GEOINFO 2018

XIX Brazilian Symposium on GeoInformatics

December 05th to 07th, Campina Grande, PB, Brazil

Proceedings

Lubia Vinhas and Cláudio Campelo (Eds.)

Dados Internacionais de Catalogação na Publicação

SI57a Simpósio Brasileiro de Geoinformática (12.: 2018: Campina Grande, PB)

Anais do 19o. Simpósio Brasileiro de Geoinformática, Campina Grande, PB,
05 de dezembro a 07 de dezembro de 2018. / editado por Lúbia Vinhas (INPE),
Claudio Campelo (UFCG) – São José dos Campos, SP:
MCTIC/INPE, 2018.
On-line
ISSN 2179-4847

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

CDU:681.3.06

Preface

This volume of proceedings contains the papers presented at the XIX Brazilian Symposium on Geoinformatics, GeoInfo 2018, held in Campina Grande, Paraíba state in Brazil, December 05-07 2018. The GeoInfo conference series, inaugurated in 1999, continues its trajectory of assembling researchers and students on geoinformatics.

In this edition, 20 papers were accepted (out of 37 high-quality submissions). Authors from 19 distinct Brazilian academic institutions and research centers and 8 international institutions are represented. Both full and short papers are assigned the same time for oral presentation at the event. Short papers, which usually reflect ongoing work, receive a larger time-share for questions and discussions.

GeoInfo has a tradition of attracting some of the most prominent researchers in the world to productively interact with our community, thus generating all sorts of interesting exchanges and discussions. This year, we have special keynote presentations by Duncan Smith from the Centre for Advanced Spatial Analysis (CASA) - Bartlett Faculty of the Built Environment, University College London (UCL), and André Lage Freitas from LaCCAN - Laboratório de Computação Científica e AnÃąlise Numérica - Universidade Federal de Alagoas (UFAL) - Brazil.

We would like to thank all Program Committee members, listed below, and additional reviewers, whose work was essential to ensure the quality of every accepted paper. At least three specialists contributed with their review for each full paper submitted to GeoInfo whereas at least two experts evaluated short papers. We would like to specially thank Daniela Seki and Adriana GonÃğalves from INPE for staffing the symposium preparation and execution.

Finally, we would like to thank GeoInfo 2018 supporters: BrazilâĂŹs Coordination for the Improvement of Higher Education Personnel (CAPES) and the Society of Latin American Remote Sensing Specialists (SELPER), which are identified at the conference's web site.

The Brazilian National Institute of Space Research (Instituto Nacional de Pesquisas Espaciais, INPE) and The Federal University of Campina Grande (Universidade Federal de Campina Grande, UFCG) are very happy to bring together the GeoInfo community, once again, to Northeast part of Brazil.

Campina Grande and São José dos Campos, Brazil.

Lubia Vinhas
Program Committee Chair

Claudio Campelo
General Chair

Conference Commitee

General Chair

Claudio Campelo
Federal University of Campina Grande, UFCG

Program Chair

Lubia Vinhas
National Institute for Space Research, INPE

Conference Staff

Daniela Seki
National Institute for Space Research, INPE

Adriana GonÃğalves
National Institute for Space Research, INPE

Organized by

UFCG - Federal University of Campina Grande
INPE - National Institute for Space Research

Supported by

SELPER - Sociedade Latino Americana de Especialistas em Sensoriamento Remoto
CAPES - Coordenação de Aperfeiçamento de Pessoal de Nível Superior

Program committee

Ana Paula Afonso, Univ. Nova de Lisboa, Portugal
Antônio Miguel Monteiro, INPE, Brazil
Carla Macario, Embrapa, Brazil
Carlos Felgueiras, INPE, Brazil
Chiara Renso, ISTI/CNR, Italy
Claudia Robbi Sluter, UFPR, Brazil
Claudio Campelo, UFCG, Brazil
Clodoveu A. Davis, UFMG, Brazil
Cristina Ciferri, USP, Brazil
Flávia Feitosa, UFABC, Brazil
Flávio Horita, UFABC, Brazil
Gilberto Queiroz, INPE, Brazil
Hugo Bendini, INPE, Brazil
João P. de Albuquerque, U. Warwick, UK
José A. Quintanilha, USP, Brazil
Jugurta Lisboa Filho, UFV, Brazil
Julio Dalge, INPE, Brazil
Jussara O. Ortiz, INPE, Brazil
Karine R. Ferreira, INPE, Brazil
Karla A. V. Borges, Prodabel, Brazil
Lubia Vinhas, INPE, Brazil
Laercio Namikawa, INPE, Brazil
Mário J. G.r da Silva, Univ. de Lisboa, Portugal
Marcus Vinicius A. Andrade, UFV, Brazil
Maria Isabel S. Escada, INPE, Brazil
Michela Bertolotto, UCD, Ireland
Pedro R. Andrade, INPE, Brazil
Rafael Santos, INPE, Brazil
Raul Q. Feitosa, PUC-RJ, Brazil
Renato Fileto, UFSC, Brazil
Ricardo R. Ciferri, UFSCAR, Brazil
Rogério Galante Negri, UNESP, Brazil
Sergio D. Faria, UFMG, Brazil
Sergio Rosim, INPE, Brazil
Silvana Amaral, INPE, Brazil
Stephan Winter, Univ. of Melbourne, Australia
Thales Sehn Körting, INPE, Brazil
Tiago G. S. Carneiro, UFOP, Brazil
Valéria C. Times, UFPE, Brazil
W. Randolph Franklin, Rensselaer Polytechnic Inst., USA

Contents

GIS and Data: Three applications to enhance Mobility

Andy S Alic[1], Jussara Almeida[2], Wagner Meira Jr.[2],Dorgival Guedes[2], Walter dos Santos[2], Ignacio Blanquer[1], Sandro Fiore[3], Nádia P. Kozievitch[9], Nazareno Andrade[4], Tarciso Braz[4], Andrey Brito[4], Carlos Eduardo Pires[4], Nuno Antunes[5], Marco Vieira[5], Paulo Silva[5], Danilo Ardagna[6], Keiko Fonseca[9], Daniele Lezzi[7], Donatello Elia[3], Regina Moraes[8], Tania Basso[8], Wilian H. Cavassin[9]

[1]Universitat Politècnica de València - CSIC

[2]Universidade Federal de Minas Gerais (UFMG), Brazil

[3]Fondazione Centro Euro-Mediterraneo sui Cambiamenti Climatici (CMCC), Italy

[4]Universidade Federal de Campina Grande (UFCG), Brazil

[5]CISUC, University of Coimbra, Portugal

[6]Politecnico di Milano, Milan, Italy

[7]Barcelona Supercomputing Center (BSC), Spain

[8]University of Campinas (UNICAMP), Brazil

[9]Universidade Tecnológica Federal do Paraná (UTFPR), Brazil

asalic@upv.es, {jussara,meira,dorgival,walter}@dcc.ufmg.br,

iblanque@dsic.upv.es, sandro.fiori@unisalento.it, nadiap@utfpr.edu.br,

{nazareno,tarciso,andrey}@computacao.ufcg.edu.br, cesp@dsc.ufcg.edu.br,

{nmsa,mvieira,pmgsilva}@dei.uc.pt, danilo.ardagna@polimi.it,

keiko@utfpr.edu.br, danieli.lezzi@bsc.es, donatello.elia@cmcc.it,

{regina,taniabasso}@ft.unicamp.br, wiliancavassin@alunos.utfpr.edu.br

Abstract. *The increasing urban population sets new demands for mobility solutions. The impacts of traffic congestions or inefficient transit connectivity directly affect public health (e.g. emissions and stress) and the city economy (e.g. deaths in road accidents, productivity, and commuting). In parallel, the advance of technology has made it easier to obtain data about the systems which make up the city information systems. This paper takes advantage of GIS and real-time data to present: 1) a web application integrating multiple services; 2) an android application for bus visualization and prediction and 3) a dashboard focused on applying exploratory data analysis techniques on ticketing data.*

1. Introduction

The steady growth of urban centers poses several challenges to human well-being, many of which are associated with urban mobility (traffic jams, longer travel times, health issues due to emissions, stress, etc.) requiring new approaches to overcome them. Thus, it is necessary to provide new tools for managing a city, reconciling the functioning of several systems so that their performance fit to better serve its inhabitants. In this scenario, the concept of *smart city* has emerged along of its several approaches of smartness [Husár et al. 2017], usually linked to efficiency in the use of natural resources [Souza et al. 2015, Azambuja 2016].

In particular, public transportation is one of the most critical areas of smart cities. In Brazil, the vehicle fleet in major cities grew more than the road structure[1]. Mobility challenges have already gained attention of the Computer Science society in Brazil[2]. The efficiency of its performance helps to reduce its operation costs and also provides social integration (such as the use of government applications and crowd-sourcing). Some people have the public transport system as their only mean of displacement among their daily trips [Weigang et al. 2001].

The increasing availability of city open data provides opportunities to explore new applications or innovative data exploitation, along with GIS techniques to enhance the cities mobility. However, processing big volume of raw data in limited time to provide timely information for services with an acceptable quality poses several challenges.

This paper describes three applications (Routes4People, Melhor Busão and City Administration Dashboard) based on GIS, cloud and parallel computing technologies to enhance mobility, not only from the citizen perspective, but also from the perspective of the city administration. The applications were developed under the *EUBra-BIGSEA* project (Europe-Brazil Collaboration of Big Data Scientific Research Through Cloud-Centric Applications), where all developments are available under Open Source licenses [3]. The rest of the paper is organized as follows: Section 2 presents related work. Section 3 offers an overview of the applications platform. Section 4 details the applications and their features. Finally, Section 5 presents the conclusion.

2. Related Work

Several applications are already available for mobility, such as Crowdbus [Sousa Junior et al. 2014], Bus Brasil[4], Cadê o Ônibus[5], Itibus[6] and Moovit[7].

Crowdbus uses resources of crowdsourcing technology to provide data about the public transportation in Recife and Maceió where the crowdsourced data are collected by users support and user's smartphones functionalities (e.g., compass, GPS, and accelerometer). Crowdbus uses data from speed, routes to generate a quality standing for each route of public transportation, processing the data to provide, in the future application,

[1] http://www1.folha.uol.com.br/cotidiano/2014/08/1503030-frota-de-\ veiculos-cresce-mais-rapido-que-a-estrutura-viaria-no-pais.shtml – Last accessed on Nov 24, 2017.

[2] http://www.sbc.org.br/documentos-da-sbc/send/141-grandes-desafios/802-grandesdesafiosdacomputaonobrasil – Last accessed on Nov 24, 2017.

[3] http://github.org/eubr-bigsea – Last accessed on Nov 24, 2017.

[4] http://www.busbrazil.com.br/ – Last accessed on Ago 22, 2018.

[5] http://www.cadeoonibus.com.br/CoO/SiteV2 Last accessed on Nov 4th, 2017.

[6] https://www.urbs.curitiba.pr.gov.br/mobile/itibus – Last accessed on Ago 22, 2018.

[7] https://moovit.com/ – Last accessed on Ago 28, 2018.

measurements of time to travel between bus stops [Sousa Junior et al. 2014].

Bus Brasil uses an application for Android smartphone that stores bus time tables from various cities in Brazil. The application provides schedules for the buses, with the closest bus coming from a determined place, such as a Bus Terminal. The application can store data in the device, providing some functionalities while the device is not connected to the Internet.

Cadê o Ônibus was developed in cross-platform modal (Android, IoS and Windows Phone). This application detects the position of buses in real-time, allowing a variable number of search requests by the user. This allows user collaboration (feature that is only showed in the Moovit and the Crowdbus apps), thanks to the use of the real-time tracking can also predict the arrival of the bus on the bus stop.

Itibus is a web-plataform application which provides the schedule of lines, lines by its code or label, itinerary of the lines in a map, real-time location of the bus, near location of bus stops and lines by stops. Besides that, the application can provide news and the balance of the client's transport card.

Moovit is another example of application which uses GIS and processes data from external sources to generate knowledge. The application operates in more than 2,500 cities, with more than 200 million users. Under the concept of Urban Mobility Analysis and Mobility as a Service (MaaS), the system provides a list with buses lines, various types of search, lines by stops, route creation, and buses that accept transport card. The system can predict the arrival and departure times of the lines in stops and terminals.

Compared to these online applications (listed in Figure 1), our three approaches present the following advantages: open source licenses, integration with several data sources (e.g. Waze, twitter, mobility open data) along with Big Data and Cloud services (among others).

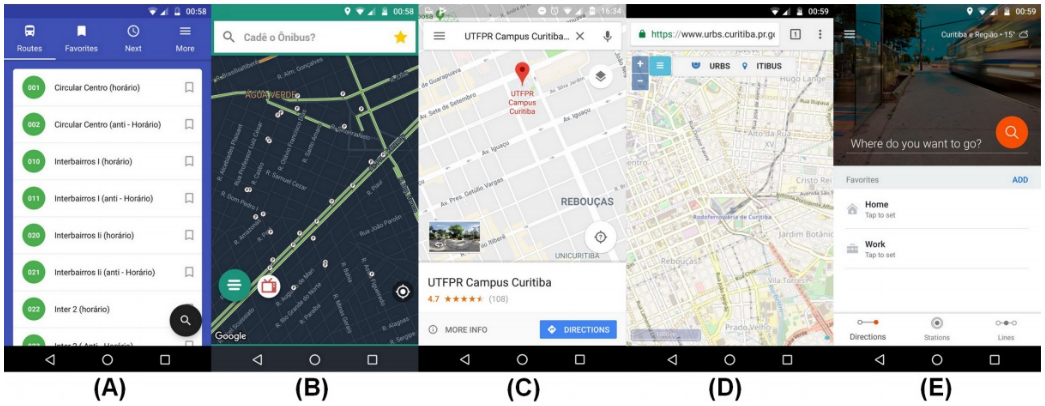


Figure 1. Home Screen of the applications: (A) Bus Brasil. (B) Cadê o Ônibus?. (C) ItiBus. (D) Moovit. Source: [Calandre et al. 2018]

Andrade et al. [Andrade et al. 2014] propose how the combination of open data, geographic information systems and cluster algorithms can bring benefits to urban mobility. This paper has the time spent in a displacement within the city of Belo Horizonte as a case study. By the cluster algorithm analysis it is possible to determine clusters of any radius and decide the best vehicle to be chosen on a given trip within the city, as well as the best place to be carried out.

In the same way, Monteiro et al. [Monteiro et al. 2017] suggest how the bus stops can be distributed in a way that bus stops are not too far apart from each other. In other words, their algorithm can be used to determine the best places to have a pick off point. To do this, they have used open data from public transportation and a Simulated Annealing algorithm.

Several other works also uses transportation scenarios, but in different contexts, such as location modeling [Li and Tong 2017], optimization [Yang et al. 2000], complex network metrics [da Silva et al. 2016, De Bona et al. 2016], and exploratory analysis [Kozievitch et al. 2016, Vila et al. 2016].

3. Infrastructure Overview

The infrastructure used to support the applications included 19 components (Figure 2). In summary, there are five different layers:

1. Modules for resource configuration, prediction of resource usage, scheduling of jobs and proactive policies for vertical and horizontal elasticity. This layer has the following items:
    - Infrastructure Manager configures the underlying infrastructure with the software required to execute the jobs from the Programming Models layer;
    - Elastic Compute Clusters in the Cloud provides the interface to deploy self-configurable scalable clusters. This is the main tool for deploying the *EUBra-BIGSEA* infrastructure, and interacts directly with IM;
    - DagSim simulator and Lundstrom predictor are two components that use information from the logs of COMPSs and Spark applications to create predictor models for estimating running time under different resource conditions;
    - Proactive Policies have two implemented components: Marathon and Chronos Framework for dealing with QoS, which adjusts the amount of resources allocated to match the expected QoS and the component to adjust the CPU CAP on hypervisors (working in both OpenNebula and OpenStack) to meet the expected deadlines.
2. Programming Models, which provide the means to write parallel data analytics programs on top of the *EUBra-BIGSEA* platform. This layer has the following items:
    - COMPSs is a programming framework that infers the inner parallelism of sequential applications dynamically, executing the different steps in parallel and taking care of data dependencies. In the frame of *EUBra-BIGSEA*, it has been extended to work as a Mesos Framework and to use HDFS as a back-end, facilitating the execution on distributed environments;
    - LEMONADE is a platform for the visual creation and execution of data analysis workflows, which produces Spark and COMPSs code.

3. Security and privacy mechanisms provide a homogeneous Authentication, Authorization and Accounting (AAA) mechanism and privacy policies for data access and processing. This layer has the following items:
    - AAAaaS is a module of Authentication, Authorization and Accounting as a Service for the *EUBra-BIGSEA* Project;
    - PRIVAaaS is a set of libraries and tools that allows controlling and reducing data leakage in the context of Big Data processing and, consequently, protecting sensible information processed by data analytics algorithms, with multiple types of anonymization techniques.
4. Big Data services is composed by the following items:
    - Ophidia - which exploits advanced parallel computing techniques and a hierarchical, distributed storage organization to execute intensive OLAP-based analysis over multi-terabyte datasets;
    - Data Quality as a Service (DQaaS) is a tool able to provide information about the quality of the analyzed Big Data sources;
    - Entity Matching as a Service (EMaaS) is a service that supports the detection and measurement of matching problems related to the linkage of large data sources
5. High-level services are composed by the following items:
    - Traffic Congestion Prediction: aims to identify traffic jams using data provided by Waze. To this end, a probabilistic graphical model equipped with Gaussian latent nodes is formulated;
    - Trip Duration Prediction: is a tool that aims to predict bus trips duration based on historical bus GPS data. We train the model using Machine Learning techniques (Support Vector Regression and Lasso Regression) on historical bus trips data, and use it to predict future trips;
    - Sentiment Analysis: transforms social media data (textual) into a quantitative estimation of the citizens expressed sentiment. Such analysis targets a specific subject, for example, traffic status or city services;
    - Trip Crowdedness Prediction is a tool that aims to predict the number of passengers (crowdedness) of a bus trip in the future, based on historical bus location and ticketing data;
    - People Paths: is an application which performs a descriptive analysis on bus GPS and passenger ticketing data, finding paths taken by urban Public Transportation users in a time period, and matching the paths origin/destination locations with city area social data.

The applications on the framework work at two levels. The final-user developed applications offer a Graphic User Interface that exposes the outcome of other components in the *EUBra-BIGSEA* platform. On the other hand, applications for descriptive and predictive data models run on top of the infrastructure. Further details of each component are presented at the *EUBra-BIGSEA* site (`http://www.eubra-bigsea.eu/`).

4. The Applications

All the applications have as use case the data from Curitiba Municipality [8]. In summary, data covering DPGS trajectories, ticketing, Weather, Social media, routes, timetables,

[8] http://www.curitiba.pr.gov.br/dadosabertos/

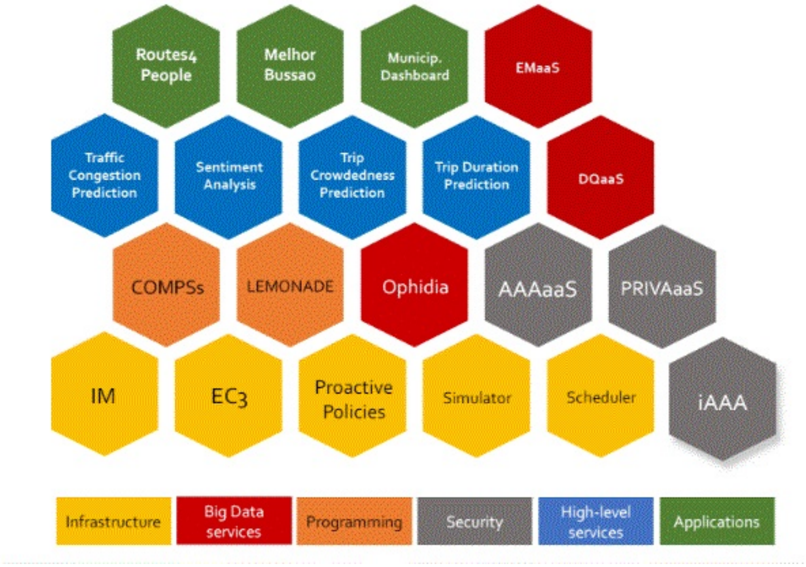


Figure 2. The 19 components from the infrastructure and its layers.

sociodemographic and environment (among others) in several formats (CSV, DOC, SHP, data from Twitter, etc.) were manipulated in several databases (PostgreSQL, MongoDB, among others) for the applications. Further details of the data sources, acquisition and integration can be found here [9]. The application ecosystem is presented in Figure 3: note that different types of data, such as file sources and databases are used.

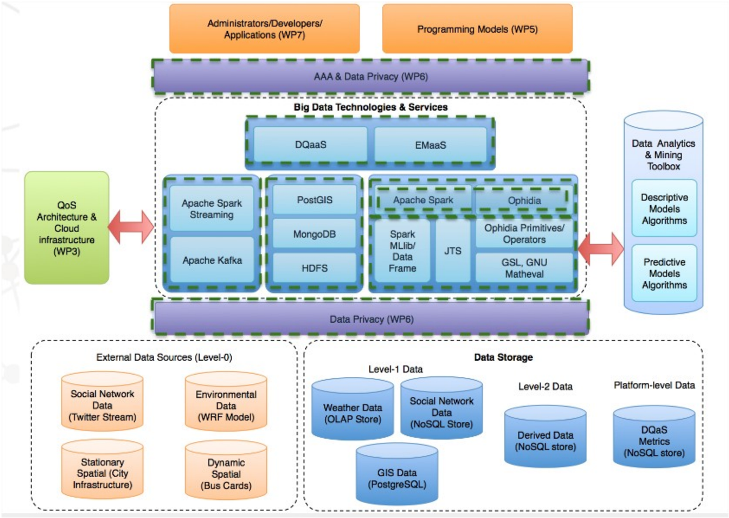

Administrators/Developers/ Applications (WP7)
Programming Models (WP5)
AAA & Data Privacy (WP6)
Big Data Technologies & Services
DQaaS
EMaaS
QoS Architecture & Cloud infrastructure (WP3)
Apache Spark Streaming
Apache Kafka
PostGIS
MongoDB
HDFS
Apache Spark
Ophidia
Spark MLlib/ Data Frame
JTS
Ophidia Primitives/ Operators
GSL, GNU Matheval
Data Analytics & Mining Toolbox
Descriptive Models Algorithms
Predictive Models Algorithms
Data Privacy (WP6)
External Data Sources (Level-0)
Social Network Data (Twitter Stream)
Environmental Data (WRF Model)
Stationary Spatial (City Infrastructure)
Dynamic Spatial (Bus Cards)
Data Storage
Level-1 Data
Weather Data (OLAP Store)
Social Network Data (NoSQL Store)
GIS Data (PostgreSQL)
Level-2 Data
Derived Data (NoSQL store)
Platform-level Data
DQaS Metrics (NoSQL store)

Figure 3. Integrated ecosystem.

On the other hand, the applications use several software components, as shown in Figure 4. Links (lines) between the relations represent the module relation to other

[9] http://www.eubra-bigsea.eu/sites/default/files/EUBRra-BIGSEA_D7.2_GES3DataIntegration_v1.pdf – Last accessed on Nov 24, 2017.

components. The top of the figure contains the Final user applications (Routes4people, Melhor Busão and Municipality Dashboard), and the bottom of the figure contains the infrastructure (Data and CPU Resources).

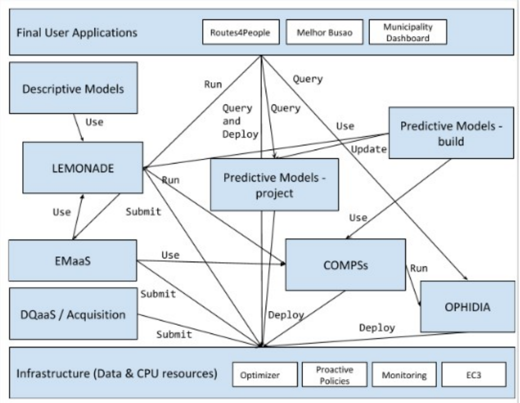


Figure 4. The software architecture for the three applications.

4.1. Routes4People

Routes4People (video available online [10]) is a web application that gathers information from the processing algorithms for Sentiment Analysis, Crowdedness prediction, Traffic congestion estimation and Route recommendation (high-level services presented in Figure 2).

The sentiment analysis service is implemented using Apache Spark, Apache Spark Streaming and Apache Kafka in two stages: 1) model learning (training of machine learning classifiers) and 2) model usage (gathering and testing data). The Crowdedness prediction [Braz et al. 2018] uses heuristics to infer where passengers alighted from the bus. The traffic congestion estimation aims to identify traffic jams using data provided by Waze.

Routes4People has as main user citizens, providing information about the best route considering standard criteria (a priori duration) and other more human criteria (forecasted crowdedness and historic duration). It mainly uses COMPSs, Lemonade, AAAaaaS and the data sources showed in Figure 3.

Within the services available, we can mention the creation of a trip, clustered visualization of all bus stops, the listing of all routes with respective schedules along with the traffic jam, sentiment analysis and feedback form. The code is available online [11] along with the web interface [12].

4.2. Melhor Busão

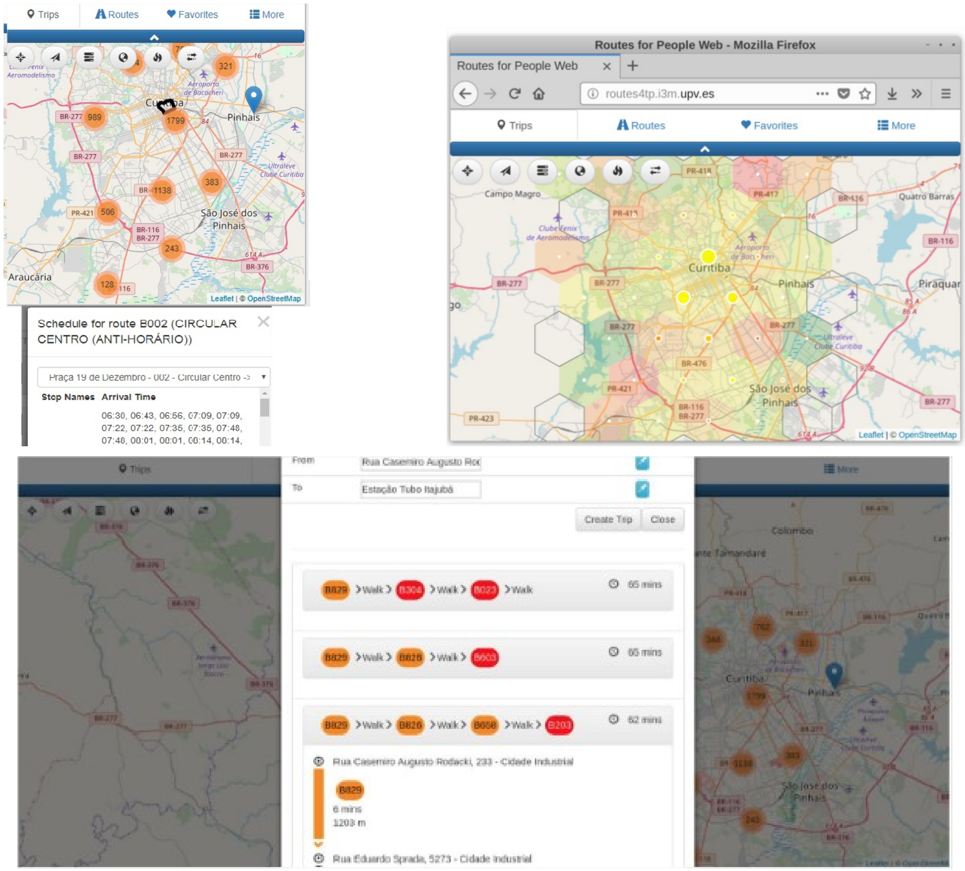
Melhor Busão (video available online [13]) is a mobile (Android) for Routes4People. Both make use of descriptive models using bus GPS and passenger ticketing data: Origin-

[10] `https://youtu.be/L5LRbq1IIho` – Last accessed on July 24, 2018.

[11] `https://github.com/eubr-bigsea/rfp-web` – Last accessed on Nov 24, 2017.

[12] `http://routes4tp.i3m.upv.es` – Last accessed on July 24, 2018.

[13] `https://youtu.be/XoWJ_BuQWmU` – Last accessed on July 24, 2018.



Schedule for route B002 (CIRCULAR CENTRO (ANTI-HORÁRIO))

Praça 19 de Dezembro - 002 - Circular Centro ->

Stop Names Arrival Time

06:30, 06:43, 06:56, 07:09, 07:09, 07:22, 07:22, 07:35, 07:35, 07:48, 07:48, 00:01, 00:01, 00:14, 00:14,

From Rua Casemiro Augusto Rod

To Estação Tubo Itajubá

B829 > Walk > B304 > Walk > B023 > Walk 65 mins

B829 > Walk > B828 > Walk > B603 65 mins

B829 > Walk > B826 > Walk > B658 > Walk > B203 62 mins

Rua Casemiro Augusto Rodacki, 233 - Cidade Industrial

B829
6 mins
1203 m

Rua Eduardo Sprada, 5273 - Cidade Industrial

Figure 5. Routes4People application.

Destination matrices, inefficiency analysis, comparison with population, income and literacy rate, implemented in LEMONADE & COMPSs.

The object was not only to present the available bus stops, time table, bus lines, but also present the prediction of the next bus in a specific area (as shown in Figure 6). The code is available online [14].

4.3. City Administration DashBoard

City Administration Dashboard (video available online [15]) has as main user the decision maker, being an application which has as basis descriptive statistics and visualization techniques in order to assist and facilitate planning and monitoring the system.

As technologies, it uses Ophidia to infer a set of bus usage statistics by means of descriptive analytics algorithms implemented in Python and exploiting the COMPSs programming model. Its modules include also anonymization phases as well as preprocessing steps based on Data Quality and Entity Matching steps.

[14] https://github.com/eubr-bigsea/bigsea-melhorbusao – Last accessed on Nov 24, 2017.

[15] https://youtu.be/AQo5O8YhssA – Last accessed on July 24, 2018.

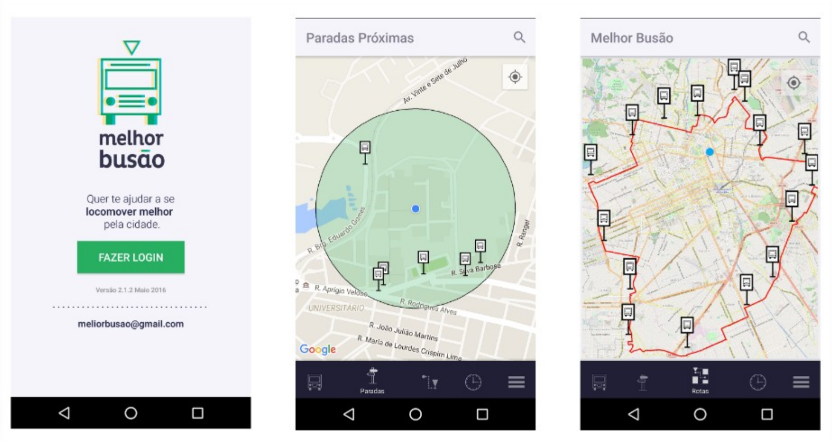


Figure 6. Melhor Busão (Opening Interface, Location of Bus Stops Nearby and Bus Stops available in a Bus Line).

In summary, it presents interactive charts related to a set of 20 statistics in the bus system usage (bus lines, bus stops and passengers) based on bus cards data, along with bus position and shape files (the application focuses on Curitiba city in Brazil) from three perspectives: overall boarding, average passenger boardings and bus stop crowdedness.

Figure 7 presents the visualization of total number of passengers by hour followed by the heatmap of the passengers bus stops. The objective of this application was to support data visualization, from the municipality point of view, and the code is available online [16] along with the web interface [17].

4.4. Discussion

The big and fast data eco-system in this proposal is a platform that can effectively support (i) different types of data processing (i.e. batch and streaming) on (ii) heterogeneous data (i.e. multidimensional, relational, NoSQL) (iii) requiring multiple data analytics and mining features (i.e. descriptive and predictive models), while also (iv) taking security, data privacy and QoS-oriented elastic cloud scenarios into account.

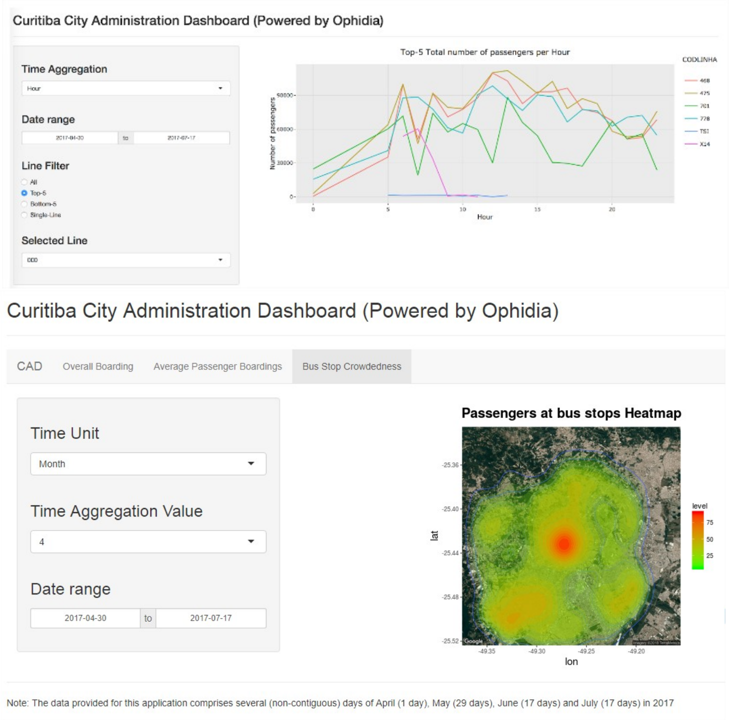
Within the main contributions of the infrastructure, we can mention: 1) specific data services, such as EMaaS [Mestre et al. 2017] (which was fundamental to match entities within the different databases integration) and DQaS [Araújo et al. 2017] (in order to check the data quality); 2) new programming technologies, such as Lemonade [d. Santos et al. 2017] (which simplifies and abstract the infrastructure and the programming task); 3) online available predictive models, such as sentiment analysis [18] and Trip Crowdedness Prediction [19], among others.

[16] `https://github.com/eubr-bigsea/ticketing-descriptive-analytics` – Last accessed on Nov 24, 2017.

[17] `https://tarcisob.shinyapps.io/city-admin-dashboard-ctba` – Last accessed on July 24, 2018.

[18] `https://github.com/eubr-bigsea/twitter-sentiment-analysis` – Last accessed on July 24, 2018.

[19] `https://github.com/eubr-bigsea/btr-spark` – Last accessed on July 24, 2018.


Curitiba City Administration Dashboard (Powered by Ophidia)

Time Aggregation: Hour

Date range: 2017-04-30 to 2017-07-17

Line Filter:
○ All
● Top-5
○ Bottom-5
○ Single-Line

Selected Line: 000

Curitiba City Administration Dashboard (Powered by Ophidia)

Time Unit: Month

Time Aggregation Value: 4

Date range: 2017-04-30 to 2017-07-17

Note: The data provided for this application comprises several (non-contiguous) days of April (1 day), May (29 days), June (17 days) and July (17 days) in 2017

Figure 7. City Administration Dashboard.

Within the contributions of the applications, we can mention: 1) their availability over open source licenses (software, documentation and video); 2) their integration with several data sources (Waze, twitter, mobility open data), along with the main challenges, such as data privacy [Basso et al. 2016]; and 3) local panels to discuss mobility applications and mobility solutions, such as MAUI Symposium [20] and Workshop on Secure Cloud and Big Data [21].

Further performance evaluation and tests for individual modules or overall applications can be found at the *EUBra-BIGSEA* project [22]. Results for the performance evaluation of the DashBoard, for example, show that the platform scalability is adequate enough to process large datasets(related to long time period) and provide useful statistics for the City Administration Dashboard application [23].

5. Conclusion

In order to accommodate the users and their transportation needs, a city must carefully analyze the several data sources to determine the citizen needs and possible changes in transportation to support those needs. Further analysis should also accommodate different

[20] `http://www.prppg.ufpr.br/site/sba-maui/` – Last accessed on July 24, 2018.

[21] `http://site.sanepar.com.br/noticias/inscricoes-para-evento-sobre\-processamento-seguro-de-dados-em-nuvens-inseguras-ate-27` – Last accessed on July 24, 2018.

[22] `http://www.eubra-bigsea.eu/menu-deliverables` –Last accessed on Sept. 09, 2018.

[23] `http://www.eubra-bigsea.eu/sites/default/files/D2.5_EUBra-BIGSEA_FinalActionPlanReport.pdf` – Last accessed on July 09, 2018.

perspectives: city administrators, general visualization and basic statistics, bus timetables, among others. This paper presented three applications resulted from the *EUBra-BIGSEA* project, from the urban mobility perspective. The first one (City Administration Dashboard) presents an historical overview of the data, from the urban traffic management perspective. The second one presents an Advanced Traveler Information System (Melhor Busão), while the third one gathers information from traffic congestion and sentiment analysis (among others) in a web interface. In summary, the three applications took advantage of several infrastructure enhancements (such as elastic computation and proactive policies), programming models (using technologies such as COMPSs and LEMONADE), security and privacy mechanisms, Big Data services (parallel computing and data quality), along with high-level services based on models (such as traffic congestion prediction and trip duration prediction). As future work, we can mention the integration of other historical data, user evaluation of the system, further tests within more data, and integration with additional scenarios (such as accidents and bumps).

Acknowledgments. We would like to thank the Municipality of Curitiba, IPPUC, CAPES, CNPq, Fapemig and *EUBra-BIGSEA* project (EC/MCTIC 3rd Coordinated Call).

References

[Andrade et al. 2014] Andrade, T. C., Pereira, M. A., and Wanner, E. F. (2014). Development of an application using a clustering algorithm for definition of collective transportation routes and times. In *XV GEOINFO*.

[Araújo et al. 2017] Araújo, T. B., Cappiello, C., Kozievitch, N. P., Mestre, D. G., Pires, C. E. S., and Vitali, M. (2017). Towards reliable data analyses for smart cities. In *Proceedings of the 21st International Database Engineering & Applications Symposium*, IDEAS 2017, pages 304–308, New York, NY, USA. ACM.

[Azambuja 2016] Azambuja, L. S. d. (2016). Dados abertos em cidades inteligentes: portais de dados abertos possibilitando o acesso e uso da informação. B.s. thesis, UFRGS.

[Basso et al. 2016] Basso, T., Matsunaga, R., Moraes, R., and Antunes, N. (2016). Challenges on anonymity, privacy, and big data. In *2016 Seventh Latin-American Symposium on Dependable Computing (LADC)*, pages 164–171.

[Braz et al. 2018] Braz, T., Maciel, M., Mestre, D. G., Andrade, N., Pires, C. E., Queiroz, A. R., and Santos, V. B. (2018). Estimating inefficiency in bus trip choices from a user perspective with schedule, positioning, and ticketing data. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–12.

[Calandre et al. 2018] Calandre, A., Pasquim, B., Santos, E., and Oliveira, J. (2018). *MYURB: Assistente de Mobilidade em Curitiba através do Transporte Público*. monography, UTFPR.

[d. Santos et al. 2017] d. Santos, W., Carvalho, L. F. M., d. P. Avelar, G., Silva, A., Ponce, L. M., Guedes, D., and Meira, W. (2017). Lemonade: A scalable and efficient spark-based platform for data analytics. In *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 745–748.

[da Silva et al. 2016] da Silva, E. L. C., d. O. Rosa, M., Fonseca, K. V. O., Luders, R., and Kozievitch, N. P. (2016). Combining k-means method and complex network analysis to evaluate city mobility. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1666–1671.

[De Bona et al. 2016] De Bona, A., Fonseca, K., Rosa, M., Lüders, R., and Delgado, M. (2016). Analysis of public bus transportation of a brazilian city based on the theory of complex networks using the p-space. *Mathematical Problems in Engineering*, 2016:1–12.

[Husár et al. 2017] Husár, M., Ondrejička, V., and Varış, S. C. (2017). Smart cities and the idea of smartness in urban development – a critical review. *IOP Conference Series: Materials Science and Engineering*, 245(8):082008.

[Kozievitch et al. 2016] Kozievitch, N. P., Gadda, T. M. C., andMarcelo O. Rosa, K. V. O. F., Gomes-Jr, L. C., and Akbar, M. (2016). Exploratory analysis of public transportation data in curitiba. In *XXXVI CSBC*, pages 1656–1666. Sociedade Brasileira de Computação.

[Li and Tong 2017] Li, R. and Tong, D. (2017). Incorporating activity space and trip chaining into facility siting for accessibility maximization. *Socio-Economic Planning Sciences*, 60:1 – 14.

[Mestre et al. 2017] Mestre, D. G., Pires, C. E. S., and Nascimento, D. C. (2017). Towards the efficient parallelization of multi-pass adaptive blocking for entity matching. *Journal of Parallel and Distributed Computing*, 101:27 – 40.

[Monteiro et al. 2017] Monteiro, C. M., Cruzeiro Martins, F. V., and Davis Junior, C. A. (2017). Optimization of new pick-up and drop-off points for public transportation. In *XVIII GEOINFO*, pages 222–233.

[Sousa Junior et al. 2014] Sousa Junior, S. R. d., Lima, R. d. S., and Cunha, R. A. H. d. (2014). Crowdbus: Aplicativo crowdsourcing para informação, localização, avaliação e fiscalização de frotas de ônibus. *SEGeT : Simpósio de Excelência em Gestão e Tecnologia*, XI.

[Souza et al. 2015] Souza, R., Oliveira, I. P., Junior, F., Sales, L., and Ferraz, F. (2015). Beyond efficiency: How to use geolocation applications to improve citizens well-being. In *SMART'2015*, pages 37 – 40.

[Vila et al. 2016] Vila, J. J. R., Kozievitch, N. P., Gadda, T. M. C., Fonseca, K., Rosa, M. O., Gomes-jr, L. C., and Akbar, M. (2016). Urban mobility challenges – an exploratory analysis of public transportation data in curitiba. *Revista de Informática Aplicada*, 12:1–14.

[Weigang et al. 2001] Weigang, L., Yamashita, Y., da Silva, O. Q., XiJun, D., dos Prazeres, M. a. T., and de Oliveira, D. C. S. (2001). Implementação do sistema de mapeamento de uma linha de ônibus para um sistema de transporte inteligente. In *SEMISH'2001*, pages 72–85.

[Yang et al. 2000] Yang, H., Bell, M. G., and Meng, Q. (2000). Modeling the capacity and level of service of urban transportation networks. *Transportation Research Part B: Methodological*, 34(4):255 – 275.

Generating Artificial Data for Bus Travel Time Predictions

Leandro S. Ribeiro[1], **Thiago P. Faleiros**[1], **Maristela Holanda**[1]

[1]Computer Science Department – University of Brasilia (UNB)
Brasília, Distrito Federal, Brazil

leandro.santos.r@gmail.com,{thiagodepaulo,mholanda}@unb.br

Abstract. *This paper proposes a simulator capable of quickly generating a large amount of data that may be used to train bus travel time predictive algorithms in an urban transport network. To validate the proposal, a case study was carried out on a bus line in the city of Brasília/DF, Brazil. In the case study, the Simulator generated data for several scenarios that differ in distinct levels of variability and these data were used to evaluate the performance of a K-Nearest Neighbor predictor in each of the scenarios.*

1. Introduction

Urban residents rely on different modes of public transportation for their daily commute. Based on that, many works have been developed aiming to improve the efficiency and the accessibility of urban transportation services [Lima and Campos 2017, Monteiro et al. 2017]. In this context, accurate and real-time travel time information for buses has an important role because, among other reasons, it can help passengers better plan their trips and minimize waiting times.

Many factors such as weather conditions, day of week, time of day, and current traffic conditions may influence bus travel times. However, the exact nature of such relationships between travel times and predictor variables is usually not known and somehow these factors need to be incorporated into prediction algorithms either indirectly through binned analyses or through direct modeling [Kormáksson et al. 2014]. Moreover, travel times in urban areas are prone to high degrees of variability due to the presence of traffic lights, congestion, geometric conditions of roads and weather conditions [Reddy et al. 2016].

High variability conditions may affect the performance of the travel time predictor. For instance, [Reddy et al. 2016] proposes bus travel time predictions under high variability conditions and concludes that Kalman Filter [Sorenson 1966] and Support Vector Machines (SVM) [Platt 1999] are promising prediction techniques to solve high variability problems. Therefore, the travel time prediction method choice should account for the degree of environmental variability.

One way to analyze the degree to which variability impacts predicting bus travel times in an urban transport network is to collect data from the network in a variety of situations with high and low degrees of variability, then predict bus travel times for each of these situations using a predictor. Finally, predictor performance can be compared for each of these scenarios. However, to run this experiment with data from a complex transport network in a large urban center and different scenarios changing traffic parameters to check simulator performance, is unfeasible, since the costs involved in creating each of the different scenarios in real world network would be prohibitive. Therefore, one way to overcome this challenge would be to use traffic simulation tools. Wen et al. in [Wen 2018] highlight safety, convenience and low cost as advantages of using a simulation.

The main challenge of this work is to propose a simulator capable of quickly generating a large amount of data that may be used to train bus travel times predictive algorithms. In addition, the simulator must be able to generate data for several scenarios, so this data may be used to evaluate the performance of prediction algorithms for each scenario. The main requirements for the simulator include: the ability to provide geographic location and velocity of buses when requested, geolocation data available through a Representational State Transfer (REST) Application Programming Interface (API), allow simulation variability degree adjustments using parameters, simple modeling and integration with other systems through a geographic database.

In order to validate the proposed tool, a case study was carried out simulating the traffic of a bus line in the city of Brasília/DF, Brazil. In this case study, the bus lines were represented as graphs in which bus stops are represented by nodes, and the roads between bus stops are the edges. The data generated by the simulation were used to train and test a machine learning algorithm to predict bus travel times. The results showed that the degree of variability affects the performance of the predictor in terms of the calculated error.

The current paper is organized as follows: in Section 2 the related works are discussed; in Section 3 the architecture of the proposed tool and its components are described; in Section 4 a case study with its respective results are presented; and finally, in Section 5 there are the conclusions.

2. Related Works

Simulation models can be classified into two types: macroscopic and microscopic [Helbing et al. 2002]. The macroscopic traffic models are restricted to the description of the collective vehicle dynamics in terms of the spatial vehicle density $\rho(x, t)$ and the average velocity $V(x, t)$ as a function of the location x and time t. In contrast, the microscopic traffic models delineate the positions $x_{\mathrm{a}}(t)$ and velocities $v_{\mathrm{a}}(t)$ of all interacting vehicles as a function of time t.

The macroscopic models are addressed to large scale traffic, treating traffic as a liquid, frequently applying hydrodynamic flow theory to vehicle behavior, such as Cellular Automaton (CA), car following model and IDM/MOBIL as widespread models used within the traffic science community [Sommer et al. 2011]. All these approaches are of equal value in terms of mobility models.

In [Wen 2018], traffic simulations of vehicles in a connected environment, traveling through a commercial area, were carried out with the PARAMICS tool to test the collection of traffic data and the prediction of travel times, four types of models were constructed for the analysis of travel times: linear regression, multivariate adaptive regression spline, stepwise regression and elastic net. The results showed that the approaches had similar performance in terms of Root Mean Square Error (RMSE).

Similarly, in [Barceló et al. 2010] it is presented a comprehensive list of traffic simulation softwares describing their approaches to model building and implementation. The microscopic approach is represented by the softwares VISSIM, AVENUE, Paramics, Aimsun, MITSIM, SUMO and DRACULA. The macroscopic approach is represented by

METANET. Although most of these tools are complete in terms of functionality, a simpler model may be implemented to generate a large volume of data with less computational effort.

When there is a concern about the exact positions of simulated nodes, macroscopic and mesoscopic models cannot offer this level of detail, then only microscopic simulations are considered. Microscopic simulations model the behavior of single vehicles and interactions between them [Sommer et al. 2011]. However, there are different disadvantages, such as a high computational time and the requirement of detailed information, which might limit the applicability of a microscopic model to medium networks and those that do not operate in real time [Adacher and Tiriolo 2018].

