geometry and coverages. Typical coverage representations such as regular grids (raster) are included in several projects, including GDAL, GRASS, OSSIM GMT and DeeGree [Ramsey 2007].

TerraLib currently offers implementations of coverage representations such as triangular irregular networks (TINs) and regular grids (raster) [Vinhas and Souza 2005], but all representations in TerraLib are referenced as geo-objects. Rather than replacing the current implementation for these representations, we propose a common interface for accessing them.

The paper is organized as follows. Section 2 discusses the representation of coverages in general. Section 3 presents the current implementation of Discrete Point Coverages. Section 4 discusses future work and conclusions.

2. Coverage Representation

2.1. General requirements

We propose the following high-level requirements for coverage representations:

1. The data structure must represent a set of geometries (the spatial domain of the coverage).
2. Each stored geometry must be associated with a value vector (the value of the coverage for the spatial region that the geometry specifies).
3. There must be support for queries that fetch the value vector associated with each stored geometry.
4. There must be support for queries that fetch the value vector in any location within the convex hull defined by the stored geometries, using an interpolation method. There must be a flexible way to change the interpolation method in use.
5. The persistent representation of geometries (in a database) must be clustered by spatial proximity to improve the performance of query processing.

Requirements 1 and 2 have to do with how to represent the coverage; requirements 3 and 4 with how to query the coverage; and requirement 5 is non-functional. Requirements 1 and 3 comply entirely with the OGC specification. We introduced requirements 4 and 5 based on what we expect to be generic application requirements.

Requirement 4 was included to provide a continuous view of the coverage, despite the fact that the support data structure is discrete (i.e., a set of geometries). Interpolation methods are used to infer the value vector at positions where it is undefined, as recommended in the discussion about discrete and continuous coverages of the OGC specification.

Requirement 5, in particular, addresses performance issues that are not considered by OGC. Clustering geometries lead to performance improvement because loading a few large chunks of data from a database is faster than loading the same amount of data in several small chunks. Besides that, if the geometries are clustered by spatial proximity, the addition of a caching mechanism will result in less frequent access to the database in a typical GIS application, in which it is reasonable to assume that access to data over time will be done by spatial proximity, e.g. a user looking at a location is more likely to move next to nearby locations.

2.2. High-level interface for coverage access

For the discrete setting, the OGC specification on coverages provides an abstract specification for the coverage function, named `DiscreteC_Function`, showed in Figure 2. The method `num` returns the number of geometries in the domain, while methods `domain` and `values` return an exhaustive list of the domain (geometries) and the range (value vectors) of the coverage function. The method `evaluate` is used to fetch the value vector associated with a specific element of the domain (a geometry).

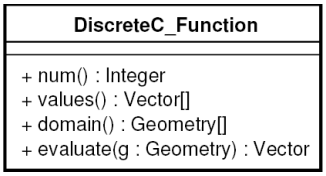


Figure 2. Discrete coverage function from the OGC abstract specification.

We follow a similar approach to define an interface for coverages in the TerraLib library, as depicted by Figure 3. This interface provides methods `begin` and `end`, which have two optional parameters of types `TePolygon` and `TeSpatialRelation`, and return objects of type `TeCoverage::iterator`. These two methods, together, play the role of the method `domain` in `DiscreteC_Function`.

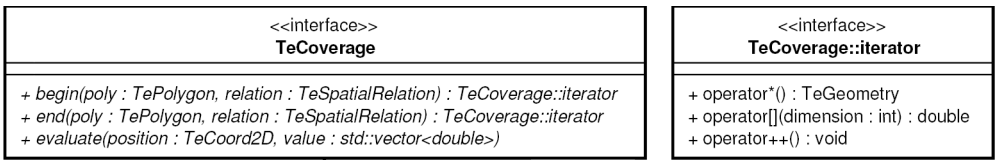


Figure 3. High-level interface for coverages in the TerraLib library.

The parameter of type `TePolygon` represents a polygon and defines a selection area. If no polygon is given, then the selection area includes the entire spatial domain. The parameter of type `TeSpatialRelation` indicates the kind of the spatial relation (e.g., intersection, crossing, overlapping) that holds between the selection area and the geometries to be selected. If no value is given, the relation defaults to intersection.

The returned iterators are used to traverse the selected geometries (instances of `TeGeometry`) and the values associated to them. Coverage iterators, represented by instances of `TeCoverage::iterator`, are manipulated using the overloaded operators "*", "[]"and "++". Dereferencing a coverage iterator with the "*" operator outputs a geometry, while the operator "[]" is used to get the value of a specific dimension. The increment operator "++" is used to advance to the next position of the iteration, i.e. to

the next selected geometry. The example code in Figure 4 shows how to use coverage iterators, assuming that we have access to an instance of `TeCoverage` named "c".

```
// Make spatial query
TeCoverage::iterator it = c.begin(poly);
TeCoverage::iterator end = c.end(poly);

// Iterate over selected geometries
while (it != end) {
    TeGeometry& geom = *it;
    double value1 = it[1];
    double value2 = it[2];
}
```

Figure 4. Example code, showing how to use coverage iterators.

The method `evaluate` provided by the interface `TeCoverage` extends the domain of the coverage beyond the discrete setting, because one is able to access the values on arbitrary locations (through the use of interpolation methods). Otherwise, the domain would be restricted to the discrete collection of geometries, as the method `evaluate` in `DiscreteC_Function`.

3. Current implementation

The first focus of our work was the definition of a representation for the simplest subtype of coverage, namely a Discrete Point Coverage. This representation assumes that the domain is a finite set of sample points. It is useful to model features such as temperature, as this kind of information is typically collected at stations distributed over the territory.

This section discusses a working implementation for a Discrete Point Coverage representation in TerraLib, including class diagrams and the database model.

3.1. Current class model

`TeDiscretePointCoverage` is the main class of the Discrete Point Coverage representation, following the TerraLib naming convention. Figure 5 shows this class, omitting several attributes and methods for clarity.

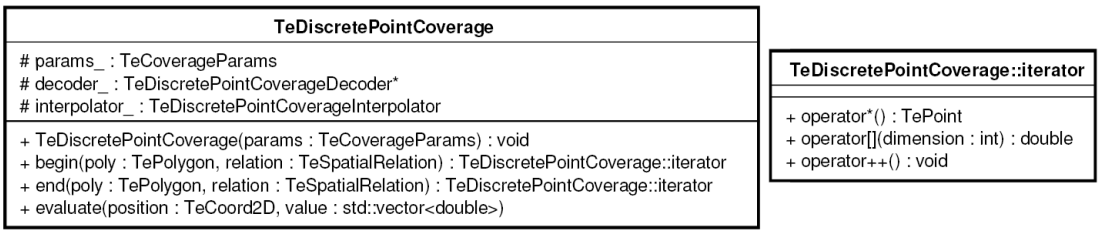


Figure 5. Class diagram for a Discrete Point Coverage.

The methods `begin` and `end` are equal to those from class `TeCoverage`, in section 2.2, except for the return of type `TeDiscretePointCoverage::iterator`, that is specialized to traverse discrete points (instances of `TePoint`).

The method `evaluate` can be applied to a position that does not belong to the discrete domain of the representation. This method delivers the call to the interpolator (instance of `TeDiscretePointCoverageInterpolator`), a class member which actually

implements interpolation. The interpolator class can be extended for implementing different kinds of interpolation, making it a flexibility point. If no interpolator object is provided by the application, the default implementation (*nearest neighbour*) is used.

The constructor of this class receives an instance of `TeCoverageParams`, which represents coverage parameters, and include information about where the data is stored (in a file or database) so that, after an instance of `TeDiscretePointCoverage` is constructed, one can start submitting spatial queries to the coverage. Figure 6 shows the class `TeCoverageParams`, along with the auxiliary data structure `TeCoverageDimension`. Methods were omitted for clarity.

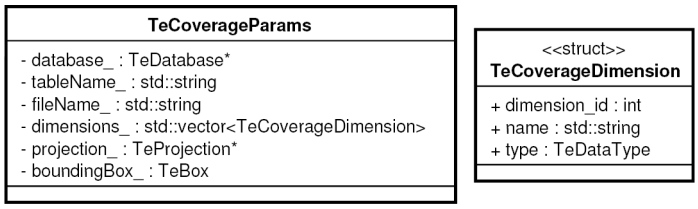


Figure 6. Class diagrams representing parameters of coverages.

An instance of `TeCoverageParams` contains information about how to access the persistent coverage representation from the database or from a file, what are the dimensions of the value vector, what is the minimum rectangle that contains all geometries of the coverage, what is the projection used and more.

3.2. Current data model

The coverage data stored in the database, including information about geometries and values associated to them, is clusterized by spatial proximity of geometries. The clustering process creates blocks, which are stored in the coverage table. The name of such a table has the form `Coverage_(layer_id)_(coverage_id)`, where `(layer_id)` means the layer identifier and `(coverage_id)` means the coverage identifier. Figure 7 shows the diagram of a coverage table.

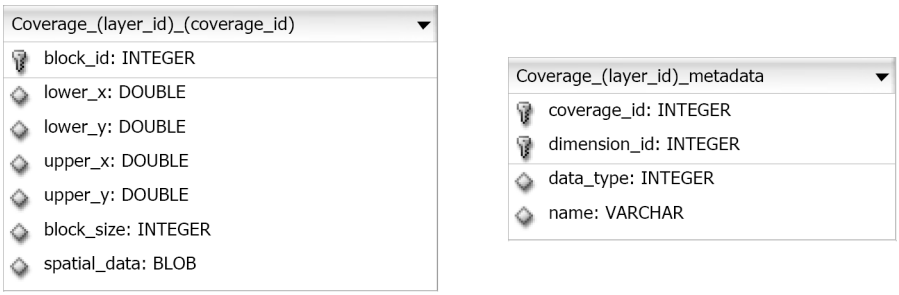


Figure 7. Diagrams for coverage tables and coverage metadata tables.

The primary key of this table is the block identifier, named `block_id`. For each stored block, there is also information about the minimum bounding box of all its geometries in the fields `lower_x`, `lower_y`, `upper_x` and `upper_y`, and information about its total size (in bytes) in the field `block_size`. The block contents are stored in a BLOB field named `spatial_data`, which contains a serialization of the geometries and values.

To decode the raw data stored in registers, the application needs to know the structure of the value vector of each coverage, i.e., what are the dimensions and the data

types used. For that matter, a metadata table is available, containing this kind of information. There is a single metadata table for all coverages in the same layer.

The primary key of this table is composed by the coverage identifier, named `coverage_id` and the dimension identifier, named `dimension_id`. The field `data_type` contains information about the data type of this dimension and the field `name` contains a name for the dimension (e.g., "Temperature", "Wind speed"). The name of this table has the form `Coverage_(layer_id)_metadata`, where `(layer_id)` means the layer identifier. Figure 7 shows the diagram of a coverage metadata table.

4. Conclusions and future work

We addressed in this paper the problem of formalizing coverage representations in the context of the TerraLib project. We described a common interface for all coverage types and implemented the simplest coverage representation, namely a Discrete Point Coverage. We started from the OGC abstract specification, but we also included requirements that addressed performance and other non-functional questions.

Future work includes the implementation of other coverage representations, such as Surface Coverages, along with adapting representations that are already part of TerraLib so that all implement a common interface. A memory caching mechanism, common for all coverage representations, is also a major issue, considering the performance of processing large amounts of data. We also want to make tests in applications that are built with TDK [TDK 2007] in order to acquire empirical evidence that *coverages* and *features with geometry* should be treated separately for better performance.TDK is an API that provides components to make it easier to implement GIS applications using Terralib.

5. References

Câmara, G. et al. (2000) "TerraLib: Technology in Support of GIS Innovation", II Workshop Brasileiro de Geoinformática, Caxambu, Brazil.

Couclelis, H. (1992) "People manipulate objects (but cultivate fields)", Proceedings of International Conference on GIS, Pisa, Italy.