Considering the trade-off between macroscopic and microscopic models, one of the challenges of this work is to propose a simulator that can provide the exact position of each simulated node, but with a good applicability in real-time applications and without the need for high computational time, merging characteristics of the macroscopic and microscopic models. Considering the high computational cost required in the general simulator softwares, and the lack of specific simulator to generate training data for bus travel time predictive algorithms, we propose a dedicated simulator able to model several external events that may occur in daily traffic and to generate data useful for experimental evaluation of predictive algorithms. The proposed simulator seeks to achieve these objectives by reducing the complexity of the model, and the results maintains the necessary coherence for the analysis of the performance of travel time predictors.

3. Simulator Architecture

The Simulator consists of four software components and a Geographic Database. The software components are: Time Controller, Line Simulator, Trip Simulator, and Location REST Service. The diagram with the components is shown in Figure 1. Each software component will be described in the following sections.

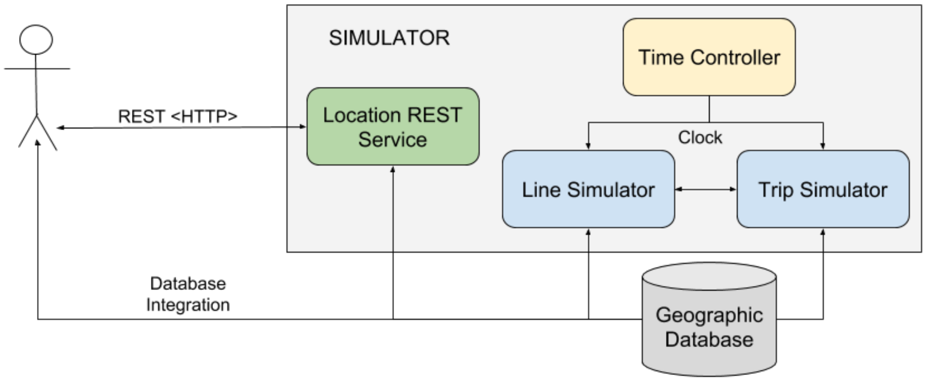


Figure 1. Simulator Architecture.

3.1. Line Simulator

The Line Simulator performs scheduled tasks according to a parameterized period. The main activities performed by this component are: updating edge status, updating neighboring edges influence, updating edges average velocities, updating node delays, and updating edges in geographic database. The responsibility of this component is to update attributes related to the behavior of roads and bus stops.

To simulate the occurrence of events, such as accidents or bad weather, each edge is classified as: normal, light event, moderate event, and severe event. The normal status represents a situation without events. In other words, it is not under the influence of external events that may negatively impact the traffic flow, whereas the other statuses represent the occurrence of events that impact the traffic flow with increasing degrees of severity from light to severe.

During the edge's status updating process, if an edge has normal status, a pseudo-random probabilistic event determines an increasing probability of severe, moderate and light events. This means the occurring probability of a light event is greater than the occurring probability of a severe event. If no event occurs, the edge retains normal status.

If an edge is under the effect of some event during the edge's status updating process, the event regression probability is evaluated. A severe event regresses to a moderate event, while a moderate event regresses to a light event and a light event regresses to the normal status. Again, the regression probability of a light event is greater than the regression probability of a moderate event, which is greater than the regression probability of a severe event. Figure 2 illustrates the edge status state machine.

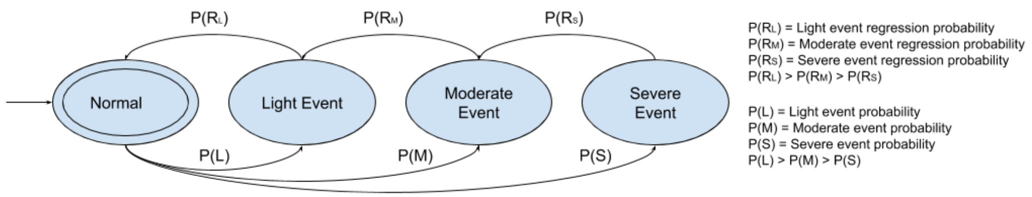

$P(R_L)$
$P(R_M)$
$P(R_S)$
Normal
Light Event
Moderate Event
Severe Event
$P(L)$
$P(M)$
$P(S)$
$P(R_L)$ = Light event regression probability
$P(R_M)$ = Moderate event regression probability
$P(R_S)$ = Severe event regression probability
$P(R_L) > P(R_M) > P(R_S)$
$P(L)$ = Light event probability
$P(M)$ = Moderate event probability
$P(S)$ = Severe event probability
$P(L) > P(M) > P(S)$

Figure 2. Edge Status State Machine.

To represent the influence of neighboring edges, four influence classifications were created: severe, moderate, light and absent. An edge is under severe influence when the edge immediately downstream has a severe event, on the other hand, an edge is under moderate influence when the edge that is after the edge immediately downstream has a severe event. Finally, an edge is under light influence when the edge immediately upstream has a severe event. If the edge does not fit into any of the above rules it will be under absent influence. Figure 3 illustrates the influence of an edge under severe event on its neighboring edges.

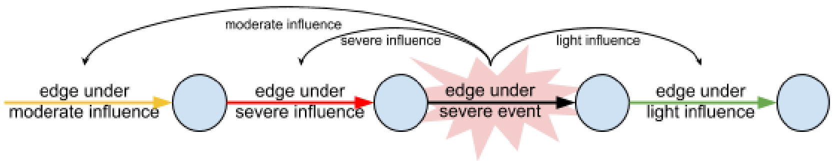

moderate influence
severe influence
light influence
edge under moderate influence
edge under severe influence
edge under severe event
edge under light influence

Figure 3. The Neighboring Edges' Influence.

After updating edge status and edge influence, edge velocity is updated. Each edge has an average velocity that is initially represented by the maximum velocity allowed in that edge. In the edge updating average velocity first step, the path maximum velocity is multiplied by a correction factor that varies according to the edge status. This factor is always a number less than or equal to one. The result velocity will always be less than or equal to the maximum edge velocity allowed. The higher the severity of the event, the lower the correction factor. This correction factor has a normal distribution so that one simulation parameter determines the mean and another determines the standard deviation.

In the second update step, the result calculated in the first step is multiplied by a peak time correction factor. This factor is not applied if the current time is outside the interval of the peak hour. Similar to the status correction factor, the peak time correction factor has a parameter that determines the Gaussian mean and a parameter that determines the Gaussian standard deviation, but in this case, the Gaussian mean is not a constant value. Rather, it is determined by a linear discontinuous function that decreases in the first half of the peak hour window and increases in the second half of this window, so that the velocity variation becomes a little smoother. The discontinuous linear function can be replaced by any other mathematical function to better model this phenomenon.

Finally, in the third update step, the result of the second step is multiplied to the influence correction factor – the greater the influence of the neighboring edge, the lower the influence correction factor and the lower the final average velocity. As with all other correction factors, this factor has a normal distribution parameterized by the mean and the standard deviation.

Node delays represent the amount of time buses spend at bus stops. These delays also have a normal distribution with mean and standard deviation parametrized. Presently, to simplify the process, the peak time window does not affect node delays. However, it is possible to apply a peak time correction factor to these delays, increasing the delay at peak hours.

The last activity of the Line Simulator is to update the Geographic Database with edge average velocities. This allows any graphical tool that can integrate with a geographic database to monitor edge status and edge average velocity at simulation time.

3.2. Trip Simulator

Similar to the Line Simulator, the Trip Simulator executes scheduled tasks according to a parameterized period. The main activities of this component are: updating bus position, updating bus velocity, and updating Geographic Database with bus positions. The responsibility of the Trip Simulator is to update the behavior attributes of the buses.

Each graph element behaves differently, whether it is an edge or a node. When a fraction of time is available to a bus that is on a node, this time is consumed while the bus stands on the bus stop. However, when the bus is on an edge and a fraction of time is available, it moves over the edge at a distance that is a function of its instantaneous velocity and the available time. The travel times spent by all the buses in each of the stops or crossing each of the edges are recorded in the Geographic Database at the exact moment that the vehicle leaves the respective graph element.

The instantaneous bus velocity is a function of the graph element average velocity. Graph nodes always have zero average velocity, while the average velocities of edges are calculated by the Line Simulator. The instantaneous velocity of the buses is the multiplication of the edge average velocity by a velocity oscillation factor. Consequently, two buses on the same edge do not necessarily have the same instantaneous velocity. The buses' instantaneous velocities have a normal distribution, with mean equals to edge average velocity and parameterizable standard deviation, which is the velocity oscillation factor.

Node delays are multiplied by a delay oscillation factor, so two buses do not necessarily remain the same time period in a certain bus stop. The delay oscillation factor also has a normal distribution with mean equals to the node average delay and parameterizable standard deviation – the delay oscillation factor.

Finally, the Geographic Database is updated with the positions of the buses, allowing other tools to monitor these positions at runtime through database integration.

3.3. Location REST Service

The Location REST Service allows retrieving information from the entire bus fleet through a Hypertext Transfer Protocol (HTTP) call. The information is made available through a JavaScript Object Notation (JSON) document and includes name, line, velocity, and location (latitude and longitude) of all buses.

This service, in addition to allowing system integration independent from the platform, the operational system, or programming language, it also makes possible to obtain bus fleet data in a real time.

3.4. Time Controller

The Time Controller is nothing more than a clock that controls the simulation time. All other system components use the Time Controller clock instead of the operational system clock. The Time Controller clock speed is set by the ”Time Multiplier” parameter. If the parameter is set to two, then the time passage in the simulation will be twice as fast as the real time passage. Therefore, this component allows the simulation to be executed both in real time and in an accelerated time, so that weeks of traffic simulation can be executed in a few hours.

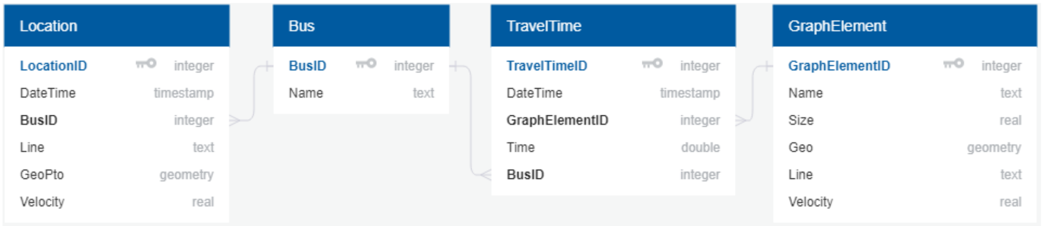
3.5. Geographic Database

The Geographic Database, illustrated in Figure 4, stores information from nodes and edges, bus locations over time, and travel time information of all edges and nodes. In addition to maintaining the entire simulation history the database also works as an integration point between the Simulator and other systems.

The ”Location” table, in Figure 4, stores the buses' geographic location and velocity at a given time. The ”TravelTime” table records the time a bus has remained in a given graph element. The ”GraphElement” table has all graph nodes and edges including each element name, size, maximum speed allowed , and the bus line name. When the element is a node, the ”Graph Element” table has the geolocated point with the node location. However, when the element is an edge this table stores the set of geolocated points and lines that represent the edge trace and position.

3.6. Simulator Parameters

Adjusting the Simulator parameters, a traffic situation with high variability can be created: constant changes in bus speeds, occurrence of many external events and many other traffic



Location	
LocationID	integer
DateTime	timestamp
BusID	integer
Line	text
GeoPto	geometry
Velocity	real

Bus	
BusID	integer
Name	text

TravelTime	
TravelTimeID	integer
DateTime	timestamp
GraphElementID	integer
Time	double
BusID	integer

GraphElement	
GraphElementID	integer
Name	text
Size	real
Geo	geometry
Line	text
Velocity	real

Figure 4. Geographic Database.

situations. Likewise, other adjustments can create a situation with low variability: with constant velocities and little or no occurrence of external events.

In addition, by adjusting the parameters it is possible to define the number of buses that will be running during the simulation, the windows of the peak hours, the Line Simulator and the Trip Simulator update periods, and the time multiplication factor. Table 1 lists all Simulator parameters.

Table 1. Simulator Parameters

Name	**Description**
fleetSize	Number of buses
severeEventProb	Severe event probability
moderateEventProb	Moderate event probability
lightEventProb	Light event probability
normalCorrectionFactor	Normal status correction factor
lightCorrectionFactor	Light event correction factor
moderateCorrectionFactor	Moderate event correction factor
severeCorrectionFactor	Severe event correction factor
correctionFactorSD	Correction factor standard deviation
severeEventEndProb	Severe event ending probability
moderateEventEndProb	Moderate event ending probability
lightEventEndProb	Light event ending probability
absentInfluence	Absent influence factor
lightInfluence	Light influence factor
moderateInfluence	Moderate influence factor
severeInfluence	Severe influence factor
influenceSD	Influence factor standard deviation
morningPeakTime	Morning peak time
afternoonPeakTime	Afternoon peak time
tripSimulatorUpdate	Trip Simulator update period
lineSimulatorUpdate	Line Simulator update period
timeMultiplier	Simulation time multiplier factor
delayOscillationFactor	Node delay oscillation factor
delayOscillationFactorSD	Standard deviation of the node delay oscillation factor
velocityOscillationFactor	Edge velocity oscillation factor
velocitOscillationFactorSD	Standard deviation of the edge velocity oscillation factor
peakTimeCorrectionFactor	Peak time correction factor
peakTimeCorrectionFactorSD	Standard deviation of the peak time correction factor

4. Case Study

A case study was carried out using simulated data to evaluate a travel time predictor performance using scenarios with different variability degrees. The simulations were performed on a bus line located in downtown Brasília / DF, Brazil.

Geolocation data from the Federal District urban transport network, on the right side of Figure 5, were obtained from a GeoServer [Contributors 2015] maintained by the Federal District Transit Department (DFTRANS). For the case study, only the ”CIRCULAR-W3-SOUTH-NORTH-L2-NORTH-SOUTH” bus line was used, on the left side of Figure 5.

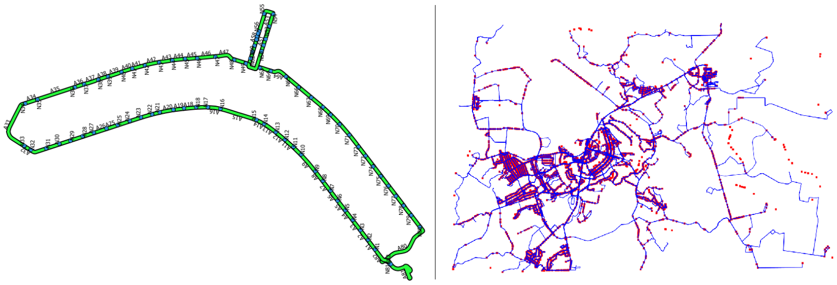


Figure 5. Federal District Urban Transport Network.

The bus line was transformed into a graph with 82 nodes named after N1 to N82 and 82 edges named after A1 to A82. The bus stops are represented by nodes, while the paths between bus stops are the graph edges. Travel time predictions take two nodes into account: the origin node (O) and the destination node (D), and the single path composed of one or more edges (A), which connects these two nodes, as shown in Figure 6.

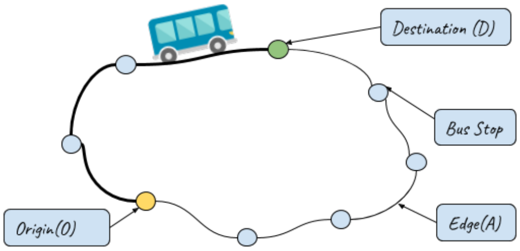


Figure 6. Bus Line Graph.

To create different levels of variability, simulator and predictor parameters were changed. The parameters were divided into 5 sets C_i, C_j, C_k, C_l, and C_m. Each of these sets has 3 different configurations. For instance, $C_i(0)$ is the parameter set C_i in configuration 0 while $C_m(2)$ is the parameter set C_m in configuration 2. The parameter sets C_i, C_j, C_k, and C_i are simulator parameters and are listed in Table 1 with their respective descriptions. The parameter set C_m has a single predictor parameter, which determines the quantity of previous travel times used to train the prediction algorithm. Since there are 4 simulator parameter sets – 1 predictor parameter set, and each one with 3 possible configurations – the total amount of simulated scenarios is 81, and the total amount of predicted scenarios is 243. Table 2 shows the 5 parameter sets and their values for each configuration.

During the experiments, the simulator ran for 3 hours for each of the scenarios described, with a fleet of 82 buses and a time multiplication factor of 60. Then, for 3 hours of simulation a little more than a week of traffic data was generated.

The prediction of travel times was performed by a K-Nearest Neighbors classifier (KNN) [Aha et al. 1991]. This classifier was chosen due to its speed and the simplicity of

Table 2. Simulation Scenarios

Ci			
Parameter	**Ci(0)**	**Ci(1)**	**Ci(2)**
severeEventProb	0,0000	0,0005	0,0010
moderateEventProb	0,0000	0,0010	0,0020
lightEventProb	0,0000	0,0020	0,0040
Cj			
Parameter	**Cj(0)**	**Cj(1)**	**Cj(2)**
lightCorrectionFactor	0,90	0,80	0,70
moderateCorrectionFactor	0,75	0,65	0,55
severeCorrectionFactor	0,60	0,50	0,40
peakTimeCorrectionFactor	0,80	0,70	0,60
Ck			
Parameter	**Ck(0)**	**Ck(1)**	**Ck(2)**
lightInfluence	1,00	0,90	0,80
moderateInfluence	1,00	0,80	0,70
severeInfluence	1,00	0,70	0,60
Cl			
Parameter	**Cl(0)**	**Cl(1)**	**Cl(2)**
delayOscillationFactorSD	0,01	0,05	0,10
velocitOscillationFactorSD	0,01	0,05	0,10
Cm			
Parameter	**Cm(0)**	**Cm(1)**	**Cm(2)**
qtdPreviousTrips	6	4	2

its implementation. This approach is one of the simplest and oldest methods used for pattern classification. It often yields efficient performance and, in certain cases, its accuracy is greater than state-of the-art predictors [Hassanat et al. 2014, Hamamoto et al. 1997]. The KNN predicts the travel time of a test example using the average past travel time among its k-nearest (most similar) neighbors in the training set.

The data used to train and test the models were: the current travel time, the period of the day represented by a real number contained in the interval $[0; 24)$, the day of the week, the average bus speed, and η previous travel times where η is determined by the predictor parameter ”qtdPreviousTrips”.

In the KNN process, the number of neighbors k needs to be determined. To determine the parameter k, an experiment was carried out creating prediction models for a random sample of 8 edges extracted from the real traffic network. The set of sampled edges are labeled as $\{A_1, A_9, A_{33}, A_{35}, A_{51}, A_{58}, A_{80}, A_{81}\}$, where A_i is the edge in the geographic database indexed by value i. The value of k was ranged from 1 to 52, and for each value of k the model was created and evaluated by calculating the absolute mean error. Figure 7 presents the absolute mean errors calculated during the evaluation of the prediction model as a function of k for the 8 sampled edges. The curves initially have a decreasing behavior, and around $k = 4$ there is an inflection point and the curves starts to increase. Due to this behavior the value 4 was chosen for parameter k.

After estimating the value of k, KNN prediction models were created for all edges and for many parameter scenarios. The simulation data were divided into 70% for training and 30% for test, and for each test example, the value of Mean Absolute Error (MAE) was calculated, corresponding the absolute difference between the predicted travel time and

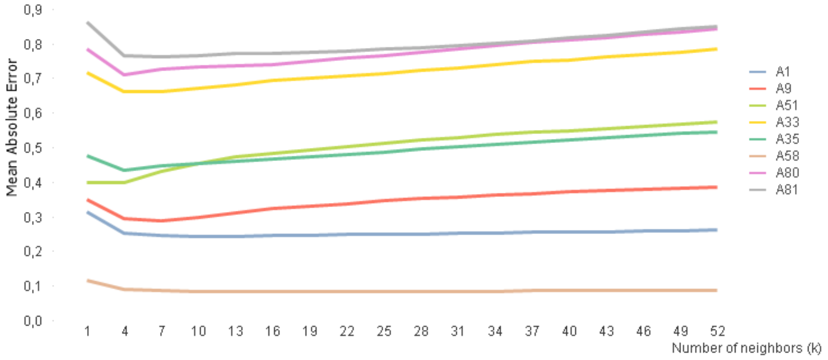


Figure 7. Mean Absolute Errors.

the real travel time.

The results predicted by KNN from generated data with several scenarios are summarized in Figure 8. Figure 8 represents a plot matrix, in which each plot shows travel time predictor mean absolute errors for each of the 243 prediction scenarios. Columns of the plot matrix represent subset of edges A_1, A_{33}, and A_{80}, while lines represent configuration groups C_i, C_j, C_k, C_l, and C_m. Numbers 0, 1 and 2 in plot matrix legends are related to simulation scenarios presented in Table 2. Not all predicted travel time results are presented here due to space limitations to plot all parameter combinations for each edge. The complete results and data can be found in `https://github.com/curupiras/results.git`. Nevertheless, the subsample presented in Figure 8 represents the pattern found in all edges.

In Figure 8, the plot matrix in line C_i shows that the greater the probability of the events, the greater the predictor error. Similarly, it can be observed in line C_j that the smaller the correction factors, the more the events and the peak time impact on bus speeds and the greater the predictor error.

Similarly, line C_k represents the impact of influence between neighboring edges on the predictor error and line C_l represents the impact of delay and velocity oscillation factors on the predictor performance. At some moments, a lower influence between neighboring edges and smaller oscillation factors end up increasing the predictor error, however this is not the rule. In most cases, the greater the oscillation factors, and the greater the influence of the neighboring edge, the lower the predictor performance, thus, the greater the error.

Finally, in line C_m, it is worthy to note that the quantity of previous travel times used in the predictor algorithm training has very little impact on its performance. However, it is possible to realize that when a greater quantity of previous travel times is used to train the model, there is a slight improvement in the predictor performance.

5. Conclusion

In this work, a simulator capable of quickly generating a large amount of data for travel time prediction algorithms was presented. The implementation was able to provide the geographic position and speed of each simulated bus, upon request, either using REST API calls or geographic database integration.

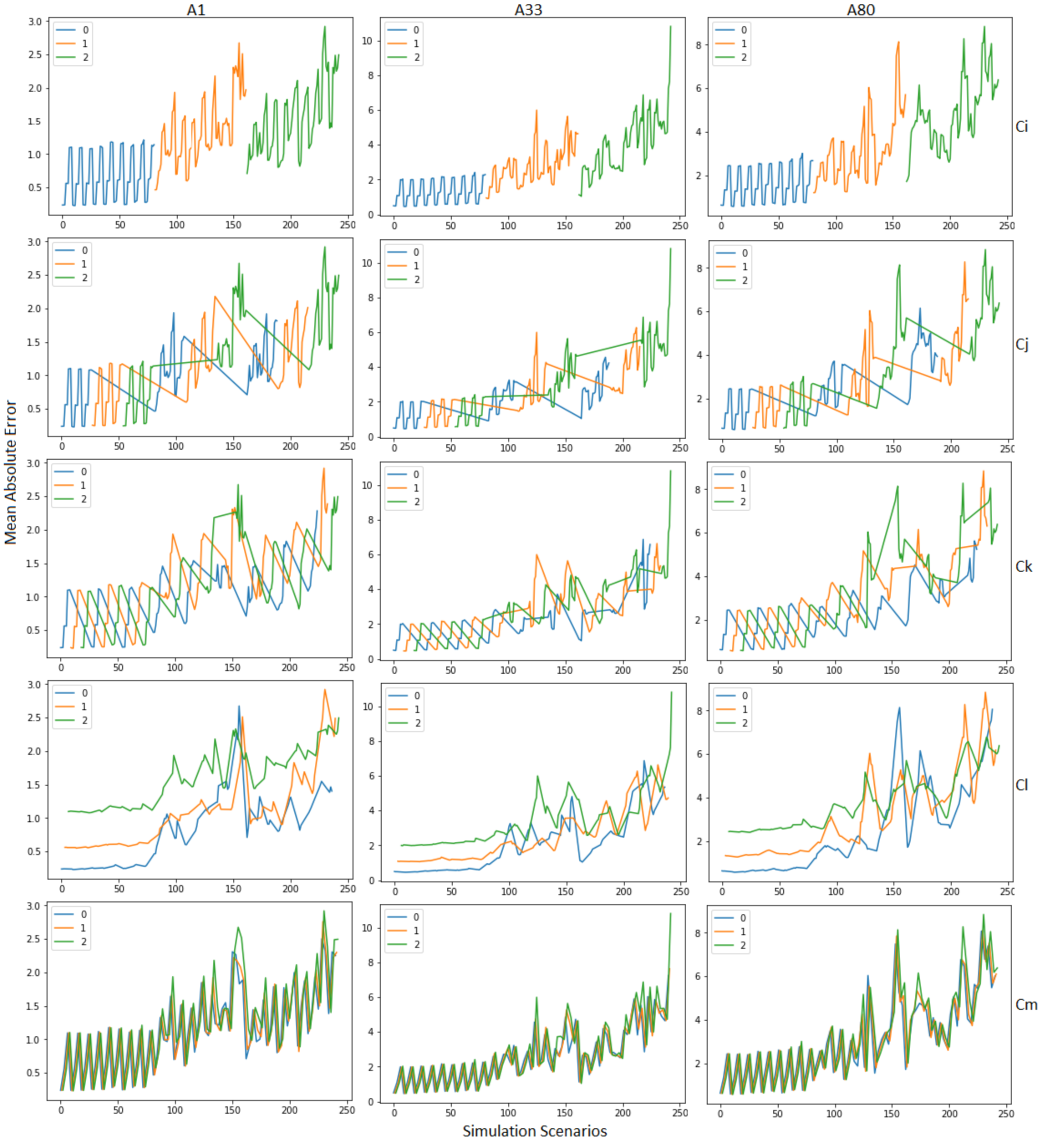


Figure 8. Predictor Performance.

Furthermore, to validate the implementation, a case study was carried out on a bus line in the city of Brasília. Over 20 months of traffic data with different levels of variability were generated in approximately 10 days, proving the Simulator's ability to generate a large quantity of data in a short period of time for several different scenarios. The simulated data were used to train and test a KNN bus travel times predictor and the predictor algorithm performance was evaluated in terms of mean absolute error. The results showed that, in general, the higher the degree of variability of the traffic environment, the lower the performance of the prediction algorithm.

Future work includes: comparing simulated data with real traffic data; testing the simulator with other bus lines; the use of other prediction algorithms and the analysis of their performance; refinements of the model to simulate traffic in multiple lanes; and to add the possibility of multiple paths choice.

References

Adacher, L. and Tiriolo, M. (2018). A macroscopic model with the advantages of microscopic model: A review of cell transmission model's extensions for urban traffic networks. *Simulation Modelling Practice and Theory*.

Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, 6(1):37–66.

Barceló, J. et al. (2010). *Fundamentals of traffic simulation*, volume 145. Springer.

Contributors, G. (2015). Geoserver–open source server for sharing geospatial data.

Hamamoto, Y., Uchimura, S., and Tomita, S. (1997). A bootstrap technique for nearest neighbor classifier design. 19:73 – 79.

Hassanat, A. B., Abbadi, M. A., Altarawneh, G. A., and Alhasanat, A. A. (2014). Solving the problem of the k parameter in the knn classifier using an ensemble learning approach. *CoRR*, abs/1409.0919.

Helbing, D., Hennecke, A., Shvetsov, V., and Treiber, M. (2002). Micro-and macro-simulation of freeway traffic. *Mathematical and computer modelling*, 35(5-6):517–547.

Kormáksson, M., Barbosa, L., Vieira, M. R., and Zadrozny, B. (2014). Bus travel time predictions using additive models. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 875–880. IEEE.

Lima, A. M. and Campos, J. (2017). How reliable is the traffic information gathered from web map services? In *GEOINFO*, pages 234–245.

Monteiro, C. M., Martins, F. V. C., and Junior, C. A. D. (2017). Optimization of new pick-up and drop-off points for public transportation. pages 222–233.

Platt, J. C. (1999). 12 fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods*, pages 185–208.

Reddy, K. K., Kumar, B. A., and Vanajakshi, L. (2016). Bus travel time prediction under high variability conditions. *Current Science (00113891)*, 111(4).

Sommer, C., German, R., and Dressler, F. (2011). Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15.

Sorenson, H. W. (1966). Kalman filtering techniques. In *Advances in control systems*, volume 3, pages 219–292. Elsevier.

Wen, X. (2018). A work zone simulation model for travel time prediction in a connected vehicle environment. *arXiv preprint arXiv:1801.07579*.

Creating Municipal Databases from OpenStreetMap: The Conceptual Database Schema

Vinícius Garcia Sperandio[1], Vitor Eduardo Concesso Dias[1],
Sérgio Murilo Stempliuc[2], Jugurta Lisboa-Filho[1]

Universidade Federal de Viçosa (UFV) - Viçosa, MG, Brazil[1]

Faculdade Governador Ozanam Coelho (FAGOC) - Ubá, MG, Brazil[2]

vinicius.sperandio@ufv.br, vitor.dias@ufv.br,
smstempliuc@gmail.com, jugurta@ufv.br

Abstract. *The systematic use of volunteered geographic information (VGI) has increased in face of the easy production of spatial data from several sources and devices. OpenStreetMap is a VGI platform that collects collaborative geographic data from any region on Earth and makes it openly available free of charge. This paper presents a method for automatic conceptual schema generation based on metadata extracted from the OpenStreetMap platform aiming to create a database to support decision-making in municipal administration. The experiments were carried out on a pilot area, which enabled testing the efficacy of the method to generate conceptual schemas.*

1. Introduction

The popularization of web 2.0 has favored the development of applications that allow users to share spatial information over the Internet. Hence, users become not only consumers, but also contributors and producers of information [Budhathoki 2007]. Ever since, platforms that allow users to be more than consumers and start acting as sensors and volunteered data producers have been standing out [Goodchild 2007]. Volunteered geographic information (VGI) has been the main source of data on platforms such as OpenStreetMap (OSM) [OSM 2018] and Wikimapia [Wikimapia 2018]. These systems allow users to map any part of the globe in a free, rapid, and intuitive manner [Haklay and Weber 2008].

According to Goodchild (2007), official geographic information production has decreased in recent decades especially due to the high cost of generating such information and the cuts in funding for cartographic services. This way volunteered geographic information has been used as a way to minimize this issue.

In Brazil, small municipalities are particularly impacted by restricted funds, besides the lack of qualified labor to produce and maintain geographic information. Such information is important for planning and decision-making toward the economic and social evolution of a city along with environmental matters. A municipality must have access to information on its territory to be administered with more quality and efficiency [Miranda et al. 2012].

These data are usually stored in spatial databases and are handled by geographic information systems (GIS) so that managers are able to carry out spatial analyses that support decision-making. However, it is important that data modeling be accurate since that is when the real-world elements are transformed into database elements [Elmasri and Navathe 2011]. Consequently, good modeling reduces the need for corrective maintenance, contributes to better understanding of the data stored and their relationships, and allows for a greater number of spatial analyses. That prevents future expenses since maintenance is the software engineering activity that generates the most cost and a high volume of effort.

The OpenStreetMap platform aims to provide and collect collaborative geographic data free of charge from the local knowledge of its users. It thus becomes a free alternative for municipal mapping to which the population itself can contribute, despite lacking training in cartography[1]. In 4 years, its taxpayers mapped 29% of the English territory, achieving 80% of similarity with cartographic bases of national agencies [Haklay 2010]. Nonetheless, in order for data to be used in a decision-making process, it is not enough to directly export the data to a geographic database, but rather the data must be properly structured to be used in spatial analyses through GIS software. For example, QGIS is a free piece of software that is able to read the file exported by the OSM platform and allows data visualization and handling, but the tables of attributes are generated according to the geographic types of each element. Therefore, elements or type *point*, for instance, are added to the same table, causing issues with data normalization. A poorly structured database hampers the understanding of the data acquired and limits the generation of spatial queries.

This paper aims to describe the automated generation process of the conceptual schema from data extracted from the OpenStreetMap platform to create a database. From a conceptual schema, a well-structured database can be automatically generated using CASE tools. Section 2 describes the OpenStreetMap platform, particularly the OSM-XML data file. Section 3 cites some works related to the research. The method used to automatically generate the conceptual schema is described in Section 4. A case study carried out on a small pilot area is presented in Section 5. Section 6 presents some conclusions and the next steps of the project.

2. OpenStreetMap Platform

The OpenStreetMap platform, released on August 9th, 2004, is a collaborative project that aims to allow users to freely and voluntarily map any region in any country [OpenStreetMap 2018]. The platform makes its data available under the Open Database License, thus they can be exported and added to GIS and database management systems (DBMS) with support to spatial data, which enables their broad use [ODbL 2018].

In most collaborative systems, the user is able to create content, add some tags related to content, and share it with other users. The OpenStreetMap platform has a tag system for the mapping elements, where important characteristics to understand the elements can be added, such as informing that the item mapped is a five-story hospital with a helipad [Ballatore and Mooney 2015]. The map features[2] provided by OSM

[1] On the OpenStreetMap platform, volunteered data are approved by moderators, which ensures a minimum quality of data.

[2] https://wiki.openstreetmap.org/wiki/Map_Features

describes and illustrates each tag available so as to increase the odds that the data contributed can be properly rendered by map visualization tools that use the platform.

Users can export the data from any area selected in the platform. The data are extracted as an OSM-XML file with three types of fundamental objects: node, way, and relation [Mooney and Corcocan 2012]. The node represents a point defined by only a pair of coordinates. Nodes are used to represent point-like objects, such as a bus stop, a traffic light, or a monument. Objects of the type way are used to represent linear structures (polyline) such as streets, roads, water courses, or closed regions (polygons), such as buildings and borders. A relation represents the relationship between the previous elements and may be a restriction, such as informing places where vehicle access is restricted, or inform multi-polygons, such as indicating that a set of buildings are part of the same condominium. Besides the elements and their characteristics, an OSM-XML file stores the timeline of updates of each element, featuring the dates and users responsible for the changes.

According to Kitchin (2014), a large volume of data is produced by national censuses and governmental records on municipalities and their citizens. However, these data are based on sampling surveys and there is usually no continuity to these surveys, besides restrictions to data access. Consequently, these volumes of data must be complemented by what can be called small data studies. Questionnaire application, case studies, and interviews are used to capture specific details on issues related to the municipality. According to Miller (2010), much of what is currently known about cities has been obtained from studies characterized by data scarcity. On the other hand, the OpenStreetMap platform seeks to provide a broader understanding of urban control, often in real time, being characterized as a Big Data project for its characteristics such as large data volume, high rate of data generation, data often referenced in time and space, relationships, etc.

3. Related Works

On the OpenStreetMap platform, contributors are free to choose the tags they deem correct to characterize a site or geographic object. The OSM Wiki[3] website has a rulebook with suggestions and instructions on how to attribute a characteristic to an object during a contribution. Davidovic et al. (2016) verified the contributions on the OSM system from 40 cities in different continents to find out whether contributors in these areas are using the rules. After selecting and analyzing ten tags, they concluded that the use of the suggestions for most tags is poor. It is possible that some users do not understand the importance of some attributes and are concerned only with informing geometric aspects of the elements.

Pruvost and Mooney (2017) explored the data model exported by the OSM platform, particularly the relationships contained in it, such as logical clustering of object of types point, line, and polygon, responsible for representing geographic relationships in the real world. The study analyzed the relationships in four European cities and assessed their complexity, composition, and flexibility of the data model in OpenStreetMap.

[3] http://wiki.openstreetmap.org/wiki

The OSM platform uses urban crowdsourcing as a paradigm in the collection of spatial information on municipalities and has proven capable of providing data that can be compared to governmental sources, but data coverage may be low and unevenly distributed across the city. Quattrone et al. (2014) modeled the spontaneous growth of digital information in some areas so as to plan means of collecting content from areas with high chances of being neglected. The research proposed a digital growth model of volunteered spatial information based on urban physical growth models used by urban planners. In order to identify the factors responsible for influencing growth and how they can change with time, the tests were carried out with data from the city of London over five years.

Almendros-Jiménez and Becerra-Terón (2018) developed a framework to analyze the quality of tags applied through the OSM platform in Spain. The evaluation method examines quality measures such as integrity, reliability, and consistency using the website Taginfo[4] as reference. The main cities in Spain were selected to be compared with some European cities and a web tool was developed to enable this type of evaluation anywhere in the world with the same quality indicators.

4. Process of Generating the Conceptual Database Schema for a Municipality

The method proposed for the generation of the conceptual schema of a municipality, using volunteered geographic information, is split into two steps.

The first step made the conceptual modeling of all object classes in the real world that can be mapped in the OSM platform. The conceptual schema was created using the UML-GeoFrame model [Lisboa-Filho and Iochpe 2008]. Figure 1 shows a simplified diagram of themes generated to illustrate the understanding of the universe at hand (municipal base), abstracting the internal content of each theme. Eleven themes and their respective relationships were identified.

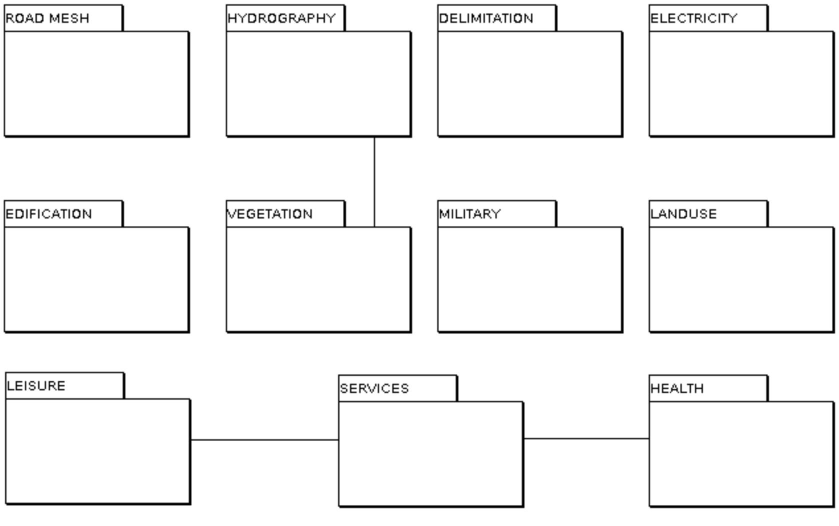


Figure 1. Overall diagram of themes of objects existing on the OSM platform

[4] https://taginfo.openstreetmap.org/

For each theme, the object classes that can be mapped on the OSM platform were modeled according to their level of affinity. Figure 2 illustrates a fragment of this schema and shows real-world object classes separated by themes. Moreover, examples of how the relationships occur among themes can be seen between themes "Leisure" and "Services" or between "Services" and "Health." It is important to point out that each class has many subclasses, which cannot be exhibited due to space constraints. For instance, class "Transportation" is a subclass of class "Amenity" and has several subclasses (e.g., "taxi," "parking," "fuel," "car_wash") that were not included in the modeling.

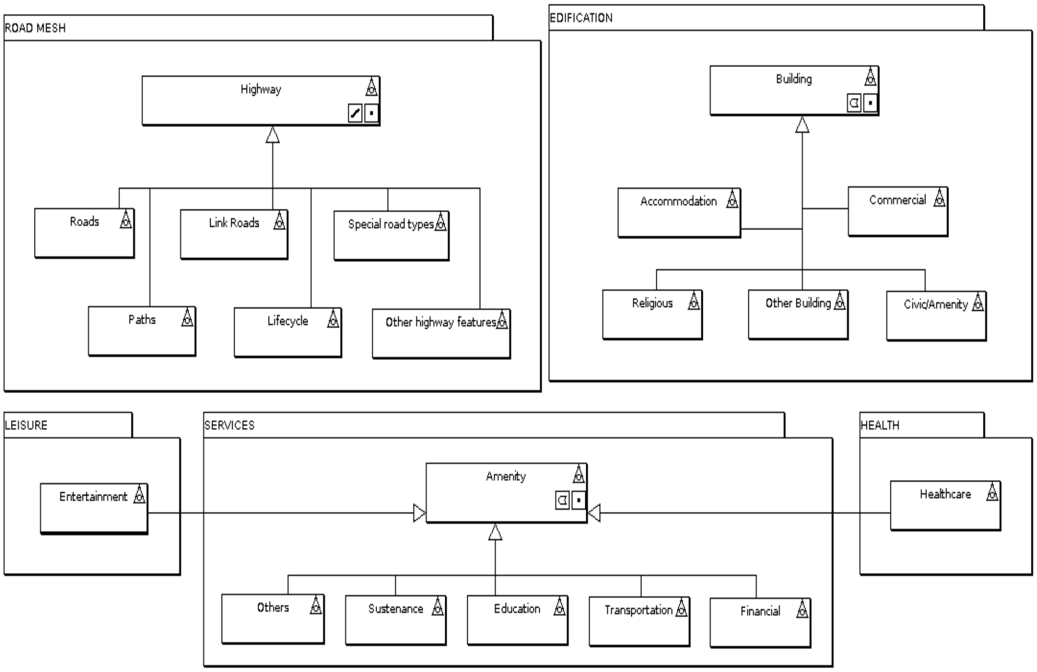


Figure 2. Examples of the class diagram of some themes modeled

The second step consists of reverse engineering from the OSM-XML metadata file extracted from the OSM platform that corresponds to a mapped area of a selected municipality for the creation of the conceptual data schema that contemplates the particularities of said municipality.

Figure 3 illustrates the simplified flowchart of the reverse engineering algorithm proposed whose input is the OSM-XML metadata file. The process starts with reading the OSM-XML file, from which the relevant information is extracted. The BeatifulSoap library [Beautiful Soap 2018] is used in this step to facilitate handling the XML file, which is parsed by the LXML library [Lxml 2018].

Next, the data obtained are tested to separate the mapped objects that have incomplete information from the others. During the separation, it is verified whether the object has a name. In case it does not, the object is stored on a list of nameless objects. This list is then saved as a text file in the format of a report containing the geographic stereotype, the coordinates, and the respective rectangle involving the incomplete objects.

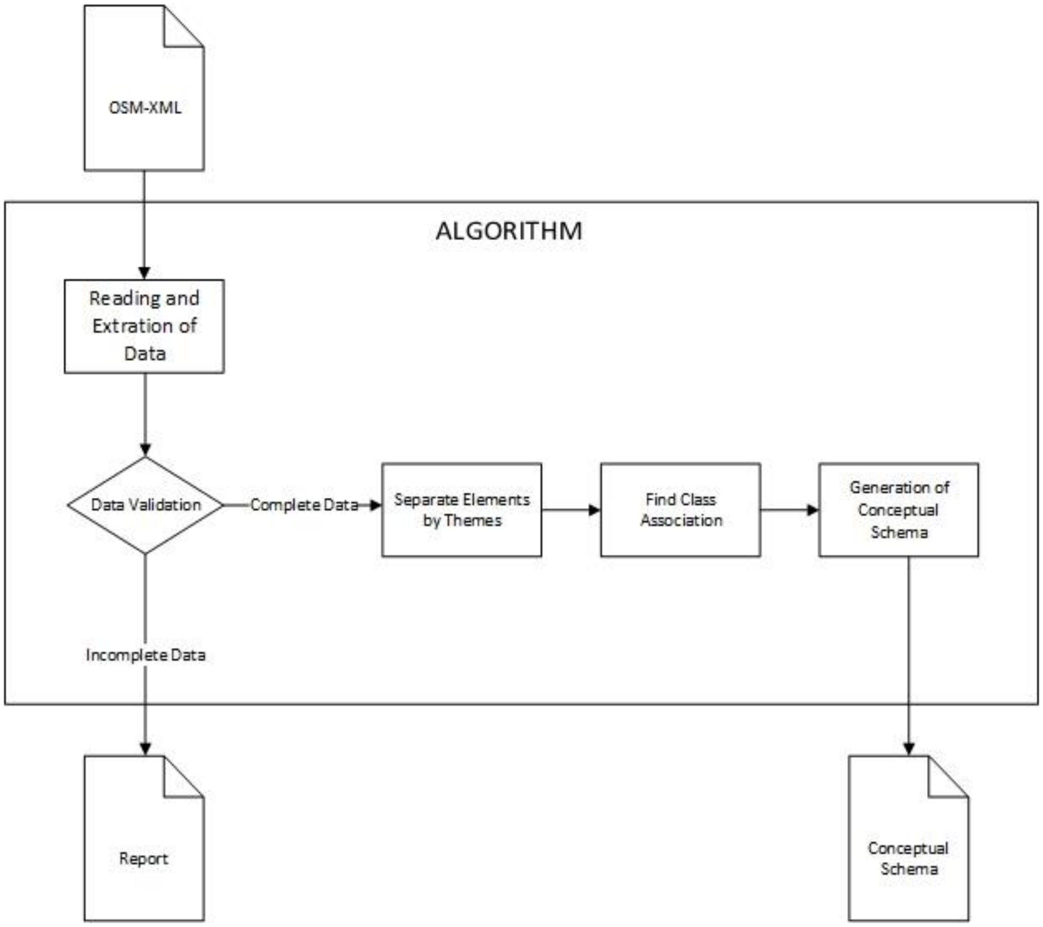


Figure 3. Simplified flowchart of the method proposed for generating the conceptual database schema

Valid objects are separated into lists according to the themes modeled (Figure 1) and the OSM-XM file provides the information regarding which superclass each object belongs to. For example, the part of the OSM-XML file that describes object “Hospital” contains a tag informing that it belongs to superclass “Amenity,” hence object “Hospital” will be added to the list that contains objects of theme “Services.”

Although the OSM-XML file provides a set of pieces of information (tags) on each object, that is still not enough to classify the exact subclass of objects. Therefore, a search must be performed among the subclasses of the superclass informed to find the exact class of the object. Taking object “Taxi” as an example, the OSM-XML file informs its superclass is “Amenity,” thus the subclasses of “Amenity” must be searched (Figure 2) to find the one that best aggregates object “Taxi.” That allows creating the relationship “Taxi” -> “Transportation” -> “Amenity.”

During this phase of searching classes and relationships, a file is generated containing the layout of the conceptual schema such as color, font, and icons in addition to objects as classes and their connections. In the end, this file is processed by the software Graphviz [Graphviz 2018], which can transform a textual conceptual schema into a graphical schema to generate the UML class diagram.

5. Case Study: Pilot Area – Medicine Department of UFV

The case study consisted in delimiting an area on OSM to enable an in-depth study of the OSM-XML file and future controlled tests. Figure 4 shows the small pilot area chosen, which is located within the Federal University of Viçosa (UFV), in the state of Minas Gerais, Brazil, and illustrates some elements such as a state public school, the Medicine Department, a parking lot, a Health Division, and some roadways.

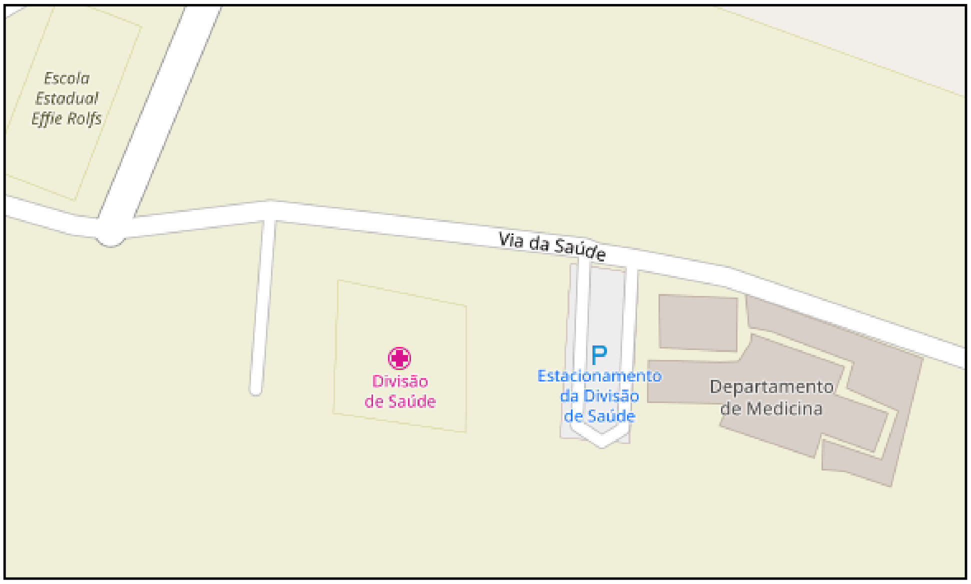
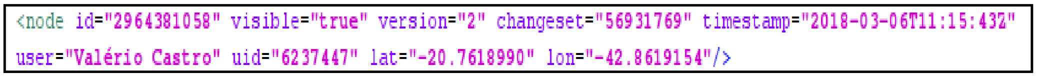

Escola Estadual Effie Rolfs
Via da Saúde
Divisão de Saúde
P
Estacionamento da Divisão de Saúde
Departamento de Medicina

Figure 4. Image of the pilot area – Medicine Department of UFV

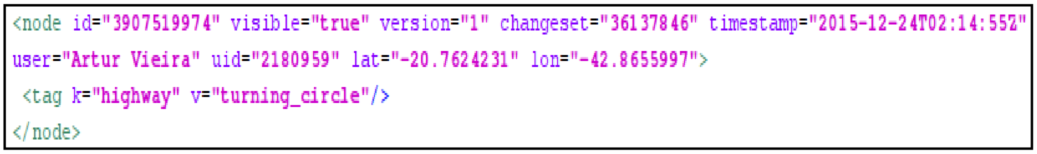
When the data of the area selected on the OSM platform is exported, the OSM-XML file is first ordered by all the *node* tags followed by the *way* tags and *relation* tags. All elements have an identifier (id) that can be used for relationships or dependencies. The *node* tag has only a single pair of coordinates (latitude and longitude), which will be used by some *way* tag, as shows Code 1. For example, the *node* tag of Code 1 refers to the coordinates of the upper left point of the Health Division and contains information on date and the volunteer user who created it.



```
<node id="2964381058" visible="true" version="2" changeset="56931769" timestamp="2018-03-06T11:15:43Z"
user="Valério Castro" uid="6237447" lat="-20.7618990" lon="-42.8619154"/>
```

Code 1. Example of node tag with metadata but with no attributes

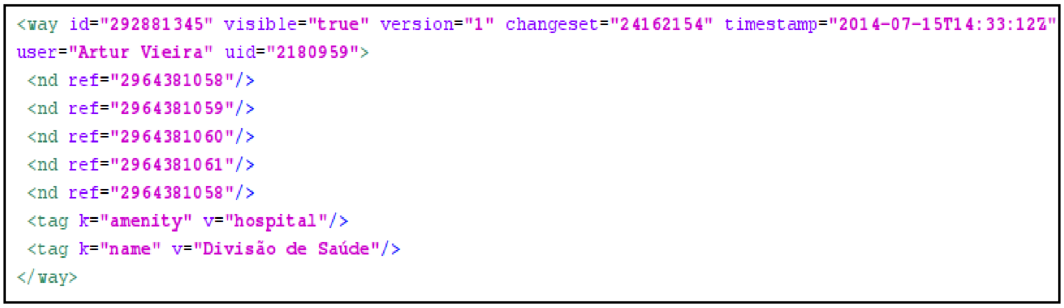
A *node* tag may contain attributes, as shows Code 2. In this example, besides having a pair of coordinates, the *node* tag has the attribute “turning_circle” of the type “highway” specified in the pair <key k, value v>. This element corresponds to the small roundabout near the public school.



```
<node id="3907519974" visible="true" version="1" changeset="36137846" timestamp="2015-12-24T02:14:55Z"
user="Artur Vieira" uid="2180959" lat="-20.7624231" lon="-42.8655997">
 <tag k="highway" v="turning_circle"/>
</node>
```

Code 2. Example of *node* tag with attributes

The *way* tag has a similar structure to the *node* tag previously exemplified. The difference lies in *nd* tags, which reference *node* tags without attributes. Code 3 illustrates the Health Division element and its attributes. The first *nd* tag references code number 2964381058, which is the id of the node described in Code 1. The last pair of coordinates of the element is the same as the first to delimit a polygon, which is why the first *nd* tag is the same as the last.



```
<way id="292881345" visible="true" version="1" changeset="24162154" timestamp="2014-07-15T14:33:12Z"
user="Artur Vieira" uid="2180959">
 <nd ref="2964381058"/>
 <nd ref="2964381059"/>
 <nd ref="2964381060"/>
 <nd ref="2964381061"/>
 <nd ref="2964381058"/>
 <tag k="amenity" v="hospital"/>
 <tag k="name" v="Divisão de Saúde"/>
</way>
```

Code 3. Example of *way* tag

Running the reverse engineering algorithm with the OSM-XML file regarding the pilot area generated the conceptual schema illustrated by Figures 5 and 6, in which all elements mapped in Figure 4 are present with their respective relationships, themes, and attributes. It is noteworthy that the modeling returned two classes of the type "University," which corresponds to the Federal University of Viçosa (UFV) in Figure 5 and to the Medicine Department building in Figure 6. That occurred because the exported area contains part of the border of UFV, observed in the upper right corner in Figure 4 by a change in the background color of the image.

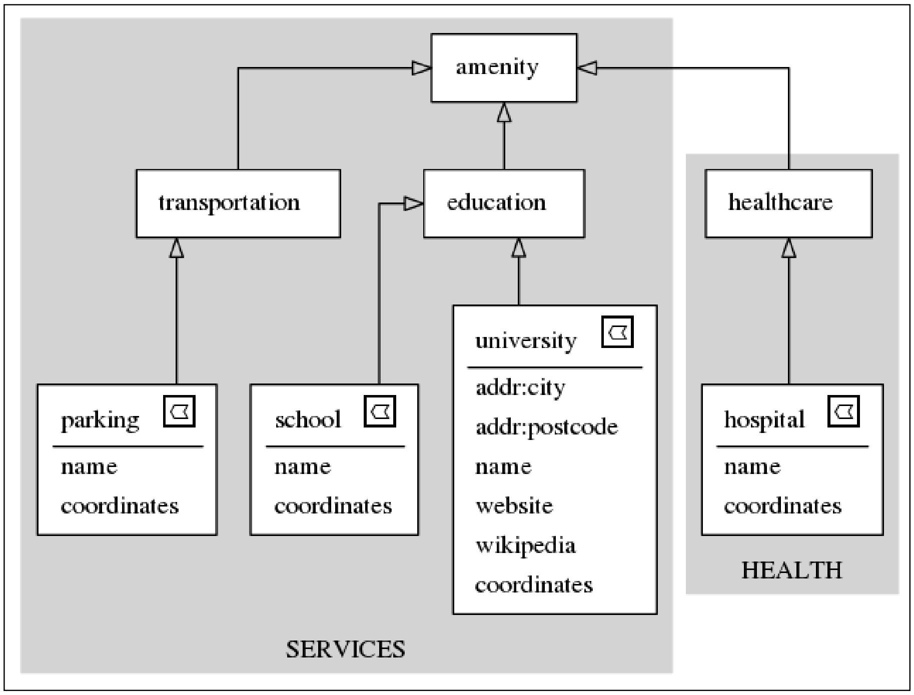


Figure 5. Conceptual schema generated from the pilot area of themes SERVICES and HEALTH

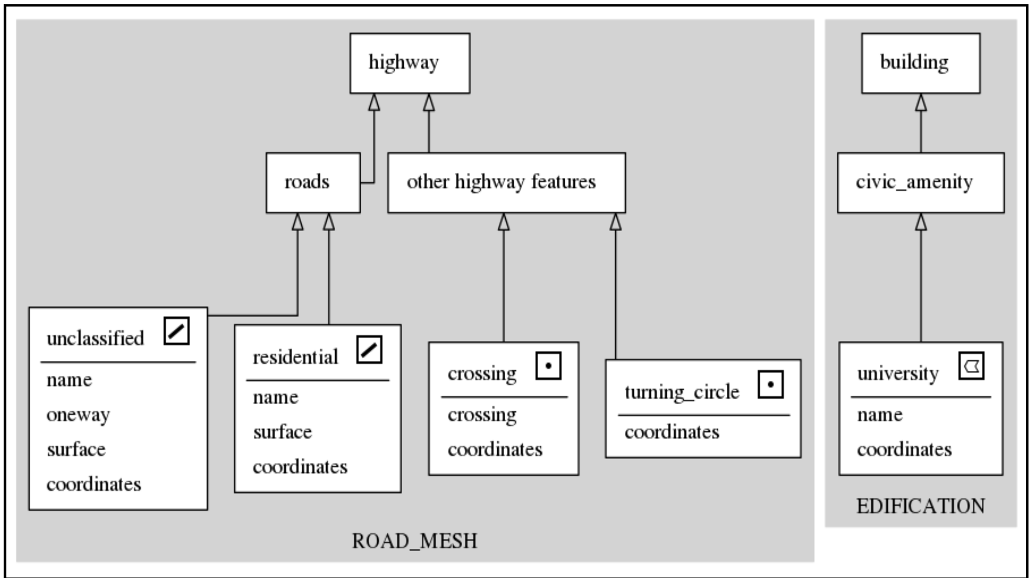
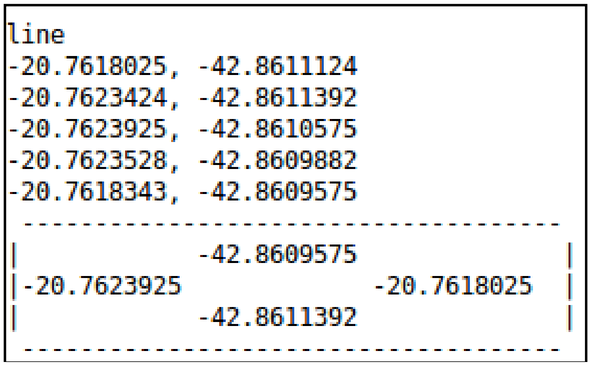


Figure 6. Conceptual schema generated from the pilot area of themes ROAD_MESH and EDIFICATION

Besides the conceptual schema, a file is generated with the elements that have no name. An excerpt of this file is illustrated by Figure 7, informing the object stereotype, its coordinates, and its involving rectangle to facilitate locating it on the map.



```
line
-20.7618025, -42.8611124
-20.7623424, -42.8611392
-20.7623925, -42.8610575
-20.7623528, -42.8609882
-20.7618343, -42.8609575
 -------------------------------------
|            -42.8609575             |
|-20.7623925             -20.7618025 |
|            -42.8611392             |
 -------------------------------------
```

Figure 7. Excerpt of the file generated with the unidentified tags

Finally, a second test was carried out with the aim of evaluating the effectiveness and efficiency of the algorithm. A load test was performed with all data available at the OSM of the city of Viçosa in Minas Gerais, with a total area of approximately 300km² and its demographic density of 241.20 inhabitants per square kilometer. The algorithm was running for 103 minutes on a personal computer, recognizing 73 entities belonging to the "ROAD MESH", "SERVICE" and "EDIFICATION" packages. The source code and the load test result are available through the link https://github.com/vinisperandio/OSM2Diagram.

6. Final Considerations and Future Works

This paper describes the process of automated generation of conceptual schemas from a file exported by the OpenStreetMap platform to create a geographic database for municipal administration based on volunteered geographic information (VGI).

The process is performed in two steps, the first consisting of creating a complete conceptual schema of the database available on the platform, i.e., contemplating all elements described in the specification of the OSM-XML file generated by this platform. The second step processes a reverse engineering algorithm that turns an XML file with data on a given selected area exported from the OpenStreetMap platform into a class diagram following the UML-GeoFrame model.

The results obtained in the pilot project allowed verifying that the work proposed presents a simple method to obtain volunteered data and an algorithm that, even in its initial version, proved capable of generating quality conceptual schemas. This is a good first step for municipalities with limited funds for cartographic services, because it allows to acquire knowledge of the features belonging to the municipality in an intuitive way, in order to facilitate administrative decision making. The method proposed is an initiative for researches that match free software and VGI systems in public administration. Silva et al (2018) show the potential of VGI such as a tool to support municipal managers in the decision-making process based on spatial analysis.

Future works include enhancing the performance of the algorithm presented and turning it into an application capable of yielding the script responsible for creating a NoSQL and relational geographic database in addition to the schemas.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

References

Almendros-Jiménez, J., and Becerra-Terón, A. (2018). Analyzing the Tagging Quality of the Spanish OpenStreetMap. *ISPRS International Journal of Geo-Information*, 7(8): 323.

Ballatore, A., and Mooney, P. (2015). Conceptualising the geographic world: the dimensions of negotiation in crowdsourced cartography. *International Journal of Geographical Information Science*, 29(12): 2310-2327.

Beautiful Soap (2018). Beautiful Soap 4.4.0 Documentation. https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Budhathoki, N. R. (2007). Reconceptualization of user is essential to expand the voluntary creation and supply of spatial information. In *Proceedings of Workshop on Volunteered Geographic Information*.

Elmasri, R. and Navathe, S. B. (2011). Sistema de Banco de Dados. Pearson AddisonWesley, 6th edition.

Goodchild, M. F. (2013). The quality of big (geo) data. *Dialogues in Human Geography*, 3(3): 280-284.

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

Graphviz – Graph Visualization Software, (2018). Welcome to Graphviz. http://www.graphviz.org/

Haklay, M., and Weber, P. (2008). Openstreetmap: User-generated street maps. *Ieee Pervas Comput*, 7(4): 12-18.

Haklay, M. (2010). How good is volunteered geographical information? A comparative study of OpenStreetMap and ordnance survey datasets. *Environment and Planning B: Planning and Design*, v. 37, n. 4, p. 682–703, doi:10.1068/b35097

Kitchin, R. (2014). The real-time city? Big data and smart urbanism. *GeoJournal*, 79(1): 1-14.

Davidovic, N., Mooney, P., Stoimenov, L., and Minghini, M. (2016). Tagging in volunteered geographic information: An analysis of tagging practices for cities and urban regions in OpenStreetMap. *ISPRS International Journal of Geo-Information*, 5(12): 232.

Lisboa-Filho, J. and Iochpe, C. (2008). Modeling with a UML profile. In Shashi Shekhar and Hui Xiong (Eds.) *Encyclopedia of GIS*, pages 691–700. Springer.

Lxml, (2018). Lxml – XML e HTML com python. https://lxml.de/

Miller, H. J. (2010). The data avalanche is here. Shouldn't we be digging? *Journal of Regional Science*, 50(1): 181-201.

Miranda, T. S., Lisboa Filho, J., Souza, W. D., Silva, O. C., and Davis Junior, C. A. (2011). Volunteered geographic information in the context of local spatial data infrastructures. In *Urban data management symposium* (UDMS), pages 123-138.

Mooney, P., and Corcoran, P. (2012). The annotation process in OpenStreetMap. *Transactions in GIS*, 16(4): 561-579.

Open Data Commons (2018). Open Data Commons Open Database License (ODbL). https://opendatacommons.org/licenses/odbl/

OpenStreetMap (2018). OpenStreetMap (OSM), https://www.openstreetmap.org , July.

Silva, L. P. et al. (2018). Bases cartográficas para municípios de pequeno porte geradas por informação geográfica voluntária. *Revista Brasileira de Cartografia* (no Prelo).

Pruvost, H., and Mooney, P. (2017). Exploring Data Model Relations in OpenStreetMap. *Future Internet*, 9(4): 70.

Quattrone, G., Mashhadi, A., Quercia, D., Smith-Clarke, C., and Capra, L. (2014). Modelling growth of urban crowd-sourced information. In *Proceedings of the 7th ACM international conference on Web search and data mining,* pages 563-572. ACM.

Wikimapia (2018). Wikimapia, http://wikimapia.org , July.

Simbology for Small Scale Reference Mapping: Automation and Sharing of Symbols

Flávia Silveira[1], Gabriele Silveira Camara[1], Silvana Philippi Camboim[1]

[1]Programa de Pós Graduação em Ciências Geodésicas – Universidade Federal do Paraná (UFPR)
Caixa Postal 19.001 – 81.531-980 – Curitiba – PR – Brazil

Flaviasilveira.silveira@gmail.com, camaragabriele@gmail.com, silvanacamboim@gmail.com

Abstract. *This work presents the stages to develop solutions to automatized the standardized simbolization of features represented in topographic maps at small scales in the Brazilian context, as well to perform the storage and sharing of symbols. The automation of the process was possible through the development of a plugin within the QGIS open source geoprocessing software environment, using Python programming language, and the storage and sharing of the symbols was done through the Github platform, which has version and free distribution and open source.*

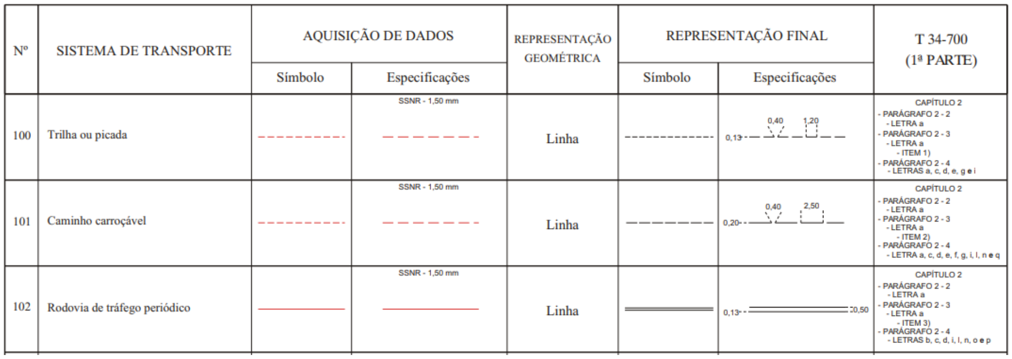
1. Introduction

According to the Brazilian Institute of Geography and Statistics (in Portuguese, Instituto Brasileiro de Geografia e Estatística - IBGE), the reference mapping aims to represent the territorial space in a systematic way, through series of topographic maps with the characteristic of being continuous, homogeneous and articulated. According to the Brazilian legislation [BRASIL 1967], it is executed by the Federal Government through the Coordination of Geographic Services (in Portuguese, Diretoria de Serviço Geográfico - DSG) and the IBGE for standard scales of 1: 1,000,000, 1: 250,000, 1: 100,000, 1: 50,000 and 1: 25,000.

The symbology is the set of definitions which determines the cartographic language used by the reference mapping for representing its features, considering the scale of the data representation and its level of generalization. These variables are related to the elements of color, level of detail and size of the cartographic symbols, in addition the specifications related to labels (font, size, and others). In the development of the reference mapping, it is necessary to be aware of the graphical presentation of the information contained in the map, because it is through the cartographic symbols that the communication of this information to the user happen [Keates 1973].

According to Sluter et al (2016), the standard cartographic symbology is an important feature for any reference mapping aiming to share information produced by different sources.

Until the present moment, in the Brazilian context, the symbology is only standardized for the cartographic representation in small scales (scales 1: 25,000 and smaller) defined by Technical Manual T34-700 - Cartographic Conventions (in Portuguese, Manual Técnico T34-700 - Convenções Cartográficas), shown in figure 1. The manual was developed by the DSG and has two parts, the first that establishes the norms for the representation of the natural and artificial features for small scales, and the second that specifies the characteristics of the standardized symbols for the use in topographic maps.



Nº	SISTEMA DE TRANSPORTE	AQUISIÇÃO DE DADOS		REPRESENTAÇÃO GEOMÉTRICA	REPRESENTAÇÃO FINAL		T 34-700 (1ª PARTE)
		Símbolo	Especificações		Símbolo	Especificações	
100	Trilha ou picada	- - - - - - - - - -	SSNR - 1,50 mm — — — — — — —	Linha	- - - - - - - - - - - - -	0,40 1,20 0,13	CAPÍTULO 2 - PARÁGRAFO 2 - 2 - LETRA a - PARÁGRAFO 2 - 3 - LETRA a - ITEM 1) - PARÁGRAFO 2 - 4 - LETRAS a, c, d, e, g e i
101	Caminho carroçável	- - - - - - - - - -	SSNR - 1,50 mm — — — — — — —	Linha	— — — — — — —	0,40 2,50 0,20	CAPÍTULO 2 - PARÁGRAFO 2 - 2 - LETRA a - PARÁGRAFO 2 - 3 - LETRA a - ITEM 2) - PARÁGRAFO 2 - 4 - LETRA a, c, d, e, f, g, i, l, n e q
102	Rodovia de tráfego periódico	———————	SSNR - 1,50 mm ———————	Linha	═══════	0,13 0,50	CAPÍTULO 2 - PARÁGRAFO 2 - 2 - LETRA a - PARÁGRAFO 2 - 3 - LETRA a - ITEM 3) - PARÁGRAFO 2 - 4 - LETRAS b, c, d, i, l, n, o e p

Figure 1. A part of Technical Manual T34-700 - Cartographic Conventions

The figure 2 shows a part of the São José dos Campos map symbolized according to the T34-700 manual.

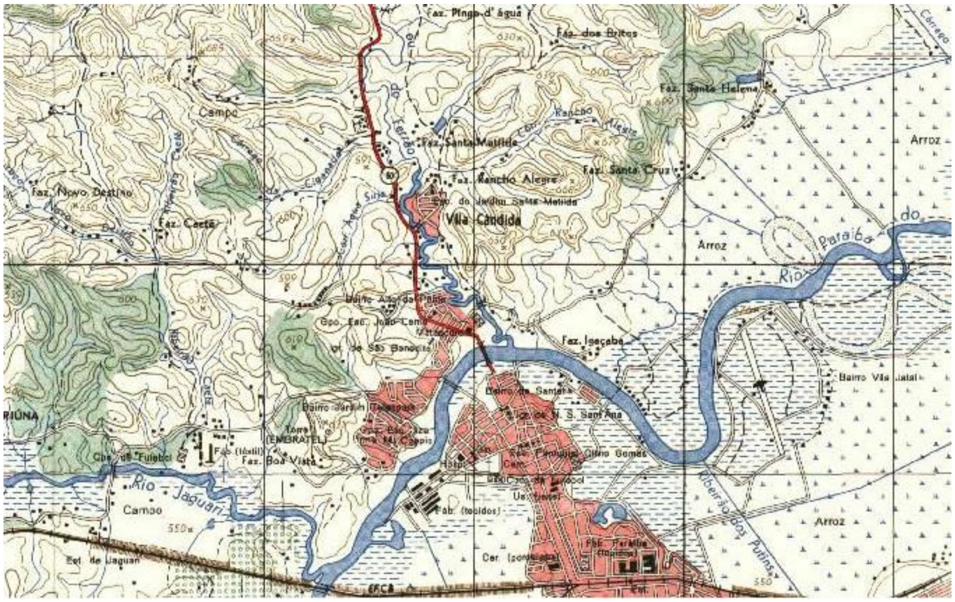


Figure 2. A part of the São José dos Campos map

The application of the symbology defined by the technical manual is a constant need for companies and institutions that develop cartographic materials supported on basemaps, especially digital products, such as interactive maps. Some symbols defined for the cartographic features representation are complex, therefore, the work of assigning the proposed symbology becomes a time-consuming work. Currently, there are only a few known tools to optimize or automate this process, and even fewer open source solutions.

The storage and sharing of symbols is also a latent demand, the symbology can be defined and stored in several formats and shared in different ways. The SLD (Styled

Layer Descriptor) established by the Open Geospatial Consortium (OGC) to represent the layers and their labels in WMS, by editing an Extensible Markup Language (XML) file. QML Layer Style File (QML) is also an important format because it is used by the open QGIS platform. So, the challenge is to store the symbols to facilitate their application and sharing in an open platform, then they can be accessed by the largest number of users, helping to ensure the standard in the symbolization of topographic maps.

The open source initiative, started in the 1990s [OSI 2018], provides a collaborative improvement in software, such as QGIS, and in open data platforms such as Open Street Map. However, when we talk about symbology, there is no open platform using user collaboration. Thus, the development of a collaborative system for symbology, with the user as an actor in the process of elaborating and sharing symbols, is important for management, improvement and propagation of the open data for cartographic representation.

This work aims to present the project and the steps for the development of a plugin for the open platform QGIS, which allows the automatic application of the symbology defined by the technical manual T34-700 for elements represented in small scales, as well as present the solution found to storage and sharing of digital symbols.

2. Symbology Description Formats and Sharing

The characteristics of the cartographic symbols can be stored in the tabular form as it happen in the Brazilian context, through the Technical Manual T34-700, as well as in the digital format, through symbol storage files, which are text files that describe the symbol properties such as graphic primitive, color, size, border thickness, and label characteristics. The most popular storage formats of symbology features and labels are: Styled Layer Descriptor (.sld), QGIS Layer Style File (.qml) and ESRI Layerfile (.lyr).

The OGC (Open Geospacial Consortium) created, in 1999, the Web Map Service (WMS), a web service that distributes matrixed tiles that are already symbolized using the SLD symbology pattern. The SLD format can be understood as a WMS profile, that together with the Symbology Encoding (SE) specification, allows the user to configure how the data will be displayed. The SE language is an XML language (Extensible Markup Language), which describes coding rules that defines symbol styles.

Currently, there is a worldwide trend to share geospatial data to facilitate the access to information of public interest, but the sharing of cartographic symbols in digital format is not yet widespread. In the international context, entities like Ordnance Survey, the British mapping agency, use platforms such as Github to share sld, qml and lyr symbology formats.

Github is the interface of the free distribution system and open source, Git, used to store and share codes [Github 2018]. In Github, the files are organized into repositories, defined by folders referring to projects created by users, which can be public or private, and repositories can belong to individual accounts or organizations.

The cartographic symbols are text files, so, they can also be shared via github to be accessed by the largest number of users and contribute to the standardization of the cartographic representation of the reference mapping.

In the Brazilian context, the only work that is known about symbol sharing is Fernandes (2012), where the author proposed a library of sld symbols, which would be available on the INDE portal, in order to facilitate the process of standardizing of topographic mapping symbology, helping the INDE to realize one of its fundamental components: data interoperability.

3. Materials and Method

3.1. Defining User Needs and Features

For the geospatial application development, the objectives were delimited based on the User Centered Design concepts, which, according to Abras *et al.* (2014) is defined as processes that end users influence the development of an application.

The plugin architecture was developed to show the structure of the application, According to Bass *et al.* (2003), the software architecture is understood as a generic standard for a project, explaining the solutions development to meet the needs of the project. The main plugin functionalities are connect to a database and offer options of symbols and scales to represent the features to the user's map.

3.2. Creation of Database and Symbology Files

The geospatial database was created on the PostgreSQL using the rules established by Technique Specification to Geospatial Vector Data (in Portuguese Especificação Técnica para Estrutura de Dados Geoespaciais Vetoriais - ET-EDGV), fundamental assumption for the plugin operation. The ET-EDGV provide a standard geographic vector data in small scales (1:25,000 or smaller) in order to guarantee the interoperability between data originated from differents sources. This Technique Specification was produced by National Spatial Data Infrastructure (in Portuguese Infraestrutura Nacional de Dados Espaciais - INDE), conceived in 2009 through Brazilian Federal Decree n° 6.666, with the goal to provide generation, storage, access, sharing, dissemination and use of geospatial data of federal, state, district and municipal origin.

In the database were entered the data referring to the classes Localities, Hydrography and Transport of the reference mapping of the Unit of the Federation of Rio de Janeiro in the scale of 1: 25,000.

Afterward, was developed on the QGIS the symbols standardized for the features of these three classes from the T34-700 Techcnical Manual. The symbols were saved in a QGIS Layer Style File format (.qml).

3.3. Plugin Implementation

The QGIS's plugin was developed using the programming language Python, the cartographic data was stored in PostGIS, which is a PostgreSQL extension, object-relational database system. The interface was designed in the QtDesigner software.

To develop the plugin it was also necessary to use the Plugin Builder complement, this QGIS's complement provides a basic model that helps to create plugins for this software. The file generated by Plugin Builder has Python extension and allows editing, so, the plugin is able to accomplish the tasks programmed by the user [GEOAPT LLC 2017].

The libraries used for the plugin development were:

- PyQGIS: Python library to develop QGIS's applications. This library allows to create custom plugins based on the QGIS Application Programming Interface (API) [QGIS 2017].
- Psycopg2: library that allows the communication between Python and PostgreSQL. This library enables the execution of SQL commands in the Python language, as well as allowing access to many resources offered by PostgreSQL [PSYCOPG 2017].

The experimental data used to the plugin development were the classes Localities, Hydrography and Transport referring to the Continuous Rio de Janeiro Cartographic Base, in 1: 25,000 scale. This data are available by the Brazilian Institute of Geography and Statistics (IBGE) at its electronic address (www.ibge.gov.br).

3.4. Storage and Sharing via GitHub Implementation

Initially, in order to centralize the information and projects related to the symbology study, an email account was created (opencartographicstyles@gmail.com), this e-mail was also used to register the github account.

Associated with the github user was created an organization called Open Cartographic Styles, initially, containing a repository referring to the symbols, mappingGrandeEscala, and another related to the symbology plugins, pluginsSimbologia.

In the mapeamentoGrandeEscala repository, the files format chosen to the sharing were SLD (Styled Layer Descriptor) and QML (QGIS Layer Style File), compatible with the Geoserver and QGIS software, respectively. To ensure the files interoperability and standardization, they were organized according to the classes established by ET-EDGV and each file was named according to the nomenclature of the category.

4. Results

4.1. System Architecture

The architecture developed for the application (Figure 3), presents the system operation, in order to understand that the shapefile format files stored in the database are loaded through the interaction between the QGIS and the developed plugin. The predefined symbols are used by the plugin to be applied in the selected layers, and those layers are represented with the desired symbology.

The predefined symbols are stored on github, where the user can download the files to their computer and allocate them in a folder, which can be referenced in the plugin and than applied to the layers stored on the database or in another projects in the QGIS.

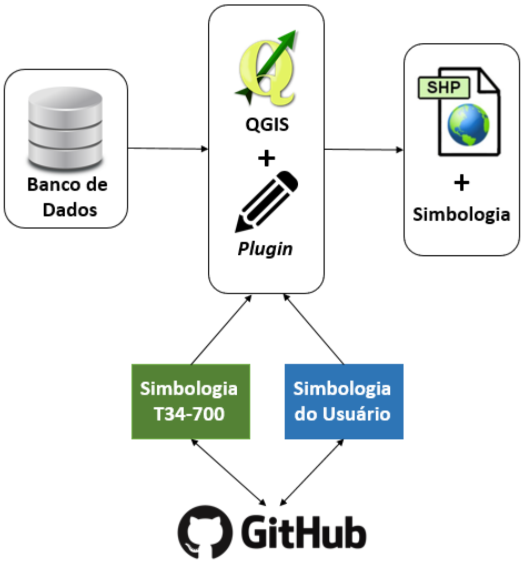


Figure 3. System architecture

4.2. Interface and Plugin Functions

The developed plugin interface (Figure 4), presents the application resources, listed and described below.

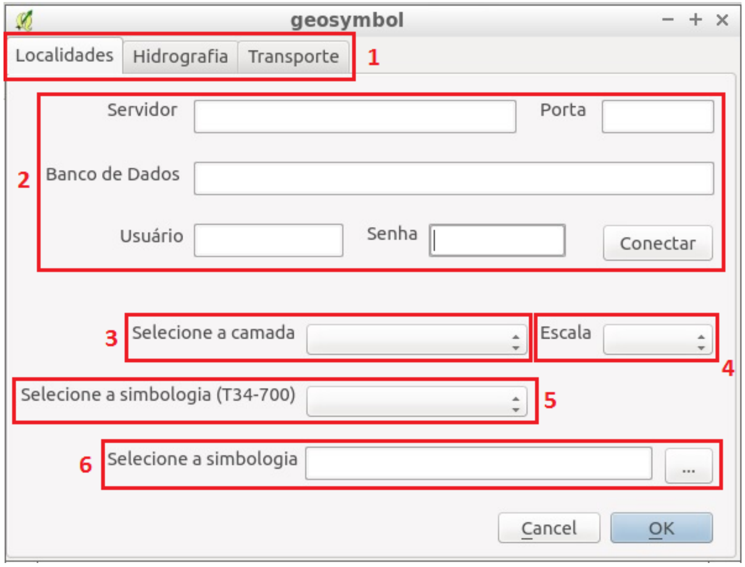


Figure 4. Plugin interface

1. To develop the plugin, three standardized classes by Technical Manual T34-700 were used (localities, hydrography and transport). Each class is represented by a tab in the plugin interface, and the user can choose one of then to work.
2. After choosing the class, it is necessary connect to the database, informing the server, port, database name, user name and password.

3. When the database is already connected, the user can chose on a combo box the layer to be symbolized. Only layers belonging to the selected tab group can be chosen in the combo box.
4. It is possible to select on a combo box the data display scale. The possibilities are the scales defined for the Brazilian systematic mapping: 1: 25,000, 1: 50,000, 1: 100,000, 1: 250,000 and 1: 1,000,000.
5. Then, it is possible to select the symbology to be applied to the previously selected layer through a combo box that presents the standard symbologies defined by the T34-700. The symbologies presented in the combobox are related to the selected layer graphic primitive, it means, if the layer selected in step 3 is represented by the point graphical primitive, only symbologies for point features can be selected in step 5, the same logic is applied to the line and polygon graphic primitives.
6. Another way to select the desired symbology is through a "button box", with it, the user is able to select ".qml" symbology file of his computer, and it will be applied to the previously selected layer.

The figure 5 shows the symbology proposed by the Technical Manual T34-700, applied automatically in scale 1:25,000 to the Hydrography, Localities and Transportation classes to a region of the state of Rio de Janeiro by the plugin developed.

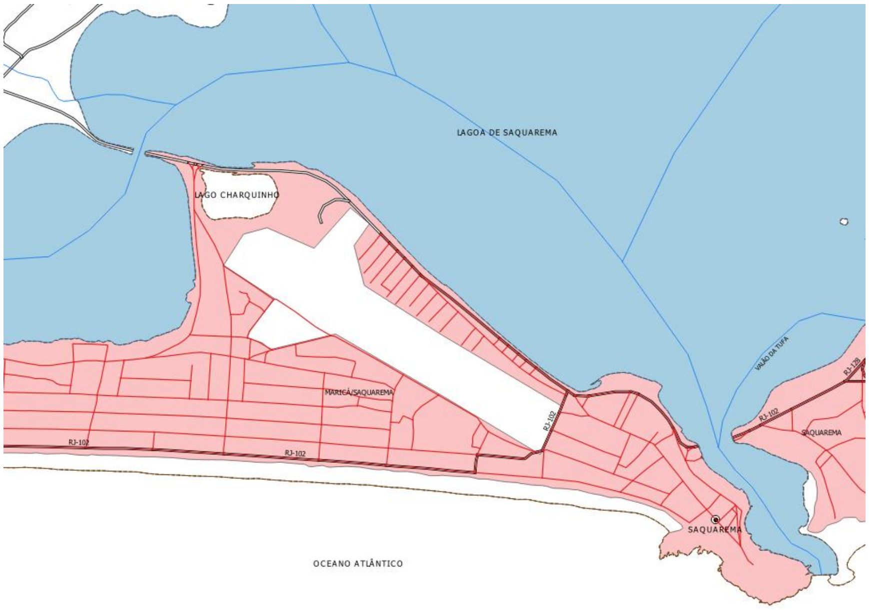
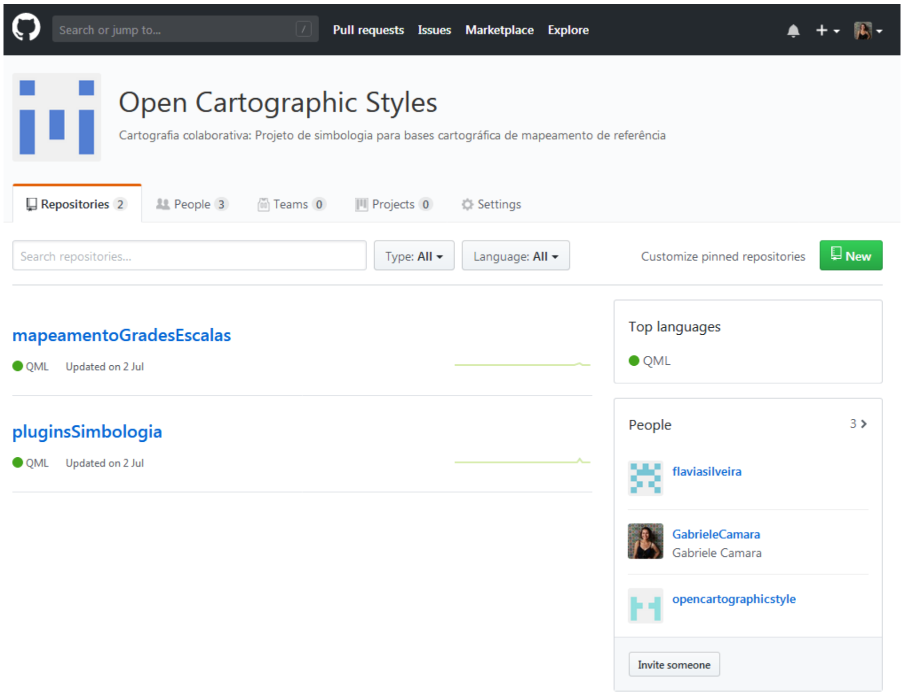


Figure 5. A part of the state of Rio de Janeiro symbolized according Technical Manual T34-700 by the plugin developed.

The benefits of the developed plugin are ensure a correct application of the symbology for small scales, data interoperability between different sources of cartographic datas and provide to a common user a tool to apply a appropriate scale according to the symbology.

4.3. Github

The initial page of Open Cartographic Styles github organization, made to storage and sharing the symbology of reference mapping and other application related (Figure 6), presents the repository related to the symbols, mapeamentoGrandesEscalas, and another associated to the plugins, pluginsSimbologia. The plugin's repository contains applications developed in python so far elaborated by symbology research group, intended for use in QGIS.

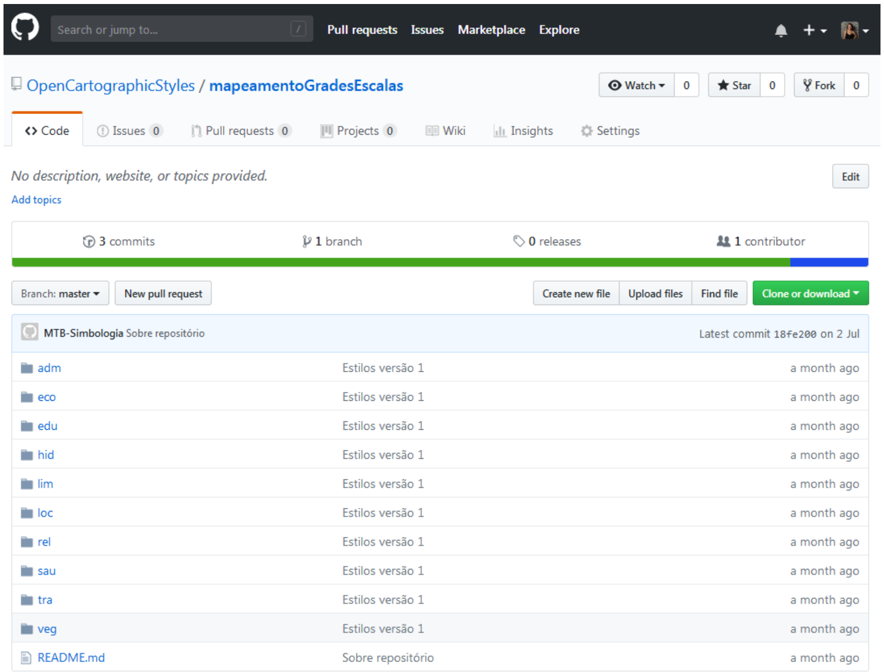


Open Cartographic Styles

Cartografia colaborativa: Projeto de simbologia para bases cartográfica de mapeamento de referência

Figure 6. Initial page of github

Within the symbology repository (Figure 7) 10 classes of symbols were stored and shared. Seven of those classes are: Administration (adm), Education (edu), Economic Structure (eco), Limits (lim), (rel), Health Structure (sau) and Vegetation (veg). The other 3 classes, included in the plugin described in this article, are: Localities (loc), Hydrography (hid) and Transportation System (tra), according to ET-EDGV.



No description, website, or topics provided.

3 commits · 1 branch · 0 releases · 1 contributor

MTB-Simbologia Sobre repositório · Latest commit 18fe200 on 2 Jul

adm	Estilos versão 1	a month ago
eco	Estilos versão 1	a month ago
edu	Estilos versão 1	a month ago
hid	Estilos versão 1	a month ago
lim	Estilos versão 1	a month ago
loc	Estilos versão 1	a month ago
rel	Estilos versão 1	a month ago
sau	Estilos versão 1	a month ago
tra	Estilos versão 1	a month ago
veg	Estilos versão 1	a month ago
README.md	Sobre repositório	a month ago

Figure 7. The repository mapeamentoGrandesEscalas related to the symbols

According to the nomenclature of the files as the ET-EDGV categories, there is a pattern, beginning with the name of the class, followed by the name of the feature, and finally, a code according to the graphic primitive.

5. Conclusion

The standard symbology is indispensable for any mapping, especially for the reference mapping, because the symbols quality ensures the cartographic information understanding, and consequently its recognition and decoding, an essential process for cartographic communication.

However, symbology standardization, in the case of Manual T34-700 for small-scale representations, it is not in accordance with the Brazilian standard ET-EDGV established by INDE. Therefore, the interoperability of data from the reference mapping is impaired, making difficult using and sharing those symbols. For this reason, it is necessary to update these symbology standards according to ET-EDGV, for small and large scales, ensuring the sharing and interoperability of data from different source.

The use of the free software QGIS and its functionalities allowed the creation of the plugin and the possibility to distribute and share it openly in the future, helping the symbology implementation process. Likewise, the use of Github ensures that the symbols storage and sharing are also open and collaborative processes. Either solutions assume the user as a collaborator, and the central figure of the process, since the user is consumer and symbology producer.

The issues of automatization of the basemaps symbolization process, storage and sharing of cartographic symbols are not yet widely discussed in the Brazilian context. Future studies on these subjects can benefit and take the present work as a basis.

In the Free Geospatial Laboratory of the Federal University of Paraná, researches are also being developed related to the symbolization of reference mapping in large scales and symbology through vector tiles.

References

Abras, C. Maloney-Krichmar, D. Preece, J. (2014) “User-Centered Design”, In: Encyclopedia of Human-Computer Interaction.

Bass, L. Clements, P. Kazman, R. (2003), “Software architecture in practice”, Addison Wesley.

Brasil. (1967), “Decreto Federal n° 243”, http://www.planalto.gov.br/civil_03/decreto-lei/1965-1988/Del0243.html, July.

Brasil. (1998), “Manual Técnico T34-700”, http://www.geoportal.eb.mil.br/index.php/inde2?id=141, July.

Brasil. (2016), “ET-EDGV - Especificações Técnicas Para Estruturação de Dados Geoespaciais Digitais”, http://www.geoportal.eb.mil.br/images/PDF/EDGV_DEFESA_F_Ter_2a_Edicao_2016_Aprovada_Publicada_BE_7_16.pdf? July.

Fernandes, W. S. (2012), “Criação de uma biblioteca de símbolos cartográficos utilizando os padrões *Symbology Encoding* (SE) e *Styled Layer Descriptor* (SLD) do OGC”, In: Encyclopedia of Human-Computer Interaction.

Geoapt LLC. (2017),

Github. (2018), “Git Cheat Sheet”, https://education.github.com/git-cheat-sheet-education.pdf, July.

Github. (2018), “Github Help”, https://help.github.com/categories/about-github/, July.

Instituto Brasileiro de Geografia e Estatística. (2018), “Bases Cartográficas Contínuas.”, https://www.ibge.gov.br/geociencias-novoportal/cartas-e-mapas/bases-cartograficas-continuas/15807-estados.html?edicao=16037, July.

Keates, J. S. (1989), “Cartographic design and production”, Longman Group.

Lima, L. A. (2005), “Eclipse tools - Ferramenta para auxílio à composição dinâmica de software”, Campina Grande.

Open Geospatial Consortium. (2018), “OGC E-Learning”, http://cite.opengeospatial.org/pub/cite/files/edu/index.html, July.

Open Source Initiative. (2018), https://opensource.org/, July.

Psycopg. (2018), http://initd.org/, July.

QGIS. (2018), “Passo-a-passo para desenvolvedor PyQGIS”, http://docs.qgis.org/2.14/pt_BR/docs/pyqgis_developer_cookbook, July.

Sluter, C. R. Elzzaker, C.P.J.M. Ivanova, I. (2016) “Requirements Elicitation for Geoinformation Solutions”, In: The Cartographic Journal.

SP-TWDTW: A New Parallel Algorithm for Spatio-Temporal Analysis of Remote Sensing Images

Sávio S. T. de Oliveira[1], Luiz M. L. Pascoal[1], Laerte Ferreira[2], Marcelo de C. Cardoso[1], Elivelton Bueno[1], Vagner J. S. Rodrigues[1], Wellington S. Martins[1]

[1]Instituto de Informática - Universidade Federal de Goiás (UFG)
Alameda Palmeiras, Quadra D, Câmpus Samambaia
131 - CEP 74001-970 - Goiânia - GO - Brasil

[2]LAPIG - CAMPUS II Samambaia - Cx. POSTAL 131
CEP: 74001-970 - Goiânia - GO - Brasil

{savioteles,luizmlpascoal,lapig.ufg}@gmail.com,
{marcelo.cardoso,elivelton.bueno,vagner}@gogeo.io, wellington@inf.ufg.br

Abstract. *In the class of computationally complex problems, the time series analysis is one of those that has high demand for computational power. The Time-Weighted Dynamic Time Warping (TWDTW) algorithm stands out as one of the best solution found in the literature in this field, but its time complexity of $O(n^2)$ makes it unfeasible for large data sets. To overcome this limitation, this work proposes a parallel algorithm, named SP-TWDTW (Spatial Parallel TWDTW), that allows the analysis of large scale time series using Manycore architectures. The SP-TWDTW considers the temporal axis and the spatial autocorrelation to determine the land use mapping in a given region. The results show that the SP-TWDTW algorithm is a promising solution with response time up to 11 times lower.*

1. Introduction

The Earth's surface is changing at an unprecedented rate. Forest ecosystems diminish at alarming speed, urban and agricultural areas expand into the surrounding natural space. Since then, time series analysis of remote sensing images has become indispensable to identify these changes. It has attracted great interest in the world scenario, becoming an important resource in several applications [Kuenzer et al. 2015].

Among the group of time series analysis algorithms, the *Time-Weighted Dynamic Time Warping (TWDTW)* is considered one of the best algorithms for searching all possible occurrences of patterns in time series of remote sensing images [Maus et al. 2016]. The TWDTW algorithm is an adaptation of the *Dynamic Time Warping (DTW)* algorithm [Sakoe 1971], a well-known method for time series analysis. The DTW compares a pattern of a known event with an unknown time series.

Unlike the DTW, the TWDTW algorithm is sensitive to the seasonal changes of the natural and cultivated vegetation types, which is extremely important in remote sensing field [Maus et al. 2016]. However, the TWDTW analyzes each pixel individually, not taking into account the neighboring pixels and, therefore, make assumptions (e.g. independent, identical distributions) which violate Tobler's first law of Geography: everything is related to everything else but nearby things are more related than distant things (i.e., independent distributions). Techniques which ignore spatial autocorrelation typically perform poorly in the presence of spatial data [Vatsavai 2008].

This paper proposes a new highly parallel solution, named *Spatial Parallel TWDTW (SP-TWDTW)*, which takes into account the temporal axis and the spatial autocorrelation to determine the land use mapping in a given region. The TWDTW algorithm has a high computational cost, with time complexity of $O(n^2)$, which makes its use unfeasible for large data sets [Xiao et al. 2013] and makes it virtually impossible to analyze the spatial axis due to the computational cost. The SP-TWDTW explore the cores available in Manycore architectures and automatically manages the usage of the memory spaces to allows the processing of large amounts of data. The main contributions of this work are listed below:

- A new parallel algorithm for spatio-temporal analysis of remote sensing images.
- The inclusion of the spatial dimension in the classification of time series.
- An automatic system to manage memory usage between CPU and GPU spaces.

This paper is organized as follows. Section 2 discusses the processing of time series analysis for remote sensing images. Section 3 describes the TWDTW algorithm used as the basis for this work. The new algorithm proposed in this paper (SP-TWDTW) is presented in Section 4. Section 5 validates the SP-TWDTW algorithm experimentally and discusses the performance of the algorithm. Finally, Section 6 presents some conclusions and future work.