OGC (2000) "The OpenGIS Abstract Specification - Topic 6: The Coverage Type and its Subtypes", version 6.0, Open Geospatial Consortium Inc, USA.

OGC (1999) "The OpenGIS Abstract Specification - Topic 5: Features", version 4.0, Open Geospatial Consortium Inc, USA.

Ramsey, P. (2007) "The State of Open Source GIS", Refractions Research Inc, Canada.

TDK (2007) "Terralib Development Kit", http://www.tecgraf.puc-rio.br/tdk.

Vinhas, L. (2006) "Um subsistema extensível para o armazenamento de geo-campos em bancos de dados geográficos", PhD Thesis, INPE, São José dos Campos, Brazil (in Portuguese).

Vinhas, L. and Ferreira, K. R. (2005) "Descrição da TerraLib", In "Bancos de Dados Geográficos", Edited by M. Casanova et al , Editora MundoGEO, Brazil (in Portuguese).

Vinhas, L. and Souza, R. C. M. (2005) "Tratamento de dados matriciais na TerraLib", In "Bancos de Dados Geográficos", Edited by M. Casanova et al, Editora MundoGEO, Brazil (in Portuguese).

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 251-256.

Representação das Características do Movimento de Objetos Móveis em Mapas Estáticos

Daniel S. Cotrim[1], Jorge Campos[1]

[1]Núcleo de Pesquisa em Redes e Computação (NUPERC) – Universidade Salvador (UNIFACS) – Salvador – BA - Brasil

daniel@gmail.com, jorge@unifacs.br

Abstract. *The widespread of global position system devices and wireless network promoted a notable evolution of technologies for storing and manipulating information related to moving objects. The large amount of data generated and the spatio-temporal characteristics of these data, however, made the job of analyzing and exploring such kind of information a complex task. The goal of this paper is to propose a new visualization model capable to communicate the characteristics of moving objects´ movement in a static map. We hope that the proposed model helps in the task of analyzing and understanding the behavior of moving objects and to identify relationships among these objects and between these objects and geographic events.*

Resumo. *A difusão de dispositivos de posicionamento global e redes sem fio motivaram uma evolução notável de ambientes e tecnologias para armazenamento e manipulação de informações relativas a objetos móveis. O grande volume de dados gerados e suas características espaço-temporal, entretanto, tornam a análise e exploração destes dados uma tarefa cada vez mais complexa. O objetivo deste trabalho é propor um modelo de visualização capaz de comunicar as características do movimento dos objetos móveis em mapas estáticos. Espera-se que o modelo proposto auxilie a análise do comportamento dos objetos móveis e das relações entre estes objetos e entre os objetos móveis e eventos geográficos*

1. Introdução

O crescente volume de dados com atributo espaço-temporal tem desafiado a capacidade dos analistas de consumir e obter conhecimento a partir destes dados [Yao 2003], [Harms, Deogun and Goddard 2003]. A análise desses dados é um campo da área de Descoberta do Conhecimento que trata da extração de conhecimento, estabelecimento de relações e identificação de padrões não armazenados explicitamente na base de dados e tem atraído interesse tanto da academia quanto da indústria [Roddick and Spiliopoulou 2002]. A inclusão de atributos espaciais e temporais, entretanto, adicionou uma complexidade substancial às técnicas tradicionais de mineração visual de dados e descoberta de conhecimento. Desta forma, a questão da espacialidade e temporalidade dos dados tornou-se um ponto crucial no entendimento dos eventos e processos geográficos.

Sistemas de Informações Geográficas (SIGs) lidam principalmente com a exploração, análise e apresentação de dados georeferenciados. Os modelos gráficos para a apresentação destes dados, entretanto, têm-se mostrado difíceis de serem utilizados em situações que requerem a análise de um crescente volume de informações que mudam com o passar do tempo [Verbree, et al. 1999].

Este artigo apresenta os resultados preliminares de um modelo capaz de representar as características do movimento de objetos móveis em mapas estáticos e está estruturado da seguinte forma. A seção dois discute algumas técnicas cartográficas para a apresentação de informações com características espaço-temporais. A seção três introduz um possível modelo para apresentar de forma visual, fácil e intuitiva as características do movimento de objetos móveis em mapas estáticos. A seção quatro apresenta as conclusões e indica trabalhos futuros.

2. Tipos de Apresentação de Objetos Móveis

SIGs podem ser definidos como uma combinação de sistemas de gerenciamento de banco de dados, um conjunto de operações para examinar estes dados e um dispositivo gráfico para apresentação e análise espacial [Rhyne 1997]. Para o propósito de apresentação e análise, as aplicações SIG utilizam na sua maioria uma interface gráfica bidimensional e estática. O crescente volume de dados espaço-temporais, entretanto, demandam um ambiente computacional mais cognitivo para lidar com este tipo de informação [Kraak et al. 1999].

Uma das principais limitações das aplicações SIG está relacionada com os modelos para apresentação e análise de objetos dinâmicos e móveis. Um objeto dinâmico é aquele no qual qualquer um dos seus atributos, espacial ou temático, muda de estado com o tempo. Um objeto móvel é qualquer objeto, pontual ou com extensão, que muda o valor do seu atributo espacial com o passar do tempo.

O maior desafio na apresentação dos objetos móveis é comunicar as características básicas do movimento do objeto, isto é, sua trajetória, velocidade e aceleração. Informações adicionais como o tempo durante o qual o objeto ficou em repouso, por exemplo, e a identificação de possíveis rotas futuras são informações que podem ajudar aos analistas no entendimento e na exploração do comportamento do objeto e na interação deste objeto com o espaço e com outros objetos. Representar visualmente estas características de forma clara e intuitiva, entretanto, é um desafio para a comunidade de Sistemas de Informações Geográficas.

Tradicionalmente, cartógrafos utilizam três modelos para representar dados geoespaciais que variam com o tempo: mapas estáticos, séries de mapas estáticos e mapas animados [Kraak 2003]. Os mapas estáticos utilizam variáveis visuais específicas e símbolos para denotar mudanças ocorridas com a passagem do tempo. A série de mapas estáticos é formada por uma seqüência temporal de mapas onde cada mapa representa um instante no tempo e juntos representam o histórico do movimento. O mapa animado é geralmente empregado na apresentação de mudanças contínuas e pode ser visto como uma seqüência mais refinada de uma série de mapas estáticos. Esta seqüência é apresentada de forma automática em uma taxa que cria a ilusão de continuidade do movimento. Este artigo está relacionado com o primeiro modelo. O desafio aqui é incorporar na apresentação de mapas estáticos informações que tradicionalmente só são exploradas em seqüência de mapas estáticos e mapas animados.

A primeira tentativa para a apresentação das características do movimento em um mapa estático foi concebida por Charles Minard [Friendly and York 1999]. O mapa de Minard ilustra a evolução das tropas Napoleônicas na campanha russa de 1812-1813 (Figura 1). Este mapa apresenta uma visão alternativa e intuitiva, diferente das apresentações tradicionais, e demonstra a perda dramática do exército de Napoleão durante a campanha contra a Rússia.

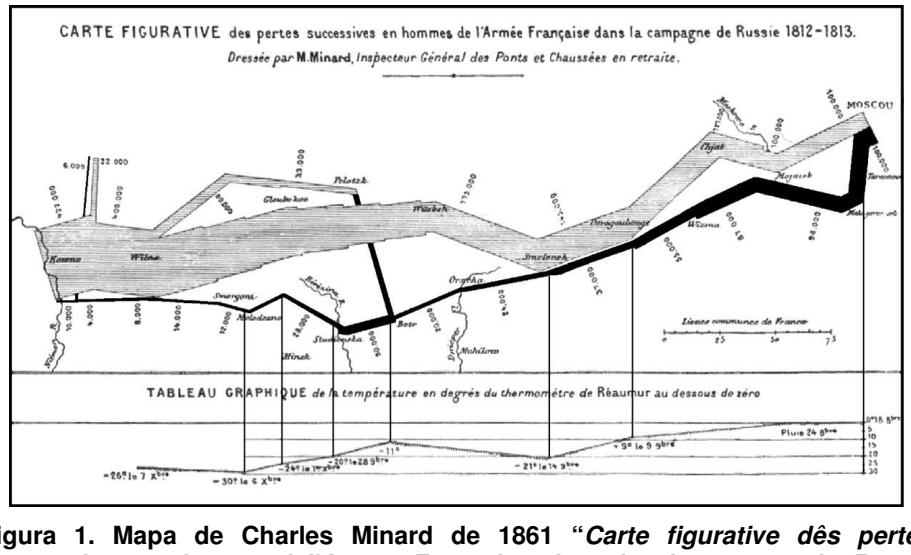


Figura 1. Mapa de Charles Minard de 1861 "*Carte figurative dês pertes successive em homes del'Armee Française dans la champagne de Russia 1812-1813*". A marcha de Napoleão em Moscou.

O mapa de Minard é um exemplo de representação de objetos móveis de forma criativa e que explora a utilização de recursos visuais e modelos gráficos para comunicar o conhecimento. Este mapa explora diversas variáveis relacionados ao movimento das tropas. A primeira variável é a trajetória do movimento, representada pelo caminho em um mapa da região. A segunda variável é o tempo. O tempo, neste exemplo, é representado de forma implícita e com um nível de granularidade bastante baixo. Pode-se inferir do mapa, por exemplo, que a campanha ocorreu entre 1812 e 1813 e que houve neste período um único avanço e retirada do exército napoleônico. O avanço e retirada são indicados pelo caminho claro e escuro respectivamente. Desta forma, não se pode precisar as datas e a duração de segmentos individuais da campanha e nem tão pouco a relação entre a duração do avanço e do retrocesso. Outro elemento gráfico explorado no mapa é a espessura da linha representando a trajetória das tropas. Esta espessura denota o contingente de soldados em cada posição. A variação da espessura evidencia as dramáticas baixas ocorridas durante a evolução da campanha. Finalmente, outros recursos gráficos são utilizados para apresentar informações sobre a campanha. Ligado ao caminho da retirada, por exemplo, tem-se um gráfico que representa a variação de temperatura. Rótulos são utilizados no mapa para identificar características geográficas relevantes e as principais batalhas.

De forma análoga aos mapas estáticos, o avanço das tropas de Napoleão pode ser demonstrado através de uma série de mapas (Figura 2). Nesta série, cada mapa apresenta a posição e configuração da tropa em diferentes instantes [Kraak 2003]. Como o movimento é representado somente por quadros considerados relevantes, não se tem a visão completa da trajetória. Desta forma, na apresentação através de série de mapas perde-se a transição entre os quadros. Além disso, os mapas estáticos e as séries de mapas são incapazes de apresentar todas as características do movimento. Informações como velocidade ou tempo em que a tropa ficou parada, por exemplo, não são capturadas por estas técnicas.

Figura 2. Dois snapshots com a localização da tropa francesa em 24 de Julho e 24 de Agosto de 1812.

Uma alternativa para expressar algumas características do movimento dos objetos móveis ou mudanças nos atributos não espaciais tem sido a apresentação em mapas animados. A despeito do formidável apelo visual, a animação não é universalmente preferida pelos analistas e podem, em alguns casos, distrair o usuário e prejudicar o entendimento dos dados [Morrison and Betrancourt 2002].

3. Apresentando Objetos Móveis em Mapas Estáticos

A concepção de um modelo para apresentação de objetos móveis em mapas estáticos fundamentou-se nos princípios da semiologia gráfica. A semiologia gráfica é uma medologia desenvolvida para codificar informações em uma linguagem gráfica. Esta metodologia se baseia em um sistema de símbolos utilizados para comunicar dados reias ou conceitos abstratos (componentes) através de modelos gráficos. A semiologia gráfica busca propiciar a percepção imediata e apreensão clara dos componentes através de um sistema semântico baseado em regras relacionadas aos signos.

Jacques Bertin, em seu trabalho seminal "Semiologia Gráfica", foi o primeiro autor a propor formalmente um conjunto de variáveis para descrever graficamente os componentes. As principais variáveis gráficas propostas por Bertin para apresentar os componentes são *localização*, *tamanho*, *valor*, *granulação*, *orientação*, *cor* e *forma*.

As variáveis gráficas podem ser classificadas em variáveis do plano e variáveis visuais. A localização é uma variável do plano enquanto as demais são classificadas como variáveis visuais. As variáveis visuais, também conhecidas como variáveis retinais, são baseadas na capacidade dos seres humanos de possuírem reações preconcebidas a essas variáveis no nível de processamento da retina.

Diversas extensões das variáveis gráficas de Bertin têm sido proposta nas últimas décadas pela comunidade cartográfica. De especial importância neste trabalho é a variável *saturação* da cor [MacEachren 1994]. A variável saturação de cor apresenta claramente uma ordem visual.

A materialização dos componentes em um mapa é feita através da combinação de uma forma geométrica e de uma ou mais variáveis gráficas. A forma de apresentação de um componente pode ser pontual, linear ou zonal. A definição das variáveis gráficas depende do significado da informação a ser transcrita.

De forma a facilitar a seleção das variáveis gráficas, os componentes grafados em quatro níveis: espacial, quantitativo, qualitativo e ordenado. O nível espacial define a posição no espaço. No nível quantitativo, a distância visual entre duas categorias é expressa por uma relação numérica. O nível qualitativo pode ser associativo ou seletivo, sendo que o primeiro exprime comparação entre os elementos, e o segundo diferenciação. O nível ordenado permite a classificação visual das categorias.

Baseado nos princípios da semiologia gráfica, este artigo propõe uma combinação de variáveis visuais e do plano para apresentar os principais componentes que descrevem o comportamento de um objeto móvel, a saber: identidade, caminho percorrido, tempo, velocidade e duração do repouso (Tabela 1).

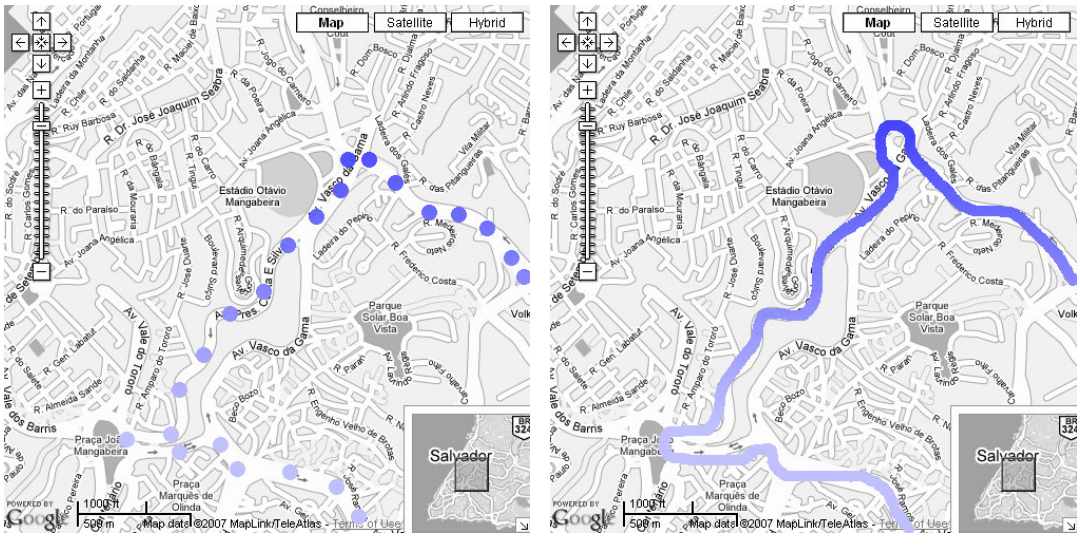
Tabela 1. Componentes e variáveis visuais

Componente	**Nível**	**Forma de Representação**	**Variável Visual**
Identidade (Objeto Móvel)	Seletivo	Linear	Cor
Caminho Percorrido	Espacial	Linear	Localização
Tempo	Ordinal	Linear	Saturação
Velocidade	Ordinal	Linear	Tamanho
Duração do Estado em Repouso	Ordinal	Pontual	Tamanho

A identidade do objeto é apresentada pela variável cor, exprimindo o nível seletivo. O *caminho percorrido* pelo objeto é um componente estritamente associado à localização e é desenhado de forma linear no mapa. O *tempo* é representado pela variável saturação, provendo o conceito de ordem. A *velocidade*, por sua vez, é representada pelo tamanho (espessura) da linha e está inserida no nível ordinal. A *duração do estado de repouso* é representada por um ponto e possui nível ordinal. Os componentes velocidade e duração do estado de repouso são classificadas como ordinais, pois o interesse do modelo é representar uma comparação visual entre as categorias de cada componente.

De forma a ilustrar a apresentação visual dos componentes citados e evidenciar a evolução do modelo proposto sobre as técnicas tradicionais de apresentação de objetos móveis, apresentamos a evolução do movimento de um veículo em um ambiente urbano (Figura 3). Em todos os mapas é aplicada uma gradação de cor que varia do tom mais escuro para o tom mais claro, associando o atributo tempo à localização do veículo. Desta forma, a tonalidade mais clara está associada ao instante inicial do movimento e o tom mais escuro ao instante final. A utilização da cor para representação do tempo permite que a variação do tempo seja percebida de forma eficaz em todas as apresentações, embora com diferentes níveis de granularidade.

A Figura 3a mostra a rota do veículo através de pontos periódicos de sua localização. Esta é a representação mais primária do movimento e possui alguns inconvenientes. A informação sobre a localização entre os pontos é perdida, isto é, esta visualização não permite captar as posições intermediárias do movimento. Desta forma, não é possível visualizar a trajetória completa do veículo. A noção de velocidade está condicionada a amostragem da localização. Se a posição do objeto é armazenada em intervalos regulares de tempo, a velocidade pode ser inferida pelo espaçamento das amostras. Se a amostragem privilegia posições significativas do movimento, a velocidade do veículo não pode ser extraída da apresentação.



a) b)

c)

Figura 3. Diversas representações em mapas estáticas do movimento de um veículo em um ambiente urbano.

A Figura 3b aperfeiçoa a apresentação da rota do veículo detalhando o percurso. Os pontos intermediários da rota já são visíveis. É possível ainda identificar mais detalhadamente a variação no tempo e a localização da viatura em cada instante. Entretanto, não se sabe o comportamento do objeto durante a trajetória. A velocidade está oculta nesta apresentação.

A Figura 3c é a apresentação do modelo proposto. Os inconvenientes das representações anteriores foram solucionados, permitindo o entendimento completo da trajetória e do comportamento do objeto. A variação da velocidade e o tempo em que o objeto ficou em repouso estão explícitos nesta representação. Entende-se que quanto mais rápido o objeto, menos contato ele tem com o meio e quanto maior o tempo parado maior a relação dele com aquela localização. Neste sentido a velocidade do objeto

é inversamente proporcional a espessura da linha da trajetória, quanto mais fina o intervalo maior a velocidade do objeto. A mesma abstração é feita para o estado de repouso do objeto. Um círculo ao redor do ponto da trajetória indica que o objeto ficou parado, quanto maior o tempo de repouso maior o círculo.

4. Trabalhos Futuros

Este trabalho apresentou os primeiros resultados de um modelo para apresentar informações em mapas estáticos que permitam a análise e exploração visual do comportamento de objetos com variação dos atributos espaciais e temporais. A concepção de modelos e de técnicas para a análise de dados dinâmicos constitui uma peça fundamental na identificação de padrões e produção de conhecimento para a comunidade SIG, em particular, e para diversos ramos da ciência, em geral.

O modelo inicialmente proposto contempla a representação das características do movimento dos objetos considerando o histórico da movimentação em um curto espaço de tempo, isto é, a nas últimas horas ou dias. O modelo, entretanto, não se mostrou adequado ou robusto o suficiente para a representação das características do movimento com dados históricos de um longo período de tempo, isto é, meses ou anos. Conseqüentemente, deverão ser realizados estudos e adaptações do modelo original para contemplar a apresentação e análise das características do movimento dos objetos de forma a tratar dados históricos que envolvam um período maior de tempo.

Agradecimento

Este trabalho é financiado pela Fundação de Amparo à Pesquisa do Estado da Bahia - FAPESB através do projeto PET0007/2005.

Referências

Friendly, M. and York University (1999) “Re-Visions of Minard”, Statistical Computing and Graphics Newsletter 11.

Harms, S. K., Deogun, J. and Goddard, S. (2003) “Building Knowledge Discovery into a Geo-Spatial Decision Support System”, ACM Symposium on Applied Computing, pages 445-449.

Kraak, M.-J. (2003) “Geovisualization illustrated”, ISPRS Journal of Photogrammetry & Remote Sensing 57, pages 390-399.

Kraak, M.-J., G. Smets and P. Sidjanin (1999) Virtual Reality, the New 3-D Interface for Geographical Information Systems. in: A. S. Câmara and J. Raper (Eds.) Spatial Multimedia and Virtual Reality, London, Taylor & Francis, pages 131-136.

MacEachren, A. M. (1994) “Visualization in modern cartography: setting the agenda”, Visualization in Modern Cartography: Pergamon, pages 1-12.

Morrison, J. B. and Betrancourt, M. (2002) “Animation: Can It Facilitate?”, International Journal of Human-Computer Studies 57(4), pages 247-262.

Rhyne, T. M. (1997) Going Virtual with Geographic Information and Scientific Visualization. Computer & Geosciences 23(4), pages 489-491.

Roddick, J. F. and Spiliopoulou, M. (2002) “A Survey of Temporal Knowledge Discovery Paradigms and Methods”, IEEE Transaction on Knowledge and Data Engineering 14(4), pages 750-767.

Verbree, E., G. V. Maren, R. Germs, F. Jansen and M.-J. Kraak (1999) Interaction in Virtual World Views - Linking 3D GIS with VR. International Journal of Geographical Information Science. 13(4), pages 385-396.

Yao, X. (2003) “Research Issues in Spatio-Temporal Data Mining”, Workshop on Geospatial Visualization and Knowledge Discovery.

Modelagem espacial de florestas estacionais do Domínio do Cerrado no Estado de Minas Gerais utilizando envelope climático

Gleyce Campos Dutra[1], Luis Marcelo Tavares de Carvalho[1], Ary Teixeira de Oliveira Filho[1]

[1]Departamento de Ciências Florestais – Universidade Federal de Lavras (UFLA)
Caixa Postal 3037 - CEP 37200-000 – Lavras – MG – Brasil

gleycedutra@yahoo.com.br, passarinho@ufla.br, ary@ufla.br

Abstract. *The general objective of this work is to map the potential distribution of seasonal forest of Cerrado in Minas Gerais, Brazil, through the species distribution modeling of its indicative species. Occurrence data and list of species of the areas were recovered in the database TreeAtlan 1.0. The environmental datasets used for the work involved climatic coverings related with temperature and precipitation. For modeling species distribution, the algorithm used was Bioclim True/False. Partial results allowed to observe that seasonal forest can cover 63% of Minas Gerais only be based in given bioclimatic dataset. The area under the ROC curve indicated a satisfactory accuracy.*

Resumo. *O objetivo geral deste trabalho é mapear a distribuição potencial de florestas estacionais do Domínio do Cerrado dentro do estado de Minas Gerais por meio da modelagem de distribuição espacial de suas espécies indicadoras. Os pontos de ocorrência e lista de espécies das áreas foram recuperados no banco de dados TreeAtlan 1.0. As bases ambientais utilizadas para o trabalho compreendem coberturas climáticas relacionadas com temperatura e precipitação. Para a modelagem de distribuição das espécies foi utilizado o algoritmo Bioclim True/False. Resultados parciais permitiram observar que a fitofisionomia possui o potencial de ocupar 63% do Estado. A área sob a curva ROC indica uma acurácia satisfatória.*

1. Introdução

O Estado de Minas Gerais possui particularidades relacionadas a clima e solo que, combinadas com as diferentes formas de relevo, proporcionam paisagens muito variadas, recobertas por vegetações características, adaptadas a cada um dos vários ambientes particulares inseridos em 3 domínios fitogeográficos brasileiros: Cerrado, Mata Atlântica e Caatinga.