2. Analysis of Time Series for Remote Sensing Images

Time series analysis comprises methods for extracting important statistics and characteristics from time series data. The DTW, which is one of the most well-known methods in this field, allows the alignment between two time series, even if they have different lengths or they are not aligned on the time axis. Given the time series A and B, the distance between them are computed as

$$DTW(A,B) = min\sqrt{\sum_{k=1}^{K} w_k} \tag{1}$$

where $w_k = (i, j)$ represents the association between the i-th and the j-th observations, say a_i and b_j, respectively time series A and B, which are equivalents according to the Euclidean Distance, $d(i, j) = \sqrt{(a_i - b_j)^2}$. The sequence $w_1, w_2, ..., w_k$ represents the association between observation pairs of the two given time series, denoted by the adjustment path. Equation 1 is subject to the following conditions: i) The first observation of one series must match the first observation of the other series, $w_1 = (1, 1)$, and the last observation of one series must match the last observation of the other, $w_k = (m, n)$; ii) Given $w_k = (i, j)$ and $w_{k+1} = (i', j')$ then $i' - i \leq 1$ and $j' - j \leq 1$; iii) Given $w_k = (i, j)$ and $w_{k+1} = (i', j')$ then $i' - i > 0$ and $j' - j \leq 0$. The DTW algorithm is not recommended for time series analysis of remote sensing images because it disregards the temporal range when finding the best alignment between two time series classification [Maus et al. 2016].

Some previous work like [Petitjean et al. 2012, Petitjean and Weber 2014, Maus et al. 2016] proposed non parallel methods using DTW to analyze time series of satellite images, while [Petitjean et al. 2012, Petitjean and Weber 2014] used a maximum time delay to avoid time distortions based on the date of the satellite images. On the other hand, [Verbesselt et al. 2010, Jamali et al. 2015] support parallel execution in Multicore architectures using the library *foreach*[1] from R programming language. The TWDTW method [Maus et al. 2016] seeks to find all possible occurrences of a particular pattern within a time series, introducing time constraints, and has been prominent in the accuracy of identifying land cover use.

The rapid growth of multicore/manycore processors has attracted the attention of many researchers. For example, the works [Xiao et al. 2013, João Jr et al. 2017, Zhu et al. 2018] presented parallel solutions to analyze time series using manycore architectures (GPUs). However, they do not address the problems identified by [Maus et al. 2016] in the remote sensing field.

It is essential to make users aware of both the spatial and temporal dimensions in a Geographic Information System (GIS), since they may reveal implicit relationships which match the reality of the analyzed data [de Oliveira and de Souza Baptista 2012]. Several techniques of spatial time analysis have been previously proposed [Cressie and Wikle 2015]. Some methods process each image independently and compare the results for different time instances [Gómez et al. 2011, Lu et al. 2016]. The technique presented in [Costa et al. 2017] builds time series of each pixel and process them independently. At the end, the algorithm chooses some seed pixels in the image and calculates the distance between the time series of these seeds to their neighbors using the DTW method, grouping similar neighbors.

Some papers performs the time series analysis through spatial interpolation [Li and Heap 2014], which is the process of using points with known values to estimate values of other unknown points. Several methods are used, such as:

1. **Nearest Neighbor:** the value of each point is determined by the nearest points [Mitas and Mitasova 1999];
2. **IDW:** gives greater weights to points close to the prediction location [Shepard 1968];
3. **Kriging:** assumes that the distance or direction between sample points reflects a spatial correlation that can be used to explain variation in the surface [Stein 2012].

3. Time-Weighted Dynamic Time Warping (TWDTW)

The TWDTW [Maus et al. 2016] is a variation of the DTW algorithm that is sensitive to seasonal changes of natural and cultivated vegetation types. It considers inter-anual climatic and seasonal variability. The TWDTW method computes the cost matrix $\Psi_{n,m}$ given the pattern $U = (u_1, ..., u_n)$ and time series $V = (v_1, ..., v_m)$. The elements $\psi_{i,j}$ of $\Psi_{n,m}$ are computed by adding the temporal cost ω, becoming $\psi_{i,j}$ = $|u_i - v_j| + \omega_{i,j}$, which $u_i \in U \; \forall \; i = 1, ..., n$ and $v_j \in V \; \forall \; j = 1, ..., m$. To calculate the time cost, the logistic model is used with a midpoint β and a bias α presented in Equation 2.

$$\omega_{i,j} = \frac{1}{1 + e^{-\alpha(g(t_i,t_j)-\beta)}}, \tag{2}$$

[1] https://cran.r-project.org/web/packages/foreach/index.html

in which $g(t_i, t_j)$ is the elapsed time in days between dates t_i for the patterns U and t_j in the time series V. From the cost matrix Ψ an accumulated cost matrix is calculated, named D by using a recursive sum of the minimum distances, as shown in equation 3

$$d_{i,j} = \psi_{i,j} + min\{d_{i-1,j}, d_{i-1,j-1}, d_{i,j-1}\}, \tag{3}$$

which is subject to the following conditions:

$$d_{ij} = \begin{cases} \psi_{i,j} & i = 1, j = 1 \\ \sum_{k=1}^{i} \psi_{k,j} & 1 < i \leq n, j = 1 \\ \sum_{k=1}^{j} \psi_{i,k} & i = 1, 1 < j \leq m \end{cases} \tag{4}$$

The kth lowest cost path in D produces an alignment between the pattern and a subsequence of V with associated distance δ_k, in which a_k is the first element and b_k the last element of k. Each minimum point in the last row of the cost matrix is accumulated, i.e. $d_{n,j}\ \forall\ j = 1, ..., m$, produces an alignment, with $b_k = argmin_k(d_{n,j}), k = 1, ..., K$ and $\delta_k = d_{n,b_k}$, in which K is the minimum number of points in the last row of D.

A reverse algorithm, equation 5, maps the path $P_k = (p_1, ..., p_L)$ along the kth "valley" to the lowest cost in D. The algorithm starts in $p_{l=L} = (i = n, j = b_k)$ and ends with $i = 1$, i.e. $p_{l=1} = (i = 1, j = a_k)$, in which L denotes the last point of alignment. The path P_k contains the elements that have been matched between the series.

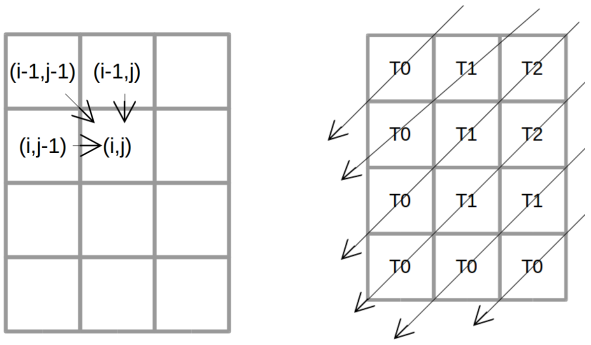
$$p_{l-1} = \begin{cases} (i, a_k = j) & \text{se } i = 1 \\ (i-1, j) & \text{se } j = 1 \\ argmin(d_{i-1,j}, d_{i-1,j-1}, d_{i,j-1}) & \text{otherwise} \end{cases} \tag{5}$$

The land cover mapping with TWDTW is performed in two steps. In the first step, the DTW algorithm is applied to each pattern in $U \in Q$ and each time serie $V \in S$. This step provides information on how many patterns match with time series intervals. In the second step, the best matching pattern found by the DTW algorithm is used for land cover mapping.

4. Spatial-Time Series Analysis of Remote Sensing Images with a Parallel Architecture

The TWDTW is a pattern-matching algorithm based on dynamic programming with time complexity $O(n^2)$. This section presents the solution proposed in this work for the parallel processing of spatial time series analysis. This solution, named SP-TWDTW (Spatial Parallel TWDTW), parallelizes the TWDTW analyzing the temporal axis of the time series, as well as the spatial axis of the neighboring pixels to classify each time series.

The accumulated cost matrix D is computed from the cost matrix Ψ using the recursive sum of the minimum distances, as shown in equation 3. The construction of D can not be trivially paralleled since the computation of each element (i, j) of the matrix depends on the previously elements $(i-1, j)$, $(i, j-1)$ and $(i-1, j-1)$. This dependency can be seen in Figure 1(a). The idea behind the SP-TWDTW algorithm is presented in Figure 1(b). Each diagonal is computed in parallel, with each thread being responsible for a diagonal cell. Since the elements are not dependent on each other within the diagonal, the calculation of the accumulated cost does not lead to an inconsistent matrix. The details of the SP-TWDTW matrix computation are presented in Algorithm 1 in Section 4.1.



(a) The computation of each element in D depends on the values of previous elements.

(b) SP-TWDTW: Parallel processing of D

Figure 1. Computation of the accumulated cost matrix D

4.1. Spatial Parallel Time-Weighted Dynamic Time Warping (SP-TWDTW)

Algorithm 1 describes the SP-TWDTW, which has as input the set of patterns Q and the set of time series S and calculates the final alignment cost matrix between each $U \in Q$ and $V \in S$. Since Q and S can be larger than available memory, the input of this algorithm admits that these sets are stored on the disk. The SP-TWDTW manages the loading of blocks of Q and S to CPU memory and subsequently to the GPU memory, so that it does not exceed the limits of them. The SP-TWDTW also receives as input the maximum size of the sets Q and S that fill in the GPU memory (bQ and bS respectively) and CPU memory (max_bQ and max_bS respectively).

The algorithm starts in the lines 2 and 3 by reading the blocks of the patterns into the $queueQ$ and blocks of the time series into the $queueS$. This work is performed by a CPU thread that manages the input queue size and the CPU memory available size. So, it is not necessary to wait to finish this step to start loading the blocks into the GPU global memory. Between lines 4 and 28, the Algorithm 1 loads the blocks into the GPU memory and computes the matrix D. This is done while there are blocks to be processed.

From line 5 until line 8, bQ patterns U and bS time series V are loaded into the GPU global memory, joining the patterns in a single $bigQ$ sequence and the time series in a single $bigS$ sequence. This union allows the GPU to perform block processing using all its computational power. In line 9, the cost matrix is constructed from $bigQ$ and $bigS$, with each GPU thread being responsible for computing an element of the array.

The calculation of matrix D is performed following the idea presented in Figure 1(b), where each thread computes the cost of each element in the diagonal. Since each element of the diagonal depends only on the two diagonals, they can be calculated independently. Given that $bigQ$ and $bigS$ have several patterns U and time series V, $bQ * bS$ threads are launched in the GPU, with each block of threads being responsible for calculating the cumulative cost between a pattern U and a time series V. Each block in the GPU has $min(sizeU, sizeV)$ threads that is the maximum size of a diagonal when comparing U and V, so that each thread performs the calculation of one diagonal element.

From line 14 to 20, the costs of the first $sizeU$ elements from the upper diagonals above the secondary diagonal are computed. In a square matrix, these first $sizeU$ diagonals are the upper triangular matrix. The Algorithm 1 assumes that the index starts at position 0. To identify each diagonal, the row index in the upper matrix is computed as $si - tid$ and the column index is determined by the thread id. The matrix D is updated for each diagonal element using the function $update\_accumulated\_cost\_matrix$.

The elements of the next $sizeV - 1$ diagonals are computed between lines 21 and 26. In a square matrix, for example, these $sizeV - 1$ diagonals are the lower triangular matrix. To identify each diagonal, the row index in the upper matrix is calculated as $sizeU - tid - 1$ and the column index is set to $sizeV - sj - tid - 1$. The matrix D is then updated for each element of the diagonal.

The function $update\_accumulated\_cost\_matrix$ updates each element $D_{i,j}$ of the accumulated cost matrix D. The calculation of $D_{i,j}$ follows the equation 3, which is the smallest value between $D_{i-1,j}$, $D_{i-1,j-1}$ and $D_{i,j-1}$ plus the cost $\Psi_{i,j}$. The index in the matrices Ψ and D related to i, j of the matrices U and V must be computed, since the matrices Ψ and D are sent as vectors for the GPU and the series U and V in blocks of size bQ and bS respectively. So, it is necessary to find the pair U and V inside D and Ψ and then find the correct position in the matrix D related to U and V. In the GPU $bQ * bS$ blocks of threads are launched with bQ on the x-axis and bS on the y-axis. So, each block of threads handles one block in Ψ and D.

The matrix D is computed following equation 6 and the global index of Ψ and D follows the equation 7, in which $(i + blockIdx.x * sizeU) * sizeV * gridDim.y$ finds the correct line in Ψ and D, wherein $blockIdx.x$ the id of the block of threads in the x-axis and $gridDim.y$ the number of blocks in the y-axis. The part of the equation $(j + blockIdx.y * sizeV)$ finds within of the matrix line the correct position of the element (i, j), being $blockIdx.y$ the id of the block of threads in the y-axis.

$$D[index_{i,j}] = min(D[index_{i-1,j}], D[index_{i-1,j-1}], D[index_{i,j-1}) + \Psi[index_{i,j}] \quad (6)$$

$$index = (i + blockIdx.x * sizeU) * sizeV * gridDim.y + (j + blockIdx.y * sizeV) \quad (7)$$

The algorithm computes the final alignment between each U and V following the equation 5 in the function $compute\_dtw\_path$ between lines 29 and 31. This cost is stored in the matrix R, which each row represents a pattern U and each column a time series V to be sorted.

Between lines 34 to 39, the SP-TWDTW analyzes the spatial axis, performing a spatial interpolation on line 36 for each alignment between U and V, that is, each element of R. We already have the cost of the alignment between U and V stored in R_{ij}, but in the line 36, the method $compute\_spatial\_interpolation(R_{i,j})$ searches for the cost of other spatially close time series to V related to the pattern U, estimate the value of R_{ij} and stores it in the matrix I at I_{ij}. There are several factors that impact on the quality of this spatial prediction [Li and Heap 2014] and, therefore, a mechanism has been made where it is possible to give weights for temporal and spatial axis in line 37 of the algorithm. This weighted cost is stored in the output R matrix.

Algorithm 1: sp-twdtw(Q, bQ, max_bQ, S, bS, max_bS, $temporal\_weight$, $spatial\_weight$)

```
Input: $Q$: set of patterns $U$
$bQ$: number of patterns $U$ in the GPU memory
$max\_bQ$: max patterns $U$ in the CPU memory
$S$: set of time series $V$
$bS$: number of time series $V$ in the GPU memory
$max\_bS$: max time series $V$ in the CPU memory
$temporal\_weight$: temporal axis weight
$spatial\_weight$: spatial axis weight
Output: $R$: final alignment cost matrix between each $U$ and $V$
while There are patterns $U$ and time series $V$ on disk do
    $queueQ \leftarrow$ load $max\_bQ$ patterns $U$ to CPU memory
    $queueS \leftarrow$ load $max\_bS$ timeseries $V$ to CPU memory
    while There are patterns in $queueQ$ and time series in $queueS$ do
        $gpu\_queueQ \leftarrow$ load $bQ$ patterns $U$ to GPU global memory
        $gpu\_queueS \leftarrow$ load $bS$ time series $V$ to GPU global memory
        $bigQ \leftarrow$ merge all patterns $U$ in $gpu\_queueQ$
        $bigS \leftarrow$ merge all time series $V$ in $gpu\_queueS$
        $\Psi \leftarrow$ compute the cost matrix between $bigQ$ and $bigS$
        $sizeU \leftarrow$ compute the pattern size of each $U$ in $bigQ$
        $sizeV \leftarrow$ compute the time series size of each $V$ in $bigS$
        $tid \leftarrow$ thread id
        if $tid < sizeU$ then
            for $si \leftarrow 0$ to $sizeU$ - 1 do
                if $tid \leq min(si, sizeV - 1)$ then
                    $i \leftarrow si - tid$
                    $j \leftarrow tid$
                    $update\_accumulated\_cost\_matrix(\Psi, D, i, j, sizeU, sizeV)$
                end
            end
            for $sj \leftarrow sizeV - 2$ to $0$ do
                if $tid \leq min(sj, sizeU - 1)$ then
                    $i \leftarrow sizeU - tid - 1$
                    $j \leftarrow sizeV - sj - tid - 1$
                    $update\_accumulated\_cost\_matrix(\Psi, D, i, j, sizeU, sizeV)$
                end
            end
        end
        for Each $U$ in $gpu\_queueQ$ and $V$ in $gpu\_queueS$ do
            $compute\_dtw\_path(R, D, sizeU, sizeV)$
        end
    end
end
for Each line $i$ in $R$ do
    for Each column $j$ in $R$ do
        $I_{i,j} \leftarrow compute\_spatial\_interpolation(R_{i,j})$
        $R_{i,j} \leftarrow R_{i,j} * temporal\_weight + I_{i,j} * spatial\_weight$
    end
end
```

The drawback of the diagonal based method is that the sizes of the diagonals vary, which causes a waste of GPU resources. When the diagonal size is lower than the number of block threads, some of these threads become idle. But, the performance gain in parallelizing the computation of the diagonal is better than sequential computation.

5. Experiments and Results

This section aims to evaluate the performance of the SP-TWDTW and TWDTW algorithms executed on Multicore (CPU) and Manycore (GPU) architectures. In the CPU tests, a machine with AMD FX-8320E (3.2 GHz, 8 MB Cache) and 8 GB of RAM was used. The GPU tests were performed on a NVIDIA GeForce GTX 1050 Ti card with 4 GB GDDR5 of available memory and 768 CUDA cores with clock of 1392 MHz. The TWDTW was implemented in C++ [2] and SP-TWDTW was implemented on the GPU using the NVIDIA CUDA language. To compare the response time between the SP-TWDTW and the TWDTW, the time series V and the patterns U were obtained from real data in Brazil [Maus et al. 2016] from MODIS sensor, specifically the Porto dos Gauchos municipality, that covers approximately 7,000 km^2 and is located in the state of Mato Grosso, Brazil, inside of the Amazon Biome. Each test was performed ten times and the response time was obtained from the average of them.

Temporal and spatial resolution of remote sensing system have increased in the last years being a great challenge for remote sensing field [Battude et al. 2016]. The results regarding the response time for high temporal resolution is presented in Figure 2(a), by comparing the TWDTW and SP-TWDTW over a pattern U and a time series V, varying their size. The response time increases considerably with TWDTW as the size of the pattern U and the time series V increases.

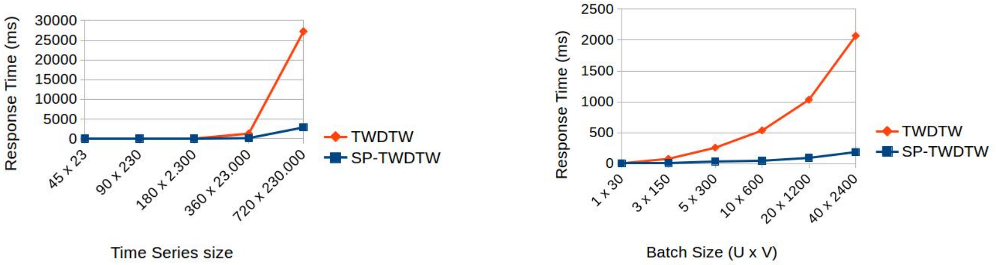
As one can see in Figure 2(a), the TWDTW algorithm works better for smaller time series (e.g. less than or equal to 90 x 230), however, when the size exceeds 90 x 230 it becomes more advantageous to use the SP-TWDTW. This is due to the fact that SP-TWDTW uses the GPU for processing and, in this case, the transfer time of U and V from the CPU memory to GPU memory overcomes the final response time of the algorithm. The SP-TWDTW algorithm presented response time up to 10 times lower than TWDTW.

Next, the results regarding the high resolution on spatial data are shown in Figure 2(b) by comparing the response time of SP-TWDTW and TWDTW using small batches of time series U and V with size of 45 and 23 respectively. As the batch size increases the difference in response time also increases. For batch size of 3x150, the response time of the SP-TWDTW was 7 times lower than the TWDTW. The SP-TWDTW used a batch size equal to 10x600, running several batches to calculate the alignment between each U and V. In this scenario, the SP-TWDTW obtained response time 11 times lower than the TWDTW, since it explored better the GPU architecture.

To compare the accuracy of the SP-TWDTW algorithm and TWDTW algorithm, we selected the Porto dos Gauchos municipality, located in the state of Mato Grosso, Brazil, inside of the Amazon Biome. Data from this study area were obtained from TWDTW's github [3] and covers approximately 7000 km^2 with 541 time series. The same

[2] The original version of TWDTW was developed in R, but, in this work, we implemented in C++ to be able to compare fairly with SP-TWDTW, since the C++ language has better performance.

[3] https://github.com/vwmaus/dtwSat



(a) Comparison between the TWDTW and the SP-TWDTW algorithms using only one pattern U and one time series V varying their size. The x axis contains the size of U and V following the format: $sizeofU$ x $sizeofV$.

(b) Comparison between the TWDTW and the SP-TWDTW algorithms with several patterns U and time series V. The x axis follows the format: number of patterns U x number of time series V. Each pattern has size 45 and time series size 23.

Figure 2. Comparing the response time between SP-TWDTW and TWDTW.

value of α and β were use for both methods. The Kriging, IDW and Nearest Neighbor(NN) spatial interpolation methods were used, with their parameters have been optimized through cross validation.

The Table 1 presents the results for time series classification using the SP-TWDTW. An analysis was performed by varying spatial and temporal axis weights and the interpolation methods. The SP-TWDTW algorithm with weight 0 for the spatial axis and 1 for temporal shows the number of time series incorrectly classified by the TWDTW. It is noteworthy that, in this case, the SP-TWDTW brings the gain of response time reduction, as presented in Figure 2, but not in accuracy. The original TWDTW algorithm implemented in R [Maus et al. 2016] obtained a response time of 9851 ms while the SP-TWDTW on the GPU took 40 ms, that is, 246 times faster than the TWDTW in R.

Regardless of the interpolation method, when the spatial axis are set with higher value of weight than the time axis, the SP-TWDTW presented lower accuracy than the TWDTW. This is explained on Table 2 by spatial closeness between the “Soybean-maize”, “Soybean-cotton” and “Cotton-fallow” crops. For “Forest” and “Soybean-millet” land usage, the SP-TWDTW didn’t misses even with spatial weight equals to 1 because the data in these land usage are clustered and independent from others land classes. Concerning to interpolation methods, the IDW performed better that the other methods due to the uniform data distribution. For sparse data, the Kriging method would probably present better accuracy than the IDW [Li and Heap 2014].

Table 2 presents the accuracy assessment for each land usage type according to the TWDTW and SP-TWDTW time series analysis methods and the IDW spatial interpolation method. The spatial and temporal weights were fixed on 0.4 and 0.6 respectively. The accuracy for SP-TWDTW was equal or better for all land use types, improving the TWDTW accuracy in a region that it already presented great results. Since the “Soybean-maize”, “Soybean-cotton” and “Cotton-fallow” crops are spatially closed and mixed, the SP-TWDTW negatively impacted on the analysis of the spatial axis.

Finally, in the results for the established scenarios, the SP-TWDTW presented a response time up to 11 times lower than the TWDTW in $C++$ and 246 times lower than

Weights		Interpolation Methods		
Spatial	Temporal	Kriging	NN	IDW
0	1	9	9	9
0,1	0,9	8	8	8
0,2	0,8	8	8	8
0,3	0,7	7	8	7
0,4	0,6	8	8	7
0,5	0,5	8	10	8
0,6	0,4	12	12	10
0,7	0,3	12	15	12
0,8	0,2	14	15	12
0,9	0,1	14	15	12
1	0	14	18	15

Table 1. Number of samples that were incorrectly classified by the SP-TWDTW method varying the spatial and temporal weights and the interpolation methods.

Method	Cotton-fallow		Forest		Soybean-cotton		Soybean-maize		Soybean-millet	
	UA (%)	PA (%)	UA (%)	PA (%)	UA (%)	PA (%)	UA (%)	PA (%)	UA (%)	PA (%)
TWDTW	95	100	100	100	100	87	95	100	100	100
SP-TWDTW	96	100	100	100	100	90	97	100	100	100

Table 2. Accuracy evaluation for each land usage type according to the TWDTW and SP-TWDTW algorithms.

the TWDTW in R, presenting a viable alternative to the time series analysis in the Big Remote Sensing Data era. The SP-TWDTW also presented similar user's and producer's accuracy than the traditional TWDTW.

6. Conclusion

In the class of complex computational problems, the time series analysis is one of the problems with increased demand for computing power [Rakthanmanon et al. 2013], due to the complexity of the algorithms and the large volume of data to be processed.

The TWDTW algorithm has been highlighted as one of the best solution found in the literature to perform the time series analysis for remote sensing images. However, TWDTW disregards the first law of Geography, processing each pixel independently. In addition, the TWDTW algorithm presents time complexity of $O(n^2)$, becoming unfeasible for large data sets.

This work presented a parallel solution capable of analyzing large volumes of time series data exploring parallel architectures. The solution, named SP-TWDTW (Spatial Parallel TWDTW), analyzes the spatial and temporal dimensions using the TWDTW algorithm to temporal analysis and the interpolation methods to spatial analysis. To support the Big Remote Sensing Data scenario, the SP-TWDTW took advantage of the Manycore architecture with the coordinated and appropriate usage of the large number of available cores. The SP-TWDTW processes the time series as batches, managing the CPU and GPU memory spaces to allows the processing of large amount of data without exceeding their capacity.

As results, the SP-TWDTW proved to be a promising solution for high temporal resolution data, with a speedup of 10 times over the traditional TWDTW and almost 11 times less response time compared to TWDTW for high spatial resolution data. Using spatial interpolation methods, SP-TWDTW was able to increase the accuracy of TWDTW for land use mapping in the Amazon region.

As future work, we intend to evaluate the SP-TWDTW algorithm using large data sets. We also intend to integrate the SP-TWDTW algorithm into the *DistSensing* platform [de Oliveira et al. 2017] exploring the resources of a cluster of computers.

References

Battude, M., Al Bitar, A., Morin, D., Cros, J., Huc, M., Sicre, C. M., Le Dantec, V., and Demarez, V. (2016). Estimating maize biomass and yield over large areas using high spatial and temporal resolution sentinel-2 like remote sensing data. *Remote Sensing of Environment*, 184:668–681.

Costa, W. S., Fonseca, L. M., Körting, T. S., SIMÕES, M., Bendini, H. N., and Souza, R. C. (2017). Segmentation of optical remote sensing images for detecting homogeneous regions in space and time. In *Embrapa Solos-Artigo em anais de congresso (ALICE)*. In: BRAZILIAN SYMPOSIUM ON GEOINFORMATICS, 18., 2017, Salvador. Proceedings... Salvador: Unifacs, 2017. p 40-51.

Cressie, N. and Wikle, C. K. (2015). *Statistics for spatio-temporal data*. John Wiley & Sons.

de Oliveira, M. G. and de Souza Baptista, C. (2012). Geostat-a system for visualization, analysis and clustering of distributed spatiotemporal data. In *GeoInfo*, pages 108–119.

de Oliveira, S. S. T., de Castro Cardoso, M., dos Santos, W., Costa, P., do Sacramento Rodrigues, V. J., and Martins, W. S. (2017). Distsensing: A new platform for time series processing in a distributed computing environment. *Revista Brasileira de Cartografia*, 69(5).

Gómez, C., White, J. C., and Wulder, M. A. (2011). Characterizing the state and processes of change in a dynamic forest environment using hierarchical spatio-temporal segmentation. *Remote Sensing of Environment*, 115(7):1665–1679.

Jamali, S., Jönsson, P., Eklundh, L., Ardö, J., and Seaquist, J. (2015). Detecting changes in vegetation trends using time series segmentation. *Remote Sensing of Environment*, 156:182–195.

João Jr, M., Sena, A. C., and Rebello, V. E. (2017). Implementação e avaliação de técnicas de paralelização no algoritmo de hirschberg para sistemas multicore. *Simpósio em Sistemas Computacionais de Alto Desempenho (WSCAD)*.

Kuenzer, C., Dech, S., and Wagner, W. (2015). Remote sensing time series revealing land surface dynamics: Status quo and the pathway ahead. In *Remote Sensing Time Series*, pages 1–24. Springer.

Li, J. and Heap, A. D. (2014). Spatial interpolation methods applied in the environmental sciences: A review. *Environmental Modelling & Software*, 53:173–189.

Lu, M., Chen, J., Tang, H., Rao, Y., Yang, P., and Wu, W. (2016). Land cover change detection by integrating object-based data blending model of landsat and modis. *Remote Sensing of Environment*, 184:374–386.

Maus, V., Câmara, G., Cartaxo, R., Sanchez, A., Ramos, F. M., and de Queiroz, G. R. (2016). A time-weighted dynamic time warping method for land-use and land-cover mapping. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(8):3729–3739.

Mitas, L. and Mitasova, H. (1999). Spatial interpolation. *Geographical information systems: principles, techniques, management and applications*, 1:481–492.

Petitjean, F., Inglada, J., and Gançarski, P. (2012). Satellite image time series analysis under time warping. *IEEE Transactions on Geoscience and Remote Sensing*, 50(8):3081–3095.

Petitjean, F. and Weber, J. (2014). Efficient satellite image time series analysis under time warping. *Ieee geoscience and remote sensing letters*, 11(6):1143–1147.

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and Keogh, E. (2013). Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(3):10.

Sakoe, H. (1971). Dynamic-programming approach to continuous speech recognition. In *1971 Proc. the International Congress of Acoustics, Budapest*.

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM.

Stein, M. L. (2012). *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media.

Vatsavai, R. R. (2008). *Machine Learning Algorithms for Spatio-temporal Data Mining*. PhD thesis, University of Minnesota, Minneapolis, MN, USA. AAI3338985.

Verbesselt, J., Hyndman, R., Newnham, G., and Culvenor, D. (2010). Detecting trend and seasonal changes in satellite image time series. *Remote sensing of Environment*, 114(1):106–115.

Xiao, L., Zheng, Y., Tang, W., Yao, G., and Ruan, L. (2013). Parallelizing dynamic time warping algorithm using prefix computations on gpu. In *High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on*, pages 294–299. IEEE.

Zhu, H., Gu, Z., Zhao, H., Chen, K., Li, C.-T., and He, L. (2018). Developing a pattern discovery method in time series data and its gpu acceleration. *Big Data Mining and Analytics*, 1(4).

A collaborative application for incidence data recording and geographic distribution of myiasis

Stefano W. P. Pontes[1], **Tiago B. Borchartt**[1], **Lívio M. Costa Júnior.**[2]

[1]Department of Informatics

[2]Department of Pathology
Federal University of Maranhão (UFMA)
São Luiz – MA – Brazil - CEP 65080-805

{stefano.walker, tiago.bonini, livio.martins}@ufma.br

Abstract. *This paper presents rMiíase, a collaborative mobile application that allows reporting cases of myiasis. To support the application, a complete structure with a web server was developed to receive, provide and synchronize the registration data of the cases reported by the users. It also has a management system for administrators to control the data registered. The application encourages users to submit or request the collection of samples that will be reviewed by laboratories for purposes of identification of myiasis-causing species. With the data obtained by the application, will be possible to study the feasibility of implantation of myiasis control plan in the affected regions.*

1. Introduction

Myiasis is one of the most common parasitic diseases that affects warm-blooded animals. In this disease, flies lay their eggs on the injured skin of animals. When the larvae hatch from eggs, they invade the subcutaneous tissue from where feed, causing severe tissue damage and leading to death [Teixeira 2013]. Myiasis can occur in both rural and urban areas, infecting animals and humans. The regions most favorable to insects appearance that causes myiasis are tropical and subtropical regions [Hall and Wall 1995].

Cochliomyia hominivorax, the main species causing myiasis, causes financial losses to brazilian cattle herds that have been estimated at about 150 million dollars a year. Losses generated by myiasis include weight loss, a decrease in milk production, leather damage and animal mortality. The prevention and treatment of these affected animals, in addition to raising the cost of production, contribute to the presence of drug residues in milk and meat, bringing harm to human health [Borja 2003].

The *C. hominivorax* had its eradication and control in the United States and part of Central America since the 1980s. Prior to eradication, the annual cost of preventive and control measures was US$ 120 million in 1960. In Panama reached US$ 43 million per year [Oliveira et al. 1982].

In Brazil, in order to implement a control plan for myiasis, it is necessary to collect data for knowledge of the following indices: the number of disease occurrence in the South America regions, the time of the year which they is most abudant, the fly specie and the animal host that is most infested. Currently, such data are non-existent. About these records, Teixeira (2013) concludes that there is a small number of health professionals who identify the agents that cause the infestation of this disease. Therefore, the occurrence rates in Brazil may be underestimated in relation to myiasis in animals include humans.

Based on this need, rMiíase (*report* myiasis) has been developed with the goal of being an accessible application for everybody record cases of myiasis. The user can register a case of myiasis by providing some basic data, such as host species, address, and location of the injured body. Then you can add the point in the map where the case is being recorded and capture an image of the injured body part by myiasis. In the end, the user is encouraged to collect a larvae sample and send it to one of the accredited laboratories, which will also appear on the application.

The user can see all the cases registered by others and the photo registered by them. Encouraged by the application, people are expected to send the samples to the accredited laboratory nearest to their locality, in order to the cases to be conffirmed, obtaining information on the species of fly that is causing the pathology. As the main objective, the application is to provide a database that will serve to study the possibility of implementing a control program of myiasis.

The remainder of this article is organized as follows: Section 2 presents some related works. In Section 3, the system architecture of the application is described. In Section 4, some analysis is done about the application. Finally, in Section 5 the conclusions of the study are presented.

2. Related Work

There are some projects that aim to function as geographic data recording platforms, using maps and allowing users to add data to the application base. Some of these works will be presented below.

As a first example, we can cite the app Fruit Map (2016), in which users can search for fruit trees in any region, besides to registering the presence of new trees. The user indicates on the map the location of the trees, selecting the type and degree of availability of the fruit. This information is available for everyone to see through the device so that other users can know which fruits are closest.

As another example, there is the app Dengue SC (2016), that is intended to reinforce the control of *Aedes aegypti* mosquito. Through this channel, the population can denounce the suspected outbreaks of dengue fever. Once the complaint is registered, the city hall has access to the event through a web tool that allows to follow the requests received and proceed with the verification.

There is the app WildHelp [Clark 2006], available for iOS devices, which helps users report sick or injured wildlife to an appropriate agency. The user captures and sends an animal photo to the application which, based on GPS coordinates, sends the photos and information to a nearby environmental agency. The information that users provide can help researchers, rescuers, and wildlife managers to understand better how and where animals get hurt.

The US Centers for Disease Control and Prevention (2017) developed the Epi Info$^{\text{TM}}$ application for the vector of mosquitoes, among them mosquitoes, *Aedes aegypti*, *Culex* and *Anopheles*. It is an application aimed at entomologists and agents to combat endemics. This application provides analysis of entomological indicators for planning and appropriate decision making on the actions of vector control measures.

These presented projects have several similarities with rMiíase, following the idea of reporting cases based on geographic location. However, the main difference between these tools and the rMiíase is the need to collect the samples from the registered cases to be analyzed in the laboratory, in order to know the species causing the myiasis. Because an eventual myiasis control plan needs to be individualized for certain species [Borja 2003]. Due to the complexity of the explanation, the other features, which allow differentiating the method from the other tools presented, will be described in the following sections as part of this article.

3. rMiíase Application

The client-server architecture, proposed by [Fielding 2002], was implemented to meet data synchronization requirements, such as registration and update data of user and cases. As shown in Figure 1, the architecture components are (1) mobile application, (2) Google Maps APIs, (3) GPS, (4) web server, (5) database and (6) web management system. Some of these components will be detailed in the following subsections.

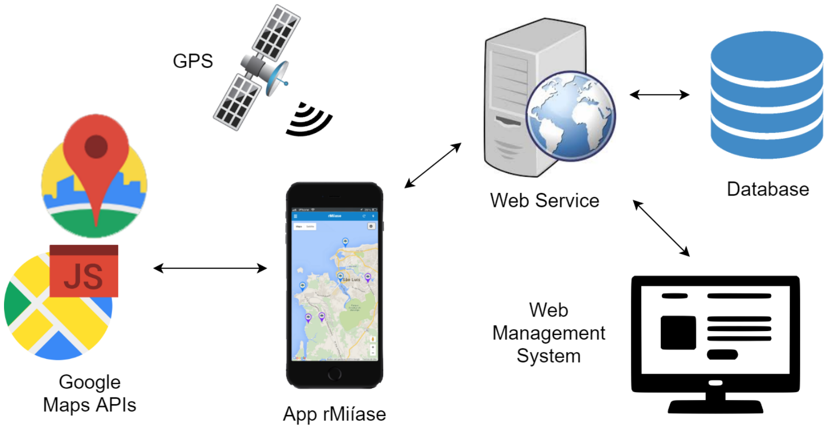


Figure 1. Application Architecture

3.1. Mobile Application

A collaborative mobile tool was designed to solve the need to create a database that records the occurrence rate of myiasis. The basic features functionality of the application are as follows:

- Allow the user to register a new case of myiasis. The case data is:
  - Species of the affected animal (canine, bovine, goat, human, etc.), the address of the occurrence and the part of the body injured.
  - The geographical location of the case, which will be through marking the location on the map provided on the screen, suggesting the current location by the GPS.
  - Image of the injured body part, which can be captured or added from the device gallery.

  - Indication of commitment to collect sample from the injured part of the animal to be sent to one of the accredited laboratories (optional).
  - Selection of the laboratory to which the user wishes to send the collected a sample (depending on whether the previous item was flagged in the affirmative).

- View on the map the cases registered by other users of the application, showing details, including the case image.

- Visualization and editing the data of the registered cases by the user. The user cannot exclude a case that has already been analyzed and approved by the laboratories.

- Registration and updating of the user's personal data (name, e-mail and telephone number).

- Access to information about myiasis disease, the importance of collecting and sending the samples, and access the website of the application.

The initial version of rMiíase is designed for Android and iOS mobile operating systems and is available on Play Store and App Store. rMiíase is a hybrid application, i.e., it is partially native, because of accesses the functionalities of the hardware, and partially web, because uses HTML, CSS, and JavaScript as web technologies. This type of development has the great advantage that the same application works on multiple platforms and is not necessary to write a different code for each one [Budiu 2013].

Apache Cordova was used in development [Wargo 2015], which is a multiplatform open source web application framework. The development of applications using HTML, CSS, and Javascript web technologies is possible with Cordova. This tool provides a set of APIs for accessing native functions of the Operating System and device hardware using Javascript without writing native code.

The development IDE used was Intel XDK [Intel 2016].The graphical interface was developed based on Framework7 (2015), that is a free and open source HTML framework for application interface development.

3.2. Google Maps APIs

Google offers several types of Application Programming Interface (API). APIs allow an application to use the features provided by it, without getting involved in the details of the software implementation.

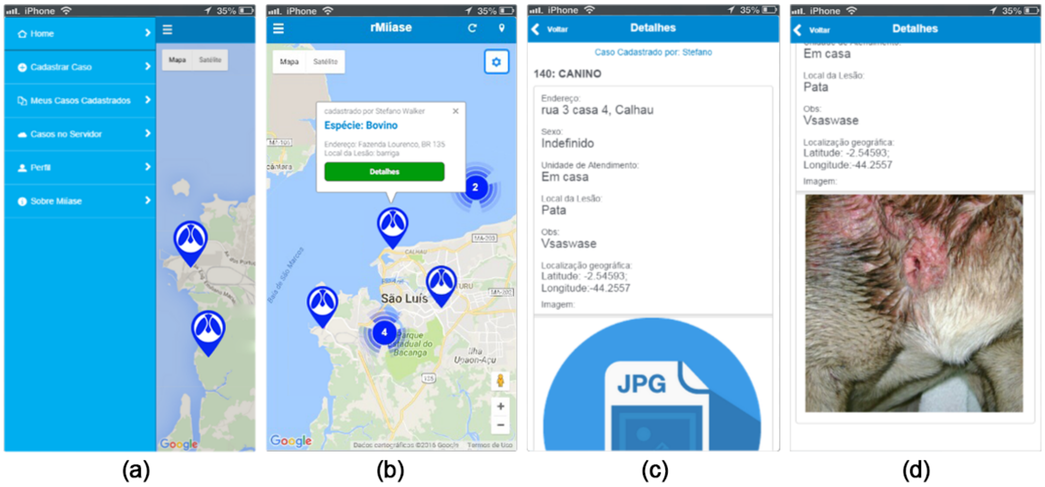
The Google Maps JavaScript API belongs to the category of web APIs and allows users to use maps and custom mapping information in their applications [Google 2016a]. Map customization styles are defined using the JSON[1] format, so the definition of the same map is supported in every web application. This customization format is dynamically processed in the application at runtime.

[1] JSON is an acronym for "JavaScript Object Notation", it is a light format for computer data interchange. JSON is a subset of JavaScript object notation, but its use does not require JavaScript exclusively (FLANAGAN, 2006).

The mobile app also uses Google Geolocation (2016) to get the location of the user's device. The Google Geolocation API returns a location and a precision radius based on GPS information, cell towers, and Wi-Fi access points that the mobile client can detect [Google 2016b]. Communication is performed by HTTPS using POST method. Both the request and response have JSON formatting and the content type of both is *application/json*.

3.3. Application Front-end

First, the app loads the Google Maps interface (2016), displaying the cases registered in nearby locations. By clicking on the case marker, the user visualizes a small frame with some data of the case as species, the user that registered and address informed. To view more details of the case, the user must click on the green button "Details" (Figure 2 (b)). Figure 2 (c) shows the possibility for user to access details of a case registered by another user, initially having the image of this case not loaded. The user can see the image only if clicks on the image download button (Figure 2 (d)).



cadastrado por Stefano Walker
Espécie: Bovino
Endereço: Fazenda Lourenco, BR 135
Local da Lesão: barriga

Caso Cadastrado por: Stefano
140: CANINO
Endereço:
rua 3 casa 4, Calhau
Sexo:
Indefinido
Unidade de Atendimento:
Em casa
Local da Lesão:
Pata
Obs:
Vsaswase
Localização geográfica:
Latitude: -2.54593;
Longitude:-44.2557
Imagem:

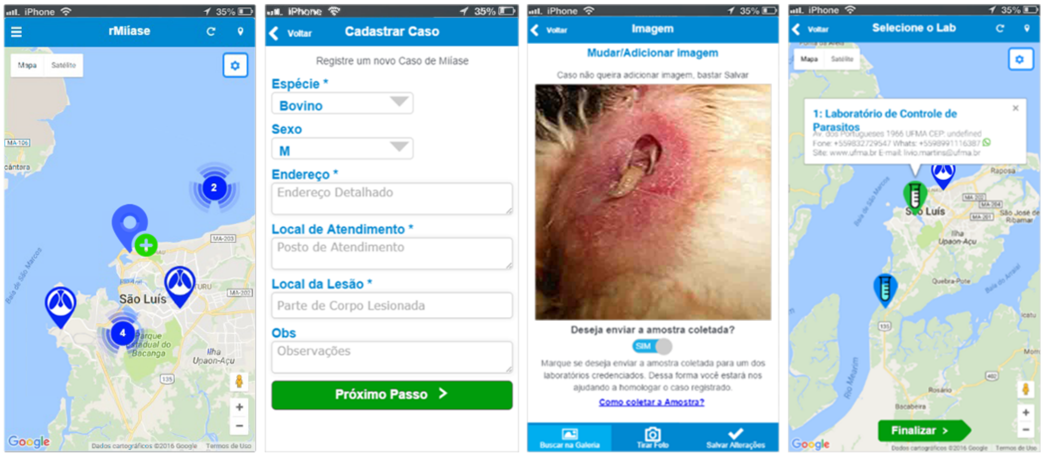
Em casa
Local da Lesão:
Pata
Obs:
Vsaswase
Localização geográfica:
Latitude: -2.54593;
Longitude:-44.2557
Imagem:

Figure 2. Application: (a) Menu; (b) Initial screen showing the map and registered case markers; (c) Details screen of a case of another user; (d) Details screen after downloading the image.

To register a new case, the user will double-click anywhere on the map, inserting a new case marker (Figure 3 (a)) which, when clicked, will start the "Register Case" function. Another alternative to access this function is through the menu. After that, the data entry screen will appear. (Figure 3 (b)).

The user is directed to the map where he should mark the locality where the case occurred after completing the form, if not previously marked. In the next step of the case registration, there is the option to insert an image through the camera or the image gallery (Figure 3 (c)). If the option "do you want to send a sample?" is marked as affirmative, before finalizing the register, the map will appear again for the user select the laboratory to send the collected sample (Figure 3 (d)). The application always suggests the nearest laboratory to location of the registered case. After saving the case, the data is stored in the device and the web server, and can be edited or deleted by the user, except when the case is approved after the sample be analyzed in the laboratory.

Figure 4 (a) shows the list of cases registered by the user, where he is allowed to see the details and also edit and delete the cases. The list of this user's cases is stored on the device and web server and is synchronized with every change made by the user or by the management system administrator. Figure 4 (c) shows the user registration screen, invoked during the registration of the first case.



Voltar Cadastrar Caso

Registre um novo Caso de Miíase

Espécie *
Bovino

Sexo
M

Endereço *
Endereço Detalhado

Local de Atendimento *
Posto de Atendimento

Local da Lesão *
Parte de Corpo Lesionada

Obs
Observações

Próximo Passo >

Voltar Imagem

Mudar/Adicionar imagem

Caso não queira adicionar imagem, bastar Salvar

Deseja enviar a amostra coletada?

SIM

Marque se deseja enviar a amostra coletada para um dos laboratórios credenciados. Dessa forma você estará nos ajudando a homologar o caso registrado.

Como coletar a Amostra?

Buscar na Galeria | Tirar Foto | Salvar Alterações

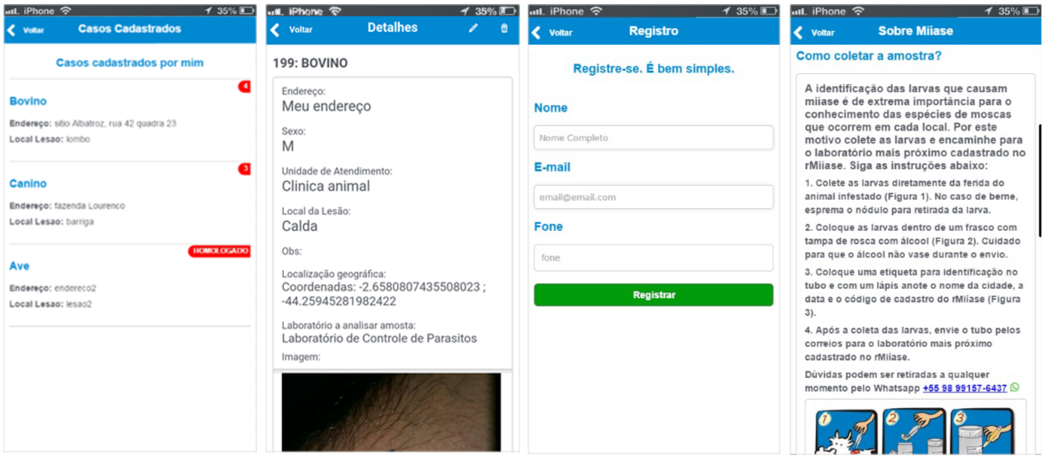
Voltar Selecione o Lab

1: Laboratório de Controle de Parasitos
Av. dos Portugueses 1966 UFMA CEP: undefined
Fone: +559832729547 Whats: +5598991116387
Site: www.ufma.br E-mail: livio.martins@ufma.br

Finalizar >

(a) (b) (c) (d)

Figure 3. rMiíase application: (a) Home screen with the addition of the case register marker. (b) Case registration form. (c) Image capture from the camera or gallery. (d) Selection of laboratories to send samples collected.



Voltar Casos Cadastrados

Casos cadastrados por mim

4
Bovino
Endereço: sitio Albatroz, rua 42 quadra 23
Local Lesao: lombo

3
Canino
Endereço: fazenda Lourenco
Local Lesao: barriga

HOMOLOGADO
Ave
Endereço: endereco2
Local Lesao: lesao2

Voltar Detalhes

199: BOVINO

Endereço:
Meu endereço

Sexo:
M

Unidade de Atendimento:
Clinica animal

Local da Lesão:
Calda

Obs:

Localização geográfica:
Coordenadas: -2.6580807435508023 ; -44.25945281982422

Laboratório a analisar amostra:
Laboratório de Controle de Parasitos

Imagem:

Voltar Registro

Registre-se. É bem simples.