Dentre esses domínios fitogeográficos, destaca-se o Cerrado, por apresentar fisionomias variadas, indo desde campos limpos desprovidos de vegetação lenhosa a cerradão, uma formação arbórea densa. Seu clima é particularmente marcante, apresentando duas estações bem definidas. A vegetação muda de Cerrado ralo para denso ou vice-versa, em função da ocorrência de queimadas, entre outros fatores. Mesmo em áreas muito próximas, as florestas substituem o Cerrado devido a mudanças

na umidade do solo e pela ocorrência de afloramentos calcáreos, onde a fertilidade é elevada (Felfili et al.,1997).

A classificação de um determinado domínio pode ser feita com facilidade em áreas representativas das diferentes formações, chamadas de áreas nucleares (Durigan, 2003). Em algumas regiões, no entanto, onde há transição entre formações ou entre associações, o mapeamento torna-se mais difícil. A diferenciação se faz, nesses casos, com o auxílio de resultados de pesquisa sobre a flora local ou com a presença de espécies indicadoras.

A modelagem de distribuição geográfica potencial de espécies, baseada em conceitos de nicho ecológico, tem se destacado com uma importante ferramenta de análise para dar suporte às políticas de conservação e ao planejamento de estratégias de recuperação de diversas áreas. É possível gerar que mapas indicam a provável presença e ausência de uma espécie, em função de variáveis ambientais relevantes (Siqueira & Peterson, 2003). Até recentemente, muitos modelos de distribuição de espécie foram baseados em técnicas de envelopes ambientais, como, por exemplo, o Bioclim.

Sendo assim, o objetivo geral deste trabalho é mapear a distribuição potencial florestas estacionais do Domínio do Cerrado dentro do Estado de Minas Gerais, por meio da modelagem de distribuição geográficas de suas espécies indicadoras utilizando o Bioclim, a partir de informações pontuais fornecidas pelos fragmentos estudados pelo Departamento de Ciências Florestais da Universidade Federal de Lavras - UFLA

2. Materiais e Métodos

Os pontos de ocorrência e lista de espécies das áreas de florestas estacionais decíduas e semidecíduas do Domínio do Cerrado foram recuperados no banco de dados TreeAtlan 1.0 para o Estado de Minas Gerais (Oliveira-Filho, 2007), totalizando 50 áreas (Figura 1). Foram utilizadas 19 variáveis climáticas com resolução espacial de 2,5º x 2,5º, relacionados à temperatura e à precipitação, disponíveis para o programa DIVA-GIS 5.1 (Hijmans et al, 2004).

As espécies indicadoras do Domínio do Cerrado foram selecionadas pela Análise de Espécies Indicadoras (ISA) (Dufrêne & Legendre, 1997), na qual um valor percentual de uma espécie particular é estimado dentro de uma região geográfica particular, que serão as áreas de ocorrência de florestas estacionais do Domínio do Cerrado em Minas Gerais. Essa análise combina informação sobre a concentração da abundância de uma espécie em um determinado grupo de unidades amostrais e da fidelidade da ocorrência desta espécie em certo grupo de amostras.

Os pontos de ocorrência de cada uma das espécies indicadoras selecionadas, associadas às bases ambientais, foram usados para modelar a sua distribuição geográfica potencial aplicando o algoritmo Bioclim True/False (Nix, 1986). As áreas que estão dentro do envelope, delineadas pelos pontos de ocorrência, foram projetadas no espaço geográfico apresentando valores de 1 para áreas de potencial presença e valor 0 para ausência das espécies. Foi estabelecido um percentil de corte de 2,5%. A modelagem foi realizada no programa DIVA-GIS 5.1. Em seguida, os modelos gerados para cada espécie indicadora selecionada foram somadas num ambiente SIG. O valor de cada pixel é igual ao número de modelos individuais combinados.

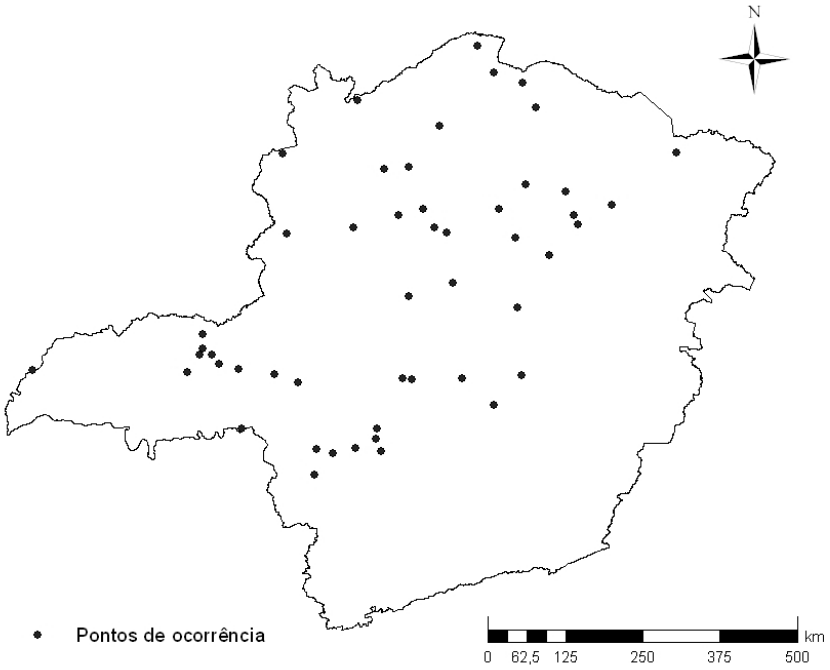


Figura 1: Localização dos fragmentos de florestas estacionais do Domínio do Cerrado amostrados no Estado de Minas Gerais

A validação dos modelos gerados foi feita testando um conjunto de dados independentes (Fielding, 1999), utilizado o chamado gráfico do receptor-operador (ROC-plot), no qual são representadas as frações dos verdadeiros positivos contra os falsos positivos. A área sob a curva é tomada como uma medida de acurácia do modelo e caracteriza o seu desempenho (Phillips et al., 2006). As amostras de validação foram coletadas em incursões a campo durante o projeto "Mapeamento e Inventário da Flora Nativa e Reflorestamentos de Minas Gerais" (Scolforo & Carvalho, 2006).

3. Resultados e Discussões

Foram selecionadas 5 espécies arbóreas indicadoras para Domínio do Cerrado (*Myracrodruon urundeuva*, *Dilodendron bipinnatum*, *Tabebuia roseoalba*, *Aspidosperma subincanum*, *Tabebuia impetiginosa*). Estas espécies foram as que apresentaram valores mais significativos pelo índice ISA e maior número de pontos de ocorrência.

Na figura 2 observa-se que o modelo de distribuição potencial das Florestas Estacionais do Domínio do Cerrado, previu uma área de 373620,860 km², aproximadamente 63% do Estado. A área nuclear, considerada aqui a área representada por pixels que possuem a combinação de 5 modelos, sendo a que, portanto, possui maior potencial de ocorrência de florestas estacionais no Domínio do Cerrado, recobre parte do Triângulo Mineiro e uma grande faixa no sentido centro-noroeste, principalmente a bacia do Rio São Francisco.

Como a projeção do modelo no espaço geográfico representa a distribuição potencial das espécies consideradas, poucas espécies ocupam todas as áreas que satisfaçam a exigência de seu nicho, ou seja, algumas áreas, apesar de apresentarem

condições climáticas favoráveis para a ocorrência das espécies indicadoras, existem outros fatores que impedem a ocorrência daquela espécie num dado local como, por exemplo, barreiras geográficas à dispersão, interações bióticas (competição, predação) e ação antrópica, que são variáveis difíceis de mensurar (Phillips et al., 2006). A região bacia do Rio Doce, que apesar de possuir certo potencial de distribuição, para este trabalho não foi encontrado nenhum registro de ocorrência das 5 espécies indicadoras, provavelmente devido a fato do complexo da Serra do Espinhaço servir como uma barreira geográfica para a distribuição destas espécies.

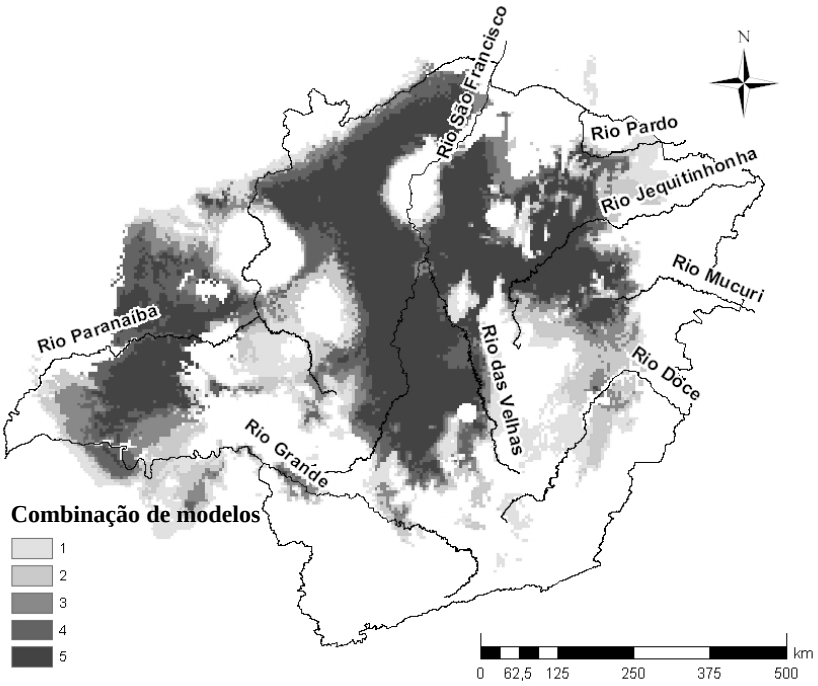


Figura 2. Distribuição potencial das Florestas Estacionais do Domínio do Cerrado no Estado de Minas Gerais. A escala de cinza determina o número de modelos combinados dentro de um pixel: do mais claro (valor 1) ao mais escuro (valor 5)

Podem-se observar, também, algumas lacunas no modelo, que podem tanto representar regiões inadequadas para a ocorrência de florestas estacionais do Domínio do Cerrado, como a faixa que segue de sul a leste no Estado, quanto a deficiência na amostragem e heterogeneidade na distribuição dos pontos de ocorrência ou limitações do algoritmo. Segundo Pearson & Dawson (2003) os modelos baseados em envelopes bioclimáticos consideram, em sua estrutura básica, somente as variáveis climáticas em seu processamento e não incluem outros fatores ambientais, como variáveis relacionadas a solos e vegetação. No entanto, esses modelos podem fornecer uma primeira aproximação muito útil nas análises iniciais.

A utilização de produtos de imagens de sensoriamento remoto auxiliaria na análise, ao incluir no modelo informações sobre habitats altamente alterados, no caso de espécies que ocorrem preferencialmente em áreas com maior ou menor densidade

vegetacional. Para isso será necessária a utilização de algoritmos mais poderosos como o GARP (Anderson et al, 2003) e o Maxent (Phillips et al., 2006).

A Curva ROC (Figura 3), compara as áreas estimadas no modelo com aquelas observadas no mesmo ponto pelas amostras de validação. Segundo Fielding (1999) quanto mais próxima de 1,0 e mais distante de 0,50 for a área sob a curva ROC, maior será a acurácia do modelo. A área sob a curva foi 0,709, o que indica que o modelo atingiu uma boa acurácia.