Nome
Nome Completo

E-mail
email@email.com

Fone
fone

Registrar

Voltar Sobre Miíase

Como coletar a amostra?

A identificação das larvas que causam miíase é de extrema importância para o conhecimento das espécies de moscas que ocorrem em cada local. Por este motivo colete as larvas e encaminhe para o laboratório mais próximo cadastrado no rMiíase. Siga as instruções abaixo:

1. Colete as larvas diretamente da ferida do animal infestado (Figura 1). No caso de berne, esprema o nódulo para retirada da larva.

2. Coloque as larvas dentro de um frasco com tampa de rosca com álcool (Figura 2). Cuidado para que o álcool não vase durante o envio.

3. Coloque uma etiqueta para identificação no tubo e com um lápis anote o nome da cidade, a data e o código de cadastro do rMiíase (Figura 3).

4. Após a coleta das larvas, envie o tubo pelos correios para o laboratório mais próximo cadastrado no rMiíase.

Dúvidas podem ser retiradas a qualquer momento pelo Whatsapp +55 98 99157-6437

(a) (b) (c) (d)

Figure 4. rMiíase application: (a) List of cases registered by the user. (b) Detail screen of a case registered by the user. (c) User registration form. (d) Section about myiasis.

3.4. Application web server

The application server has been implemented in PHP language with a MySQL database and has the same modeling structure as the internal database of the mobile application.

The web server can be accessed by POST requests, which return data or error parameters. The possible types of requisitions are detailed below:

- *uploadUser*: Receives user registration data from the application to the web server.
- *uploadCase*: Receive the case data.
- *deleteCase*: Delete case on server.
- *seeHomologatedCase*: Checks if the case has already been approved by laboratories (called before deleteCase).
- *updateUser*: Receive user data and update on the server.
- *updateCase*: Receives the case data and updates on the server.
- *downloadPicture*: Receives the case id and returns the corresponding image.

All cases registered in the server appear on the map application in the form of markers (Figure 2 (b)). The application accesses the web server to retrieve the case data in JSON format and load the markers. The same process is performed to access the list of laboratories.

The web server API is responsible for data integration. The server provides the database information for the mobile application platform and for the management system via the GET request. This service may provide two types of information: the data (exception for the image) of the registered cases and the data of the registered laboratories. When the API is accessed, it queries the data in the database and mounts a JSON package to a web address, allowing the mobile application to capture that package to list the data at a given time. The request to access the API data is done through the Ajax[2] request, taking advantage of its asynchronous characteristic that allows the loading of isolated components.

Images of cases registered by other users are not loaded with the map markers. Otherwise, this process would require a lot of server data flow, and a smartphone would not support a large amount of data loaded into its RAM. Therefore, the images of the cases are only displayed if the user accesses the details of the case and click the specific button for image display. The download and upload of the images are also done through Ajax requests.

3.5. Web Management System

The app has a homepage[3]. This page contains information about the project, application download links, and contact information. The Web Management System has restricted access by login and password. The administrators of this web system can perform the following functions:

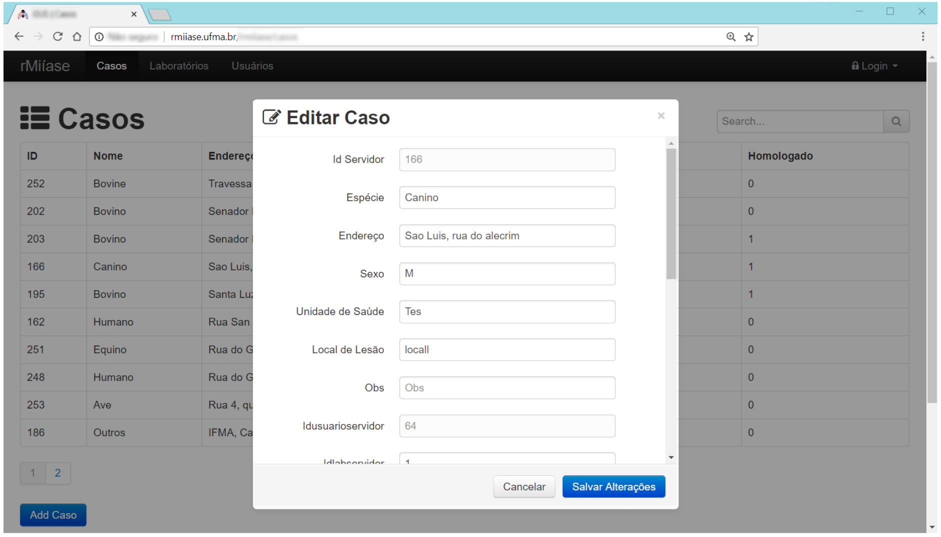
- edit or delete the cases registered by any user of the application;

[2] Asynchronous JavaScript and XML. Ajax basically uses JavaScript, XML and HTML dynamically. It allows, immediately, the validation of forms, suggestion of automatic completion and other exchange of data, without having to reload the entire web page.

[3] Available in `http://www.rmiiase.ufma.br`

- edit, delete or block any user;
- manage the list of accredited laboratories;
- homologate the registered cases, when the sample sent by the user is analyzed by one of the laboratories;
- download the database into spreadsheets.

The management system was developed using Phrezee (2016), a framework for developing CRUDs[4] in PHP. Phreeze generates a complete generic system in the MVC (Model-View-Controller) and ORM (Object Relational Mapping). Figure 5 presents the web management system screen, in the case editing functionality.



Casos

ID	Nome	Endereç	Homologado
252	Bovine	Travessa	0
202	Bovino	Senador	0
203	Bovino	Senador	1
166	Canino	Sao Luis,	1
195	Bovino	Santa Lu:	1
162	Humano	Rua San	0
251	Equino	Rua do G	0
248	Humano	Rua do G	0
253	Ave	Rua 4, qu	0
186	Outros	IFMA, Ca	0

Editar Caso

Id Servidor: 166
Espécie: Canino
Endereço: Sao Luis, rua do alecrim
Sexo: M
Unidade de Saúde: Tes
Local de Lesão: locall
Obs: Obs
Idusuarioservidor: 64
Idlabservidor: 1

Figure 5. Web Management System: "Edit Case" functionality.

4. Application Analyzes

The main tests of the application were made in the Android platform, in smartphone considered of intermediate level, composed of a processor with 1.2GHz of clock, 2GB of RAM, the screen of 5 inches and rear camera with 13Mpx.

For the Android platform, the app was produced with backward compatibility up to version 4.0 (API Android 14), reaching approximately, in this year, 94% of Android devices. For iOS, it reaches up to version 8.0 and has compatibility with 96.5% of iOS devices. The application, in the versions for Android and iOS, presented a good graphic interface layout with Framework7.

The images captured by the camera provided enough quality photos for the proposal. The storage size of the images obtained was in the range of 40 to 90 KB, with the approximate resolution of 500x500 pixels, after the application of compression algorithm. The small file size makes easy transfer data and leaves the quality of the photos at an acceptable level.

[4] CRUD is an acronym for Create, Read, Update, and Delete, the four basic operations used in relational databases.

Images in JPG format are stored in the gallery (in the default storage directory) and are also stored in the application's internal database in the format Base64[5]. The computational cost dedicated to Base64 file format conversion is small and produces a binary file with approximately 75% compression. In tests, the process of downloading registry data of 18 cases, spent 6.3KB of connection traffic to the server, an average of 0.35KB per case. All cases data of other users are not permanently stored on the user smartphone that visualizes.

The data synchronization scheme was implemented to address the situation of no connection to the internet. Synchronization routines are triggered in all register and edit functions, as well as in application initialization events (*OnStart()*), background return (*onResume()*) and connection to the internet (*onOnline()*).

The application dynamically loads cases. When the user accesses the map, the cases are viewed in a grouped way, according to the zoom of the map screen, and downloads the data registered by other users as the user navigates the map. Finally, rMiíase has support for the Portuguese, Spanish and English languages, in order to reach users from the Americas.

5. Final Discussions

The rMiíase has proven to be a good solution for registering cases of myiasis around the world, especially in South America. The application will provide reliable data for myiasis studies in this continent, along with the implementation of the web system for homologation/confirmation of cases, providing a great contribution to the area of public and animal health.

The tool has the advantage of requiring little data usage when connecting to the server. In addition, the rMiíase works offline, allowing the manipulation of data of registered cases and their subsequent synchronization, once it is connected to the internet. The application does not require professionals to be scattered across regions by collecting samples, the system administrators can receive regional data directly from affected populations.

The application has advantages of maintaining source code, by using the web-based development, which allows the same source code to be used in the production of a cross-platform application. This results in a reduction in software update time and cost.

As future enhancements, the authors suggest adding a dynamic data visualization tool to the management system, providing disease statistics as graphs, filtering queries, and exporting reports to web system administrators.

The rMiíase is registered[6] in the National Institute of Industrial Property (INPI) and is expected to begin to be used for official data collection in 2018.

6. Acknowledgment

The authors thanks the technical support provided by the Information Technology Nucleus (NTI) of UFMA and the financial support by FAPEMA, MAPA and CNPq for the execution of this work.

[5] Base64 is a method for encoding data for download on the Internet.

[6] Computer program registration number: BR5120160018321 (`http://www.inpi.gov.br/`)

References

Borja, G. E. M. (2003). Erradicação ou manejo integrado das miíases neotropicais das américas. *Pesquisa Veterinária Brasileira, SciELO Brasil*, vol. 23.

Budiu, R. (2013). Mobile: Native apps, web apps, and hybrid apps. `https://www.nngroup.com/articles/mobile-native-apps/`. [Accessed on October 13, 2016].

CIASC (2016). Dengue SC. `http://www.ciasc.sc.gov.br/302/`. [Accessed on November 8, 2016].

Clark, S. (2006). New WildHelp app helps users report sick or injured animals. Mercury News. `https://www.mercurynews.com/2016/06/24/`. [Accessed on June 20, 2018].

Fielding, R. (2002). Principle design of the modern web architecture. `http://dl.acm.org/citation.cfm?id=514185/`. [Accessed on August 13, 2016].

Framework7 (2016). Framework7. `https://framework7.io/`. [Accessed on January 13, 2017].

FuitMap (2016). FruitMap. `https://fruitmap.app/`. [Accessed on November 8, 2016].

Google (2016a). The Google Maps API. `https://developers.google.com/maps/`. [Accessed on August 24, 2018].

Google (2016b). The Google Maps Geolocation API. `https://developers.google.com/maps/documentation/geolocation/`. [Accessed on August 24, 2018].

Hall, M. and Wall, R. (1995). Myiasis of humans and domestic animals. *Advances in parasitology*, 35:257–334.

Intel (2016). Intel xdk. `http://xdk-software.intel.com`. [Accessed on December 13, 2017].

Oliveira, C., Mova, G., and Mello, R. (1982). Flutuação populacional de Cochliomyia hominivorax no município de Itagual. *Brazilian journal of veterinary research*, Rio de Janeiro.

Phreeze (2016). Phreeze Framework. `http://www.phreeze.com/`. [Accessed on December 27, 2017].

Teixeira, D. G. (2013). Principais dípteros causadores de miíases. *Programa de Pós-graduação em ciência animal*.

Wargo, J. M. (2015). *Apache Cordova 4 Programming*. New York, Pearson Education.

An Efficient Flash-aware Spatial Index for Points

Anderson Chaves Carniel[1], George Roumelis[2], Ricardo Rodrigues Ciferri[3], Michael Vassilakopoulos[2], Antonio Corral[4], Cristina Dutra de Aguiar Ciferri[1]

[1]Department of Computer Science – University of São Paulo – Brazil

accarniel@gmail.com, cdac@icmc.usp.br

[2]Department of Electrical and Computer Engineering – University of Thessaly – Greece

groumelis@uth.gr, mvasilako@uth.gr

[3]Department of Computing – Federal University of São Carlos – Brazil

ricardo@dc.ufscar.br

[4]Department on Informatics – University of Almeria – Spain

acorral@ual.es

Abstract. *Spatial database systems often employ spatial indices to speed up the processing of spatial queries. In addition, modern spatial database applications are interested in exploiting the positive characteristics of flash-based Solid State Drives (SSDs) like fast reads and writes. However, designing spatial indices for SSDs (i.e., flash-aware spatial indices) has been a challenging task because of the intrinsic characteristics of these devices. In this paper, we propose the eFIND xBR$^+$-tree, a novel flash-aware spatial index for points. The eFIND xBR$^+$-tree combines the efficient indexing method of the xBR$^+$-tree with the sophisticated data structures and algorithms of eFIND to handle points in SSDs efficiently. Experiments carried out considering real and synthetic spatial data showed that the eFIND xBR$^+$-tree overcame its closest competitor by reducing the elapsed time to construct the index from 28.4% to 83.5% and to execute spatial queries up to 34.6%.*

1. Introduction

The use of a spatial index is essential for processing spatial queries because the search space is greatly reduced [Gaede and Günther 1998]. The main assumption of several spatial indices is that the spatial objects are stored in magnetic disks (i.e., *Hard Disk Drives* - HDDs). Hence, they often consider the slow mechanical access and the high cost of search and rotational delay of disks in their design. We term spatial indices designed for magnetic disks as *disk-based spatial indices*.

A wide range of disk-based spatial indices has been proposed in the literature [Gaede and Günther 1998]. The R-tree and its variants, such as the R$^+$-tree and the R*-tree, are well-known spatial indices. The efficient indexing of multidimensional points has been a main focus of several indices because of the use of points in real spatial database applications [Gaede and Günther 1998]. Among the existing disk-based spatial indices, we highlight the xBR$^+$-tree [Roumelis et al. 2015], which provides data structures and algorithms for handling points efficiently. In fact, extensive experimental evaluations [Roumelis et al. 2017] showed that the xBR$^+$-tree outperforms variants of the R-tree (the R*-tree and the R$^+$-tree) when processing different types of spatial queries.

On the other hand, advanced database applications are interested in using modern storage devices like *flash-based Solid State Drives* (SSDs) [Mittal and Vetter 2016]. This includes spatial database systems that employ spatial indices to efficiently retrieve spatial objects stored in SSDs [Carniel et al. 2017a]. The main reason of this interest is because SSDs, in contrast to HDDs, have smaller size, lighter weight, lower power consumption, better shock resistance, and faster reads and writes.

However, SSDs have introduced a new paradigm in data management because of their intrinsic characteristics [Jung and Kandemir 2013, Mittal and Vetter 2016]. A well-known characteristic is the asymmetric cost of reads and writes, where a write requires more time and power consumption than a read. Further, SSDs are able to write data to empty pages only, which means that updating data in previously written pages requires an erase-before-update operation. Other factors that impact on SSD performance are the processing of interleaved reads and writes, and the execution of reads on frequent locations. These factors are related to the internal controls of SSDs, such as the internal buffers and the read disturbance management [Jung and Kandemir 2013].

To deal with the intrinsic characteristics of SSDs, spatial indices specifically designed for SSDs have been proposed in the literature. However, designing spatial indices for SSDs, termed here as *flash-aware spatial indices*, has been a challenging task. A common strategy is to mitigate the poor performance of random writes by storing index modifications in a write buffer. Whenever this buffer is full, a flushing operation is performed. Among existing flash-aware spatial indices proposed in the literature (see Section 2), FAST-based indices [Sarwat et al. 2013] and eFIND-based indices [Carniel et al. 2017b, Carniel et al. 2018] distinguish themselves. FAST and eFIND are generic frameworks that transform disk-based hierarchical indices into flash-aware hierarchical indices. They also provide support for data durability by using a log-structured approach that allows to recover its write buffer after a fatal problem (e.g., power failure). Comparing FAST to eFIND, the former does not fully exploit SSD performance because it does not consider several intrinsic characteristics of SSDs. On the other hand, eFIND contains managers based on a set of design goals that are developed to fully take into account the intrinsic characteristics of SSDs. Hence, we consider eFIND as the state-of-the-art method for porting disk-based spatial indices to SSDs.

Considering the aforementioned state-of-the-art methods, an open question is how to efficiently port the xBR$^+$-tree to SSDs using eFIND. In this paper, we answer this question by proposing the *eFIND xBR$^+$-tree*, a flash-aware spatial index for points. This novel index combines the efficient spatial organization of xBR$^+$-trees with the sophisticated managers of eFIND specifically designed for SSDs. That is, the eFIND xBR$^+$-tree is designed as an integration of the xBR$^+$-tree's hierarchical structure with the eFIND's data structures. We measure the efficiency of this porting by conducting experimental evaluations, considering real and synthetic datasets, against the FAST xBR$^+$-tree, the porting of the xBR$^+$-tree to SSDs using FAST. Our performance results show that the eFIND xBR$^+$-tree ports the xBR$^+$-tree to SSDs efficiently, guaranteeing smaller elapsed times to process insertions and intersection range queries.

The rest of this paper is organized as follows. Section 2 surveys related work. Section 3 presents the eFIND xBR$^+$-tree. Section 4 discusses the conducted experiments. Finally, Section 5 concludes the paper and presents future work.

2. Related Work

A few flash-aware spatial indices have been proposed in the literature. In this section, we summarize the characteristics of the main flash-aware spatial indices as follows.

The *RFTL* [Wu et al. 2003] ports the R-tree to SSDs using a write buffer to avoid random writes. The main problem of RFTL is the flushing operation because it flushes all modifications stored in the write buffer, requiring high elapsed times. Another problem is related to the data durability. This means that the modifications stored in the write buffer are lost after a system crash or power failure.

FAST [Sarwat et al. 2013] distinguishes itself because it generalizes the write buffer to store modifications of any hierarchical index. Hence, it transforms any disk-based hierarchical index into a flash-aware index. Further, FAST provides a specialized *flushing algorithm* that picks only a set of nodes, termed *flushing unit*, to be written to the SSD instead of writing all modifications contained in the write buffer. FAST also provides support for data durability. However, FAST faces several problems. First, its flushing algorithm might pick nodes without modifications, resulting in unnecessary writes to the SSD. This is due to the static creation of flushing units as soon as nodes are created in the index. Second, its write buffer stores the modifications in a list possibly containing repeated entries, impacting negatively the performance of retrieving modified nodes. Finally, FAST does not improve the performance of reads.

The *FOR-tree* [Jin et al. 2015] improves the flushing algorithm of FAST by dynamically creating flushing units containing modified nodes only. It also abolishes splitting operations by allowing overflowed nodes. Whenever a specific number of accesses in an overflowed node is reached, a merge-back operation is invoked. This operation eliminates overflowed nodes by inserting them into the parent node, growing up the tree if needed. However, the number of accesses of an overflowed root node is never incremented in an insertion operation. As a consequence, the construction of a FOR-tree, inserting one spatial object by time, forms an overflowed root node instead of a hierarchical structure. This critical problem disallowed us to create spatial indices over large and medium spatial datasets.

Specific flash-aware spatial indices for points have also been developed. *MicroHash* and *MicroGF* [Lin et al. 2006] are data structures for flash-based sensor devices. Due to the low processing capabilities of sensor devices, they deploy write buffers only. The *F-KDB* [Li et al. 2013] employs a write buffer that stores modified entries of the K-D-B-tree, called logging entries. Its main problem is the complex operation to retrieve nodes because the entries of a node might be stored in different flash pages. Finally, the *Grid file for flash memory* [Fevgas and Bozanis 2015] employs a buffer strategy based on the LRU to cache modifications of the grid file. A flushing operation writes to the SSD only those index pages that are classified as cold pages. However, the quantity of modifications is not considered, leading to a possibly high number of flushing operations.

eFIND [Carniel et al. 2017b, Carniel et al. 2018] is a generic framework that efficiently transforms any disk-based spatial index into a flash-aware spatial index. It is based on distinct design goals that considers the intrinsic characteristics of SSDs. eFIND employs efficient in-memory data structures to handle index modifications and a specialized flushing operation that smartly picks a number of nodes to be written to the SSD. Further, eFIND prevents reads on frequent locations and avoids interleaved reads and writes. Due to its advantages, we consider eFIND as the state-of-the-art method for porting disk-based spatial indices to SSDs. Hence, we employ eFIND to port the xBR$^+$-tree to SSDs.

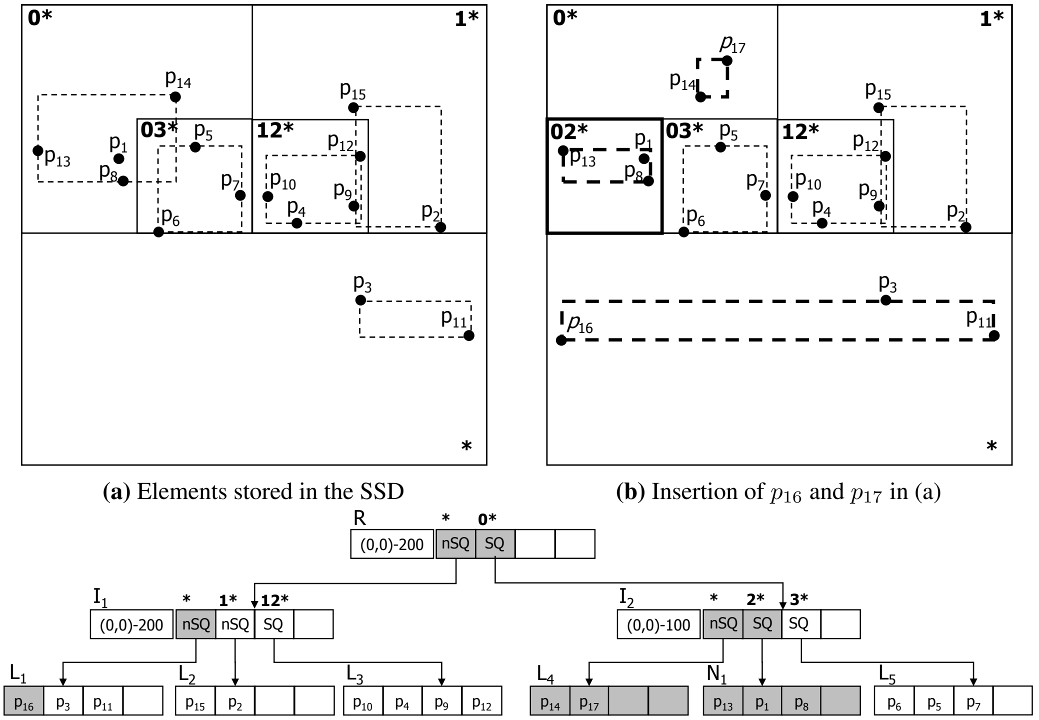
3. The eFIND xBR$^+$-tree: An Efficient Flash-Aware Spatial Index for Points

3.1. The Tree Structure

eFIND does not change the underlying tree structure of the ported index. Hence, the tree structure of the eFIND xBR$^+$-tree is the same as the xBR$^+$-tree. The xBR$^+$-tree is a hierarchical index based on the regular decomposition of space of Quadtrees [Gaede and Günther 1998] able to index multidimensional points. Hence, it is a *space-driven access method*. For bidimensional points, the xBR$^+$-tree decomposes recursively the space by 4 equal quadrants, called *sub-quadrants*. Figure 1a depicts an example of an eFIND xBR$^+$-tree that indexes 15 points (i.e., p_1 to p_{15}) and is whole stored in the SSD. Figure 1b shows the eFIND xBR$^+$-tree with a set of adjustments, represented by thick lines, after the insertion of two new points, p_{16} and p_{17}. These points and the resulting adjustments are modifications stored in the main memory (Section 3.2) and are also highlighted in the hierarchical representation of Figure 1c. We detail the structure of this eFIND xBR$^+$-tree as follows.

There are two types of nodes, internal nodes and leaf nodes. Internal nodes consist of entries in the following format $(p, DBR, qside, shape)$. Each entry of an internal node refers to a child node that is pointed by p and represents a sub-quadrant of the original space. DBR refers to the data bounding rectangle that minimally encompasses the points stored in such sub-quadrant. $qside$ stores the side length of the sub-quadrant corresponding to the child node's entry. Finally, $shape$ is a flag that indicates if the sub-quadrant is either a complete square or a non-complete square. The entries of an internal node are also sorted by their *addresses*. Each address is calculated by using $qside$ and DBR, and consists of a sequence of *directional digits* representing a sub-quadrant. The directional digits 0, 1, 2, and 3 respectively symbolize the NW, NE, SW, and SE sub-quadrants of a relative space. Hence, it follows the Z-order.

Figure 1c depicts a tree with 3 internal nodes, R, I_1, and I_2. Each internal node has also a header containing data about its sub-quadrant. For instance, the origin point of the sub-quadrant of R is $(0, 0)$ with a side length of 200. The address of each entry of an internal node is showed in bold (but, this is not actually stored). For instance, the right child of R that points to I_2 is the NW quadrant of the original space, denoted as *0** (* is used to mark the end of the address). Further, it represents a complete square (i.e., *SQ*). Its *DBR* consists of a minimum bounding rectangle containing the points p_5 to p_8, p_{14}, p_{17}, p_{13}, and p_1. The left child of R represents a region derived from the spatial difference between the original space and the region of the NW quadrant. Hence, it has address equal to * (i.e., empty) and represents a non-complete square (i.e., *nSQ*). Finally, addresses of entries of internal nodes determine a sub-quadrant in relation to the region of their node. For instance, the address *3** (in node I_2 of Figure 1c) represents the SE sub-quadrant of the NW sub-quadrant of the original space (the region of I_2, denoted by *0** in R of Figure 1c).



(a) Elements stored in the SSD

(b) Insertion of p_{16} and p_{17} in (a)

(c) Hierarchical representation of (b)

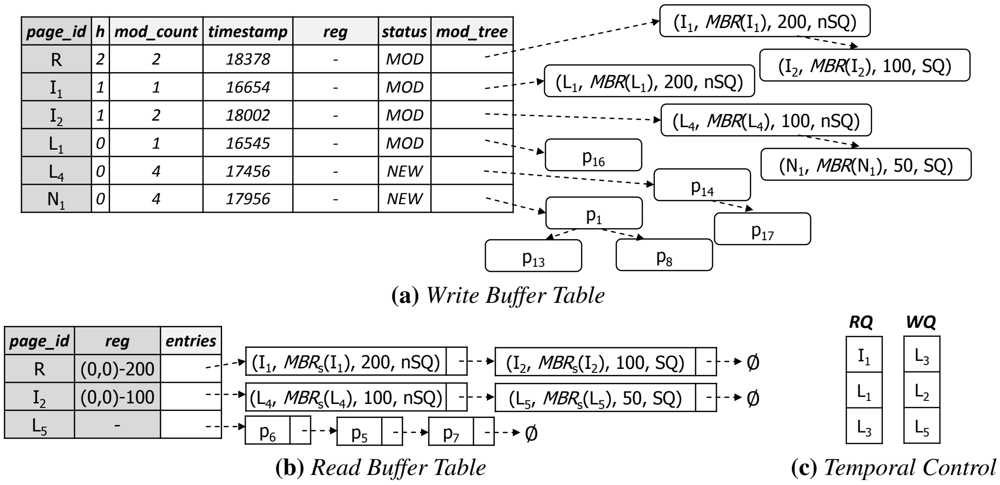
Figure 1. An example of an eFIND xBR^{+}-tree.

Leaf nodes contain entries in the format (p, id), where p is the multidimensional point and id is a pointer to the register of p. These entries are sorted by X-axis coordinates of the points, allowing the use of the *plane sweep technique* in specific spatial query types. For instance, the leaf node L_1 in Figure 1c contains the points p_{16}, p_3, and p_{11}, which are sorted by their X-axis coordinates depicted in Figure 1b. The pointers to the registers of these points are omitted.

When the capacity of a leaf or internal node is achieved, the quadrant encompassing the overflowed node is partitioned into two sub-quadrants according to a Quadtree-like hierarchical decomposition. Different criteria for this partitioning are conceivable, as discussed in Roumelis et al. 2017. For instance, Figure 1b depicts the creation of a new sub-quadrant with address *02** (i.e., node N_1 in Figure 1c) resulting from a splitting operation after inserting p_{17}.

3.2. Employed Data Structures

eFIND provides specific data structures to fulfill its design goals [Carniel et al. 2018]; they are: (i) a write buffer, (ii) a read buffer, (iii) a log file, and (iii) read and write queues. To deal with the xBR^{+}-tree, we extend the eFIND's data structures as follows: (i) we adapt the write and read buffers to store specific data related to internal nodes, (ii) we generalize the storage of index modifications according to the sorting properties of internal and leaf nodes (Section 3.1), and (iii) we adjust the structure of log entries to recover the write buffer after a system crash. We detail these extensions as follows.



page_id	*h*	*mod_count*	*timestamp*	*reg*	*status*	*mod_tree*
R	*2*	*2*	*18378*	-	*MOD*	
I_1	*1*	*1*	*16654*	-	*MOD*	
I_2	*1*	*2*	*18002*	-	*MOD*	
L_1	*0*	*1*	*16545*	-	*MOD*	
L_4	*0*	*4*	*17456*	-	*NEW*	
N_1	*0*	*4*	*17956*	-	*NEW*	

(I_1, *MBR*(I_1), 200, nSQ)
(I_2, *MBR*(I_2), 100, SQ)
(L_1, *MBR*(L_1), 200, nSQ)
(L_4, *MBR*(L_4), 100, nSQ)
(N_1, *MBR*(N_1), 50, SQ)
p_{16}
p_{14}
p_{17}
p_1
p_{13}
p_8

(a) *Write Buffer Table*

page_id	*reg*	*entries*
R	(0,0)-200	
I_2	(0,0)-100	
L_5	-	

(I_1, *MBR*$_S$(I_1), 200, nSQ)
-
(I_2, *MBR*$_S$(I_2), 100, SQ)
-
∅
(L_4, *MBR*$_S$(L_4), 100, nSQ)
-
(L_5, *MBR*$_S$(L_5), 50, SQ)
-
∅
p_6
-
p_5
-
p_7
-
∅

RQ	*WQ*
I_1	L_3
L_1	L_2
L_3	L_5

(b) *Read Buffer Table*

(c) *Temporal Control*

Figure 2. Data structures to handle the eFIND xBR$^+$-tree of Figure 1.

The write buffer is implemented as a hash table named *Write Buffer Table* and stores the modifications of nodes that were not applied to the SSD yet. Its main goal is to avoid random writes to the SSD. The key of this hash table is the identifier of a node (*page_id*) and its value stores modifications in the format (*h*, *mod_count*, *timestamp*, *reg*, *status*, *mod_tree*). Here, *h* stores the height of the modified node; *mod_count* is the quantity of in-memory modifications; *timestamp* informs when the last modification was made; *reg* is the sub-quadrant of a newly created internal node; and *status* is the type of modification made and can be NEW, MOD, or DEL for representing newly created nodes in the buffer, nodes stored in the SSD but with modified entries, and deleted nodes, respectively. If *status* is equal to DEL, *mod_tree* is null. Otherwise, it is a red-black tree containing the most recent version of modified entries. Each element of this red-black tree has the format (*e*, *mod_result*), where *e* is the key and corresponds to the unique identifier of an entry and *mod_result* stores the latest version of an entry, assuming null if *e* was removed. The comparison function to determine the order of the elements in the red-black tree is defined to deal with the specific sorting of entries of internal and leaf nodes (Section 3.1). This is important when retrieving nodes (Section 3.3).

Figure 2a shows the *Write Buffer Table* for the eFIND xBR$^+$-tree of Figure 1b. In this figure, MBR means the rectangle that encompasses all points of a sub-quadrant considering the modifications stored in the write buffer. The elements of the *mod_tree* employ the same format as an entry of the underlying index. For instance, the first line of the hash table in Figure 2a shows that R, located in the *height* 2, has the *status* MOD, and stores 2 in-memory modifications in the *mod_tree*. Hence, the most recent version of the two entries of R are now the entries of the red-black tree, i.e., $(I_1, MBR(I_1), 200, nSQ)$ and $(I_2, MBR(I_2), 100, SQ)$.

The read buffer is implemented as another hash table named *Read Buffer Table* and caches nodes stored in the SSD that are frequently accessed. The key of this hash table is the unique node identifier (*page_id*) and its value stores a list of entries of the node (*entries*) and its sub-quadrant, if it is an internal node (*reg*). Figure 2b depicts that R, I_2, and L_5 are cached in the *Read Buffer Table*. In this figure, MBR_S refers to the stored data bounding rectangle of a child node. For instance, the entries of the cached version of I_2 consists of two entries, even after the creation of N_1.

To provide data durability, all modifications are also stored in a log file. The format of a log entry is the same as a hash entry in the *Write Buffer Table* to rebuild the write buffer after a system crash. The cost of keeping the log of the modifications is very low because it requires sequential writes only [Sarwat et al. 2013, Carniel et al. 2018].

The temporal control of eFIND remains unchanged. The read and writes queues, named ***RQ*** and ***WQ***, are employed to provide the temporal control of eFIND. Each queue is a First-In-First-Out data structure. ***RQ*** stores identifiers of the nodes read from the SSD, while ***WQ*** keeps the identifiers of the last nodes written to the SSD. Figure 2c shows that the last read nodes are I_1, L_1, and L_3, and the last flushed nodes are L_3, L_2, and L_5.

3.3. Methods for Handling the Index Operations

eFIND provides generic algorithms to execute the following operations: (i) maintenance operation, which is responsible for reorganizing the index whenever modifications are made on the underlying spatial dataset (i.e., insertions, deletions, and updates); (ii) search operation, which is responsible for executing spatial queries; (iii) flushing operation, which selects a set of modifications stored in the write buffer to be written to the SSD according to a flushing policy; and (iii) restart operation, which rebuilds the write buffer after a fatal problem and compacts the log file. To deal with the xBR$^+$-tree, we extend eFIND as follows: (i) we generalize the retrieval algorithm of eFIND to return valid internal and leaf nodes, respecting their sorting properties (Section 3.1), and (ii) we detail the management of splits, which improves the space utilization of the write buffer.

To retrieve a node N, the eFIND xBR$^+$-tree takes two sorted lists as input: (i) the modified entries stored in the *Write Buffer Table*, and (ii) the entries stored in the SSD. The former is empty if N has not modifications, while the latter is empty if there exists a hash entry of N in the *Write Buffer Table* with *status* equal to NEW. If one list is empty, the other non-empty list is directly returned. The second list is always sorted because its first flushing happens when its status in the *Write Buffer Table* is equal to NEW.

The following merge operation should be performed if these lists are not empty. It is based on the classical merge operation between sorted files [Folk et al. 1997]. Let i, j be two integer values, where i indicates the position in the first list and j indicates the position in the second list. A loop is then processed, starting with $i = j = 0$. If the element in the position i on the first list, called E_a, goes before the element in the position j on the second list, called E_b, this means that the merge operation appends E_a to N and increments i by 1 since an element of the first list has been processed. If the inverse happens, i.e., E_b goes before E_a, the merge operation appends E_b to N and increments j by 1. Evaluating the order of two node entries requires the execution of the same comparison function employed by the red-black trees (Section 3.1). If E_a and E_b point to the same entry (i.e., their unique identifier are equal), the merge operation appends only E_a to N if its value (i.e., *mod_result* in the *mod_tree*) is different to null and increment both i and j by 1. This is done because the result should only maintain the latest version of the entry and non-null entries. The loop is finished if i (j) is equal to the number of entries in the first (second) list. Finally, the entries that were not evaluated by the loop are appended to N, which is returned as the final step of the merge operation.

The merge operation requires a cost of $\mathcal{O}(n + m)$, where n is the number of elements in the first list and m is the number of elements in the second list. The use of a red-black tree for storing modified entries is essential for the merge operation and represents a main advantage compared to FAST. First, it guarantees the order between the entries stored in the write buffer. Hence, the resulting node is valid. Second, it has an amortized cost of inserting and updating entries stored in the main memory. Finally, the space allocated in the main memory is better managed because it does not allow repeated elements. All these factors combined to the other eFIND's managers lead to a better performance compared to porting the xBR$^+$-tree using FAST, as reported in our experiments (Section 4).

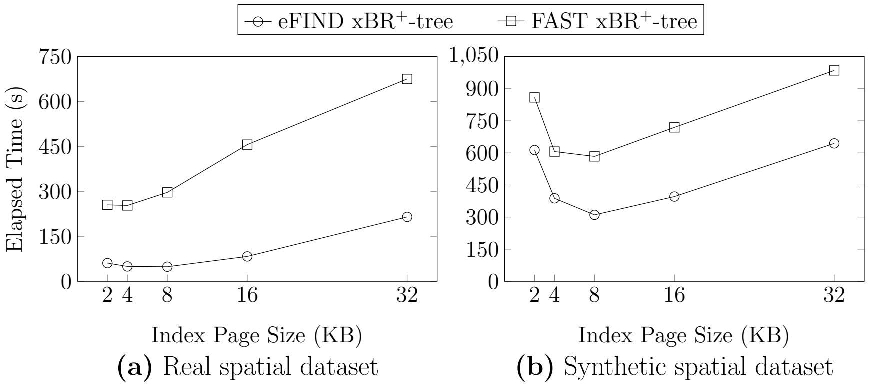
Handling splitting operations in the write buffer is performed as follows. Let A be an overflowed node. First, if A has a hash entry in the *Write Buffer Table*, it assumes *status* equal to DEL, deleting previous modifications of A and thus freeing some space in the write buffer. Otherwise, a new hash entry, with *status* equal to DEL, in the *Write Buffer Table* is created. Then, after completing the splitting operation in the main memory, A has a new set of entries and a new node, called B, is created. Hence, the hash entry of A in the *Write Buffer Table* becomes NEW and the entries of A are added in its corresponding *mod_tree*. A similar procedure for B is employed. This strategy for handling splitting operations is important because of the management of the write buffer space. An example of handling of a splitting operation is depicted in Figure 1c, after inserting p_{17}. As a result, L_4 has 4 modifications (fifth line in the *Write Buffer Table* of Figure 2a), where one modification is related to its deletion, another modification for its creation, and then two modifications for inserting its two entries. Further, N_1 is newly created in the write buffer (last line in the *Write Buffer Table* of Figure 2a).

4. Experimental Evaluation

4.1. Experimental Setup

Datasets. We used two spatial datasets. The first one is a real spatial dataset, called *brazil_points2017*, containing 770,842 points that represent geographical locations of Brazil like public telephones, ATMs, and towers. This dataset was extracted from the OpenStreetMap and its statistical description can be found in Carniel et al. 2017c. The second one is a synthetic dataset containing 1,000,000 points equally distributed in 125 clusters uniformly distributed in the range $[0, 1]^2$. The points in each cluster (i.e., 8,000 points) were located around the center of each cluster, according to Gaussian distribution.

Configurations. We compared two configurations: (i) the *FAST xBR$^+$-tree*, which is our closest competitor (Section 2), and (ii) the *eFIND xBR$^+$-tree*, which is our proposed index. We created the FAST xBR$^+$-tree by extending FAST in an analogous way to the extensions we performed to eFIND. However, due to space limitations, this extension is not presented here. Both configurations had a buffer of 512KB, log capacity of 10MB, and employed index page sizes (i.e., node sizes) from 4KB to 32KB. For the FAST xBR$^+$-tree, we used the FAST* flushing policy, which provided the best results according to Sarwat et al. 2013. For the eFIND xBR$^+$-tree, we employed the best parameter values according to our experiments [Carniel et al. 2018]: the use of 60% of the oldest modified nodes to create flushing units, a flushing policy using the height of nodes as weight to choose one flushing unit to be written, and the allocation of 20% of the buffer for the read buffer.



(a) Real spatial dataset (b) Synthetic spatial dataset

Figure 3. The eFIND xBR$^+$-tree showed the fastest elapsed times when building spatial indices over both spatial datasets.

Finally, both configurations employed the flushing unit size equal to 5 since this value commonly provide good results for FAST and eFIND [Carniel et al. 2018].

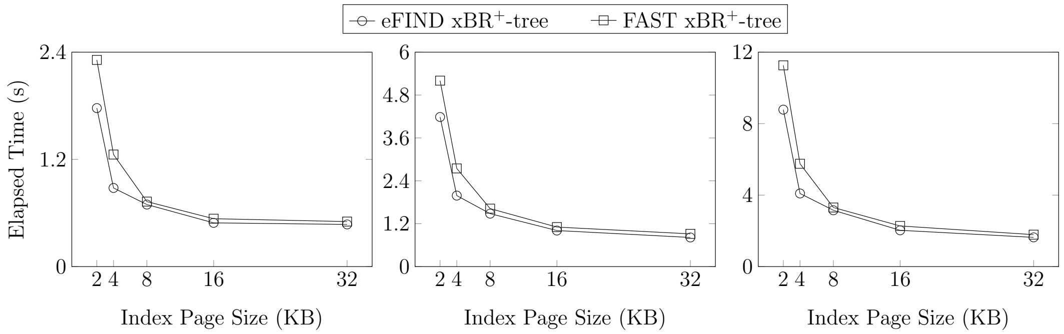
Workloads. We executed two workloads: (i) index construction, and (ii) execution of 300 intersection range queries (IRQs). An IRQ retrieves the points contained in a given rectangular query window, including its borders. Three different sets of query windows were used, representing respectively 100 rectangles with 0.001%, 0.01%, and 0.1% of the area of the total extent of the dataset being used by the workload. We generated different query windows for each dataset using the algorithms described in Carniel et al. 2017c. This method allows us to measure the performance of spatial queries with distinct selectivity levels. We consider the selectivity of a spatial query as the ratio of the number of returned objects and the total objects; thus, the three sets of query windows built IRQs with low, medium, and high selectivity, respectively. We executed the workloads as a sequence, that is, the index construction followed by the execution of IRQs. For each configuration and dataset, this sequence was executed 5 times. We avoided the page caching of the system by using direct I/O. For the first workload, we collected the average elapsed time. For the second workload, we calculated the average elapsed time to execute each set of query windows.

Running Environment. We employed a server equipped with an Intel Core® i7-4770 with a frequency of 3.40GHz, 32GB of main memory, and the SSD Kingston V300 of 480GB. The operating system used was Ubuntu Server 14.04 64 bits.

4.2. Performance Results

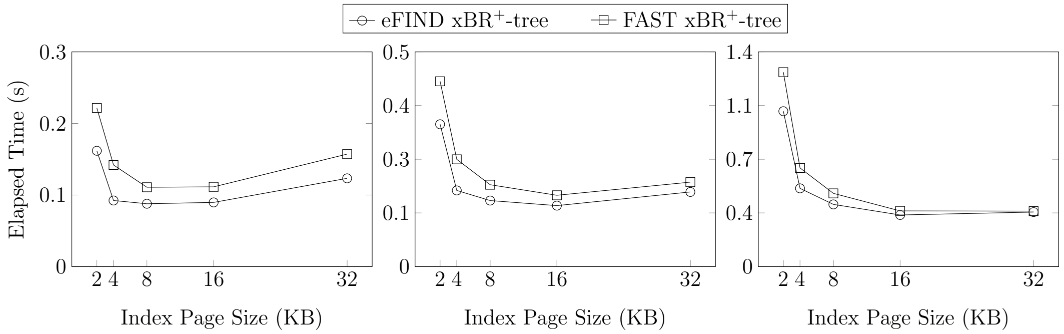
Index Construction. Figure 3 depicts that the eFIND xBR$^+$-tree overcame the FAST xBR$^+$-tree for both spatial datasets. The performance gains of the eFIND xBR$^+$-tree ranged from 68.1% to 83.5% for the real spatial dataset (Figure 3a) and from 28.4% to 46.5% for the synthetic spatial dataset (Figure 3b). A performance gain shows how much a configuration reduced the elapsed time from another configuration.

The eFIND xBR$^+$-tree exploited the benefits of the SSD because it leverages specific data structures and sophisticated methods that take into account the intrinsic characteristics of SSDs. We highlight three main contributions. First, the use of the read buffer



(a) Low selectivity (b) Medium selectivity (c) High selectivity

Figure 4. Performance results when processing IRQs on the real spatial dataset. The eFIND xBR$^+$-tree outperformed the FAST xBR$^+$-tree for all selectivity levels, showing expressive performance gains.



(a) Low selectivity (b) Medium selectivity (c) High selectivity

Figure 5. Performance results when processing IRQs on the synthetic spatial dataset. The eFIND xBR$^+$-tree showed better elapsed time than the FAST xBR$^+$-tree for all selectivity levels.

avoided several reads on frequent locations of the SSD, even using a small portion of the whole buffer size. Second, the merge operation accelerated the retrieval of the most recent version of modified nodes. This operation also naturally guaranteed the order of node entries. Finally, the eFIND xBR$^+$-tree avoided interleaved reads and writes.

Building spatial indices over the synthetic spatial dataset required more time because it is larger than the real spatial dataset. In both spatial datasets, the eFIND xBR$^+$-tree provided the best elapsed time by using the page size equal to 8KB. The use of larger page sizes faced the problem of writing big flushing units [Sarwat et al. 2013, Carniel et al. 2018], while the use of smaller page sizes introduced the management of a high number of nodes.

Spatial Query Processing. Figures 4 and 5 depict that the eFIND xBR$^+$-tree always provided the best performance results when processing all selectivity levels of IRQs. For the real spatial dataset (Figure 4), the eFIND xBR$^+$-tree showed performance gains up to 29.4%, 27.2%, and 28.5% for the high, medium, and high selectivity levels, respectively. For the synthetic spatial dataset (Figure 5), it showed performance gains up to 34.6%, 28.6%, and 20.2% for the high, medium, and high selectivity levels, respectively. Similarly to our previous discussions, these performance gains were obtained thanks to the effective use of the merge operation and read buffer.

Processing IRQs over the synthetic dataset required much less time than processing IRQs over the real dataset because of its specific spatial distribution. In most of the cases, better elapsed times were obtained by using large page sizes (i.e., 16KB and 32KB) because more entries are loaded into the main memory with a few reads. IRQs returning more points (i.e., with high selectivity) exhibited higher elapsed times. This is due to the traversal of multiple large nodes in the main memory, requiring more CPU time than queries with low selectivity. This fact also contributed to a similar time among the configurations when processing IRQs with high selectivity using the page size of 32KB.

5. Conclusions and Future Work

This paper proposes the eFIND xBR$^+$-tree, a novel flash-aware spatial index for points. eFIND allowed to efficiently port the xBR$^+$-tree to SSDs because its data structures fit well the properties and spatial organization of the xBR$^+$-tree. To accomplish this porting, eFIND has been generalized to deal with the sorting properties of nodes and to efficiently handle modifications produced by the xBR$^+$-tree.

The eFIND xBR$^+$-tree has empirically evaluated against the FAST xBR$^+$-tree, which employed FAST to port the xBR$^+$-tree to SSDs. The eFIND xBR$^+$-tree provided performance gains from 28.4% to 83.5% when building spatial indices and up to 34.6% when processing IRQs. In general, the page size of 16KB was the best configuration. Although this page size required more time to build an index compared to smaller page sizes, it provided the best results to execute the IRQs. Hence, the cost of its construction can be suppressed by its efficiency when processing spatial queries.

The efficiency of the eFIND xBR$^+$-tree is obtained mainly because of two reasons. First, the internal structure of the xBR$^+$-tree was completely integrated to eFIND, guaranteeing all the properties of the xBR$^+$-tree that offer good spatial indexing performance. Second, eFIND is based on distinct design goals that fully exploit SSD performance.

Our future work includes to evaluate the eFIND xBR$^+$-tree against other spatial organizations, such as the data partitioning strategy of eFIND R-trees [Carniel et al. 2018]. Another future work is to extend our experiments to consider workloads that mix insertions and other types of queries, such as point queries.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. This work has also been supported by CNPq and by the São Paulo Research Foundation (FAPESP). Anderson C. Carniel has been supported by the grants #2015/26687-8 and #2018/10687-7, FAPESP. Ricardo R. Ciferri has been supported by the grant #311868/2015-0, CNPq. Cristina D. A. Ciferri has been supported by the grant #2018/22277-8, FAPESP.

References

Carniel, A. C., Ciferri, R. R., and Ciferri, C. D. A. (2017a). Analyzing the performance of spatial indices on hard disk drives and flash-based solid state drives. *Journal of Information and Data Management*, 8(1):34–49.

Carniel, A. C., Ciferri, R. R., and Ciferri, C. D. A. (2017b). A generic and efficient framework for spatial indexing on flash-based solid state drives. In *European Conf. on Advances in Databases and Information Systems*, pages 229–243.