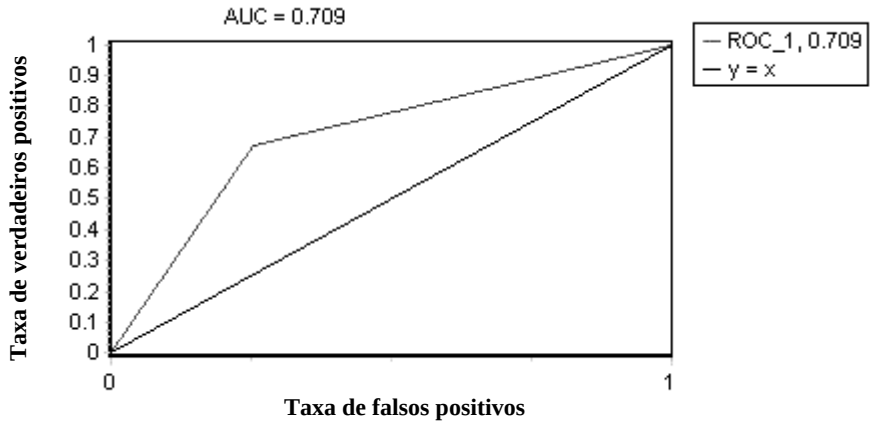


Figura 3. Curva ROC para o modelo de distribuição geográfica de florestas estacionais do Domínio do Cerrado. A reta na diagonal representa um modelo gerado aleatoriamente

4. Considerações finais

Para uma análise inicial, o Bioclim foi eficiente ao modelar a distribuição de florestas estacionais do Domínio do Cerrado, demonstrando que espécies indicadores de um ambiente são capazes de explicar sua distribuição geográfica.

As análises seguintes incluirão a atualização da base de dados, com a adição de pontos de ocorrência de fragmentos estudados pelo projeto "Mapeamento e Inventário da Flora Nativa e Reflorestamentos de Minas Gerais" e utilização de mais outra variedade de bases ambientais, que consistem em dados relativos a relevo e solos e, também, produtos de imagens do sensor MODIS.

Para maior refinamento da análise, esses dados também serão analisados utilizando os algoritmos GARP e Maxent.

5. Referências Bibliográficas

ANDERSON, R. P., D. LEW, & PETERSON, A. T. (2003). Evaluating predictive models of species' distributions: criteria for selecting optimal models. **Ecological Modelling** 162: 211-232

DUFRÊNE, M. & LEGENDRE, P. (1997). Species assemblages and indicator species: the need for a flexible asymmetrical approah. **Ecological Monograhs** 67 (3): 345-366.

DURIGAN, G. (2003) Métodos para análise de vegetação arbórea. In: L. CULLEN Jr.; R. RUDRAN; C. VALLADARES-PÁDUA. (Org.). **Métodos de Estudos em Biologia da Conservação e Manejo da Vida Silvestre**. Curitiba: UFPR / Fundação Boticário de Proteção à Natureza. p. 455-479.

FELFILI, J. M.; SILVA, M. C.; REZENDE, A. V.; NOGUEIRA, P. E.; WALTER, B. M. T.; FELFILI, M. C.; Silva, M. A. & ENCINAS, J. I. (1997). Comparação do cerrado (*sensu stricto*) nas Chapadas Pratinha e dos Veadeiros. *In*: LEITE, L. L. & SAITO, C. H. (Eds.). **Contribuição ao conhecimento ecológico do cerrado**. Departamento de Ecologia, Universidade de Brasília, Brasília-DF. p. 6-11.

FIELDING A.H. How should be accuracy be mensured? (1999) In: FIELDING A.H, ed.. **Machine learning methods for ecological applications**. Boston: Kluwer Academic. p.209-223.

HIJMANS, R.J., CAMERON, S.E.C., PARRA, J.L., JONES, P.G. & JARVIS, A. (2004). **The WorldClim interpolated global terrestrial climate surfaces**. Version 1.3. Disponível em http://www.diva-gis.org/Data.htm. Acessado em: maio/2007

NIX, H. (1986). A biogeographic analysis of Australian elapid snakes. Atlas of Elapid Snakes of Australian. Australian Government Publishing Service, Canberra, Australia, 4-15. Disponível em: http://cres.anu.edu.au/outputs/anuclim/doc/bioclim.html

OLIVEIRA-FILHO, A.T de. (2007) **TreeAtlan:** Flora arbórea da Mata Atlântica e domínios adjacentes: Um banco de dados envolvendo geografia, diversidade e conservação. Disponível em: http://www.treeatlan.dcf.ufla.br. Acessado em: maio/2007.

PEARSON, R. G.; DAWSON, T. P. (2003). "Predicting the impacts of climate change on the distribution of species: are bioclimate envelope models useful?" **Global Ecology & Biogeography** 12: 361–371.

PHILLIPS, S.J., R.P. ANDERSON & SCHAPIRE, R.E. (2006) Maximum entropy modeling of species geographic distributions. **Ecological Modelling**, 190:231-259.

SCOLFORO, J. R. S.; CARVALHO, L. M. T. (2006). **Mapeamento e inventario da flora nativa e reflorestamentos de Minas Gerais**. 1. ed. Lavras: Editora UFLA. v. 1. 288 p.

SIQUEIRA, M. F.; PETERSON, A. T. (2003). Consequences of global climate change for geographic distributions of cerrado tree species. **Biota Neotropica**, v. 3, n. 2, p. 1-14.

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 263-276.

ANÁLISE ESPACIAL DA DISTRIBUIÇÃO DO *AEDES AEGYPTI* (DIPTERA: CULICIDAE) EM DIFERENTES ÁREAS DA CIDADE DO RIO DE JANEIRO

Reis IC [1]; Honório N.A.[1]; Codeço C.T. [3], Barcellos C[2], Magalhães M.A.F.M[2]

[1]Laboratório de Transmissores de Hematozoários –Instituto Oswaldo Cruz, [2]Laboratório de Geoprocessamento – Fundação Oswaldo Cruz, [3]Programa de Computação Cientifica – Fundação Oswaldo Cruz.

(izareis@hotmail[1].com; honorio@fiocruz.br[1] ,codeco@fiocruz.br[3]; xris@fiocruz.br[2]; monica@cict.fiocruz.br[2])

ABSTRACT. This work aimed at assessing the spatial distribution of the dengue fever mosquito vector, *Aedes aegypti* and its association with Strategic Sites (SS). In each of three localities of Rio de Janeiro city, 80 traps were installed (40 for adult mosquitos and 40 for immatures) in randomly chosen households. Moreover, two further traps we installed close to each SS. These traps were localized with GPS and visited every week for 11 weeks, in the summer of 2007. Based on these data, mosquito infestation maps were created by applying Gaussian Kernels (ArcGis 9.0) to the weekly data. We also used non-parametric methods to test the hypothesis that traps located closer to the SSs would catch more mosquitos than distant ones. Our results suggest that unusual sites, as those used for recycling and a boat factory, showed significant association with infestation *hotspots*. As a conclusion, our study point to the importance of not standard types of human activity as potential new categories to be included in the surveillance of SSs.

RESUMO. Este trabalho tem por objetivo avaliar a distribuição espacial do mosquito vetor da dengue, *Aedes aegypti*, em três localidades da cidade do Rio de Janeiro, e sua associação com a presença de pontos estratégicos (PE). Para isso foram implantadas, aleatoriamente, um total de 80 armadilhas em cada área, sendo 40 para coletar formas imaturas (Ovitrampas) e 40 para mosquitos adultos (MosquiTraps) no peridomicilio e duas armadilhas em cada PE. Estas foram mapeadas com GPS e monitoradas semanalmente durante 11 semanas, no verão de 2007. Para análise dos resultados foram feitos mapas de kernel (ArcGis 9.0) por semana. Além disso, utilizamos métodos não paramétricos para testar tendências de aglomeração de armadilhas positivas próximo aos PE. Os resultados sugerem um papel importante para PEs associadas à reciclagem de materiais (na localidade de Palmares), à fabricação de barcos (em Tubiacanga) e a transportadora em Higienópolis.

1. INTRODUÇÃO

Em virtude das mudanças demográficas ocorridas nos países subdesenvolvidos, a partir da década de 60, com os intensos fluxos migratórios rural-urbanos, houve um inchaço das cidades propiciando, principalmente, nas periferias das grandes cidades condições inadequadas ou insuficientes de saneamento básico, abastecimento de água e coleta de lixo. Uma das conseqüências desta situação é o aumento do número de criadouros potenciais do principal vetor da dengue, *Aedes aegypti* (Tauil, 2001). Esse mosquito é altamente adaptado a ambientes urbanos e suburbanos onde a concentração populacional humana é elevada, há grande concentração de casas e moderada cobertura vegetal (Braks et al. 2003).

Tais características permitem que estes vetores sejam abundantes nas cidades e facilmente levados para outras áreas de forma passiva, por meios de transportes, aumentando assim sua dispersão que pode ocorrer em todas as fases do seu desenvolvimento - ovo, larvas, pupa e adulto (Teixeira et al. 1999; Forattini 2002; Honório et al. 2003).

As formas imaturas desse mosquito se mantêm, principalmente, em pneus, caixa d'água, vasos de plantas, latas, garrafas, bebedouros de animais, objetos que retenham água. Esses criadouros artificiais são encontrados mais facilmente em pontos estratégicos (PE), que segundo o Programa de Vigilância e Controle dos Vetores de Dengue e Febre Amarela no estado de São Paulo (Sucen, 2002), são os imóveis de maior importância na geração e dispersão ativa e passiva do *Ae. aegypti*, pois podem apresentar grandes quantidades de recipientes em condições favoráveis a proliferação de larvas (depósitos de

pneus usados e de ferro velho, borracharias, cemitérios, entre outros) ou pequenas quantidades de recipientes tais como transportadoras, rodoviárias, portos e aeroportos, dentre outros. Esses pontos estratégicos merecem grande atenção dos órgãos de saúde, pois podem contribuir para o aumento do vetor e da doença na área. Recentemente, a utilização do Sistema de Informação Geográfica (SIG) na saúde tem contribuído para o desenvolvimento de modelos que visam a prevenção da transmissão de várias doenças no espaço urbano através do mapeamento dos seus casos e dos locais mais vulneráveis à presença do vetor primário do dengue, o *Ae. aegypti* (Barcellos et al. 2005).

2. OBJETIVO

O objetivo deste trabalho é avaliar, no período referente ao verão, a distribuição de *Ae. aegypti* em três localidades do Rio de Janeiro e sua associação com a presença de pontos estratégicos.

3. MATERIAL E MÉTODOS

3.1. Área de estudo - O estudo foi realizado no período de 08/01/07 a 22/03/2007, que corresponde ao verão. Foram escolhidas três áreas do Rio de Janeiro: Higienópolis, Vargem Pequena (Palmares) e Ilha do Governador (Tubiacanga), que são áreas com diferentes níveis de densidade populacional, cobertura vegetal e histórico de dengue.

O bairro de Higienópolis (22º52’25’’S 43º15’41’’W) está localizado na zona norte da cidade do Rio de Janeiro e apresenta uma área altamente urbanizada com 16.587 habitantes,

baixa cobertura vegetal e regularização de serviços de limpeza e sistema geral de esgoto (IBGE, 2000). Tubiacanga (22º47'08''S 43º13'36''W), comunidade localizada na Ilha do Governador, é cercada parcialmente pela Baía de Guanabara e apresenta uma população de aproximadamente 2.900 habitantes, com moderada cobertura vegetal e irregularidades no abastecimento de água (Marciel-de-Freitas, et al. 2006). Vargem Pequena, bairro onde a comunidade de Palmares está situada, possui 11.536 habitantes (IBGE, 2000). A comunidade de Palmares (22º 59'26'' S 43º 27' 36'' W) apresenta alta cobertura vegetal, irregularidades no abastecimento de água e a principal atividade econômica é a reciclagem de material utilizado pela comunidade e áreas vizinhas.

3.2. Inquérito Entomológico - Em cada área de estudo, foram selecionados, aleatoriamente, 80 domicílios e localizados todos os pontos estratégicos. Foram instaladas 40 armadilhas de oviposição (Ovitrampas) e 40 armadilhas para a coleta de adultos (MosquiTrap) em 80 domicílios, perfazendo um total de 240 armadilhas nas três localidades, com o propósito de monitorar a distribuição do *Ae. aegypti*. Cada residência selecionada recebeu 1 ovitrampa ou 1 MosquiTrap, que ficou implantada no ambiente peridomiciliar por 7 dias. Nos pontos estratégicos foram implantadas duas armadilhas (1 MosquiTrap e 1 ovitrampa) totalizando 10 armadilhas por área, que foram mapeadas utilizando GPS (Global Position System). As armadilhas foram monitoradas, semanalmente, as paletas das ovitrampas e os cartões das MosquiTraps foram transportados para o Laboratório de Transmissores de Hematozoários do Instituto Oswaldo Cruz e Setor de Controle de Vetores e Pragas-Uadema/Dirac/IOC. No laboratório, as formas imaturas e os adultos foram identificados através da chave dicotômica de Consoli & Lourenço-de-

Oliveira (1994) e contados. Foram obtidos os dados pluviométricos das estações mais próximas de cada área de estudo: para Higienópolis utilizamos os dados da Penha, Ilha do Governador (Tubiacanga) e Riocentro (Palmares), que foram obtidos através da GeoRio.

3.3 Análise dos dados - Na análise estatística dos dados, foram construídos mapas de kernel para cada semana utilizando o *software* ArcGis 9.1 no laboratório de Geoprocessamento – ICICT. Para testar a associação entre ponto estratégico e positividade das armadilhas, foram construídos gráficos para a distribuição acumulada de armadilhas positivas em função da distância das armadilhas em relação a cada ponto estratégico. A partir deste gráfico, calculou-se os quartis da distribuição, isto é, a distância do ponto estratégico onde se encontravam 25%, 50% e 75% das armadilhas positivas. Para testar se esta distância era menor do que seria observado por acaso, foi utilizado método de randomização.

4. RESULTADOS E DISCUSSÃO

Ao todo, foram identificados 2.023 mosquitos adultos, sendo 1.242 (61,4%) *Ae. aegypti*, 91 (4,5%) *Ae. albopictus* e 690 (34,1%) *Culex quinquefasciatus* coletados nas MosquiTraps. Nas ovitrampas foram identificados 116.777 ovos que corresponderam a 106.775 larvas, sendo 99.989 (93,6%) de *Ae. aegypti*, 2.708 (2,5%) *Ae. albopictus* e 4.078 (3,9%) *Culex quinquefasciatus.* Não observamos forte correlação entre pluviosidade e densidade do *Ae. aegypti* nas áreas de estudo.

Das três áreas de estudo, Tubiacanga apresentou a maior densidade de larvas (50,4%) e adultos (26,9%) de *Ae. aegypti*, seguido por Higienópolis (29,5% e 23,6 respectivamente). Apesar de Tubiacanga não ser um ambiente totalmente urbano, sendo cercado pela Baía de Guanabara e com moderada cobertura vegetal, o armazenamento de água em tonéis, caixa d'água e recipientes na maioria dos casos de forma precária tem

favorecido o estabelecimento do *Ae. aegypti* e dessa forma tem contribuído para o aumento de criadouros na área sendo vulneráveis a manutenção do vetor.

Através dos mapas de kernel, por semana, considerando todas as etapas de desenvolvimento do mosquito (ovo, larva e adulto) observamos em Tubiacanga a existência de *hotspots* constantes e próximos aos pontos estratégicos: fábricas de barcos localizados próximos a Baía de Guanabara e ferro velho. Na localidade de Palmares há *hotspots* muito próximos as áreas utilizadas para reciclagens, principal fonte de renda dessa localidade. Enquanto que em Higienópolis existe uma dinâmica dos *hotspots* em toda área de abrangência, observando em algumas semanas uma possível associação com os pontos estratégicos.

A análise estatística da distribuição das armadilhas positivas em relação aos pontos estratégicos sugere que a oficina de barcos em Tubiacanga, a área utilizada para armazenamento e triagem de material para reciclagem, em Palmares, e a transportadora em Higienópolis, foram significativamente associados a altos índices de infestação.

Um estudo realizado em São José do Rio Preto detectou a presença de 191 PE, dos quais apenas 14 foram positivos para larvas do mosquito *Ae. aegypti*, que foram coletadas em borracharias, depósitos de pneus, recauchutadoras, lojas e depósitos de matérias de construção (Neto 1997). Lopes *et al* (1993), realizou um estudo em Lodrina-PR em 30 tipos de ambientes de risco incluindo três favelas da região para *Ae. aegypti* onde observou a presença de imaturos de *Ae. aegypti* apenas no ferro velho (40 espécimes) e no terreno baldio (275 espécimes).

Os resultados desta pesquisa demonstraram que as fábricas de barcos em Tubiacanga, e as reciclagens localizadas dentro da comunidade de Palmares, merecem

grande atenção dos agentes de saúde, devido a presença de *hotspot* próximos a esses pontos estratégicos.

5. AGRADECIMENTOS

Ao Setor de Controle de Vetores e Pragas-UADEMA/DIRAC/IOC e aos técnicos da Prefeitura do Rio de Janeiro, da Fundação Nacional de Saúde (FUNASA) pela ajuda na coleta e identificação do material. Ao Dr. Álvaro Eiras por ceder as MosquiTraps.

6. REFERENCIAS BIBLIOGRAFICAS

Barcellos C, Pustai AK, Weber MA, Brito MRV (2005). Identificação de locais com potencial de transmissão de dengue em Porto Alegre através de técnicas de geoprocessamento. Rev. Soc. Bras. Med. Trop. vol.38 no.3

Braks MAH, Honório NA, Lourenço-de-Oliveira R, Juliano AS, Lounibos LP (2003). Convergent habitat segregation of *Aedes aegypti* and *Aedes albopictus* (Diptera: Culicidae) in Southeastern Brazil and Florida. *J Med Entomol 40*: 785-794.

Consoli RAGB, Lourenço-de-oliveira R (1994). Principais mosquitos de importância sanitária no Brasil. *Editora Fiocruz. Rio de Janeiro, Brasil. 225pp.*

Forattini OP (2002). Culicidologia Médica**.** São Paulo. Editora da Universidade de São Paulo. V. 2.

Honório NA, Silva WC, Leite PJ, Gonçalves JM, Lounibos LP, Lourenço-de-Oliveira R (2003). Dispersal of *Aedes aegypti* and *Aedes albopictus* (Diptera:Culicidade) in a

urban endemic dengue area in the state of Rio de Janeiro, Brazil. *Mem Inst Oswaldo Cruz 98:191-198.*

Instituto brasileiro de geografia e estatística 2000. Disponível em: <http://www.ibge.gov.br>. Acessado em: 22 de agosto de 2007.

Lopes J, Silva M AN, Borsato A M, Oliveira V DRB, Oliveira F J A (1993). *Aedes (Stegomyia) aegypti* L. e a culicideofauna associada em área urbana da região sul, Brasil. Rev Saúde Pública 27 (5): 326-33.

Maciel-de-Freitas R, Eiras AE, Lourenço-de-Oliveira R (2006). Field evaluation of effectiveness of the BG-Sentinel, a new trap for capturing adult *Aedes aegypti* (Diptera: Culicidae). Mem. Inst. Oswaldo Cruz vol.101 no.3 Rio de Janeiro.

Neto F C (1997). Descrição da colonização de Aedes aegypti na região de São José do Rio Preto, São Paulo. Revista da Sociedade Brasileira de Medicina Tropical. 30(4): 279-285

Superintendência de controle de endemias (Sucen) (2002). Normas, orientações e recomendações técnicas para a vigilância e controle de *Aedes aegypti* no Estado de São Paulo. São Paulo: Secretaria de Estado de Saúde.

Tauil, PL. (2001) Urbanização e ecologia do dengue. Cadernos de saúde pública, Rio de Janeiro, 17:99-102

Teixeira MG, Barreto ML, Guerra Z (1999). Epidemiologia e Medidas de prevenção de dengue. *Informe epidemiológico do SUS. V. 8, nº 4.*

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 251-264.

SPATIAL ANALYSIS OF *AEDES AEGYPTI* (DIPTERA: CULICIDAE) DISTRIBUTION IN DIFFERENT AREAS OF THE CITY OF RIO DE JANEIRO

Reis IC [1]; Honório N.A.[1];Codeço C.T. [3], Barcellos, C[2], Magalhães, M.A.F.M [2]

1Laboratory of Transmitters of Hematozoa –Institute Oswaldo Cruz, 2Laboratory of Geoprocessing – Institute Oswaldo Cruz, 3Program of Cientifica Computation – Institute Oswaldo Cruz.

(izareis@hotmail[1].com; honorio@fiocruz.br[1];codeco@fiocruz.br[3]; xris@fiocruz.br[2]; monica@cict.fiocruz.br[2])

ABSTRACT. This work aimed at assessing the spatial distribution of the dengue fever mosquito vector, *Aedes aegypti* and its association with Strategic Sites (SS). In each of three localities of Rio de Janeiro city, 80 traps were installed (40 for adult mosquitos and 40 for immatures) in randomly chosen households. Moreover, two further traps we installed close to each SS. These traps were localized with GPS and visited every week for 11 weeks, in the summer of 2007. Based on these data, mosquito infestation maps were created by applying Gaussian Kernels (ArcGis 9.0) to the weekly data. We also used non-parametric methods to test the hypothesis that traps located closer to the SSs would catch more mosquitos than distant ones. Our results suggest that unusual sites, as those used for recycling and a boat factory, showed significant association with infestation *hotspots*. As a conclusion, our study point to the importance of not standard types of human activity as potential new categories to be included in the surveillance of SSs.

1. INTRODUCTION

Intense demographic changes that occurred in the developing countries since the 60's, driven in part by the intense rural-to-urban flow, has lead to the fast swelling of cities and a large contingent of people with inadequate access to sanitation services, water supply and garbage collection. This, together with modern changes in habits, has led to an increase of the number of potential breeding sites for mosquitoes such as the main vector of dengue fever, *Aedes aegypti* (Tauil, 2001). This species is highly adapted to urban and suburban environments where population and household densities are high, vegetation coverage is relatively low (Braks et al. 2003).

Good oviposition sites for *Aedes aegypti* are tires, water boxes, vases of plants, cans, bottles, water throughs of animals, and any object that holds water. Areas were such artificial containers are found in high abundance are named Strategical Sites (SS), according to the National Vector Control Program (Sucen, 2002). These areas are relevant due to their ability to produce large mosquito populations and act as source of mosquitos for areas with low production potential. Areas classically called as SS are areas with large amount of containers (deposits of used tires, junk yard, tire dealer, cemeteries, among others) or areas with large amounts of small and medium size containers, in favorable conditions (transporter, road, ports and airports among other).

The identification of strategical sites is key for the development of optimized vector control strategies. Recently, the use of the Geographic Information System (GIS) in the health has contributed for the development of models that aim at the prevention of the transmission of some illnesses in the urban space through the mapping of its cases and the places most vulnerable the presence of the *Ae. Aegypti* (Barcellos et al. 2005).

1. OBJECTIVE

The objective of this work is to assess, in the summer period, the distribution of the *Ae. aegypti* in three localities of Rio de Janeiro, and its association with the presence of strategicals sites.

3. MATERIAL AND METHODS

3.1. Study area - The work was carried during the summer of 2007, from 08/01/07 to 22/03/2007. Three areas of Rio de Janeiro were chosen: Higienópolis, Vargem Pequena (Palmares) and Ilha do Governador (Tubiacanga), which are characterized by distinct levels of population density, vegetation coverage and history of dengue fever.