Carniel, A. C., Ciferri, R. R., and Ciferri, C. D. A. (2017c). Spatial datasets for conducting experimental evaluations of spatial indices. In *Satellite Events of the Brazilian Symp. on Databases - Dataset Showcase Workshop*, pages 286–295.

Carniel, A. C., Ciferri, R. R., and Ciferri, C. D. A. (2018). A generic and efficient framework for flash-aware spatial indexing. *Information Systems*. https://doi.org/10.1016/j.is.2018.09.004.

Fevgas, A. and Bozanis, P. (2015). Grid-file: Towards to a flash efficient multi-dimensional index. In *Int. Conf. on Database and Expert Systems Applications*, pages 285–294.

Folk, M. J., Zoellick, B., and Riccardi, G. (1997). *File Structures: An Object-Oriented Approach with C++*. Addison Wesley, 3rd edition.

Gaede, V. and Günther, O. (1998). Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231.

Jin, P., Xie, X., Wang, N., and Yue, L. (2015). Optimizing R-tree for flash memory. *Expert Systems with Applications*, 42(10):4676–4686.

Jung, M. and Kandemir, M. (2013). Revisiting widely held SSD expectations and rethinking system-level implications. In *ACM SIGMETRICS Int. Conf. on Measurement and Modeling of Computer Systems*, pages 203–216.

Li, G., Zhao, P., Yuan, L., and Gao, S. (2013). Efficient implementation of a multi-dimensional index structure over flash memory storage systems. *The Journal of Supercomputing*, 64(3):1055–1074.

Lin, S., Zeinalipour-Yazti, D., Kalogeraki, V., Gunopulos, D., and Najjar, W. A. (2006). Efficient indexing data structures for flash-based sensor devices. *ACM Transactions on Storage*, 2(4):468–503.

Mittal, S. and Vetter, J. S. (2016). A survey of software techniques for using non-volatile memories for storage and main memory systems. *IEEE Transactions on Parallel and Distributed Systems*, 27(5):1537–1550.

Roumelis, G., Vassilakopoulos, M., Corral, A., and Manolopoulos, Y. (2017). Efficient query processing on large spatial databases: A performance study. *Journal of Systems and Software*, 132:165–185.

Roumelis, G., Vassilakopoulos, M., Loukopoulos, T., Corral, A., and Manolopoulos, Y. (2015). The xBR+-tree: an efficient access method for points. In *Int. Conf. on Database and Expert Systems Applications*, pages 43–58.

Sarwat, M., Mokbel, M. F., Zhou, X., and Nath, S. (2013). Generic and efficient framework for search trees on flash memory storage systems. *GeoInformatica*, 17(3):417–448.

Wu, C.-H., Chang, L.-P., and Kuo, T.-W. (2003). An efficient R-tree implementation over flash-memory storage systems. In *ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*, pages 17–24.

A User-centric View of Distributed Spatial Data Management Systems

João Pedro de Carvalho Castro[1], Anderson Chaves Carniel[1], Cristina Dutra de Aguiar Ciferri[1]

[1]Department of Computer Science – University of São Paulo – Brazil

jp.carvalhocastro@usp.br, accarniel@gmail.com, cdac@icmc.usp.br

Abstract. *Distributed spatial data management systems (DSDMSs) represent a new technology capable of managing huge volumes of spatial data using parallel and distributed frameworks. An increasing number of DSDMSs have been proposed in the literature, requiring a comparison among them. However, comparisons available in the literature only provide a system-centric view of DSDMSs, which is essentially based on performance evaluations. Thus, there is a lack of comparisons based on the user-centric view, which is aimed to help users to understand how the characteristics of DSDMSs are useful to meet the specific requirements of their spatial applications. In this paper, we fill this gap in the literature. We provide a user-centric comparison of Hadoop-GIS, SpatialHadoop, SpatialSpark, GeoSpark, SIMBA, LocationSpark, SparkGIS, and Elcano, using as a basis an extensive set of criteria related to the characteristics of spatial data handling and to the aspects inherent to distributed systems. Based on this comparison, we introduce a set of guidelines to help users to choose an appropriate DSDMS. We also describe a case study to illustrate the use of these guidelines.*

1. Introduction

The analysis of spatial data is a core issue for corporations that use geographic location to take strategic decisions and to enhance the user experience. These corporations have a massive advantage over their competitors and are able to react quickly to business conditions changes. Nowadays, the volume of spatial data is growing increasingly fast, mainly due to the wide variety of applications that harvest this data, such as mobile and Internet of Things applications. Therefore, there is a demand for new technologies capable of managing huge volumes of spatial data.

Distributed spatial data management systems (DSDMSs) have emerged as a solution to this demand. They provide specialized functionalities aimed to process and index huge volumes of vector spatial data using parallel and distributed frameworks, such as the Apache Hadoop MapReduce[1] and the Apache Spark[2] [García-García et al. 2017]. The Apache Hadoop MapReduce is based on a generic programming model composed of map and reduce functions, while the Apache Spark is based on in-memory computation and on a Resilient Distributed Dataset (RDD) abstraction. DSDMSs are developed on the top of these frameworks, inheriting their characteristics and advantages, and providing extended functionalities to deal with spatial data.

[1] https://hadoop.apache.org/

[2] https://spark.apache.org/

Several DSDMSs have been proposed in the literature, which are classified as Hadoop- or Spark-based systems. The most remarkable Hadoop-based systems are Hadoop-GIS [Aji et al. 2013] and SpatialHadoop [Eldawy and Mokbel 2015]. The state-of-the-art Spark-based systems include SpatialSpark [You et al. 2015], GeoSpark [Yu et al. 2015], SIMBA [Xie et al. 2016], LocationSpark [Tang et al. 2016], SparkGIS [Baig et al. 2017], and Elcano [Engélinus and Badard 2018]. However, each DSDMS has its own characteristics and introduces different functionalities to deal with spatial data in parallel and distributed environments. Hence, they integrate two different perspectives, the characteristics of spatial data handling [Güting 1994, OGC 2018] and the aspects inherent to distributed systems [Pandey et al. 2018].

Due to the variety of DSDMSs and provided functionalities, users who design, develop, and implement spatial applications for corporations face the challenge of choosing one system over the others. Indeed, the arguments behind choosing a given DSDMS depend on the purpose of the application. There are spatial applications that process ad-hoc spatial queries (e.g., [Wiemann et al. 2018]). For these applications, the chosen system should provide support for processing different types of spatial operations, such as topological predicates (e.g., contains, inside, and meet), geometric set operations (e.g., union, intersection), and numerical operations (e.g., area, distance). Other spatial applications require interoperability among different systems (e.g., [Lee and Reichardt 2005]); thus, the existence of different representations (e.g., textual and binary) for spatial objects (e.g., points, lines, and regions) is a requirement. Further, there are applications that require quick answers to specific spatial queries [Pandey et al. 2018]. In this case, the chosen system should provide indices specifically designed to answer these queries efficiently.

A review of related work aimed to compare DSDMSs shows that existing studies focus on comparing these systems experimentally (see Section 2). That is, they provide a *system-centric* view of DSDMSs based on performance evaluations. However, it is also important for users to understand how the characteristics of these systems are useful to meet the specific requirements of their spatial applications. To the best of our knowledge, there is no related work that provides a comparison based on this *user-centric* view.

The main goal of this work is to fill this gap in the literature by analyzing, from the *user-centric* point of view, the following DSDMSs: Hadoop-GIS, SpatialHadoop, SpatialSpark, GeoSpark, SIMBA, LocationSpark, SparkGIS, and Elcano. We introduce the contributions described as follows.

- Comparison of the DSDMSs using an extensive set of criteria related to the characteristics of spatial data handling and to the aspects inherent to distributed systems.
- Proposal of a set of guidelines, based on the comparison, to help users to identify the systems that most meet the specific requirements of their spatial applications.
- Description of a case study using GeoSpark to illustrate the use of the guidelines.

This paper is organized as follows. Section 2 reviews related work. Section 3 defines the set of criteria and compares the analyzed DSDMSs. Section 4 introduces our guidelines. Section 5 describes the case study. Finally, Section 6 concludes the paper.

2. Related Work

We survey related work considering two groups. The first one refers to approaches that introduce DSDMSs. Because the proposal of these systems depends on computational advances such as those related to parallel and distributed frameworks for processing big data, the first DSDMSs available in the literature are Hadoop-based. Here, we are interested in Hadoop-GIS [Aji et al. 2013] and SpatialHadoop [Eldawy and Mokbel 2015]. The latest systems have been Spark-based, i.e., SpatialSpark [You et al. 2015], GeoSpark [Yu et al. 2015], SIMBA [Xie et al. 2016], LocationSpark [Tang et al. 2016], SparkGIS [Baig et al. 2017], and Elcano [Engélinus and Badard 2018]. Further, the work of García-García et al. (2017) introduces algorithms for optimizing distance join queries and implements them using SpatialHadoop and LocationSpark. Differently from our work, these approaches only briefly and technically summarize system by system. That is, they do not conduct a comparison among them considering the characteristics of spatial data handling and the aspects inherent to distributed systems. They also do not propose guidelines for users.

The second group refers to approaches aimed to provide a performance comparison among DSDMSs. In this context, there are only two related works that have been proposed in the literature. In Hagedorn et al. (2017), a performance evaluation is conducted focusing on the spatial filter and join operators for the following DSDMSs: Hadoop-GIS, SpatialHadoop, SpatialSpark, GeoSpark, and STARK[3], which is a spatial-temporal query processing extension that integrates into any Spark application. A broader performance evaluation is introduced in Pandey et al. (2018). First, the authors briefly survey SpatialHadoop, Hadoop-GIS, SpatialSpark, GeoSpark, Simba, Magellan[4], and LocationSpark. Then, they present extensive experiments involving the last five DSDMSs, considering five different spatial queries (i.e., range query, kNN query, spatial joins between distinct spatial data types, distance join, and kNN join) and four different data types (i.e., points, lines, rectangles, and polygons). However, Hagedorn et al. (2017) and Pandey et al. (2018) provide a *system-centric* view of DSDMSs, which is aimed to compare these systems based on their performance only. On the other hand, we compare DSDMSs considering the *user-centric* view, which is aimed to help users to understand how the characteristics of these systems are useful to meet the specific requirements of their spatial applications. Another differential of our work is that we also compare SparkGIS and Elcano, which are recently published DSDMSs.

In this paper, we analyze DSDMSs with publications in the literature and that support spatial data only. Thus, Magellan and STARK are not included in our comparisons. Further, carrying out performance evaluations of the DSDMSs is out of the scope of our paper due to the *user-centric* view. In this context, we consider the work of Pandey et al. (2018) as the state-of-the-art *system-centric* view. We use their findings in Section 3.2 and in Section 4 to complement our work.

3. User-Centric Comparative Analysis

In this section, we introduce a detailed *user-centric* comparison among the following DSDMSs: Hadoop-GIS, SpatialHadoop, SpatialSpark, GeoSpark, SIMBA, LocationSpark, SparkGIS, and Elcano. We use an extensive set of criteria that integrate two different perspectives: (i) the characteristics of spatial data handling (Section 3.1); and (ii) the aspects inherent to distributed systems (Section 3.2).

[3] `https://github.com/dbis-ilm/stark`

[4] `https://github.com/harsha2010/magellan`

Table 1. A *User-centric* view of DSDMSs, considering the characteristics of spatial data handling.

DSDMS	Spatial Data Types	Representation of Spatial Objects	Geometric Set Operations	Topological Predicates	Numerical Operations	Spatial Indexing
Hadoop-GIS	✓	textual only	union and intersection only	✓	✓	✓
SpatialHadoop	simple points, rectangles, and polygons only	textual only	union only	limited	for queries only	✓
SpatialSpark	✓	textual only	✓	✓	✓	✓
GeoSpark	✓	✓	✓	✓	✓	✓
SIMBA	simple points and rectangles only	textual only	no	limited	✓	✓
LocationSpark	simple points and rectangles only	user should implement	no	limited	for queries only	✓
SparkGIS	at least simple spatial data types	not specified	not specified	not specified	at least for queries	✓
Elcano	✓	textual only	✓	✓	✓	✓

3.1. Support for Spatial Data Types and Their Operations

Table 1 compares the DSDMSs considering the following characteristics of spatial data handling [Güting 1994, OGC 2018]: spatial data types, representation of spatial objects, geometric set operations, topological predicates, numerical operations, and spatial indexing techniques. A *checkmark icon* in a cell indicates that the DSDMS completely fulfills the corresponding criterion. A *not specified* expression indicates that no information is provided in the research paper that introduces the DSDMS, and the criterion was not further analyzed because the implementation of the system is not available yet.

Before detailing the motivation and the context behind each aforementioned criterion, it is important to note that the underlying library used by the DSDMSs to handle spatial objects impacts directly on their capabilities regarding the management of spatial data. SpatialSpark, GeoSpark, and Elcano are based on the JTS library[5]. Hence, they fulfill almost all the criteria since this library follows the OGC specifications [OGC 2018].

Spatial data types. The support for spatial data types is fundamental to adequately represent geographical phenomena of different dimensions, such as points, lines, and regions (i.e., polygons) [Güting 1994]. Regions might also be specialized to other types, such as rectangles. Further, spatial data types can be simple or complex; simple spatial data types provide only single-component objects, while complex spatial data types provide versatile spatial objects with finitely many components. Hadoop-GIS, SpatialSpark, GeoSpark, and Elcano provide support for simple and complex spatial data types. The remaining DSDMSs limit their scope. For instance, SIMBA and LocationSpark provide support for simple points and rectangles only. SpatialHadoop, however, is an exception. Despite its

[5] `https://locationtech.github.io/jts/`

limited support for spatial data types, it can be extended, allowing users to define their own geometry data types. Regarding SparkGIS, it may provide features to manipulate complex spatial data types, but this cannot be affirmed since its implementation is not available yet.

Representation of spatial objects. Spatial objects can assume distinct representations that are used for different purposes, such as interoperability between applications, loading of spatial objects into DSDMSs, and visualization of spatial objects. The OGC standard specifies several representations [OGC 2018]. For instance, GML and GeoJSON are textual representations of spatial objects useful for visualizing objects in web-based applications. Another example is WKB, a binary representation that is useful to make fast data transferring between applications. Only GeoSpark supports textual and binary representations. Regarding LocationSpark, it does not provide encapsulated functions capable of processing any type of representation, burdening the user to implement these functions.

Geometric set operations. This type of spatial operation calculates the geometric intersection, geometric union, and geometric difference of two spatial objects [Güting 1994]. The result of these operations is another spatial object that can be used in further spatial analysis; thus, allowing users to take more flexible strategic decisions and enhancing the user experience. With the exception of the DSDMSs that employ the JTS library, geometric set operations are not fully supported by Hadoop-GIS and SpatialHadoop, which include up to two operations, or are not supported at all by the others systems.

Topological predicates. Spatial queries commonly required in spatial applications often make use of topological predicates [Gaede and Günther 1998], such as overlap, contains, and intersects. Examples of typical spatial queries include: (i) spatial selections that return a set of spatial objects satisfying a topological predicate given a search object, (ii) range queries that yield all spatial objects satisfying a topological predicate given a rectangular-shaped object called query window, and (iii) spatial joins that combine different sets of spatial objects according to a topological predicate. From the *user-centric* view, it is important to a DSDMS to offer native support for topological predicates because users can define the specific queries required by their applications. Hadoop-GIS, SpatialSpark, GeoSpark, and Elcano satisfy this requirement. On the other hand, SpatialHadoop, SIMBA, and LocationSpark do not allow the specification of ad-hoc spatial queries with topological predicates. Instead, they only support a subgroup of optimized spatial queries with a predetermined set of predicates. For instance, range and spatial join queries in SpatialHadoop can only be executed with the predicate overlap.

Numerical operations. This type of operation returns numbers calculated from geometric properties of spatial objects [Güting 1994], such as the length of lines and the area of regions. Further, numerical operations can be employed to execute distance-based spatial queries, such as the k-nearest neighbor query that returns a set of k spatial objects nearest to an origin location. For instance, distance-based spatial queries are useful in spatial neighbourhood analysis. All DSDMSs provide at least some support for numerical operations. In SpatialHadoop, LocationSpark, and SparkGIS, the numerical operations are performed only within encapsulated functions that execute spatial queries. However, SparkGIS may provide other features to manipulate numerical operations, but this cannot be affirmed since its implementation is not available yet. Finally, regarding numerical op-

Table 2. A *user-centric* view of DSDMSs, considering the aspects inherent to distributed systems.

DSDMS	Underlying Technology	Doc. Completeness	Spatial Partitioning	Distributed Indexing	Query Language	Visualization Module
Hadoop-GIS	Hadoop-based	✓	✓	✓	HiveSQL-based	limited
SpatialHadoop	Hadoop-based	✓	✓	✓	Pigeon-based	for simple spatial data types only
SpatialSpark	Spark-based	limited	✓	local indices only	no	no
GeoSpark	Spark-based	✓	✓	local indices only	SQL-based	for simple spatial data types only
SIMBA	Spark-based	limited	✓	✓	SQL-based	no
LocationSpark	Spark-based	limited	✓	✓	no	no
SparkGIS	Spark-based	unavailable	✓	✓	no	limited
Elcano	Spark-based	unavailable	not specified	not specified	SQL-based	no

erations that extract geometric characteristics, only SpatialSpark, GeoSpark, and Elcano provide this type of support since they are based on the JTS library.

Spatial indexing techniques. The use of a spatial index is one of the most common techniques employed to accelerate spatial query processing [Gaede and Günther 1998]. Many different spatial indices have been proposed in the literature, such as the R-trees and the Quadtrees. In general, DSDMSs employ spatial indices for two main purposes: (i) to process spatial queries in slave nodes; and (ii) to distribute data among slave nodes and possibly reduce the number of partitions visited during a spatial query (Section 3.2). Because of these advantages, spatial indices are supported by all compared DSDMSs.

3.2. Support for Aspects Inherent to Distributed Systems

Table 2 compares the DSDMSs considering the following aspects inherent to distributed systems: underlying technology, documentation completeness, spatial partitioning, distributed indexing, and query language. A *checkmark icon* is employed if a DSDMS completely fulfills a corresponding criterion. A *not specified* expression indicates that no information is provided in the research paper that introduces the DSDMS, and the criterion was not further analyzed because the implementation of the system is not available yet. We discuss the motivation and the context behind each criterion as follows.

Underlying technology. The underlying technology used to implement a DSDMS impacts on the performance of spatial applications because of the I/O cost. Hadoop-based systems need to write intermediate data to disk, while Spark-based systems store intermediate data in the main memory through the use of RDDs. Thus, according to Pandey et al. (2018), Spark-based systems usually deliver better performance.

Documentation completeness. A complete, accurate, and up-to-date documentation is a required prerequisite to help users who design, develop, and implement spatial applications, especially when these users are dealing with state-of-the-art systems. The lack of documentation may impact negatively in the application development, requiring extra time to develop the application and burdening the users. Hadoop-GIS, SpatialHadoop, and GeoSpark stand out since these systems provide a dedicated website with expressive content, such as list of features, detailing of installation procedures, and several tutorials describing how to execute spatial operations. The available documentations of SpatialSpark, SIMBA, and LocationSpark provide only a short description and a quick example of some of their operations. SparkGIS and Elcano are exceptions. They do not have a public version released yet, and therefore do not provide documentation.

Spatial partitioning. The partitioning of data across cluster nodes is a technique frequently used in parallel and distributed computing. This technique usually speed up query processing by taking advantage of data locality. By using spatial partitioning techniques, a DSDMS is able to use the spatial characteristics of the dataset as the criterion for the partitioning, improving the performance of spatial queries. Several DSDMSs employ spatial partitioning techniques, with Hadoop-GIS, SpatialHadoop, SIMBA, and SparkGIS providing the most expressive quantity of algorithms available.

Distributed indexing. The concept of spatial indexing in a parallel and distributed environment is strongly related to spatial partitioning. Commonly, two data structures are created: (i) a global index, located on the master node, pointing to each data partition, and (ii) multiple local indices, each located on a data partition, pointing to the data inside the partition. The global index, which is created with the same data structure used for spatial partitioning, is applied before executing a spatial query in order to prune unnecessary partitions. This global index, however, is not employed by SpatialSpark and GeoSpark, which include only local indices. On the other hand, global and local indices are employed by Hadoop-GIS, SpatialHadoop, SIMBA, LocationSpark, and SparkGIS, introducing several benefits as discussed in Section 3.1.

Query language. Extending existing query languages is an essential characteristic that a DSDMS should provide to simplify the manipulation of spatial objects. DSDMSs that extend well-known query languages, such as SQL, usually reduce the learning curve of users. That is, they enable users to quickly familiarize themselves with the features of the system. GeoSpark, SIMBA, and Elcano distinguish themselves because they extend SparkSQL to support the execution of spatial queries. Hadoop-GIS and SpatialHadoop also extend existing query languages, but those are based on other standards (HiveSQL and Pigeon, respectively). The remainder DSDMSs do not offer query language, requiring extra efforts from users to retrieve and manage spatial objects.

Visualization module. Providing a module for visualizing results of spatial queries is an important aspect that enhances the user experience in spatial applications. For instance, visualizing the output of a spatial query in a map instead of in a text file enables users to visible interpret its content and to take quicker decisions over the identified problems. To provide a precise and complete map visualization of spatial objects, the generation of high-resolution maps is needed. Currently, only SpatialHadoop and GeoSpark fully support this feature. However, this support is restricted to simple spatial data types. On the other hand, Hadoop-GIS only provides a simple tool that allows users to visualize the boundaries of partitions. Regarding SparkGIS, its support for visualizing spatial objects depends on the implementation of plugins that extend its functionalities.

4. User-centric Guidelines

In this section, we propose a set of *user-centric* guidelines to help users to identify the most appropriate DSDMSs to their needs, according to the analyses described in Section 3. Because the context and objectives behind the development of applications are very variable, we provide a small but general characterization that can be later specialized based on the requirements of each application. Thus, we propose guidelines defined on the *focus of spatial applications*. Each guideline also lists the DSDMSs that meet its specification. Non-listed DSDMSs may offer some related functionalities, but should require extra implementation efforts from users.

Guideline 1. Focus on executing ad-hoc spatial queries. This guideline is based on spatial applications that need to process spatial queries without specific formats. An example is an application for analyzing heterogeneous and distributed spatial data for environmental monitoring [Wiemann et al. 2018]. To fulfill Guideline 1, a DSDMS should provide support for an expressive variety of spatial operations, such as geometric set operations, topological predicates, and numerical operations. Based on our analyses, the following DSDMSs fulfill Guideline 1: SpatialSpark, GeoSpark, and Elcano.

Guideline 2. Focus on the interoperability among different systems. This guideline is based on spatial applications that need to communicate with each other, such as those that integrate heterogeneous spatial data from different sources. For instance, the integration of public and private urban transportation spatial data [Smarzaro et al. 2017]. To fulfill Guideline 2, a DSDMS should provide support for all spatial data types since the representation of spatial phenomena can be different in the sources. Further, spatial objects should be exchangeable by using textual or binary representations. Based on our analyses, GeoSpark fulfills Guideline 2. Hadoop-GIS, SpatialSpark, and Elcano partially fulfill this guideline because they provide support for textual representations only.

Guideline 3. Focus on characteristics based on well-known standards. This guideline is based on spatial applications that require the use of well-known and accepted concepts, such as terms and expressions employed in the literature, query languages, and representations of spatial objects. An example is the application of some open standards for homeland security networks [Lee and Reichardt 2005]. Another example is the web-based application detailed in Wiemann et al. (2018). To fulfill Guideline 3, a DSDMS should employ well-known and accepted concepts in its design. It also should follow Guideline 2. Based on our analyses, GeoSpark fulfills Guideline 3. SpatialSpark, SIMBA, and Elcano also consider several aspects of this guideline, but introduce limitations related to the lack of a binary representation of spatial objects based on well-known standards.

Guideline 4. Focus on spatial data visualization. Spatial applications usually require the use of graphical user interfaces through which users are able to manipulate spatial objects, share findings by plotting spatial objects in maps, and enrich their decision-making. For instance, applications analyzing traffic data require the visualization of spatial data to better understand transportation systems [Chen et al. 2015]. To fulfill Guideline 4, a DSDMS should provide visualization modules without restricting the type of spatial data being manipulated. To the best of our knowledge, there is no compared DSDMS that completely fulfills this guideline. Hadoop-GIS, SpatialHadoop, GeoSpark, and SparkGIS only provide a limited support for visualizing spatial objects.

Guideline 5. Focus on efficiently processing spatial queries. Reducing the time spent to process spatial queries is a common requirement of spatial applications. For instance, García-García et al. (2017) propose algorithms for optimizing distance joins queries. To fulfill Guideline 5, a DSDMS should include optimized- and specialized-algorithms for processing spatial queries, including the use of indices. Here, the findings about the performance evaluation of DSDMSs described in Pandey et al. (2018) should be used as foundation. Pandey et al. (2018) also provide the best parameters of these systems to process several types of spatial queries.

5. Case Study

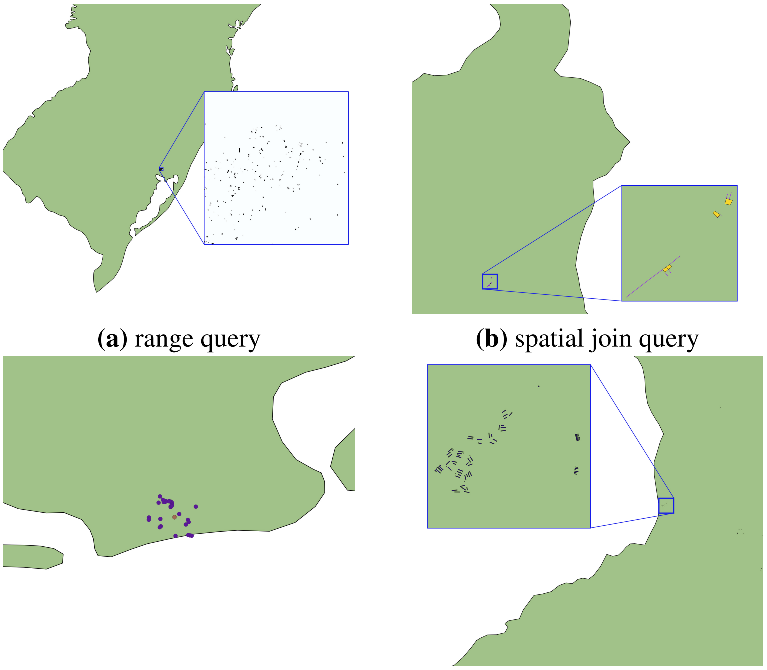
In this section, we describe a case study that illustrates the use of the proposed guidelines. We use a spatial application containing real spatial objects extracted from OpenStreetMaps that correspond to buildings, highways, and single locations of Brazil [Carniel et al. 2017]. They are represented by regions, lines, and points respectively. This application handles these objects in spatial queries that analyze the infrastructure situation of different places, such as farms, schools, and roads. Users who design, develop, and implement this spatial application should consider the following requirements to decide which DSDMS to choose. The application should manage spatial objects represented by simple and complex data types. The spatial objects are stored in CSV files. Each line of these files has the format $(geo, desc)$, where *geo* is the WKT representation of the spatial object and *desc* is its description. The application should also support ad-hoc spatial queries possibly containing geometric set operations, topological predicates, and numerical operations. Further, the application should provide good performance results. Finally, it is important to note that users have some previous knowledge of SQL.

The application's requirements indicate that users should take into account Guidelines 1, 3, and 5 (Section 4). As a result, users choose GeoSpark as the most appropriate DSDMS since it fulfills the requirements. Regarding Guideline 5, users should apply the same values of parameters as described in Pandey et al. (2018) to guarantee the best elapsed times for spatial queries. These parameters are: (i) the R-tree as the local index, and (ii) the Quadtree as the spatial partitioning technique.

Data loading. First, spatial objects from the CSV files are loaded into GeoSpark by using GeoSparkSQL. Because this module is an extension of SparkSQL, the files are loaded into a structure called DataFrame, which resembles a relational table. Next, the column that stores the textual representation of the spatial data should be transformed into a geometry column. GeoSparkSQL provides a function capable of transforming a WKT representation into a spatial object, which can be manipulated in SQL queries (Guideline 4). The following query is an example of how the column that contains the WKT representation of the regions stored in $brazil\_buildings$ can be transformed into a geometry column:

```
SELECT ST_GeomFromWKT(brazil_buildings.geo) AS geo,
       brazil_buildings.desc AS desc
FROM brazil_buildings
```

After loading all spatial objects, users can execute ad-hoc spatial queries on the DataFrame (Guideline 1). We define four ad-hoc spatial queries as samples for our application. These queries are described as follows.



(a) range query **(b)** spatial join query

(c) k-nearest neighbours query **(d)** geometric set manipulation query

Figure 1. Visualization of the query results.

Range query. The first query returns all schools located inside a query window (QW), which is a rectangle that corresponds to 0.001% of the total extent of Brazil. The results of this query enable users to analyze the school coverage in a given range. The command that express this query employs the topological predicate *contains* as follows:

```
SELECT brazil_buildings.geo
FROM brazil_buildings
WHERE ST_Contains(QW, brazil_buildings.geo)
      AND brazil_buildings.desc = 'school'
```

Spatial join query. The next query returns all tracks (i.e., rough road used by agricultural or similar vehicles) that intersect a building. This query is useful to analyze if tracks should be improved or not. The command that express this query employs the topological predicate *intersects* as follows:

```
SELECT brazil_buildings.geo, brazil_highways.geo
FROM brazil_buildings, brazil_highways
WHERE ST_Intersects(brazil_buildings.geo, brazil_highways.geo)
      AND brazil_highways.desc = 'track'
```

K-nearest neighbours query. The next query yields the 30 nearest energy towers from the Olympic Arena of Rio de Janeiro (PT) to analyze the infrastructure situation of an important neighborhood. To this end, the following distance-based query can be written:

```
SELECT brazil_points.geo, ST_Distance(brazil_points.geo, PT) as d
FROM brazil_points
```

```
WHERE brazil_points.desc = 'tower'
ORDER BY d
LIMIT 30
```

Geometric set manipulation query. The final query returns the total area of all farms in Brazil. Hence, we need to compute the geometric union among all farms and then calculate its area, as follows:

```
SELECT ST_Area(ST_Union_Aggr(brazil_buildings.geo))
FROM brazil_buildings
WHERE brazil_buildings.desc = 'farm'
```

Figure 1 depicts the spatial objects returned by these queries. Zooming was applied to better visualize portions of the result; thus, the result was not completely displayed for some queries. We use QGIS[6] to show the queries' results because the visualization module GeoSpark-Viz does not allow the visualization of complex spatial objects.

6. Conclusions and Future Work

In this paper, we provide a comparative analysis of the following up-to-date DSDMSs: Hadoop-GIS, SpatialHadoop, SpatialSpark, GeoSpark, SIMBA, LocationSpark, SparkGIS, and Elcano. Because the analysis is performed from the *user-centric* view, it is aimed to help users to understand how the characteristics of DSDMSs are useful to meet the specific requirements of their spatial applications. Based on the comparisons, we propose a set of guidelines to help users to choose an appropriate DSDMS to design, develop, and implement their spatial applications. Finally, we describe a case study using GeoSpark to illustrate the use of the guidelines. Future work includes the *user-centric* comparison of other DSDMSs such as Magellan and STARK. Another future work is to describe the case study using each surveyed DSDMS.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. This work has also been supported by CNPq and by the São Paulo Research Foundation (FAPESP). Anderson C. Carniel has been supported by the grant #2015/26687-8, FAPESP. Cristina D. A. Ciferri has been supported by the grant #2018/22277-8, FAPESP.

References

Aji, A., Wang, F., Vo, H., Lee, R., Liu, Q., Zhang, X., and Saltz, J. H. (2013). Hadoop-GIS: A high performance spatial data warehousing system over MapReduce. *VLDB Endowment*, 6(11):1009–1020.

Baig, F., Vo, H., Kurç, T. M., Saltz, J. H., and Wang, F. (2017). SparkGIS: Resource aware efficient in-memory spatial query processing. In *ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*, pages 28:1–28:10.

Carniel, A. C., Ciferri, R. R., and Ciferri, C. D. A. (2017). Spatial datasets for conducting experimental evaluations of spatial indices. In *Satellite Events of the Brazilian Symposium on Databases - Dataset Showcase Workshop*, pages 286–295.

[6] http://qgis.osgeo.org

Chen, W., Guo, F., and Wang, F. (2015). A survey of traffic data visualization. *IEEE Trans. on Intelligent Transportation Systems*, 16(6):2970–2984.

Eldawy, A. and Mokbel, M. F. (2015). SpatialHadoop: A MapReduce framework for spatial data. In *Int. Conf. on Data Engineering*, pages 1352–1363.

Engélinus, J. and Badard, T. (2018). Elcano: A geospatial big data processing system based on SparkSQL. In *Int. Conf. on Geographical Information Systems Theory, Applications and Management*, pages 119–128.

Gaede, V. and Günther, O. (1998). Multidimensional access methods. 30(2):170–231.

García-García, F., Corral, A., Iribarne, L., Mavrommatis, G., and Vassilakopoulos, M. (2017). A comparison of distributed spatial data management systems for processing distance join queries. In *European Conf. on Advances in Databases and Information Systems*, pages 214–228.

Güting, R. H. (1994). An introduction to spatial database systems. *The VLDB Journal*, 3(4):357–399.

Hagedorn, S., Götze, P., and Sattler, K. (2017). Big spatial data processing frameworks: Feature and performance evaluation. In *Int. Conf. on Extending Database Technology*, pages 490–493.

Lee, K. B. and Reichardt, M. E. (2005). Open standards for homeland security sensor networks. *IEEE Instrumentation Measurement Magazine*, 8(5):14–21.

OGC (2018). OpenGIS® Implementation Standard for Geographic Information - Simple Feature Access - Part 1: Common Architecture. Open Geospatial Consortium. Available at: `http://www.opengeospatial.org/standards/sfa`.

Pandey, V., Kipf, A., Neumann, T., and Kemper, A. (2018). How good are modern spatial analytics systems? *VLDB Endowment*, 11(11):1661–1673.

Smarzaro, R., Lima, T. F. M., and Davis, Jr., C. A. (2017). Could data from location-based social networks be used to support urban planning? In *Int. Conf. on World Wide Web Companion*, pages 1463–1468.

Tang, M., Yu, Y., Malluhi, Q. M., Ouzzani, M., and Aref, W. G. (2016). LocationSpark: A distributed in-memory data management system for big spatial data. *VLDB Endowment*, 9(13):1565–1568.

Wiemann, S., Karrasch, P., and Bernard, L. (2018). Ad-hoc combination and analysis of heterogeneous and distributed spatial data for environmental monitoring - design and prototype of a web-based solution. *International Journal Digital Earth*, 11(1):79–94.

Xie, D., Li, F., Yao, B., Li, G., Zhou, L., and Guo, M. (2016). Simba: Efficient in-memory spatial analytics. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 1071–1085.

You, S., Zhang, J., and Gruenwald, L. (2015). Large-scale spatial join query processing in cloud. In *Int. Conf. on Data Engineering Workshops*, pages 34–41.

Yu, J., Wu, J., and Sarwat, M. (2015). GeoSpark: a cluster computing framework for processing large-scale spatial data. In *ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*, pages 70:1–70:4.

Environmental Vulnerability of the Environmental Protection Area of the Mamanguape River Bar - PB

Iara dos S. Medeiros[1], Hugo Y. E. G. de Assis[1], Mayara de S. Dantas[1], Tiago S. Clemente[1], Nadjacleia V. Almeida[1]

iaraeco2015@gmail.com; hugo.ecologia@gmail.com; mayaradantas-@outlook.com;tiagoclemente288@gmail.com; nadjageo@gmail.com

[1]Laboratório de Cartografia e Geoprocessamento – Universidade Federal da Paraíba (CCAE/UFPB). Av. Santa Elisabete, s/n, Centro. Rio Tinto-PB, Brasil. CEP 58297-000

Abstract. *The fragility of the environment is the join of the vulnerabilities of each component present in nature. The present work aimed to identify the environmental fragility of the Environmental Protection Area (EPA) of the Mamanguape River Barrier, in order to identify the vulnerability present in the area. The research used fundamentals of the geosystemic method, observing the interconnections of the parameters geology, geomorphology, pedology, rainfall and land use and coverage. The EPA was classified as low fragility (1%) and medium fragility (69%). The remaining of the EPA was classified as high fragility (30%). The diagnosis of the environmental fragility of the EPA showed that only geomorphology is stable, but geology, pedology and land use and cover are unstable.*

1. Introduction

In an intrinsic way the environment has a potential (natural) fragility, that is, it presents vulnerable aspects in the composition of the elements that constitute it. Thus, the fragility of the environment is the junction of the vulnerabilities of each component present in nature.

The theory of eco-dynamics proposed by Tricart (1977), studies the dynamics of the environment based on its morphogenetic (relief) and pedogenetic characteristics (soil modification). Crepani et al (2001, p.83) point out that "a natural landscape unit is considered stable when the natural events that occur in it favor the processes of pedogenesis".

The elements present in the landscape of the Environmental Protection Area (EPA) of Mamanguape River Barrier are: mangroves and floodplains, coastal reefs, Atlantic forest, forest of restinga, dunes, and cliffs. Although the EPA is a conservation unit of sustainable use that seeks to reconcile nature conservation with the sustainable use of natural resources, allowing activities involving the use of natural resources, practiced in a way that the permanence of renewable environmental resources and ecological processes is assured (MMA, 2015), environments are changing. External factors such as human activities can intensify the fragility of the environments, making them even more susceptible to intense changes that can cause irreversible damage.

In view of the importance of EPA ecosystems, the present work aimed to identify the environmental fragility of the EPA of the Mamanguape River Barrier, in order to identify the vulnerability present in the area.

These variations are reflected in the landscape that is constantly altered by the most diverse factors, be they positive or negative, that influence directly or indirectly on any system, which, even in the face of perturbations, tends to remain in a dynamic equilibrium. These characteristics can be spatialized through geotechnologies such as Remote Sensing, Geographic Information System (GIS), Global Positioning System (GPS) and Geoprocessing, which aid in the analysis of the data, allowing an integration of the elements present in the landscape.

2.Material and Methods

2.1Area of study

The Environmental Protection Area of the Mamanguape River Barrier was implemented by Decree 924 of September 10, 1993. The EPA has an area of 14,640 ha and is located in the mesoregion of the forest zone, north coast of the State of Paraíba, (Figure 1).

EPA is home to the main mangrove remnants of northeastern Brazil and has natural habitats that house endemic and endangered species. The EPA is also represented by floodplains, coastal reefs, Atlantic forest, restinga forest, dunes and cliffs (BRAZIL 1993, ICMBIO 2015, EMBRAPA 2008).

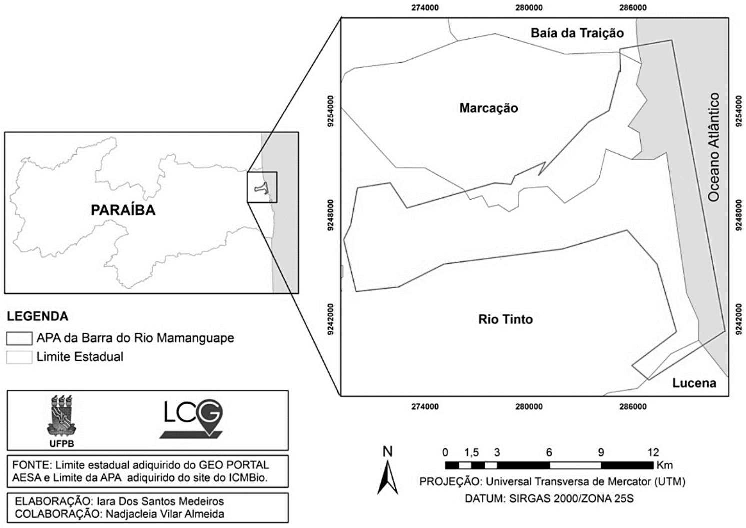

274000
280000
286000
Baía da Traição
Marcação
Rio Tinto
Lucena
Oceano Atlântico
9254000
9248000
9242000
PARAÍBA
LEGENDA
APA da Barra do Rio Mamanguape
Limite Estadual
UFPB
LCG
FONTE: Limite estadual adiquirido do GEO PORTAL AESA e Limite da APA adiquirido do site do ICMBio.
ELABORAÇÃO: Iara Dos Santos Medeiros
COLABORAÇÃO: Nadjacleia Vilar Almeida
N
0 1,5 3 6 9 12
Km
PROJEÇÃO: Universal Transversa de Mercator (UTM)
DATUM: SIRGAS 2000/ZONA 25S

Figure 1. Location map of the APA of Barra do Rio Mamanguape, Paraíba, Brazil.

2.2 Methodological Procedures

The research used fundamentals of the geosystemic method, observing the interconnections of the parameters geology, geomorphology, pedology, rainfall, and land use and coverage. In the methodological process, the geoenvironmental and eco-dynamic analyzes were carried out where the procedures outlined in figure 2 were applied, which were performed using the arcgis software version 10.6.1.

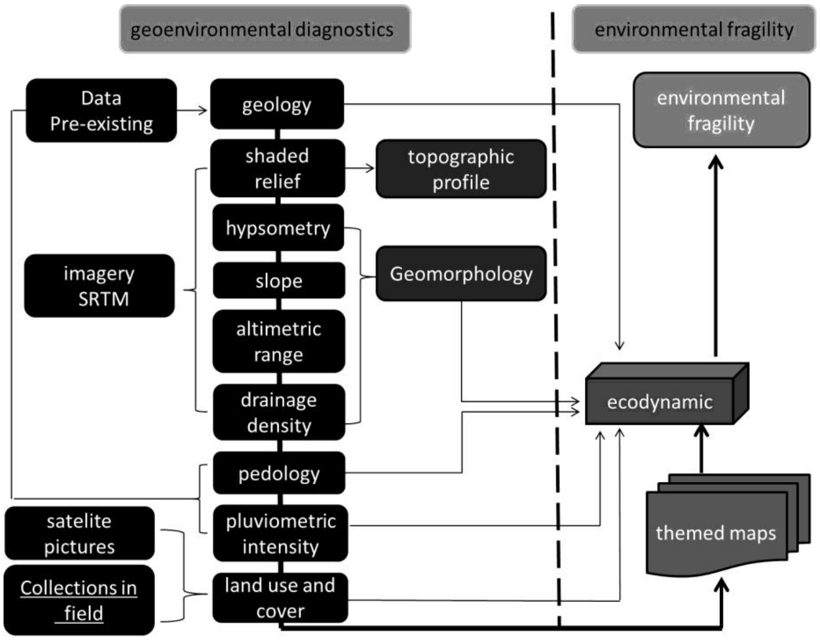


Figure 2. methodological procedures

For the geoenvironmental diagnosis, a vector file containing EPA geological information made available by the Geological Survey of Brazil (CPRM) was used through the GEOBANK1 website.

The geomorphological characteristics were analyzed through three cartographic products: slope, drainage density, and altimetric amplitude, all of which were generated based on SRTM (Radar Topography Mission) with the spatial resolution of 30 meters, referring to SB-25- YA provided by the TOPODATA2 website which contains the morphometric data of Brazil. The slope was established through the slope tool being expressed in percentage values according to the classification of Embrapa (1979). The altimetric amplitude based on Crepani et al. (2001) and drainage density according to Christofoletti (1980, p. 115).

The value of the rainfall intensity was obtained by dividing the average annual rainfall value (in mm) by the duration of the rainy season (in months). The spreadsheet with the data was exported to the GIS and through the interpolation tool, the data were converted to pixels generating the EPA pluviometric intensity map. The method of interpolation chosen was the IDW-Inverse of the Weighted Distance because it represents better the results for the study area and because it is a widely used method. According to Miranda (2005), this method estimates a value for a non-sampled location as an average of the data values within a neighborhood.

The soil and physical aspects of soil formation were considered for pedological characterization. The soil map of the State of Paraíba, dated 1997, with a scale of 1: 1,200,000, was used as the Cartographic basis.

For the analysis of land use and land cover, a LANDSAT 8 image with a spatial resolution of 30x30m dated 11/02/2016, made available by the United States Geological Survey (USGS) 3 was used. The steps of radiometric conversion, generation of false color composition (RGB), fusion of the multispectral image to color Panchromatic in order to convert the spatial resolution of the used satellite image, which initially was of 30x30, resulted in an image of 15x15 meters, ending with supervised classification using the maximum likelihood method, where it is considered that "objects belonging to the same class will present spectral responses close to the average values for that class" (RIBEIRO et al., 2007).

Based on the theory of eco-dynamics proposed by Tricart (1977) and adapted by Crepani et al. (2001) all components were analyzed separately. Through reclassification procedures, vulnerability values were assigned to each theme.

Tabela 1. Categorias ecodinâmicas e seus respectivos valores de vulnerabilidade

Ecodynamic Categories	**Relationship between pedogenesis and morphogenesis**	**Vulnerability Values**
Stable	Pedogenesis Prevails	1,0 – 1,3
Moderately Stable		1,4 – 1,7
Average Stability	Pedogenesis / Morphogenesis Balance	1,8 – 2,2
Pedogenesis / Morphogenesis Balance		1,8 – 2,2
Unstable	Morphogenesis Prevails	2,7 – 3,0

Source: adapted from Almeida (2012).

After applying the eco-dynamics for each theme individually, through equation 1, the general fragility of EPA was obtained.

$$\boldsymbol{v} = \frac{(G+R+S+VG+C)}{5}$$ Equation 1.

At where:

V = Vulnerability; G = vulnerability to the theme Geology; R = vulnerability to the theme Geomorphology; S = vulnerability to the theme Solos; Vg = vulnerability to vegetation; C = Climate Vulnerability

Thus, the units with the highest stability are represented by values closer to 1.0, intermediate values by values around 2.0 and the most vulnerable units present values closer to 3.0 (CREPANI, 2001).

3.Results and Discussion

The fragility diagnosis consists in the classification of the studied environments in degrees of vulnerability and is based on the information obtained from the geoenvironmental diagnosis, being of primary importance to guide the use and occupation of the land and to identify the activities that cause negative impacts in the EPA.

According to the geological diagnosis, the EPA is in the domain of the Pernambuco-Paraíba Basin (Brazil, 2002). The study area has three lithologic units: colluvium-eluvial reservoirs and fluvial-marine and barriers group. The colluvium-eluvial deposits and the fluvial-marine deposits are classified as unstable because they are unconsolidated sediments and because they are in constant modification are easily removed and deposited by means of chemical, physical and biological weathering. The Barriers Group presents as moderately unstable because it is composed of siltstones and argillites, which, although fragile, are little more resistant than the fluvial-marine and colluvial-eluvial deposits.

When analyzing the geological fragility of the Tambaba EPA on the southern coast of the state of Paraíba composed by the Barriers Indiviso Group, marine deposits, and the continental deposits, Almeida (2012) found results similar to those presented here and classified the environment as geologically unstable.

According to Crepani et al. (2001, p.60), rocks considered unstable are "poorly cohesive, prevailing erosive processes, modifiers of relief forms (morphogenesis)". Thus, we can say that the geology of the EPA is fragile and susceptible to transformations that can directly influence all other elements of the landscape since the geological formations are the substrate where the whole environment develops.

The geomorphological analysis performed from the arithmetic mean of the morphometric indices, altimetric amplitude, slope, and drainage density confer stability and moderate stability to the studied area, presenting a greater stability when compared to the geology. It is a flat area with moderate dissection where pedogenesis prevails. But this stability is not static, and the most endangered geomorphological units are the dunes that suffer from vehicular traffic and the building of houses beyond the natural action of the wind and the cliffs that are suffering from the ravine process. According to

Medeiros et al (2018), morphometric indexes, whether high or low, are directly related to erosion.

The types of soil present in the EPA are Red-Yellow Argisol, Quartzarenic Neosols, and Flubic Neosols, most of them confer instability because they are poorly drained soils, possessing high Vulnerability and corroborate with the instability found in the APA geology since the soils are formed from the weathering of the geological substrate.

The Red-Yellow Argisol with medium stability has characteristics such as deep soil, very porous, strongly or strongly drained, which means that it has a medium degree of stability. In the case of the Quartzeneic Neosols, Mangrove Soils, and Flossic Neosols, both have similar characteristics, eg., they are poorly drained soils, to which instability is conferred. being classified as unstable.