Higienópolis (22º52' 25" S 43º15' 41" W) is located in the north region of the city, in a densely urbanized area, with 16.587 inhabitants, low vegetation coverage and adequate garbage and sewage services (IBGE, 2000). Tubiacanga (22º47' 08" S 43º13' 36" W) is located in Ilha do Governador, at the shore of Guanabara Bay. This neighborhood has approximately 2.900 inhabitants, living in an area with moderate vegetion coverage and irregular water supply (Marciel-de-Freitas, et al. 2006). Vargem Pequena, at last, has 11.536 inhabitants. Palmares (22º 59' 26" S 43º 27' 36" W) is a small favela, located at an extreme of Vargem Pequena (IBGE, 2000), close to the Atlantic Rain Forest. Water supply is irregular and the main economic activity is storage of discarded material for recycling.

3.2. Survey - In each study area, 80 traps were installed in randomly selected households. Moreover, five strategic sites per area received two traps each. Half of the traps were ovitraps (capturing immature forms) and 40 were MosquiTraps (capturing adults). Traps were placed outdoors and visited every 7 days, during 11 weeks. Ovitrap paddles and the MosquiTraps cards were taken to the Laboratory of Hematozoa Transmitters, at the Institute Oswaldo Cruz. Immatures forms and adults were identified (Consoli & Lourenço-of-Oliveiras, 1994) and counted. Daily pluviometric data were obtained from GeoRio, for three stations located close to the study sites.

3.3. Data analysis. Infestation maps were created by applying Gaussian kernel to the weekly data, using the *software* ArcGis 9.1. To test for associations between SSs and infestation level, we calculated the cumulative distribution of captured individuals by distance from each SS. From this distribution, we obtained the distance from the SS at each 25%, 50% and 75% of mosquitos were found. To test if such distances were significantly lower than expected from the random distribution, we used randomization methods.

4. RESULTS AND DISCUSSION

A total of 2.023 mosquitoes were collect from MosquiTraps, being 1.242 (61.4%) *Ae. aegypti*, 91 (4,5%) *Ae. albopictus* and 690 (34,1%) *Culex quinquefasciatus*. From the ovitraps, 116.777 eggs were captured, being 99.989 (93,6%) of *Ae. aegypti*, 2.708 (2,5%) *Ae. albopictus* and 4.078 (3,9%) *Cx quinquefasciatus*. We did not find strong correlation between rainfall and density of the *Ae. aegypti* in the study areas.

From the three study areas, Tubiacanga presented the highest density of larvae (50.4%) and adults (26.9%) of *Ae. aegypti*, followed by Higienópolis (29.5% and 23,6 respectively). The weekly infestation maps of Tubiacanga suggest the maintainance of infestation *hotspots* next to strategic sites: factory boats next the Bay to Guanabara and a junk yard. In Palmares, *hotspots* were found very close to the recycling material storage areas. In Higienópolis, on the other hand, the location of *hotspots* varied from week to week.

The analysis statistics of the distribution of the positive traps in relation to the strategicals sites suggests that the boats factory in Tubiacanga, the area used for storage and selection of material for recycling, in Palmares, and the transporter in Higienópolis, had been the high indices of infestation significantly associates.

A similar study carried out in São Jose do Rio Preto, São Paulo, detected the presence of 191 strategic sites, of which only 14 were positive for larvae of *Ae aegypti* (a tire dealer, deposits of tires, a tire repair shop, a building supply store) (Neto 1997). Lopes et al (1993), carried through another study in Lodrina, Paraná, in 30 localities and observed the presence of immature of *Ae. aegypti* only in junk yards (40 specimens) and in an empty lot (275 specimens).

The results of this research suggest that unusual sites, as a boat factory in Tubiacanga, and a recycling material storage area in Palmares, should be included as important strategic sites.

5. ACKNOWLEDGEMENTS

To Setor de Controle de Vetores e Pragas-UADEMA/DIRAC/IOC and profissional of Prefeitura do Rio de Janeiro, da Fundação Nacional de Saúde (FUNASA) by fiel assistence and identified of the larvae and adult of mosquitoes. To Dr. Álvaro Eiras by MosquiTraps.

6. REFERENCES

Barcellos C, Pustai AK, Weber MA, Brito MRV (2005). Identificação de locais com potencial de transmissão de dengue em Porto Alegre através de técnicas de geoprocessamento. Rev. Soc. Bras. Med. Trop. vol.38 no.3

Braks MAH, Honório NA, Lourenço-de-Oliveira R, Juliano AS, Lounibos LP (2003). Convergent habitat segregation of *Aedes aegypti* and *Aedes albopictus* (Diptera: Culicidae) in Southeastern Brazil and Florida. *J Med Entomol 40*: 785-794.

Consoli RAGB, Lourenço-de-oliveira R (1994). Principais mosquitos de importância sanitária no Brasil. *Editora Fiocruz. Rio de Janeiro, Brasil. 225pp.*

Forattini OP (2002). Culicidologia Médica**.** São Paulo. Editora da Universidade de São Paulo. V. 2.

Honório NA, Silva WC, Leite PJ, Gonçalves JM, Lounibos LP, Lourenço-de-Oliveira R (2003). Dispersal of *Aedes aegypti* and *Aedes albopictus* (Diptera:Culicidade) in a urban endemic dengue area in the state of Rio de Janeiro, Brazil. *Mem Inst Oswaldo Cruz 98:191-198.*

Instituto brasileiro de geografia e estatística 2000. Disponível em: <http://www.ibge.gov.br>. Acessado em: 22 de agosto de 2007.

Lopes J, Silva M AN, Borsato A M, Oliveira V DRB, Oliveira F J A (1993). *Aedes (Stegomyia) aegypti* L. e a culicideofauna associada em área urbana da região sul, Brasil. Rev Saúde Pública 27 (5): 326-33.

Maciel-de-Freitas R, Eiras AE, Lourenço-de-Oliveira R (2006). Field evaluation of effectiveness of the BG-Sentinel, a new trap for capturing adult *Aedes aegypti* (Diptera: Culicidae). Mem. Inst. Oswaldo Cruz vol.101 no.3 Rio de Janeiro.

Neto F C (1997). Descrição da colonização de Aedes aegypti na região de São José do Rio Preto, São Paulo. Revista da Sociedade Brasileira de Medicina Tropical. 30(4): 279-285

Superintendência de controle de endemias (Sucen) (2002). Normas, orientações e recomendações técnicas para a vigilância e controle de *Aedes aegypti* no Estado de São Paulo. São Paulo: Secretaria de Estado de Saúde.

Tauil, PL. (2001) Urbanização e ecologia do dengue. Cadernos de saúde pública, Rio de Janeiro, 17:99-102

Teixeira MG, Barreto ML, Guerra Z (1999). Epidemiologia e Medidas de prevenção de dengue. *Informe epidemiológico do SUS. V. 8, nº 4.*

IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 277-282.

Extensão do WEKA para Métodos de Agrupamento com Restrição de Contigüidade

Carlos Eduardo R. de Mello, Geraldo Zimbrão da Silva, Jano M. de Souza

Programa de Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro (UFRJ)
Caixa Postal 68.511 – Zip Code: 21945-970 – Rio de Janeiro – RJ – Brazil

{carlosmello,zimbrao,jano}@cos.ufrj.br

Abstract. *This work addresses the shortage of open-source data mining tools that implement spatial data mining methods. Therefore, this work presents the development of a WEKA extension for contiguity-constrained clustering method.*

Resumo. *Este trabalho aponta a pouca disponibilidade de ferramentas de mineração de dados de código-aberto que implementam métodos de mineração de dados espaciais. Portanto, o objetivo deste trabalho, para tentar resolver esse problema, é apresentar o desenvolvimento de uma extensão da ferramenta WEKA para métodos de agrupamento com restrição de contigüidade.*

1. Introdução

O uso de Sistemas de Informação Geográfica e de Bancos de Dados Espaciais permitiu que grandes quantidades de dados fossem coletadas e armazenadas. Entretanto, extrair conhecimento desses dados de maneira manual torna-se inviável, sendo necessário a utilização de métodos de Descoberta de Conhecimento (também conhecido com Mineração de Dados).

Existem vários métodos e ferramentas para Descoberta de Conhecimento em Bases de Dados que tratam dados convencionais. No entanto, além de dados convencionais, os dados geográficos armazenam suas geometrias e relações entre os objetos espaciais. Essa característica faz com que ferramentas e métodos específicos de descoberta de conhecimento em bases de dados espaciais sejam necessários.

Existem várias ferramentas que implementam os algoritmos clássicos de mineração de dados disponíveis no mercado, entretanto, a maioria delas são pagas ou não possuem código-aberto. As ferramentas que implementam algoritmos de mineração de dados em bases de dados espaciais são raras, acarretando a necessidade de desenvolvimento de novas ferramentas ou a implementação dos algoritmos de mineração de dados espaciais em ferramentas de código-aberto existentes.

Uma ferramenta de código-aberto bastante utilizada é o WEKA (*Waikato Environment for Knowledge Analysis*). Esta ferramenta, desenvolvida na Universidade de Waikato de Hamilton, Nova Zelândia, implementa mais de vinte algoritmos diferentes de mineração de dados convencionais. Pelo fato dessa ferramenta ser de código-aberto, muitas iniciativas de extensão têm sido realizadas. Em [Bogorny et al., 2006], temos o desenvolvimento de uma extensão do WEKA para suportar a extração de regras de associação espaciais.

O objetivo deste trabalho é apresentar uma extensão do WEKA que implementa o algoritmo de agrupamento de dados espaciais com restrição de contigüidade, utilizando as funcionalidades já implementadas no WEKA.

Na seção 2, apresentamos uma visão geral de Mineração de Dados. Na seção 3, apresentamos os métodos de agrupamento e o algoritmo de agrupamento com restrição de contigüidade. Em seguida (seção 4), apresentamos a extensão do WEKA e o algoritmo de agrupamento implementado. Finalmente, na seção 5, descrevemos as conclusões desse trabalho e os trabalhos futuros.

2. Mineração de Dados

Mineração de Dados, ou Descoberta de Conhecimento em Bases de Dados, é definida por [Fayyad et al., 1996] como o processo não-trivial de descoberta de padrões válidos, novos, potencialmente úteis e compreensíveis a partir de dados. Segundo [Ester et al.,2000], o processo de Mineração de Dados é interativo e iterativo englobando várias atividades, como as seguintes:

Seleção: seleção do subconjunto de todos os atributos e do subconjunto de todos os dados em que o conhecimento possa ser descoberto;

Redução: redução das dimensões dos atributos ou técnicas de transformação para reduzir o número efetivo de atributos a serem considerados;

Mineração de dados: a aplicação de algoritmos apropriados que, sob um limite aceitável de eficiência computacional, produzem uma enumeração particular de padrões sobre os dados; e

Análise: interpretação e análise dos padrões descobertos com respeito a sua utilidade em uma dada aplicação.

Embora muitos estudos tenham sido realizados em bancos de dados relacionais (uma visão mais geral pode ser encontrada em [Chen et al.,1996]), ainda há uma grande demanda em outras áreas de aplicação de bancos de dados, incluindo bancos de dados espaciais, bancos de dados temporais, bancos de dados para *multimedia*, etc.

O processo de mineração de dados espaciais é mais complexo que o de dados relacionais, tanto pelo aspecto de eficiência dos algoritmos, como pelo aspecto da complexidade da descoberta de possíveis padrões [Ester et al., 2001]. Uma das razões da complexidade para a descoberta de padrões está no fato que os algoritmos de mineração de dados espaciais devem levar em consideração as relações de vizinhança entre os objetos espaciais para extrair informação útil. Isto é necessário porque as relações de um objeto com os seus vizinhos podem influenciar significativamente o próprio objeto [Ester et al., 2001]. Por outro lado, a eficiência dos algoritmos de mineração de dados espaciais está relacionada à grande quantidade de dados espaciais, à complexidade dos tipos de objetos espaciais e aos métodos de acesso aos dados espaciais [Koperski et al., 1996].