A natural landscape unit is considered to be vulnerable when relief modifying processes prevail (morphogenesis) and, therefore, there is a predominance of erosion processes to the detriment of soil formation and development processes.

According to Bertoni, (2010), precipitation is the most important climatic element in the process of soil erosion, with the intensity being the precipitation factor determining erosion. After analyzing the pluviometric intensity data, the entire territory of the PA was classified as medium stability. According to Crepani et al. (2001, p. 95), "the greater the values of rainfall intensity the greater the rainfall erosivity and we can create a rainfall erosivity scale that represents the influence of climate on morphodynamic processes." Thus, we can affirm that the influence of rain on the erosive processes of EPA is medium. However, this influence can be aggravated if the soil is directly exposed to raindrops, with no vegetation present to intercept or cushion the erosive effects of rainfall.

For the use and land cover, six classes were identified: forest, water, mangrove, board vegetation, restinga, cultivation, and soil exposed. Two were classified as stable, the remainder ranging from moderately stable to unstable.

According to Crepani et.al (2001, p.88), "the density of the vegetation cover is of paramount importance to avoid morphogenetic processes, so the high coverage densities are close to 1.0". According to ICMBio (2014, 84), "the increase of sugarcane cultivation in the Mamanguape EPA has increased the degradation of the forest remnants of the Coastal Table and of the Atlantic Forest, generating discontinuous fragments, highly impacted by trails and paths along the woods. "

Pessoa (2016), when analyzing the vegetation of the EPA of the Mamanguape River Barrier, found that between 1974 and 2013 there was a dense vegetation loss of 54.3 km² corresponding to 36.43%. With the exception of the Oiteiro forest, all other fragments have a reduced area and, consequently, with little or no presence of core areas, these fragments are fragile from the ecological point of view, since the fact that they are small fragments and isolates prevent or hinder the permanence of some species of flora and mainly of fauna that needs bigger and interconnected areas so that there is a greater availability of habitats and food that facilitates the gene flow among the species.

Water: Although not a class of vegetation, it was inserted in the mapping because it is one of the main elements and with greater representativity in the CU. It was classified as stable because it represents a type of cover of the soil and for propitiating the maintenance of adjacent ecosystems such as mangrove.

The removal of the riparian forest that has been replaced by sugarcane is the main cause of the degradation of the rivers, since it generates silting of the rivers and migration of the springs causing them to have their flow reduced, this causes several consequences that reflect not only the hydrography of the EPA more in the species of fauna and flora associated to her. Thus, it was observed that these classes have an ecological fragility that minimizes their role in the ecosystem.

Mangrove: The mangrove vegetation is dense and has an arboreal stratum, so it receives the value of 1.7 being classified as moderately stable. Because it is a very specialized environment with fluvial-marine influence, type of soil, specific fauna and flora, it needs a balance that allows its full development. However, this balance is being threatened by human activities. With this, it can be stated that the mangrove may eventually lose its stability and become an unstable environment, as a consequence of the continuity and intensification of negative impacts.

Board Vegetation: It is an ecosystem constituted of two strata, one arboreal-shrub, and another herbaceous, drained and discontinuous, (Environmental Sensitivity Primer: Ecosystems of Rio Grande do Norte, 2016). Because it is a transition environment (ecotone), it is more fragile, because it develops in an environment such as soil, relief, and vegetation. Therefore it is classified as average stability receiving value of 2.2.

Restinga: According to Assis (2014) the EPA restingas are very susceptible to degradation due to the activities developed by the local population and tourists, mainly by the trampling and illegal transit of motor vehicles and where they concentrate houses built for summer, changing their features and negatively impacting the environment.

Cultivation: being a introduced vegetation with shrub stratus is classified as unstable. In the case of the EPA, as most of the crop is composed of sugarcane, this causes other problems, making the environment even more fragile.

According to Costa and Andrade (2012, p.10) "the sugar and ethanol industries through the use of agrochemicals cause contamination of soil, rivers, and aquifers, as well as harming human health, biodiversity and causing damage to one's own agriculture. "This means that in addition to being fragile, it contributes to making other fragile elements such as soil and rivers.

Soil Exposure: Much of the soil classified as exposed represents areas of sugarcane cultivation that was cut, leaving the soil without vegetation.

3.1 Environmental Fragilityof the EPA of the MamanguapeRiverBarrier

Only a small portion of the EPA, in the Oiteiro forest region, was classified as low fragility (0.53km2, 0%). In the areas where the mangrove is inserted, most of the rivers, the forest, the lower altitudes, and declivities were classified as average fragility occupying 79.62 km² and 69% of the UA. The rest of the EPA was classified as high fragility with 34.77 km² totaling 30% of the area that corresponds to the regions of soil without vegetation, more pronounced slopes and recent soils (figure 3).

According to Almeida (2012 p.70), "the vulnerability of geo-environments to erosive processes (predominance of morphogenesis) reflects geoenvironmental (potential) fragility ... which means that the more fragile the more vulnerable to erosion is the". Thus, through this diagnosis, it is possible to identify the environmental fragility of EPA.

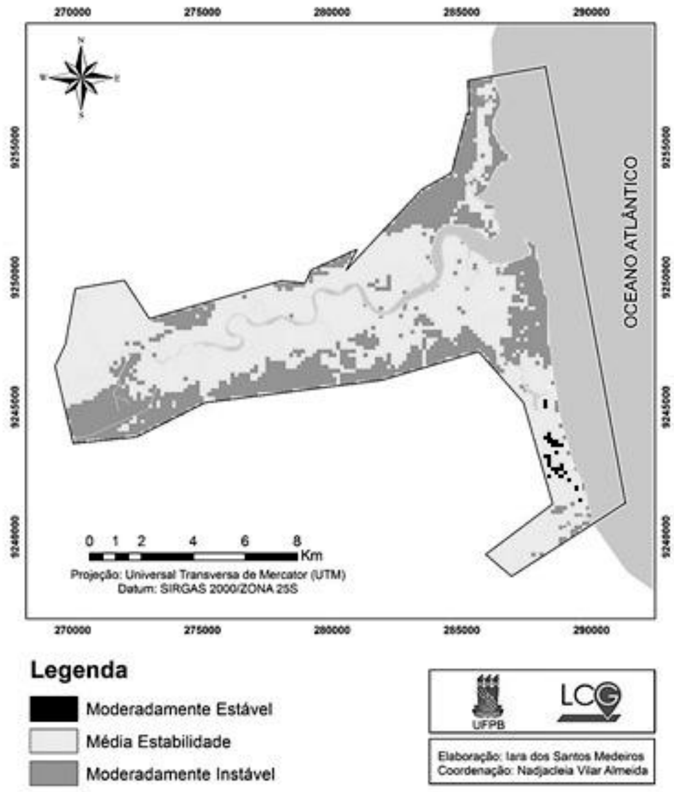


Figure 3. Vulnerability of the APA of Barra do Rio Mamanguape, Paraíba, Brazil.

There are some activities that cause direct negative impacts and may intensify the fragility of EPA. The following stand out:

Urbanization: The city of Rio Tinto, the indigenous villages of Jaraguá, Brejinho, Caieira, Camurupim, Tramataia, and Akaju-tibiró, and the communities of Aritingui, Barra de Mamanguape, Cravassu and Lagoa de Praia, Pacaré, Praia de Campina, Taberaba, Tanques, Tatupeba and Tavares (ICMBIO, 2014).

Agriculture: Silvestre et al (2011, p.30) say that the EPA is an "area surrounded by extensive cane fields. The great deforestation of the Atlantic Forest has motivated expansion of cane farms, resulting from the implementation of the Pró-alcohol program in 1970 by the Federal Government. "

Carciniculture: Carciniculture is present in a greater number of tanks in the northern portion of the EPA, of indigenous domain, which became an important source of local income, but the little planning carried out for the implantation caused that the tanks were abandoned in a short space and the replanting of the flora species is not carried out, leaving a huge void in the mangrove, fragmenting it. There is also a presence in the southern portion, but in smaller number, totalizing two farms, where one is not in operation due to IBAMA intervention, however, it was not reforested. This crop poses some risks to the local biota due to antibiotics and other chemicals harmful to the environment, besides, of course, the deforestation of the mangrove for the implantation of the tanks, (ASSIS 2014).

Deforestation: "The great deforestation of the Atlantic Forest in this area was motivated by the expansion of cane farms, resulting from the implementation of the pro-alcohol program in 1970 by the Federal Government," (ICMBIO 2014).

6. Conclusion

The diagnosis of the environmental fragility of EPA showed that only geomorphology is stable, but geology, pedology, and land use and cover are unstable. The rainfall intensity of the UC was only in the middle-class stability, with this it is observed that the landscapes that compose the EPA are in the limit between stability and instability, tending naturally to keep in balance, but the anthropic activities that act in the area, intensify and accelerate morphogenetic processes, making the environment more conducive to instability.

Therefore, a more targeted management is necessary, so that the fragile environments are restored and do not exceed their recovery threshold and the environments that are still stable can be preserved, thus ensuring the maintenance of the ecosystems present in the UC.

References

Almeida, N. V, (2012). Geoenvironmental Territorial Planning of the Taperoá River Basin / Semi-Arid Paraibano. Thesis (Doctorate). UFF / POSGEO. Niterói-RJ.

Assis, H. Y. E. G. (2014). Analysis of the APA landscape classes of Barra do Rio Mamanguape-PB. Monography (Undergraduate). UFPB / CCAE. Rio Tinto, PB.

Environmental Awareness Booklet: Rio Grande do Norte Ecosystems. Available at: http: //adcon.rn.gov.br/ACERVO/idema/DOC/DOC000000000007179.PDF. Accessed on: September 20, (2016).

Costa, I. M.; Andrade, M. O.; (2012) OVERVIEW OF BARA DE MAMANGUAPE AND TI POTIGUARA MONTE MOR-PB: analysis of environmental legislation and conflicts with the activities of the sugar and alcohol industry. Annals: III Brazilian Congress of Environmental Management Goiânia / GO.

Crepani, E.; Medeiros, J.S .; Hernandez filho, P .; Florenzano, T.G .; Duarte, V & Barbosa, C.C.F,(2001) Remote Sensing and Geoprocessing Applied to Ecological-Economic Zoning and Territorial Planning. São José dos Campos: INPE.

EMBRAPA-Research and Development Bulletin. Territorial Environmental Management in the Environmental Protection Area of Barra do Rio Mamanguape (PB),(2008).

ICMBio - Chico Mendes Institute for Biodiversity Conservation. Brasília, (2014). Management Plan of the Environmental Protection Area of the Mamanguape River and Area of Relevant Ecological Interest of Mangroves in the Foz do Rio Mamanguape.

ICMBIO - CHICO MENDES INSTITUTE OF CONSERVATION OF BIODIVERSITY. Available at: <http://www.icmbio.gov.br/portal/>. Accessed on: April, (2015).

MMA - MINISTRY OF THE ENVIRONMENT. Available at: <http://www.mma.gov.br/>. Accessed on: April, (2015).

Pessoa, A. F.(2016) Spatial-temporal dynamics of vegetation cover at the APA of Barra do Rio Mamanguape - PB.Monografia (Graduação). UFPB / CCAE. Rio Tinto - PB.

Rickfles, R. E. A. THE ECONOMY OF NATURE ed. 5th. Rio de Janeiro. Guanabara Koogan. (2003).503p.

Silvestre, L. C .; farias, D. L. S .; Lourenço, J. D. S .; barros, S. C. A .; braga, N. M. P,(2011) Diagnosis of Environmental Impacts Aware of Anthropogenic Activities at the APA of Barra do Rio Mamanguape. ENCYCLOPEDIA BIOSPHERE, Centro Científico Conhecer - Goiânia, vol.7, N.12 .

Townsend, C.R .; Begon, M .; Harper, J. L. translation of Leandro da silva Duarte. Fundamentals of Ecology. 3. Ed. Porto Alegre: Artmed,(2010) 576p.

Tricart, J.,(1977) Ecodynamics. Rio de Janeiro, IBGE, Technical Board, SURPREN. 97p.

Ucha, J.M .; hadlich, G.M .; celino, J. J.(2008) Apicum: transition between slope and mangrove soils. Revista E.T.C.

VGI Protocol and Web Service for Historical Data Management

Rodrigo M. Mariano[1], Karine R. Ferreira[1], Luis A. C. Ferla[2]

[1] National Institute for Space Research (INPE)
São José dos Campos – SP – Brazil

[2]Federal University of São Paulo (UNIFESP)
Guarulhos – SP – Brazil

{rodrigo.mariano,karine.ferreira}@inpe.br, ferla@unifesp.br

Abstract. *Volunteered Geographic Information (VGI) is a phenomenon that uses the web to produce, assemble and disseminate geographic information provided by volunteers. VGI techniques generate detailed geographical data with low cost, taking advantage of citizens local knowledge. The definition of a VGI protocol is crucial to improve the quality of citizen-derived geographical data sets collected by a project. Protocols are also important to facilitate the reuse of VGI data for other projects and applications different from what was originally collected. This paper presents a VGI protocol that was defined for the Pauliceia 2.0 project and a web service that was built based on this protocol. Pauliceia 2.0 project aims to use VGI and crowdsourcing techniques to produce historical geographical data sets of São Paulo city from 1870 to 1940.*

1. Introduction

VGI, citizen science, crowdsourcing and collaborative mapping are examples of different terms used to refer to the general subject of collaborative work and citizen-derived geographical information. See et al. [See et al. 2016] present a good review of these terms and categorize them according to three main aspects: (1) information or process; (2) active or passive contributions; and (3) spatial or non-spatial user-generated information.

The term VGI was first defined by Goodchild [Goodchild 2007] as *"the harnessing of tools to create, assemble, and disseminate geographic data provided voluntarily by individuals"*. Goodchild and Li [Goodchild and Li 2012] define VGI as a version of crowdsourcing, focused on manipulating geographical information. Estellés-Arolas and Guevara [Estellés-Arolas and González-Ladrón-de Guevara 2012] define crowdsourcing as *"a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit."*.

There are many projects that use VGI techniques to collect data, such as OpenStreetMap, Wikimapia and Flickr. OpenStreetMap is the most well-known general platform that implements VGI successfully [Goodchild and Li 2012]. It is an editable map of the world, provided by volunteers, being possible to handle free geographic data [OpenStreetMap 2017a]. It adopts the local expertise of the users to make updated maps.

Mapping agencies use robust protocols that drive the geographic data collection, while VGI projects regularly contain lack of standards or just supply vague instructions. The definition of a VGI protocol is crucial to improve the quality of citizen-derived geographical data sets collected by a project and to facilitate the reuse of these data sets for other projects and applications [Mooney et al. 2016].

Mooney et al. [Mooney et al. 2016] propose a generic protocol to drive VGI projects. This protocol establishes crucial issues that must be defined in the context of a VGI project in order to improve the understanding of volunteers and so the quality of the data produced by them. These issues include vector geographical data collection and management, user control, self-assessment and quality metrics and feedback to the community.

Pauliceia 2.0 project aims to build a computational platform for collaborative historical research based on VGI and crowdsourcing techniques [Ferreira et al. 2017]. Through this platform, citizens can contribute to produce historical geographical information of São Paulo city from 1870 to 1940. These contributions can be done in different ways, for example, by doing the vectorization of streets and buildings from historical maps or by uploading photos and information about historical places. Besides that, this platform allows historians to share data sets resulting from their researches.

This paper presents a VGI protocol for historical data that was defined for the Pauliceia 2.0 project and a web service based on this protocol, called VGI Management Web Service (VGIMWS), that was built in the Pauliceia platform. Ferreira et al. [Ferreira et al. 2017] introduce the Pauliceia project and its platform generally; while this paper presents a detailed description of the VGI protocol for historical data and VGIMWS service.

2. Related Work

This section presents projects that also use VGI and crowdsourcing techniques to produce historical geographical information, similar to Pauliceia 2.0 project.

OpenHistoricalMap [OpenStreetMap 2018] and HistOSM[1] projects are built on the OpenStreetMap (OSM) platform. OpenHistoricalMap is an effort to use the OSM infrastructure to produce a universal and historical map of the world. HistOSM is a web application to explore the historical objects of OSM, such as castles, ruins, monuments and memorials.

Building Inspector[2] is a web-based platform that allows citizens to produce, correct and analyze data from historical maps of New York city from 1853 to 1930. In this project, Budig et al. [Budig et al. 2016] propose a consensus polygon algorithm to extract a single polygon to represent each building from all polygons provided voluntarily.

ATLMaps[3] is an web portal of the Atlanta Explorer project that handles historical information of Atlanta city for post Civil War to 1940 [Page et al. 2013]. This portal

[1] http://histosm.org/

[2] http://buildinginspector.nypl.org/

[3] https://atlmaps.org/

allows user to visualize and explore historical maps, events and places. Users can produce their own projects using the layers that are ready to use in the portal and contribute with audios, annotations or images related to them.

Many VGI projects use the OSM infrastructure and its API (Application Programming Interface) to build their web platforms. In the beginning of the Pauliceia 2.0 project, our team evaluated if it was possible to build the Pauliceia 2.0 platform using the OSM infrastructure and its API. However, after some studies, we concluded that the OSM data model and operation are not suitable for the Pauliceia project due to the following reasons:

1. In the Pauliceia platform, historians can share data sets resulting from their researches and these data sets can not be edited by anyone. In OSM, data can be updated by anyone.
2. The historical features in the Pauliceia database are spatiotemporal, that is, they have a period to indicate when they existed. There are features that do not exist today anymore. In OSM database, the entities are not spatiotemporal. OSM considers that all entities stored in its database exist today.
3. In the Pauliceia platform, the historical data sets are organized in layers, while OSM data sets are not.
4. The community and domain of the Pauliceia project are very specific and structured, while the OSM is very general. The Pauliceia project has a specific domain with a particular spatial and temporal scope, generating a structured community.

Therefore, we decided not to use the OSM infrastructure and API. We defined a specific VGI protocol for the Pauliceia project and built a web service for VGI data management based on this protocol. These protocol and web service are crucial parts of the Pauliceia platform and they are described in the next sections.

3. VGI Protocol for Historical Data

Mooney et al. [Mooney et al. 2016] propose a generic protocol that organizes the issues related to citizen-derived geographical data management in five main stages, that are shown in Figure 1: (1) Initialisation; (2) Vector data collection; (3) Self-assessment and quality control; (4) Data submission; and (5) Feedback to the community.

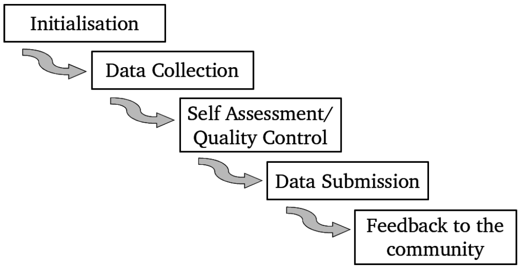


Figure 1. Main stages of VGI Protocol [Mooney et al. 2016]

A VGI protocol defines crucial issues that improve the understanding of volunteers about the project and all its mechanisms and methods to collect, manage and assess the quality of the citizen-derived geographical data. Thus, this helps to improve the quality of the data sets collected by volunteers in a project.

In this work, we define a VGI protocol specific for the Pauliceia 2.0 project following the guide proposed by Mooney et al. [Mooney et al. 2016]. The Pauliceia 2.0 VGI protocol is described in next sections.

3.1. Data types and initialization

In the Pauliceia 2.0 project, volunteers can upload and edit vector geographical data. The Pauliceia platform provides tools that allow citizens to include and edit geometries, such as points, lines and polygons, as well as textual and numerical values associated to geographical entities or features. The platform does not provide tools to edit and create raster data types.

One of the project goals is to use VGI and crowdsourcing techniques for the vectorization of features, such as streets and buildings, from historical raster maps. In this case, the data set gathered by volunteers can have a set of distinct geometries to represent the same feature. To extract the most accurate geometry to represent a single feature from this data set, we intend to employ methods that compute a single geometry that represents the majority opinion, as proposed by Budig et al. [Budig et al. 2016].

The users access the Pauliceia 2.0 platform through an online browser. Before starting the contribution, the collaborator needs to register himself/herself to the platform and accept a ”Use Term” that describes mainly that the portal is not responsible for the collected data sets and that these data sets are public. The registration can be done by creating a new user or by using a social login through Google and Facebook accounts. Everybody can access the platform and visualise its data sets freely, however just registered volunteers can edit or add new historical data sets.

The Pauliceia 2.0 database is made available under the Creative Commons Attribution-ShareAlike 4.0 license (CC BY-SA)[4]. In a nutshell, this license authorises the people freely copy, share, modify and use the data for any propose, since the users cite Pauliceia 2.0 and its contributors. If the user reproduces the data, he or she must use the same license for the results.

To motivate volunteers to vectorize streets and buildings as well as historians to share their historical data sets, we intend to organize events oriented to this purpose, following the same idea of the mapathon events promoted by Google Maps [Tech2 2014] and OpenStreetMap [OpenStreetMap 2017b]. These events, called HistMapathon (Historic Mapping Marathon), will be organized in universities with historians and their students to promote the mass contribution of geographical data in the Pauliceia 2.0 platform.

3.2. Data model

Figure 2 shows the concepts of the Pauliceia 2.0 VGI protocol and their relationships, using an entity-relationship diagram. Its main concepts are: user, layer, reference, keyword and notification.

In the platform, the data sets are organized in layers as in GIS (Geographic Information System). A layer groups geographical features related to a subject that are described by the same set of properties. Each feature has spatial and non-spatial properties [Herring 2006]. The spatial properties are represented by geometric data types, such as

[4] https://creativecommons.org/licenses/by-sa/4.0/

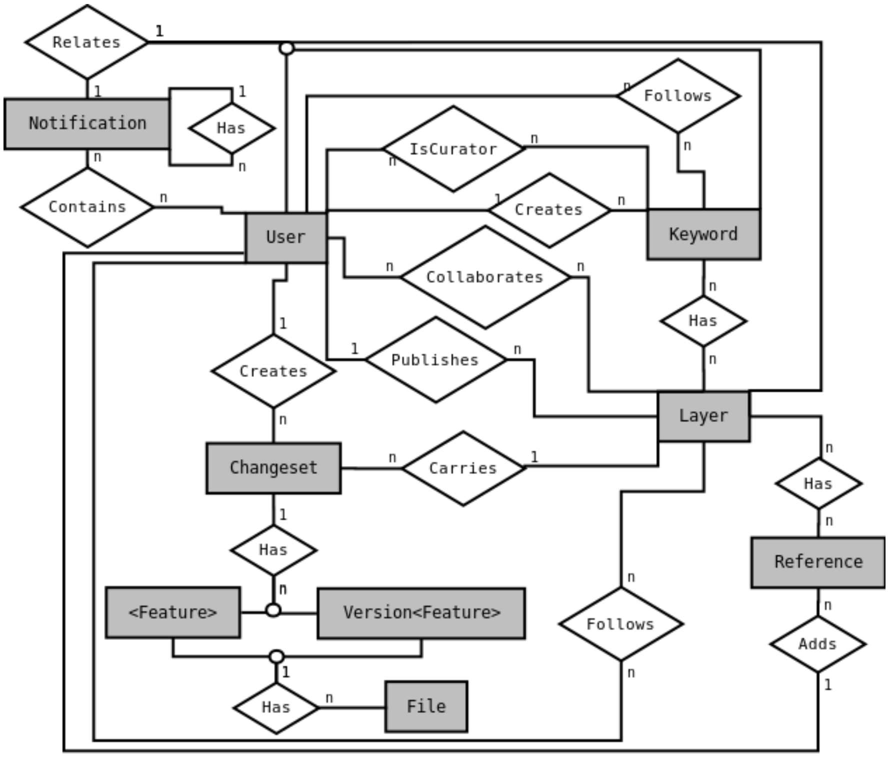


Figure 2. The Pauliceia 2.0 project data model

points, lines and geometries. The non-spatial properties are represented by alphanumerical data types, such as texts and numbers. A non-spatial property of a feature can contain links to media or documents, e.g photos and videos, that are stored in other repositories, like Google Drive, YouTube or Dropbox.

A layer contains a set of features and their versions along time. The changeset entity controls the features of a layer and their versions along time, keeping the history about when and what user realized each change. A changeset is a group of changes related to the features of a layer made by users in a period.

A layer can be associated to one or more keywords (e.g. crimes or factories) and to one or more bibliographical references, such as book, thesis or article. In the platform, the keywords are used in the search mechanism to select layers associated to specific themes. Each layer has an owner user who creates it in the platform and a set of collaborators. Collaborators are users that have permission to edit, delete and include new features into a layer. The collaborators of a layer are defined by its owner. Users can only edit data sets in layers where they are collaborators.

The communication among the platform users, called Pauliceia community, is done through notifications. Notifications can be reviews of data, comments or denunciation. Users can write notifications about a specific layer or about an another notification. Besides that, they can write general notifications for all Pauliceia community.

In the platform, there are two types of users: logged and unlogged ones. The unlogged users can visualize, search and download the platform data sets as well as read all its notifications. The logged users can be of three types: normal, curator and administrator. The normal users can edit and contribute with new data sets, creating new layers and associating them with keywords and references. The curator users can edit all layers of the platform by by adding new keywords to them. The administrator users have permission to create, edit and remove all entities of the platform.

3.3. Data collection methods

There are two ways of data collection methods in the Pauliceia 2.0 platform: manual contribution and bulk import.

In the manual edition, users manually create and edit the spatial locations or boundaries of features by clicking on the historical maps presented in the web portal. Besides that, users can edit manually all attribute values associated to the features.

In the bulk importing, users can upload a group of features stored in well-known file formats of vector geographical data, such as shapefile or geojson. In the manual edition, users have to inform all metadada associated to the features. In the bulk importing, some types of metadada can be extracted automatically by the platform from the file content.

The shapefiles generated by the spatiotemporal geocoding tool of the Pauliceia platform can be directly used in the bulk import. Using the spatiotemporal geocoding tool, a user can upload a CSV (Comma Separated Values) file that contains a set of textual historical addresses and get a shapefile with all spatial locations of these addresses produced by the geocoder. So, this shapefile can be imported in the platform, creating a new layer in the Pauliceia database.

3.4. Quality control

In the literature, there are several proposals to evaluate VGI data quality. In a nutshell, these methods are described as quality measures, quality indicators and quality approaches. Quality measures verify the accuracy of VGI data in relation to the authoritative data provided by mapping agencies. Quality indicators measure the quality of VGI data in an abstract way when there is not authoritative data for comparison [Senaratne et al. 2017]. Quality approaches determine the degree of a fact, if it is possible to be true, and it can be automated or used of human intervention [Goodchild and Li 2012]. One proposal of VGI data quality introduced by Goodchild and Li [Goodchild and Li 2012] is the crowdsourcing approach. One interpretation of this approach is the use of the Linus Law[5]. Linus Law is the ability of using the people to verify the contributed data of VGI to converge to the truth.

To use quality measures, it is necessary to use authoritative data for comparison [Senaratne et al. 2017]. In the Pauliceia 2.0 project context, there is no authoritative data and so we can not use quality measures. Thus, the Pauliceia 2.0 team evaluated different types of quality indicators and approaches to evaluate the citizen-derived geographical data of the project, such as gamification (e.g. ranking) [Senaratne et al. 2017], trustworthiness and user reputation [D'Antonio et al. 2014]. Nevertheless, a consensus

[5] "Given enough eyeballs, all bugs are shallow" [Raymond 2017]

was reached that the best thing would be to adopt a crowdsourcing approach, using notifications and denunciations provided by the Pauliceia 2.0 community.

In the Pauliceia 2.0 platform, a notification is a comment from a user related to a layer or to another comment. A user can write a notification describing the positives or negatives points of a layer, warnings or suggestions to improve it, such as suggestions of new bibliographic references related to the layer. A denunciation is a special kind of notification made to alert administrators that a layer contains inappropriate data (e.g. copyright data or owned by another researcher). The administrators of the platform receive these reports, evaluate the layers associated to denunciations and can remove them from the platform as well as its owner user.

3.5. Feedback to the Community

A collaborative project become better as more users assist in it. Hence, it is important that users supply feedbacks about their experience with the project [Mooney et al. 2016]. In the Pauliceia 2.0 platform, volunteers are encouraged to make comments, give opinions and observations about their experiences on the platform, indicating the positive aspects and suggestions to improve it. This feedback can be done by the available mailing list and social networks that are managed by the Pauliceia 2.0 team. The feedback is important to improve the platform.

Using the Pauliceia 2.0 platform, researchers can disseminate and share their historical data sets as well as receive feedback from other researchers about them through notifications. Such data sets will be freely available on the portal and thus will achieve a greater visibility in the scientific community and dissemination.

Historians can write notifications on their layers or on the layers of other users, providing feedback on their status, such as remarks, praise or hints (e.g. indicate a new reference for that layer). Users can also write general notifications for all members of the Pauliceia 2.0 community, such as research event announcements. If a user finds unappropriated data in the platform, he or she can report it through denunciations. Besides that, users may follow layers of interest from other authors and receive notifications about them by e-mail.

4. VGI Management Web Service for Historical Data

Figure 3 shows the Pauliceia 2.0 platform architecture [Ferreira et al. 2017]. The platform is open source, online and service oriented. Service-oriented systems are well appropriate to supply a better interoperability among applications. The spatiotemporal data sets of the project are stored in a PostgreSQL database with spatial extension PostGIS (vector data) and in GeoTIFF files (raster data).

The platform architecture contains two groups of web services. The first group contains standards of geographic web services specified by the Open Geospatial Consortium (OGC), such as Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS) and Catalogue Service Web (CSW). The second group is composed of two web services designed and implemented to augment the functionalities of the OGC standard services, attending to specific and crucial demands of the Pauliceia 2.0 project.

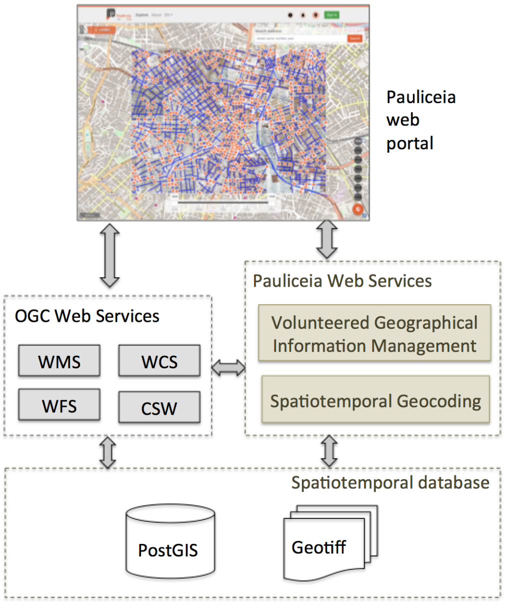


Figure 3. Pauliceia 2.0 platform architecture. [Ferreira et al. 2017]

This section describes the Volunteered Geographical Information Management Web Service (VGIMWS) that was designed and built based on the Pauliceia VGI protocol described in section 3. It provides all necessary functionalities for dealing with historical citizen-derived geographical information, such as user control, spatiotemporal features management as well as user edition of notifications and denunciations.

Figure 4 shows the architecture of VGIMWS. It is a RESTful web service developed in Python language. The chosen standard for data exchange is GeoJSON and JSON, that handle data with geographic information or without, respectively. It provides function to create, edit and remove all concepts described in section 3, such as user, layer, features and notifications.

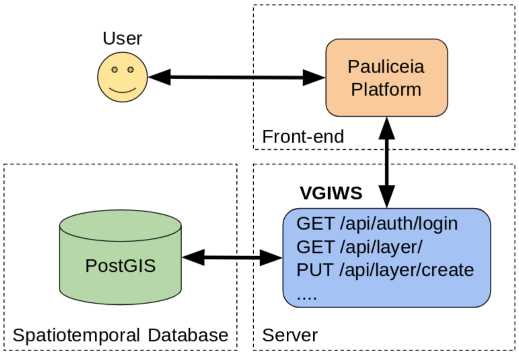


Figure 4. VGIMWS architecture.

All software codes of Pauliceia 2.0 project are free and open source and can be found at the Github of the project [6]. This Github contains the source code of the VGIMWS[7], its documentation[8] and instruction about how to run the web service and to install its dependencies.

Figure 5 shows a sequence diagram of one function of the VGIMWS that creates a new layer. First, the user tries to log in the platform using one URL and the VGIMWS returns a HTTP status, success or error. If the user is able to log in the platform, he or she can enter on the dashboard, create a new layer and associate keywords and references to it. After that, the volunteer can import a shapefile using the bulk import or create an empty layer. Lastly, the user must inform other metadata about the layer, such as its temporal columns.

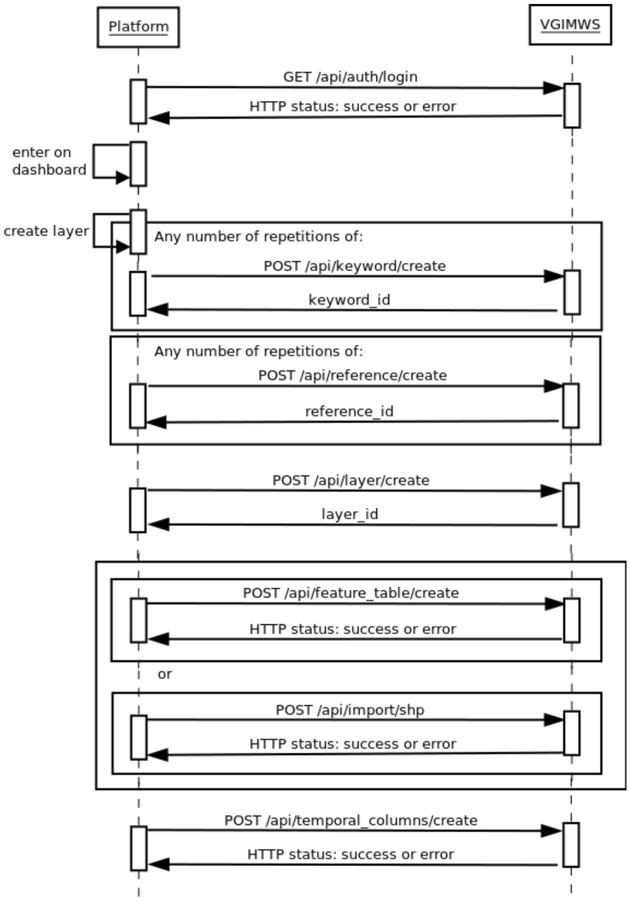

Platform
VGIMWS
GET /api/auth/login
HTTP status: success or error
enter on dashboard
create layer
Any number of repetitions of:
POST /api/keyword/create
keyword_id
Any number of repetitions of:
POST /api/reference/create
reference_id
POST /api/layer/create
layer_id
POST /api/feature_table/create
HTTP status: success or error
or
POST /api/import/shp
HTTP status: success or error
POST /api/temporal_columns/create
HTTP status: success or error

Figure 5. Add a new layer.

A set of tables is proposed to store metadata related to the temporal information and media attributes of the layers, shown in Figure 6. It is an extension of the Simple Feature Access Model proposed by OGC. The GEOMETRY_COLUMNS and the

[6] https://github.com/Pauliceia

[7] https://github.com/Pauliceia/vgiws

[8] https://github.com/Pauliceia/vgiws/blob/master/doc/README.md

FEATURE TABLE are tables defined by OGC, while the MEDIA_COLUMNS, TEMPORAL_COLUMNS and MASK are proposed in this work.

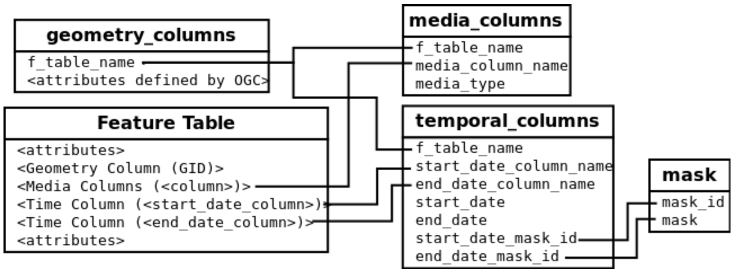


Figure 6. Feature metadata tables.

FEATURE TABLE is a table that stores features, where the columns are the attributes and the rows are the features. GEOMETRY_COLUMNS is a table that contains metadata about the geometry properties of a feature table [Herring 2006]. MEDIA_COLUMNS is a table that contains the metadata about the properties of a feature table that are links to media, such as photos or videos, that are stored in other repositories, as Google Drive, YouTube or Dropbox. TEMPORAL_COLUMNS is a table that defines the temporal attributes of the feature table. Its attributes are the start date, end date and the temporal bounding box of a feature. MASK is a table that saves the possible masks for the start date and end date, susch as ”YYYY-MM-DD”. Both MEDIA_COLUMNS and TEMPORAL_COLUMNS contain a reference to a feature table that is registered in the GEOMETRY_COLUMNS.

Figure 7 presents the complete spatiotemporal database model of Pauliceia 2.0 project. This model express the conceptual model described in section 3.2. It contains tables to store the concepts of the Pauliceia 2.0 project, such as user, layer, reference, keyword, notification, changeset, feature table, media, temporal information and followers. This database was built using PostgreSQL with the spatial extension PostGIS.

5. Conclusion

VGI has emerged with the purpose of collecting geographical data sets fast and with low cost. However, to improve the quality and the reuse of these data sets, it is necessary to define protocols to guide VGI projects.

This paper presents a VGI protocol for historical data that was defined in the context of Pauliceia 2.0 project. This project aims to develop an online platform for collaborative research of historical data, using VGI and crowdsourcing techniques. Besides that, this paper describes a RESTful web service, called VGIMWS, that was built in the Pauliceia platform based on the VGI protocol. VGIMWS manipulates all the protocol concepts through specific URLs.

The VGI protocol helps to increase the quality of the historical citizen-derived geographical data of the Pauliceia 2.0 platform. It defines crucial issues that improve the understanding of volunteers about the Pauliceia project and all its mechanisms and methods to collect, manage and assess the quality of the citizen-derived geographical data. The proposed VGI protocol and the VGI management web service are generic, so both can be used to other collaborative historical project.

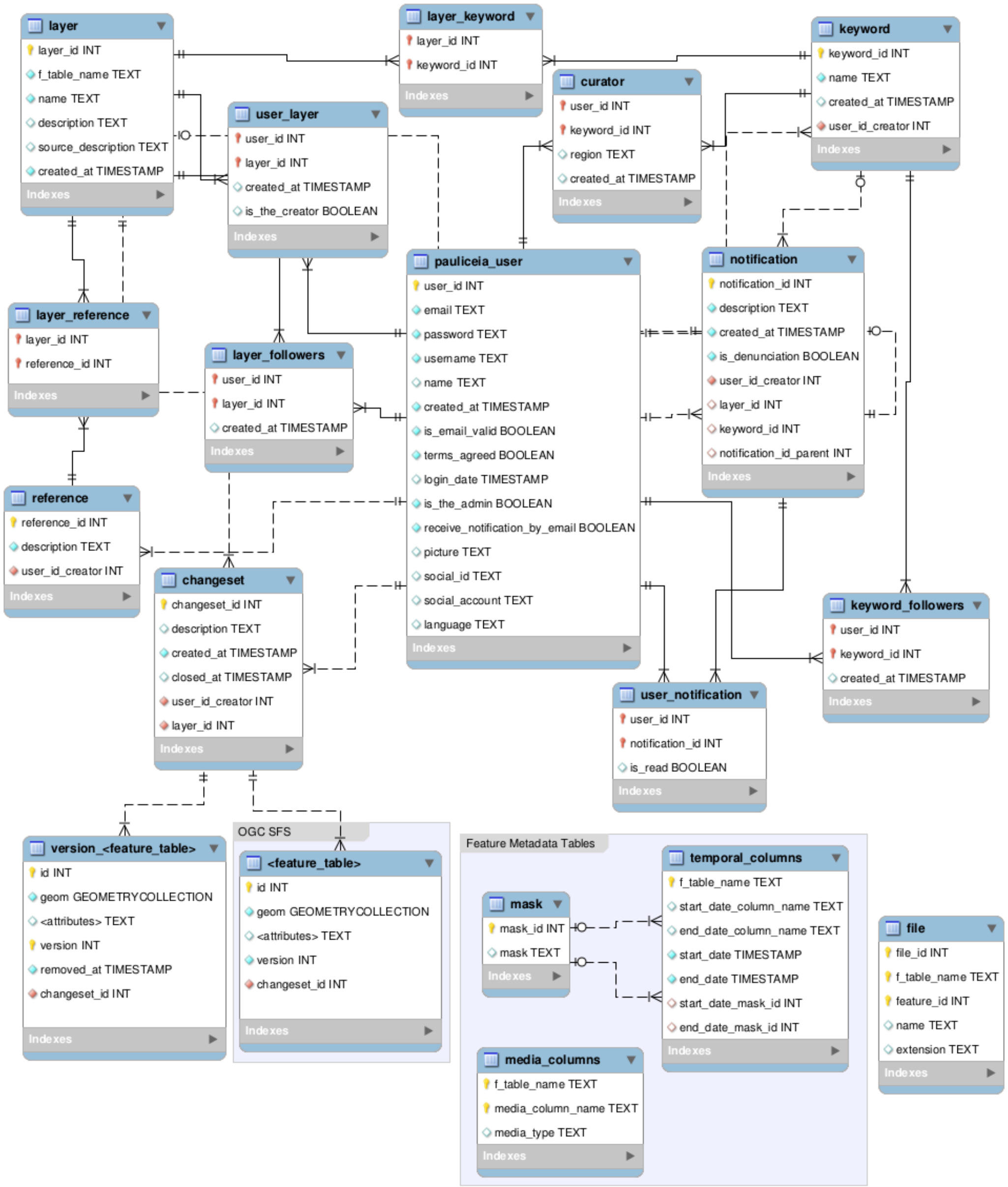


Figure 7. Database Model of the Pauliceia 2.0 project.

6. Acknowledgment

Pauliceia 2.0 project is funded by São Paulo Research Foundation (FAPESP) eScience Program, grant #2016/04846-0. The authors thank FAPESP for granting student scholarship #2017/03852-9.

References

Budig, B., van Dijk, T. C., Feitsch, F., and Arteaga, M. G. (2016). Polygon consensus: smart crowdsourcing for extracting building footprints from historical maps. In

Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, page 66. ACM.

D'Antonio, F., Fogliaroni, P., and Kauppinen, T. (2014). Vgi edit history reveals data trustworthiness and user reputation.

Estellés-Arolas, E. and González-Ladrón-de Guevara, F. (2012). Towards an integrated crowdsourcing definition. *Journal of Information science*, 38(2):189–200.

Ferreira, K. R., Ferla, L., de Queiroz, G. R., Vijaykumar, N. L., Noronha, C. A., Mariano, R. M., Wassef, Y., Taveira, D., Dardi, I. B., Sansigolo, G., Guarnieri, O., Musa, D. L., Rogers, T., Lesser, J., Page, M., Britt, A. G., Atique, F., Santos, J. Y., Morais, D. S., Miyasaka, C. R., de Almeida, C. R., do Nascimento, L. G. M., Diniz, J. A., and dos Santos, M. C. (2017). Pauliceia 2.0: A computational platform for collaborative historical research. *Proceedings XVIII GEOINFO, December 04th to 06th, 2017*, pages 28–39.

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

Goodchild, M. F. and Li, L. (2012). Assuring the quality of volunteered geographic information. *Spatial statistics*, 1:110–120.

Herring, J. (2006). Opengis implementation specification for geographic information-simple feature access-part 2: Sql option. *Open Geospatial Consortium Inc*.

Mooney, P., Minghini, M., Laakso, M., Antoniou, V., Olteanu-Raimond, A.-M., and Skopeliti, A. (2016). Towards a protocol for the collection of vgi vector data. *ISPRS International Journal of Geo-Information*, 5(11):217.

OpenStreetMap (2017a). About. `https://www.openstreetmap.org/about`. Accessed on 02/08/2018.

OpenStreetMap (2017b). Mapathon. `http://wiki.openstreetmap.org/wiki/Mapathon`. Accessed on 18/08/2018.

OpenStreetMap (2018). Open historical map. `https://wiki.openstreetmap.org/wiki/Open_Historical_Map`. Accessed on 05/08/2018.

Page, M. C., Durante, K., and Gue, R. (2013). Modeling the history of the city. *Journal of Map & Geography Libraries*, 9(1-2):128–139.

Raymond, E. S. (2017). Release early, release often. `http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/ar01s04.html`. Accessed on 10/08/2018.

See, L., Mooney, P., Foody, G., Bastin, L., Comber, A., Estima, J., Fritz, S., Kerle, N., Jiang, B., Laakso, M., et al. (2016). Crowdsourcing, citizen science or volunteered geographic information? the current state of crowdsourced geographic information. *ISPRS International Journal of Geo-Information*, 5(5):55.

Senaratne, H., Mobasheri, A., Ali, A. L., Capineri, C., and Haklay, M. (2017). A review of volunteered geographic information quality assessment methods. *International Journal of Geographical Information Science*, 31(1):139–167.

Tech2 (2014). Why is google's mapathon in hot waters in india? all you need to know. http://www.firstpost.com/tech/news-analysis/why-is-googles-mapathon-in-hot-waters-in-india-all-you-need-to-know-3655197.html. Accessed on 18/08/2018.

Does keyword noise change over space and time? A case study of social media messages

Sidgley C. de Andrade[1], Lívia Castro Degrossi[2], Camilo Restrepo-Estrada[3]
Alexandre C. B. Delbem[2], João Porto de Albuquerque[4]

[1]Federal University of Technology, Paraná (UTFPR)
Toledo – PR – Brazil

[2]Institute of Mathematical and Computing Sciences (ICMC)
University of São Paulo (USP) – São Carlos – SP – Brazil

[3]Faculty of Economic Sciences
University of Antioquia – Medellín – Colombia

[4]Centre for Interdisciplinary Methodologies (CIM)
University of Warwick – Coventry – UK

sidgleyandrade@utfpr.edu.br, degrossi@icmc.usp.br
camilo.restrepo@udea.edu.co, acbd@icmc.usp.br, J.Porto@warwick.ac.uk

Abstract. *Social media is a valuable source of information for different domains, since users share their opinion and knowledge in (near) real-time. Moreover, users usually use different words to refer to a particular event (e.g., a rain event). These words may be later employed to filter social media messages regarding new occurrences of the event and, thus, to reduce the number of unrelated messages. These words, however, may have different meanings and, thus, may not reduce the number of messages. In this work, we conduct a case study to measure which rain- or flood-related keywords are less relevant to reduce the number of unrelated messages. The results show that the keywords change over space, due to local language/culture, and time, specially in different time scales.*

1. Introduction

In the last few years, there has been a growing interest in social media data since it is a valuable source of (near) real-time information that can be used to detect, monitor and predict different types of events [Steiger et al. 2015]. For instance, in the field of flood management, social media messages could be employed to cover areas where there are an insufficient number of physical sensors and a lack of accurate and updated official data. Moreover, social media may improve the situational awareness through eyewitnesses [Vieweg et al. 2010].

In general, social media users utilize a variety of terms to refer to an event that they observe. However, because of the great amount of data, retrieving relevant and meaningful data is not a straightforward task. Keyword-based filtering approach has been widely employed to remove duplicate, unreliable and unrelated data. In this work, we define duplicate and unrelated messages as noise, i.e. messages that contain rain- or flood-related keywords, but are not related to an event indeed or are duplicated. The noise usually occurs when the keywords have different definitions and/or meanings. In Brazil, for example, the term “Santos” can refer to the coastal city or the soccer team. A context analysis can reveal the true meaning of the term; nonetheless, it is a hard computational task because of the variations, misspelling and typos inherent in social media messages. Furthermore, ambiguous terms can lead to a second type of noise, i.e., false-positive messages[1], that hereafter we refer as noise.

Hence, this work addresses the following question: *Does keyword noise change over space and time?* To answer this question, we carried out a case study to measure the signal and noise rate of the keywords. The case study was supported by an exploratory content analysis of a rain- and flood-related data sample from Twitter.

This paper is structured as follows: in Section 2, we describe the methodology. In Section 3, we present the results. Finally, in Section 4, we discuss the results, address some conclusions and make suggestions for future work.

2. Methodology

2.1. Study area

The city of São Paulo (Brazil) was selected as the study area because it registers several rain events that cause flash floods. The city is known as "the land of drizzle" by Brazilians and has a population of approximately 12 million people [IBGE 2010]. Furthermore, the city is divided in 96 districts, which were used as spatial units of observation for the exploratory content analysis.