Os principais algoritmos de mineração de dados espaciais cobrem os problemas de classificação, generalização, agrupamento e regras de associação [Ester et al., 2001].

3. Algoritmos de Agrupamento

Nesta seção apresentamos uma visão geral sobre algoritmos (ou métodos) de agrupamento e em seguida descrevemos o método de agrupamento com restrição de contigüidade.

Uma das técnicas mais utilizadas em Mineração de Dados é o Agrupamento. O objetivo dessa técnica é separar objetos ou observações em grupos, onde os objetos mais semelhantes estejam em um mesmo grupo e objetos distintos estejam em grupos diferentes. Portanto, seu objetivo principal é identificar estruturas ou grupos presentes em dados [Ng and Han, 1994].

Os algoritmos de agrupamento podem ser classificados em duas categorias principais: *métodos hierárquicos* ou *métodos de particionamento* [Ng and Han, 2002].

Os *métodos hierárquicos* se dividem em *aglomerativos* ou *divisivos*. Dados *n* objetos para serem agrupados, nos métodos *aglomerativos* começamos o algoritmo com *n* grupos, cada qual formado por um objeto. A cada passo do algoritmo, dois grupos semelhantes são aglomerados transformando-se em um novo grupo. Este processo é repetido até que exista apenas um único grupo contendo todos os *n* objetos. Nos métodos *divisivos*, dados *n* objetos, o algoritmo começa com um único grupo contendo todos os *n* objetos. A cada passo do algoritmo, os grupos formados são divididos de acordo com a similaridade dos objetos contidos neles. Este processo é repetido até que *n* grupos com apenas um objeto sejam formados. Esses dois métodos, *aglomerativos* e *divisivos*, são chamados hierárquicos, pois criam uma relação de hierarquia entre os grupos formados. A partir da visualização da hierarquia desses grupos, o usuário pode decidir com quais grupos deseja trabalhar.

Os *métodos de particionamento* trabalham com um número fixo de grupos. Dados *n* objetos para serem agrupados em *k* grupos, os *métodos de particionamento* tentam encontrar as *k* melhores partições para os *n* objetos. Segundo [Ng and Han, 2002], é muito comum encontrar casos onde os *k* grupos encontrados pelo *método de particionamento* são de melhor qualidade (*i.e.*, mais similares) que os grupos encontrados nos *métodos hierárquicos*. Por isso, os *métodos de particionamento* têm recebido maior atenção da área de análise de grupos. Além disso, muitos *métodos de particionamento* baseados no *k-means* e no *k-medoid* têm sido desenvolvidos.

O *k-means* (ou *k-médias*) é o *método de particionamento* que utiliza o ponto médio da distância entre os objetos no espaço para representar o centro do grupo (ou centróide). Por outro lado, o *k-medoid* utiliza o objeto espacial dentro do grupo mais próximo do ponto médio da distância euclidiana para representar o centro do grupo. O método *k-medoid* é mais robusto que o *k-means* com relação aos *outliers* e os grupos formados por esse independem da ordem com que os objetos são examinados durante sua execução.

3.1. Métodos de Agrupamento com Restrição de Contigüidade

A maioria dos algoritmos de agrupamento de dados espaciais utiliza a distância física entre os objetos espaciais para calcular a similaridade entre os objetos. Em [Ng and Han, 2002], Ng e Han apresentam como agrupar objetos espaciais de geometria poligonal convexa utilizando o algoritmo CLARANS e três maneiras diferentes de calcular a distância entre dois polígonos convexos.

Os métodos de agrupamento com restrição de contigüidade, além dos dados convencionais, também utilizam a informação espacial para restringir os grupos de objetos formados. Em [Gordon, 1995], são definidas duas abordagens para incorporar a informação de proximidade dos objetos espaciais aos métodos de agrupamento: distância geográfica ou grafo (ou matriz) de contigüidade.

Na primeira abordagem, uma medida de distância entre os objetos espaciais é adotada para descobrir a proximidade entre os objetos. Um método de agrupamento clássico pode ser adaptado para considerar, além dos dados convencionais (os atributos dos objetos), a distância física entre os objetos. Alternativamente, esta abordagem pode ser realizada em dois estágios. No primeiro, um algoritmo de agrupamento tradicional pode ser aplicado aos dados convencionais. No segundo estágio, é realizada uma reavaliação dos grupos formados, levando em consideração a distância entre os objetos espaciais para a realocação de objetos.

Na segunda abordagem, a informação topológica dos objetos espaciais é representada através de dispositivos auxiliares como grafo ou matriz. Nessa representação, os nós dos grafos são os objetos espaciais e as arestas são as relações de vizinhança entre eles. Uma aresta *a* ligando os nós *A* e *B* em um grafo indica que os objetos espaciais *A* e *B* são vizinhos. Essa abordagem também pode ser realizada em dois estágios. No primeiro estágio, um algoritmo de agrupamento clássico é aplicado aos dados convencionais. No segundo estágio, os grupos formados são reavaliados utilizando as informações de vizinhança entre objetos contidos no grafo (ou matriz), que foram carregadas a partir um banco de dados espacial.

Em [Neves et al., 2002], é apresentado o método de agrupamento com restrição de contigüidade por árvore geradora mínima. Esse método começa com a construção de um grafo, onde regiões são os nós do grafo e as arestas são as relações de vizinhança entre as regiões. Em seguida, são dados pesos para as arestas desse grafo. O valor dos pesos é a medida de dissimilaridade entre os vetores dos atributos das regiões que são representadas no grafo. A partir do grafo das regiões com os pesos das arestas, é executado um algoritmo para encontrar uma árvore geradora mínima para o grafo. Então, a árvore geradora mínima sofre uma poda. As arestas mais caras da árvore são retiradas, formando sub-árvores desconectadas. Essas sub-árvores desconectadas representam os grupos de regiões formados. Á medida que as arestas mais caras da árvore geradora mínima são retiradas, novos grupos são formados.

Segundo [Neves et al., 2002], o método de agrupamento com restrição de contigüidade por árvore geradora mínima funciona de maneira eficiente para a detecção de regiões. A ferramenta SKATER desenvolvida na Universidade Federal de Minas Gerais implementa esse método [SKATER] [Neves et al., 2002].

4. Extensão do WEKA

Nesta seção vamos apresentar a nossa extensão do WEKA, que implementa o algoritmo de agrupamento com restrição de contigüidade por árvore geradora mínima.

O WEKA possui implementados vários métodos de agrupamento para dados convencionais, entretanto, não encontramos na literatura nenhuma iniciativa de implementação de métodos de agrupamento espaciais. O objetivo do presente trabalho é implementar uma extensão do WEKA para o método de agrupamento com restrição de contigüidade por árvore geradora mínima.

Parte da implementação do trabalho realizado em [Bogorny et al., 2006] foi utilizada em nossa extensão. Nesse trabalho foi desenvolvida uma extensão do WEKA para extração de regras de associação espaciais. Para isso, foi implementado um algoritmo de pré-processamento das informações espaciais, que armazena os dados da topologia dos objetos espaciais em uma tabela do banco de dados e em arquivos.

Através do resultado do pré-processamento descrito em [Bogorny et al., 2006], informações da topologia de vizinhança entre as regiões são extraídas do banco de dados. Essas informações são carregadas em uma matriz de contigüidade, onde as dimensões dessa matriz são iguais ao número de regiões a serem agrupadas. Para cada par de regiões vizinhas é calculado o valor da dissimilaridade entre elas e armazenado na matriz.

O cálculo da dissimilaridade entre os objetos espaciais é realizado através da função de dissimilaridade já implementada no WEKA. A medida adotada para isso é a distância euclidiana entre os vetores formados pelos dados dos atributos das regiões. Portanto, quanto mais semelhantes são as regiões, mais próximo de zero será o valor da dissimilaridade entre elas.

A matriz de contigüidade preenchida pode ser interpretada como um grafo. Portanto, desenvolvemos um algoritmo para encontrar a árvore geradora mínima (AGM) a partir da matriz. Com isso, as arestas de maior custo que geram ciclos são eliminadas do grafo.

A AGM resultante representa as relações de vizinhança mais fortes, isto é, conectam as regiões mais similares. Assim, cada aresta podada da árvore gera duas sub-árvores. O critério utilizado para a poda das arestas consiste em eliminar as arestas de maior peso em ordem descrente, separando as regiões vizinhas menos similares. Cada árvore da floresta representa um grupo de regiões vizinhas e similares.

A implementação do algoritmo foi realizada através da classe chamada *ClusterContiguityConstraintTree*. Além disso, esta classe também é responsável por carregar as informações da topologia que estão na tabela gerada pelo pré-processamento dos dados espaciais e as informações dos dados contidos no arquivo *ARFF* do WEKA.

Portanto, nossa extensão do WEKA baseou-se em utilizar uma série de funcionalidades já existentes dentro da própria ferramenta e de parte da implementação realizada por [Bogorny et al., 2006], contemplando o método de agrupamento com restrição de contigüidade por árvore geradora mínima.

5. Conclusões e Trabalhos Futuros

Neste trabalho apontamos o problema da pouca disponibilidade de ferramentas mineração de dados em código-aberto que implementam métodos de mineração de dados espaciais. A principal contribuição deste trabalho, para tentar resolver esse problema, foi o desenvolvimento de uma extensão do WEKA que implementa o método de agrupamento de dados espaciais com restrição de contigüidade por árvore geradora mínima. Como trabalhos futuros, estamos realizando experimentos e avaliando os resultados obtidos de nossa implementação.

6. Agradecimentos

Agradecemos ao CNPQ pelo apoio financeiro.

Referências

Bogorny, V., Palma, A., Engel, P. and Alvares, L.O. (2006) "Weka-GDPM: Integrating Classical Data Mining Toolkit to Geographic Information Systems.", In: SBBD Workshop on Data Mining Algorithms and Aplications(WAAMD'06), pp.9-16, Florianopolis, Brazil, October 16-20.

Chen, M., Han, J. and Yu, P. S. (1996) "Data Mining: An Overview from Database Perspective", IEEE Transactions on Knowledge and Data Eng., 8(6):866-883, December.

Ester, M., Frommelt, A., Kriegel, H. P. and Sander, J. (2000) "Spatial Data Mining: Database Primitives, Algorithms and Efficient DBMS Support.", Data Mining and Knowledge Discovery Vol. 4, No. 2, July, pp. 193-216.

Ester, M., Kriegel, H. P. and Sander, J. (2001) "Algorithms and Applications for Spatial Data Mining", invited chapter for Geographic Data Mining and Knowledge Discovery, Research Monographs in GIS, Taylor and Francis.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996) "Knowledge discovery and data mining toward a unifying framework.", In Proceeding of The Second Int. Conference on Knowledge Discovery and Data Mining, pages 82-88.

Gordon, A.D. (1996) "A survey of constrained classification.", Computational Statistics & Data Analysis, v. 21, p. 17-29.

Koperski, K., Adhikary, J. and Han, J. (1996) "Spatial data mining: Progress and challenges survey paper", In Proc. ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Montreal, Canada.

Neves, C.M., Câmara, G., Assunção, R.M. and Freitas, C.C. (2002) "Procedimentos Automáticos e Semi-automáticos de Regionalização por Árvore Geradora Mínima.", In: Simpósio Brasileiro de Geoinformática, GeoInfo.

Ng, R. and Han, J. (1994) "Efficient and Effective Clustering Methods for Spatial Data Mining", Proceedings of 20th International Conference on Very Large DataBases, pp 144-155.

Ng, R. and Han, J. (2002) "CLARANS: A Method for Clustering Objects for Spatial Data Mining", IEEE Trans. Knowledge & Data Engineering , 14, 5, pp 1003-1016, September.

SKATER, http://www.est.ufmg.br/leste/skater.htm.

Author Index

Author Index