2.2. Twitter data and keywords

We developed a crawler tool to retrieve public tweets through Twitter Stream API. Moreover, we defined two bounding-box filters covering the city of São Paulo, one north (-46.95, -23.62, -46.28, -23.33) and one south (-46.95, -23.91, -46.28, -23.62). A total of 11,848,923 million tweets were retrieved from 7 November 2016 to 28 February 2017 (UTC time), where only 891,367 were geotagged (7,52%).

After retrieving the tweets, we filtered the geotagged ones based on a set of keywords (Table 1) – using a substring-searching approach. We selected the ones that contained at least one of the keywords and aggregated them by district (Figure 1). Though some tweets geotagged within the bounding-box may be referring to other places, we identify and remove them in the next subsection.

Table 1. Keywords in Brazilian-Portuguese with their English meaning in parentheses. The keywords were chosen based on previous works and a preliminary analysis of the tweets. Similar terms as "chuva" (rain) and "chuvaaa" (rainn) were aggregated. Keywords with grammar mistakes were take into account as long as the frequency was equal or greater than 10 (e.g. "chuvendo").

alagamento (flood), alagado (flooded), alagada (flooded), alagando (it's flooding), alagou (flooded), alagar (to flood), chove (rain), chova (rain), chovia (had been rained), chuva (rain), chuvarada (rain), chuvosa (rain), chuvoso (rainy), chuvona (heavy rain), chuvinha (drizzle), chuvisco (drizzle), chuvendo (it's raining), dilúvio (heavy rain), garoa (drizzle), inundação (flood), inundada (flooded), inundado (flooded), inundar (to flood), inundam (flood), inundou (flooded), temporal (storm), temporais (storms)

[1] The false-positive messages correspond to messages that contain the keywords but are not related to event.

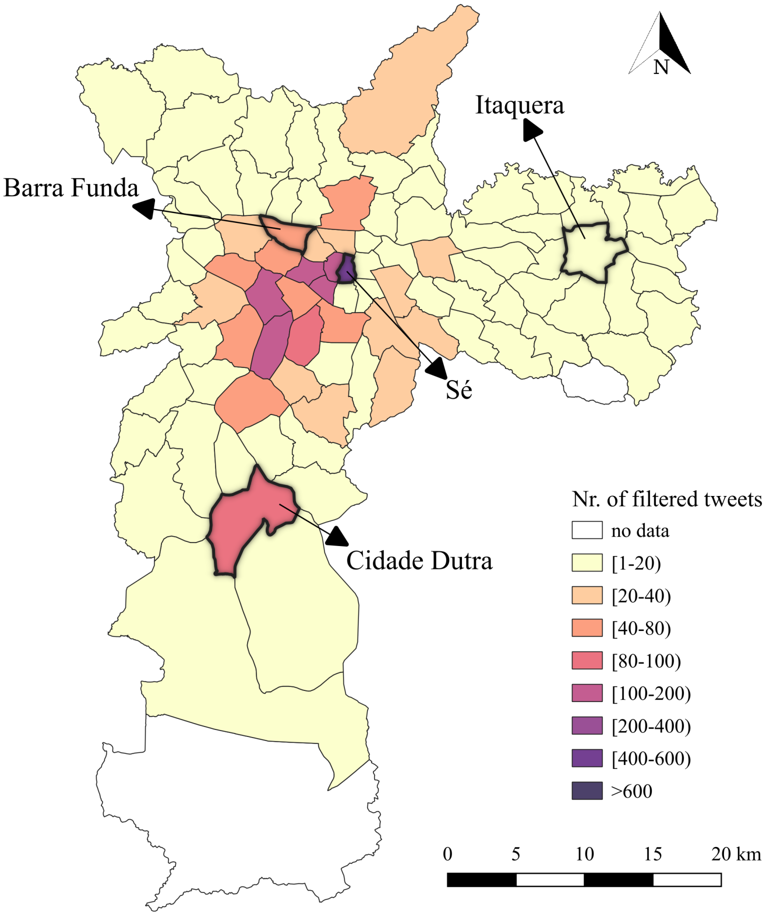


Figure 1. Choropleth map of the number of filtered tweets per district.

2.3. Exploratory content analysis

The exploratory content analysis consisted of two steps: (i) labeling the 5,408 filtered tweets as on-topic and off-topic, and (ii) building the time series of the signal and noise of the keywords.

First, five raters manually labeled 3,964 tweets as on-topic (related to local rain or flood), and 1,444 as off-topic (not related to local rain or flood). In the following, we measured the degree of agreement among raters by means of the Krippendorff's alpha coefficient, a statistical measure of the degree of agreement among two or more raters [Krippendorff 2004]. A value equal to 0.72 was obtained, which indicates a good agreement among raters. A coefficient equals to 0 (zero) indicates an absence of agreement and 1 (one) a perfect agreement.

Second, for each district and keyword, we built two time series, called signal and noise, that correspond to the on-topic and off-topic tweets, respectively. For this, we used 6 time scales: (i) 30 min., (ii) hour, (iii) 12 hours, (iv) day, and (v) week. After, a third time series was built with the ratio between the on-topic (signal) or off-topic (noise) time series. Finally, we analyzed the time series over time and across the districts. The main idea was to evaluate the difference between the time series of the signal and noise of each keyword.

3. Results

The results show that the keyword noise changes over time and space, leading to a depreciation or increase of the keyword signal (Figure 2). Moreover, the signal tends to increase (appear) for large time scales (e.g., weeks) and decrease (disappear) for small time scales (e.g., minutes). In addition, there is a spatial dependence of the keyword signal across the districts, i.e., the signal and noise are usually more similar in near districts than distant ones. For example, The Sé district is similar to the Barra Funda district, whereas the Cidade Dutra district is different from both districts (Figure 2). However, the distance sometimes does not influence the similarity among the districts. For example, the Sé and Itaquera districts are similar and far away from each other. That means that the amount of tweets posted within these districts (Figure 1) does not explain totally the signal of the keywords. Other variables such as the interconnection areas of the underground railway system and economic factors could also be describe the signal.

When the keywords are examined, we can see that some of them do not vary over time, such as “chuvinha” (raining a little), “chuvosa” (rainy) and “inundação” (flood), i.e., they do not often vary from signal to noise or vice-versa. On the other hand, some keywords reveal greater noise in short time intervals, such as the keyword “chuva” (rain).

Furthermore, some keywords have potential to compose a (good) signal, however they create noise. This could be explained by the fact that some words have special association to local language/culture or atypical events. The keyword “garoa” (drizzle), for instance, might be strongly related to a drizzle phenomenon, however, most messages refer to the codinome of the city of São Paulo (“the land of drizzle”). Other interesting example occurred during the concert of the rock band Guns and Roses at Allianz Park in the Barra Funda district. On November 12th, 2016, there was a frequency peak of the keyword “chuva” (rain) when the band played one of their most famous songs, “November Rain”. Messages like “chuva de novembro” and “a chuva veio antes pra colocar todo mundo no clima da November Rain!...” were reported by people who were attending the concert.

The underlying problem behind using keywords is their reproducibility from one area to another or at the same area over time. Rzeszewski (2018) refers to this behaviour as a change of the perception of the physical space. As shown in Figure 2, the behaviour changes in terms of time and space. Hence, keywords should be selected with caution, considering local issues, such as language/culture, and, specially, atypical events.

4. Discussion and Conclusion

This work analyzed the signal and noise of rain- and flood-related keywords that are used to filter social media messages. The results evidence that the keywords are sensible to time and space. At the first sight, all predefined keywords had potential to filter rain- and flood-related messages; however, our analysis demonstrated that some keywords are noisy and may introduce false-positive messages. This implies a lack of quality of the filtered messages. For example, people usually post messages with the keyword “garoa” (drizzle) as reference to the city of São Paulo (“the land of drizzle”), which could lead to a noisy dataset. Therefore, the type of keyword can influence the keyword-based filtering technique, an useful technique to reduce the amount of social media messages, because it could cause more noise than others. Thus, firstly, an analysis of keywords noise should be carried out in order to support the selection of them.

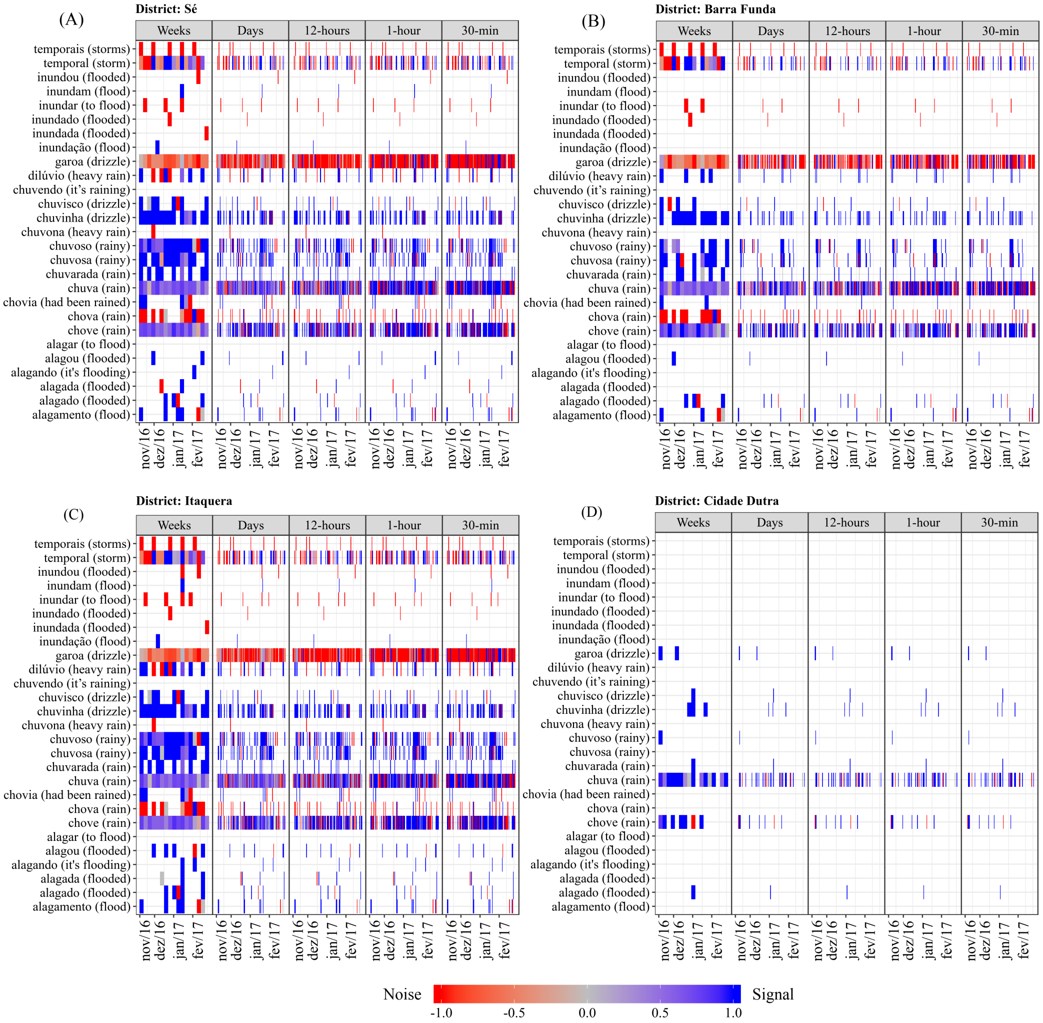


Figure 2. Hovmöller-based diagram depicting the signal and noise of the keywords over the entire period of analysis and across the four highlighted districts in Figure 1. The x and y axes show the time slices and the keywords, respectively. The blue color represents the signal intensity, whereas the red color represents the noise intensity. White color represents no data. The signal and noise were measured as the fraction between on-topic and off-topic tweets and all the tweets posted within the district (relative frequency) and, later, rescaled to [-1, 1].

Future work should further extend this exploratory content analysis by incorporating other cities in order to understand the noise of the rain- and flood-related keywords. Once the noises are understood, keywords can be selected to filter the social media messages more accurately. Finally, skip-gram models (e.g., word2vec) could be used to address the ambiguity problem of terms in social media.

Acknowledgements

The authors are grateful for the computational resources provided by of the Center for Mathematical Sciences Applied to Industry (CeMEAI), funded by the São Paulo Research

Foundation (FAPESP) (grant no. #2013/07375-0) and also to DAEE/FCTH for making the weather radar data available for this study. The authors would also like to express their thanks for the financial support provided by the Coordination for higher Education Staff Development (CAPES) (Edital Pró-Alertas 24/2014, grant no. #88887.091744/2014-01 and #88887.091743/2014-01). S.C.A. would like to thank the following agencies - FAPESP (grant no. #2017/15413-0), Araucária Research Foundation in Support of Scientific, and Technological Development in the State of Paraná (FAPPR), and the State Secretariat of Science, Technology and Higher Education of Paraná (SETI) - for its financial support.

References

de Andrade, S. C., Restrepo-Estrada, C., Delbem, A. C. B., Mendiondo, E. M., and de Albuquerque, J. a. P. (2017). Mining rainfall spatio-temporal patterns in twitter: A temporal approach. In Bregt, A., Sarjakoski, T., van Lammeren, R., and Rip, F., editors, *Societal Geo-innovation*, pages 19–37, Cham. Springer International Publishing.

IBGE (2010). *Censo Demográfico 2010*. Brazilian Institute of Geography and Statistics, Rio de Janeiro.

Krippendorff, K. (2004). Reliability in content analysis: Some common misconceptions and recommendations. *Human Communications Research*, 30(3):411–433.

Restrepo-Estrada, C., de Andrade, S. C., Abe, N., Fava, M. C., Mendiondo, E. M., and ao Porto de Albuquerque, J. (2018). Geo-social media as a proxy for hydrometeorological data for streamflow estimation and to improve flood monitoring. *Computers & Geosciences*, 111:148–158.

Rzeszewski, M. (2018). Geosocial capta in geographical research – a critical analysis. *Cartography and Geographic Information Science*, 45(1):18–30.

Steiger, E., de Albuquerque, J. a. P., and Zipf, A. (2015). An advanced systematic literature review on spatiotemporal analyses of twitter data. *Transactions in GIS*, 19(6):809–834.

Vieweg, S., Hughes, A. L., Starbird, K., and Palen, L. (2010). Microblogging during two natural hazards events: What twitter may contribute to situational awareness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1079–1088, New York, NY, USA. ACM.

A Performance Comparison Between two GIS Multi-Criteria Decision Aid methods: a Case Study of Desertification Evaluation

Heithor Alexandre de Araujo Queiroz[1], Bruno Cardoso Dantas[1], Cícero Fidelis da Silva Neto[1], Thiago Emmanuel Pereira[2], Ricardo da Cunha Correia Lima[1]

[1]Department of Geoinformatics – Instituto Nacional do Semiárido (INSA)
Caixa Postal 10067 – Campina Grande – PB – Brazil

[2] Computer and Systems Department – Universidade Federal de Campina Grande (UFCG) Campina Grande – PB – Brazil

{heithor.queiroz, bruno.dantas, cicero.fidelis, ricardo.lima@insa.gov.br, temmanuel@computacao.ufcg.edu.br}

Abstract. *Desertification is widely recognized as one of the most relevant environmental problems to be evaluated. In many cases, it requires processing large amounts of data and is also computing intensive. The present study sheds light on this problem in the context of a desertification analysis of the Brazilian Semiarid, using the PROMETHEE Multi-Criteria Decision Aid method, which is a multicriteria analysis method used to identify the outranking relation for a pair of alternatives tackling spatial problems such as site selection problem and land use/suitability analysis. We describe the design and implementation of a practical solution to this problem, based on state-of-the-art theoretical advances and further improvements to deal with large datasets. We compare the performance of our solution with the GRASS software environment. The performance evaluation indicates that our solution can address the problem; it is up to 720 times faster than the GRASS alternative, for the evaluated scenario.*

1. Introduction

Desertification is an environmental problem that is highlighted to be assessed by the most important agencies and institutions all over the world, such as IPCC, ONU, USGS, NASA (GEIST, 2017; IPCC, 2007). Desertification is featured by the soil degradation, which impacts negatively the environmental, social and economic spheres of the countries (TOMASELLA et al. 2018; BESTELMEYER, et al. 2015; OLAGUNJU 2015).

Regarding the desertification evaluation, the high amount of variables which is commonly required to assess the desertification process usually leads to the generation of large datasets to be analysed, directly impacting the computational costs of the analysis (BRITO, et al. 2018; MARIANO, et al. 2018; VIEIRA, et al. 2015).

A recent development (LIMA, 2017) which has applied the Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE), which is a Multi-Criteria Decision Aid (MCDA) method, based on 27 criteria (including land concentration, social inequality, deforestation and others), to analyse the desertification of the Seridó Region (part of the Brazilian Semiarid - BSA) illustrates this problem. The Seridó region which is composed of 32 municipalities, for a total area of 11.194,696 km^2, has a total of 187.000 pixels (considering a 300m spatial resolution). Considering this number of pixels and the 27 criteria, the total number of alternatives is up to 5.000.000. The size of the dataset for the Seridó region is up to 35MB. Even for this small region, the GRASS software environment (OSGeo project, 2015) took a dozen hours to execute its PROMETHEE analysis on a workstation. The analysis of the whole Brazilian Semiarid dataset, which is up to 350GB, would be infeasible to execute using the GRASS system (since its PROMETHEE implementation has a quadratic complexity).

Furthermore, although recently approximation methods have been developed to reduce the complexity of the calculation of PROMETHEE, for example, the use of piecewise linear functions (EPPE and DE SMET, 2014), we designed and developed an optimized PROMETHEE implementation based on a subquadratic exact solution of the PROMETHEE algorithm presented in Calders and Van Assche (2018). Our implementation attests that is possible to improve the computational cost efficiency by preserving the exact PROMETHEE method. In addition to this improved complexity, our implementation also adopted some optimizations to handle large datasets.

In this study, we briefly describe our solution (Section 2) and provide a performance comparison with the GRASS system (Section 3). The results obtained indicated that, for the datasets analysed, our solution is up to 720 times faster than the GRASS alternative (in fact, this speed up would increase as the dataset grows, due to the improved complexity). Finally, in Section 4, we discuss relevant future work.

2. MCDA Tools

In this section, we introduce the GRASS system and our optimized MCDA tool highlighting the differences between them. Although the GRASS system includes not only MCDA features, we restraint the discussion to its implementation of the PROMETHEE method.

2.1. GRASS

The Geographic Resources Analysis Support System (GRASS) GIS is a widely used (thus a suitable alternative to our performance comparison described in Section 3) open source software for geospatial management, data analysis and image processing (OSGeo project, 2015). The design of GRASS is based on a plugin architecture (add-ons) which allows extending its feature set. Its PROMETHEE plugin, which follows the original proposition of the method (VINCKE and BRANS, 1985), is implemented in the C language. Despite GRASS popularity and overall quality, its

MCDA implementation has a performance limitation that turns it unsuitable to our scenario.

2.1. Optimized Implementation

Our tool is a C++ optimized implementation of the PROMETHEE method designed to process large GIS datasets[1]. It is also important to highlight that although the method optimized in the present study is the PROMETHEE II (once it considers the fluxes differences), in the remaining of the text it is named as PROMETHEE rather than PROMETHEE II, only to simplify the reading. Our implementation is based on a linear algorithm that improves the original PROMETHEE II method (which has quadratic complexity) for the linear and level preference functions (CALDERS and VAN ASSCHE, 2018). In addition to the speed up provided by adoption of the sub-quadratic algorithm, our implementation dealt with a practical aspect of its implementation when analysing large datasets: how to keep the data in memory during the execution of the analysis; in some cases, the datasets are larger than the amount of available memory. To this end, we design and developed two optimizations. First, for each criterion, the analysis of alternatives is made up in a partial fashion (to avoid keeping the whole dataset in memory) and stored in stable storage. Second, we avoid loading into the memory segments of the dataset which show consecutive alternatives of the same value.

3. Performance evaluation

In this section, we describe the experiments we have executed to compare the performance of the GRASS (version 7.4.1) system and our optimized solution. In the first experiment, we aimed to analyze how these solutions behave as the number of alternatives grows. To this end, we executed the multi-criteria analysis on synthetic samples, made of randomly generated values, of 4096, 16385, and 65536 alternatives (in all these cases, we analysed a single criterion). In the second experiment, we compared the average time to execute a multi-criteria analysis of a sample of the target study area (the Seridó region), considering only two criteria (instead of 27); the duration of experiment, considering the whole dataset, would be prohibitive to execute using the GRASS. To ease the reproducibility of results, we made available both datasets used in these experiments[2].

We configured an experimental environment based on a Linux workstation which runs both the GRASS and our optimized solution. The workstation runs the Linux kernel version 4.4.0-134, based on the Ubuntu 16.04.5 release. The workstation has an octa-core Intel i7-4770 3.10GHz CPU with 8GB of main memory, and a 1TB SEAGATE 7200 RMP hard disk, ST1000DM003 model.

In both experiments, the performance was given as the duration to run the MCDA. This duration is given by the elapsed time between the start of the program until the time it finished (after it writes its output to stable storage). Each execution starts by flushing the operating system memory caches. By flushing these caches, we avoid that one execution affects the subsequent one.

[1] https://github.com/simsab-ufcg/Promethee2

[2] https://github.com/simsab-ufcg/landsat-samples/tree/master/geoinfo-2018

3.1 Results

Figure 1 shows how the duration of the multi-criteria analysis varies, on GRASS and in our optimized implementation, according to the number of alternatives evaluated (from 4096 to 65536). The Figure shows the results of 10 analysis, for each configuration of the number alternatives, for both the implementations. The duration is given in logarithm scale.

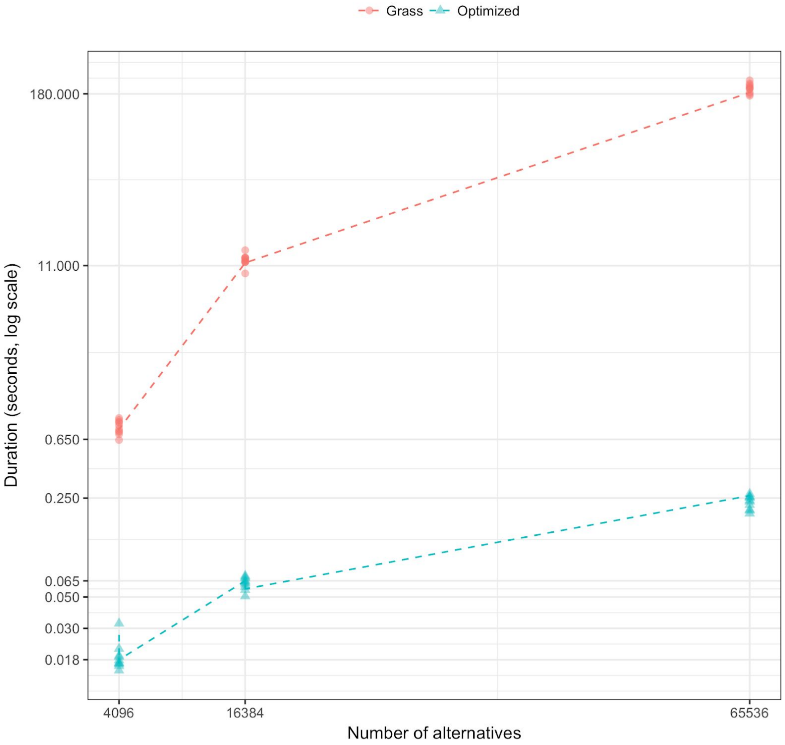


Figure 1. Duration of the analysis for the GRASS and our optimized implementation. The experiments considered three different scenarios: 4096, 16386, and 65536 alternatives. The optimized is no less than 21 times faster than the GRASS tool. For the largest scenario, the optimized solution is 720 times faster.

Considering the optimized solution, the duration of the analysis for the scenario of 4096 alternatives is up to 0.03 seconds, up to 0.065 seconds for 16384 alternatives, and no more than 0.25 seconds for the largest scenario, of 65536 alternatives; all the executions are in the subsecond range. Due to the inherent, unnecessary complexity of the GRASS implementation, the duration of the analysis is 0.65, 11, and 180 seconds, respectively for the scenarios of 4096, 16384 and 65536 alternatives. For the smallest scenario (4096 alternatives), the optimized implementation is up to 21 times faster than the GRASS, and for the lager scenario (65536), it is 720 times faster.

Table 1 shows the duration of the multi-criteria analysis, for the Seridó region, on both the GRASS system and in our optimized implementation. We considered two criteria in this analysis, thus 350000 alternatives in total. The duration and its standard deviation are given for an average of 10 executions. The results for our optimized solution are still in the subsecond range, 0.004 minutes (0.26 seconds), while for the GRASS the mean duration is more than 30 minutes. Note that, the duration of our optimized solution is almost the same duration for the experiments shown in Figure 1, with 65536 alternatives, even though the current dataset is about five times larger. The reasons for this speed-up are twofold: (i) the experiments shown in Table 1 analyze more than one criteria, and, in this case, our solution can take advantage of the multiple processors of the workstation used in the experiment (the analysis of each criterion runs in parallel); (ii) differently from the dataset analyzed for the first experiment, which was generated randomly, the data from the Seridó region has some degree of duplication, which leads to less data loading into memory during the execution.

	Duration in minutes (mean; std deviation)
Grass	(30.64; 0.19)
Optimized	(0.004; 4.21x10^{-5})

Table 1. Duration of the analysis of the Seridó region for the GRASS and our optimized implementation. The mean and standard duration are based on the execution of 10 experiments. The experiments considered two criteria, totalizing more than 350000 alternatives. For the GRASS alternative, the mean duration is approximately 30 minutes, while for our optimized solution is approximately 0.004 minutes (0.26 seconds).

4. Conclusions and Future Work

In this work, we considered the challenge of performing the multi-criteria analysis of large GIS datasets. In doing so, we provided two major contributions: (i) we developed and made publicly available an implementation of the algorithm proposed by Calders and Van Assche (2018), which provides exact solutions instead of approximate ones such as the piecewise linear functions (EPPE and DE SMET, 2014); to the best of our knowledge, there was no such implementation available yet; (ii) we designed further optimizations on the original proposal to cope with the analysis of large datasets including the partial computation of the analysis (on chunks of the dataset) and the use of a compact data format that avoids the store (and analysis) of duplicated alternatives.

The initial assessment described in this work can be extended to characterize our proposed design better. For example, a hardware resource utilization analysis could help us to identify opportunities for further improvements (e.g. to better parallelise the execution of the algorithm). In addition to that, we plan to improve our evaluation of the data compression feature by studying how the variability of the input data affects the performance of our tool. Also, we plan to compare our approach with parallel data processing tools (such as hadoop), as a comparison baseline; note that, however it is feasible to process the PROMETHEE analysis in a cluster/distributed environment, the associated costs (or resource usage) would be much higher than in our proposed solution.

5. References

BESTELMEYER, Brandon T. et al. Desertification, land use, and the transformation of global drylands. Frontiers in Ecology and the Environment, v. 13, n. 1, p. 28-36, 2015.

BRITO, S. S. B. et al. Frequency, duration and severity of drought in the Semiarid Northeast Brazil region. International Journal of Climatology, v. 38, n. 2, p. 517-529, 2018.

CALDERS, T.; VAN ASSCHE, D. PROMETHEE is not quadratic: An O (qnlog (n)) algorithm. Omega, v. 76, p. 63-69. 2018

EPPE, Stefan; DE SMET, Yves. Approximating Promethee II's net flow scores by piecewise linear value functions. European journal of operational research, v. 233, n. 3, p. 651-659, 2014.

GEIST, Helmut. The causes and progression of desertification. Routledge, 2017.

INTERGOVERNMENTAL PANEL ON CLIMATE CHANGE (IPCC). Working Group II: Impacts, adaptation, and vulnerability. 2007.

LIMA, R. C. C. Sistema de avaliação e comparação espacial do processo de desertificação no Seridó potiguar e paraibano, semiárido brasileiro. Tese (Doutorado em Recursos Naturais) – Programa de Pós-Graduação em Recursos Naturais, Centro de Tecnologia e Recursos Naturais, Universidade Federal de Campina Grande. Campina Grande, 2017, 150 f.

MARIANO, Denis A. et al. Use of remote sensing indicators to assess effects of drought and human-induced land degradation on ecosystem health in Northeastern Brazil. Remote Sensing of Environment, v. 213, p. 129-143, 2018.

OLAGUNJU, Temidayo Ebenezer. Drought, desertification and the Nigerian environment: A review. Journal of Ecology and the Natural Environment, v. 7, n. 7, p. 196-209, 2015.

OSGeo project. (24 de 02 de 2015). GRASS GIS - Bringing advanced geospatial technologies to the world. https://grass.osgeo.org/documentation/general-overview/.

TOMASELLA, Javier et al. Desertification trends in the Northeast of Brazil over the period 2000–2016. International Journal of Applied Earth Observation and Geoinformation, v. 73, p. 197-206, 2018.

VIEIRA, RM da Silva Pinto et al. Identifying areas susceptible to desertification in the Brazilian northeast. Solid Earth, v. 6, n. 1, p. 347-360, 2015.

VINCKE, J. P.; BRANS, Ph. A preference ranking organization method. The PROMETHEE method for MCDM. Management Science, v. 31, n. 6, p. 647-656, 1985.

Spatial Analysis and Statistics of Pedestrian Injuries in Curitiba through a Grid Database

Cassiano Bastos Moroz[1], Jorge Tiago Bastos[1]

[1]Departamento de Transportes – Universidade Federal do Parana (UFPR)
Postal Code 80050-540 – Curitiba – PR – Brazil

cassianomoroz@gmail.com, jtbastos@ufpr.br

Abstract. *This study addresses the pedestrian exposure to traffic accidents in the Municipality of Curitiba, in Brazil, by spatially analyzing fatal traffic accidents during the period 2010-2016. The proposed methodology englobes the application of GIS through the implementation of a grid database that correlates, on a macroscale, traffic accidents intensity – calculated by Kernel Density Estimator – with spatial factors of the built environment. Although the statistical analysis has not been concluded, preliminary results demonstrate a relation between population density, monthly income and traffic injuries.*

1. Introduction

Since the 1980s, in the midst of globalization, Brazilian urban areas have been experiencing the increasing advance of motorized vehicles with the encouragement of the city's "highway" configuration. Simultaneously, there was a stagnation of urban public transport, driven by the fragile economic growth, portrayed in the form of informal vans and motorcycle taxis [Maricato 2008]. Only two decades later there were efforts to create an effective legislation to promote sustainable urban mobility through the valorization of non-motorized modes of transportation, including pedestrian and cyclists, and the increase of an efficient and affordable public transport system. Among these legislations, stand out the Law nº 10.257 from July 10th, 2001, and the Law nº 12.587 from January 3rd, 2012.

In this context, it is essential to highlight the participation of pedestrians in Brazilian urban life. According to the report published by the Brazilian National Public Transport Agency [ANTP 2016], in 2014, non-motorized transportation corresponded to 40% of total urban displacements in Brazil, 36% of which refers to pedestrians and 4% to cyclists. Although the representativeness of this transport is higher in small municipalities, the use of non-motorized transportation is still considerable in large Brazilian urban centers with populations of more than 1 million inhabitants, covering 36.3% of total displacements [ANTP 2016].

Despite their expressive participation in urban life, pedestrians are highly vulnerable to traffic accidents. According to the Datasus Mortality Information System (SIM), pedestrians constituted 22.83% of total deaths caused by traffic accidents in Brazil in 2015. The numbers are even worst when observing that traffic accidents in 2015 were among the main causes of death in the country, with a mortality rate of 18.9 per 100,000 inhabitants [Datasus 2015]. For all these reasons, there are lots of aspects to be improved in the road infrastructure of Brazilian urban areas.

Considering the severity of this scenario, this study aims to map the exposure of pedestrians to traffic accidents within the Municipality of Curitiba, in Brazil, through an empirical analysis. Therefore, the methodology income is based on previous fatal traffic accidents registered within Curitiba during the period 2010-2016.

2. Literature Review

Several studies related to road safety use the Geographic Information Sciences to spatially analyze traffic accidents in urban areas. Anderson (2009) identified hotspots of traffic accidents in Greater London, in the United Kingdom, by using the Kernel Density Estimator (KDE) and a 5-year database from 1999 until 2003.

Similarly, Schuurman et al. (2009) studied the hotspots of traffic accidents involving pedestrians in Vancouver, Canada, also by adopting the KDE. From the defined hotspots, they had verified a range of features of the built environment, such as road infrastructure and land use, thus identifying their influence on the intensity of those accidents.

In this context, Druck et al. (2004) emphasize the importance of a density analysis to verify systematic patterns of a selected spatial data. According to them, the identification of patterns, such as hotspots, implies that individual occurrences, when in proximity, may be associated with common causes, that they are spatially dependent [Anderson 2009]. Therefore, it is noticeable that the spatial analysis of traffic accidents is an important tool to understand the effects caused by different urban features, thus providing valuable outcomes for a safer urban planning and design.

3. Methodology

3.1. Determination of Traffic Accident Hotspots

For the empirical density analysis, it was adopted a database of traffic accidents provided by the Life in Transit Project (in Portuguese, *Projeto Vida no Trânsito*) from the Ministry of Health of the Brazilian Government. This database englobes a georeferenced group of fatal traffic accidents occurred during the period 2010-2016 in the Municipality of Curitiba. In the context of this study, there were analyzed uniquely the occurrences involving pedestrians.

First, in order to identify the hotspots of the analyzed traffic accidents, their spatial distribution was determined on GIS software by using the Kernel Density Estimator (KDE). This methodology is widely adopted in a range of studies, such as those described in the previous section [Anderson 2009; Schuurman et al. 2009]. The KDE method uses two inputs in order to conduct the analysis: the spatial data, as a group of georeferenced points – in this study, it refers to the traffic accidents – and the bandwidth.

It is important to highlight that several authors [Anderson 2009; Schuurman et al. 2009; Hashimoto et al. 2016] comment on the subjectivity in the definition of the bandwidth for the accurate analysis of traffic accidents. It can be observed that its variation can lead to significant changes in results, as pointed out by Druck et al. (2004). While a low value can create an irregular surface, a high bandwidth should make it softened. On this study, a 200-meter bandwidth was defined based on similar analysis carried out in previous publications.

Finally, after applying the KDE, a 1-meter spatial resolution raster file was created as an output, highlighting the hotspots of fatal vehicle-pedestrian traffic accidents. This raster file is represented in Section 4 of this article.

3.2. Spatial Analysis Grid

In order to allow the correct association of the spatial factors to the corresponding accident intensities, a 300-meter-spatial-resolution grid, in shapefile format, was created covering the whole area of the Municipality of Curitiba so that different values could be attributed to each pixel.

The spatial resolution was determined seeking a balance between low values, which would prevent the effective analysis of the characteristics of the road system, such as the number of intersections and road density, and high values, which would result in a loss of accuracy of punctual data as is the case of the socioeconomic index.

After creating the grid, a filtering process was conduct in order to eliminate the pixels located over areas that are not urbanized. This process was conducted taking into consideration the 2015 Zoning Plan of the Municipality of Curitiba, made available by the Research and Planning Institute of Curitiba (IPPUC). In this way, all pixels located in Environmental Protection Areas and industrial, service and military zones were deleted. Figure 1 schematically shows the filtering process, presenting the resulting grid.

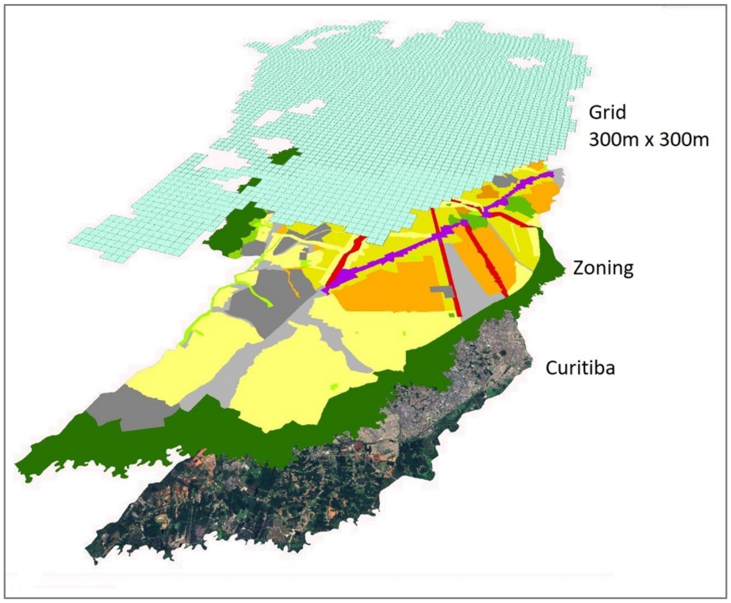


Figure 1. Spatial analysis 300-meter-resolution grid

3.3. Determination of the Independent and Dependent Variables

To effectively map pedestrian vulnerability to traffic accidents in the Municipality of Curitiba, it is crucial to determine which features of the built environment are relevant for the analysis. First, the factors related to the configuration of the road system were analyzed based on the road system shapefile of the Municipality of Curitiba, made available by IPPUC. This file assigns a hierarchical classification to all the roads within the city, associating a higher hierarchy (number 5) to highways and a lower hierarchy (number 1) to local roads.

By taking into consideration this shapefile, it was determined the road density, the road maximum hierarchy and the difference between maximum and minimum hierarchies within each pixel. The road density was calculated as the ratio between total road length within the pixel and its area. In its turn, the maximum hierarchy was extracted using spatial analysis tools and refers to the highest road hierarchy registered in the analyzed pixel. In this way, regions crossed by highways present higher values in relation to regions crossed only by local roads. Finally, the difference between maximum and minimum hierarchies refers to the subtraction between the highest and the lowest values recorded in a pixel, thus indicating a probability of intersections of roads from different hierarchies, which may culminate in a higher susceptibility to traffic accidents.

In addition to the characteristics of the road system, the per capita income and the population density were calculated for each pixel. Both variables were obtained through the census tracts of the 2010 Census, provided by the Brazilian Institute of Geography and Statistics (IBGE). The conversion of these values from the shapefile of census tracts to the 300-meter-resolution grid was performed through the following methodology:

a. the census tracts – already containing the values associated with the variables – were converted into a raster file with a spatial resolution of 1 meter;
b. the spatial statistics tool was used to calculate the mean of all 1-meter-resolution pixels inserted in the 300-meter-resolution pixel from the grid. Thus, for each grid pixel, the average of 90,000 values for population density and per capita income was calculated and attributed to it.

Therefore, by following the methodology described above, five features of the built environment were analyzed, calculated and attributed to the grid. Figure 2 shows schematically these variables, already converted to the grid format.

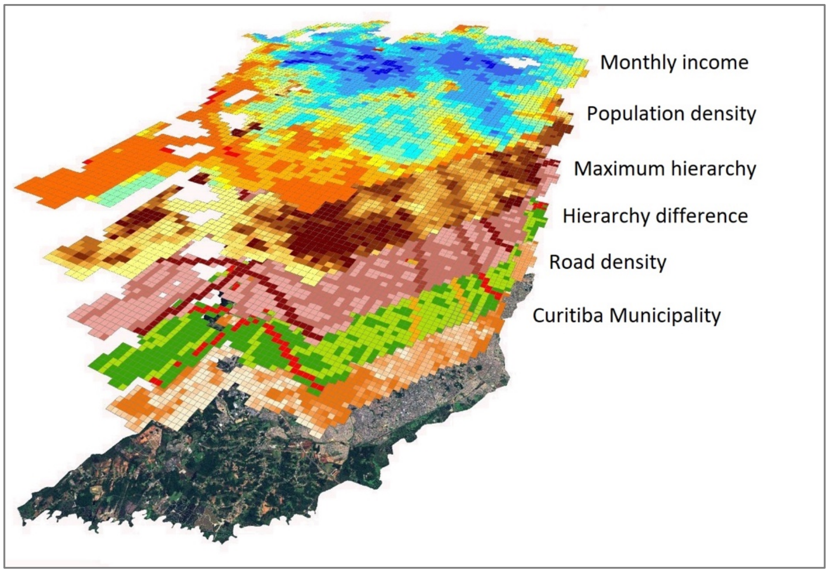


Figure 2. Independent variables attributed to the grid

Finally, the intensity of fatal pedestrian-vehicle traffic accidents – previously obtained by KDE – was also attributed to the grid. The methodology for this assignment consisted of the following steps:

a. initially, we tried to understand the theory of the KDE. This method distributes the unit value of the accident in a circular area, of radius pre-established by the bandwidth, where all the pixels generated around an accident, when summed, produce the unit value of that accident;
b. therefore, to attribute the value corresponding to the intensity of traffic accidents to each 300-meter-resolution pixel from the grid, the raster file of hotspots was analyzed and all 1-meter-resolution pixels within the grid were summed. This means that a total of 90,000 values have been summed to generate a result of the intensity of traffic accidents for each pixel from the grid.

4. Results and Discussion

For this study, a hotspot analysis was conducted by applying the Kernel Density Estimator and based on the traffic accidents database from the Municipality of Curitiba, as presented in Figure 3. Even though the statistical analyzes that relate spatial variables to the intensity of accidents in the region have not been concluded, it is already possible to visually identify spatial patterns relevant to the research.

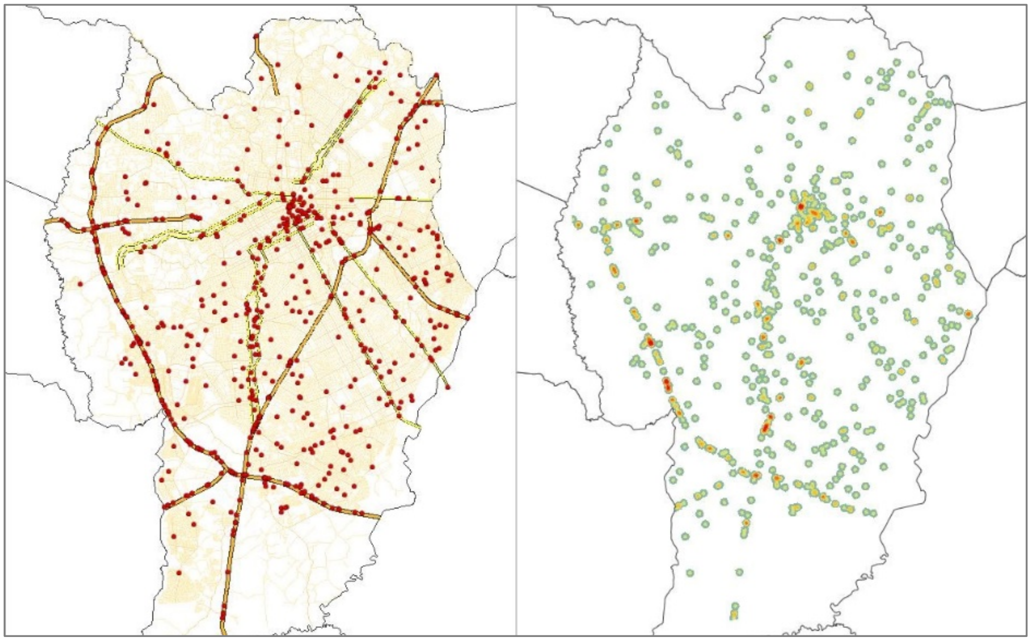


Figure 3. Pedestrian injuries database (in red, left) and hotspots obtained through Kernel Density Estimator (right)

There is a high concentration of fatal traffic accidents involving pedestrians in the central region of Curitiba, which has a higher population density. Additionally, hotspots (shown in red in the figure on the right) are also identified along highways and structuring streets, especially in the south, which is a region characterized by a lower per capita income when compared to the center and north of the city.

In this way, it is believed that through statistical analysis, especially a multiple linear regression associating the dependent variable (intensity of traffic accidents) with the independent variables (features of the built environment), it will be possible to map the exposure of pedestrians to traffic accidents within Curitiba in the next stages of this study.

5. Conclusion

This article seeks to understand the influence of the features of the built environment on the intensity of vehicle-pedestrian traffic accidents within Curitiba. As a database for the study, a georeferenced group of traffic accidents registered during the period 2010-2016 were spatially analyzed through KDE. Finally, the identified hotspots were associated with spatial factors of the urban area, including the characteristics of the road system, the per capita income and the population density.

In the next steps, the study aims to use statistical analysis of the attributes obtained through this methodology in order to map the pedestrian exposure to traffic accidents within Curitiba. This analysis includes the use of multiple linear regression, associating the intensity of fatal vehicle-pedestrian traffic accidents (dependent variable) with the features of the built environment (independent variables).

6. Acknowledgement

A special thanks to Márian, my internship supervisor, who provided me a little more time for imaging and making this article possible.

7. References

ANDERSON, Tessa K. Kernel density estimation and K-means clustering to profile road accident hotspots. Accident Analysis and Prevention. Brisbane, p. 359-364. maio 2009.

BRASIL. ASSOCIAÇÃO NACIONAL DE TRANSPORTES PÚBLICOS. Sistema de Informações da Mobilidade Urbana: Relatório Geral 2014. São Paulo, 2016.

BRASIL. MINISTÉRIO DA SAÚDE. Sistema de Informações sobre Mortalidade. Brasília, 2018.

CENSO DEMOGRÁFICO 2010. Características da população e dos domicílios: resultados do universo. Rio de Janeiro: IBGE, 2011.

DRUCK, S.; CARVALHO, M.S.; CÂMARA, G.; MONTEIRO, A.V.M. (eds). Análise Espacial de Dados Geográficos. Brasília, EMBRAPA, 2004.

HASHIMOTO, Seiji et al. Development and application of traffic accident density estimation models using kernel density estimation. Journal Of Traffic And Transportation Engineering. Xi'an. p. 262-270. jun. 2016.

SCHUURMAN, Nadine et al. Pedestrian injury and the built environment: an environmental scan of hotspots. Burnaby: BMW Public Health, 2009.

Spatial Database to Store Years of Earth Observation Information Obtained from Field Expeditions in the Amazon

Gabriel Crivellaro Gonçalves[1], Lucas Augusto, Maria Isabel Sobral Escada[1], Silvana Amaral [1]

[1] Remote Sensing – National Institute for Space Research (INPE)
Av. dos Astronautas,1758, CEP 12227-010, São José dos Campos, SP, Brazil.

gabriel.goncalves@inpe.br, lucasaugusto@gmail.com, {isabel, silvana}@dpi.inpe.br

Abstract. *Earth observation information obtained through remote sensing and field expeditions generate large volumes of data. These information are usually stored on physical drives such as hard drives. However, this data must be available to all researchers and be stored at safe places. This paper presents the preliminary results of modeling a spatial database (SDB), and some tools for storing Amazonian field expedition information of land use and localities features. Now, an information subset is already being stored in the SDB using the web platform for management. This work contributes to highlights the relevance of database approach, making the management, access and query of this information easier to research purposes.*

1. Introduction

Spatial Databases (SDB) are computer systems created for store, manage, read and write information with spatial attributes in a database. Unlike a common database (DB), SDB is capable to model and store spatial attributes such as location, extent and shape of geographical features. SDB also enable different spatial operation with spatial data, like union, intersection, spatial cutout, overlay operation and others [WINSTANLEY, 2009]. These are capabilities very useful for different knowledge areas, such as land-use and land cover studies based on remote sensing and spatial data.

For the last 10 years, the Laboratory for the Investigation of Socio-Environmental Systems (LiSS) from National Institute for Space and Research (INPE) has been studying the processes involved in Land Use and Land Cover Changes (LULCC) through Remote Sensing (RS) in the Brazilian Amazon, specially at Pará state, as described in the respective research reports [AMARAL et al., 2012; DAL´ASTA et al., 2011; ESCADA et al., 2013]. Remote Sensing data requires validation *in loco* and additional information to understand the LULCC process.

A large amount of data has been generated and obtained for each field expedition: remote sensing images, vector files, spreadsheets, photos, audio recordings, videos and GPS points and tracks. A total of 700 gigabytes of information is already saved in physical store units like hard drives. However, this strategy of information storage is risky, once this hardware is defective, all information may be lost. On the other hand, since several researchers in the group need to access the data, is it essential to have an integrated platform from where everyone could access the information.

To cope with this problem, this paper describes the spatial database model developed to store field-acquired information and remote sensing data, and its initial results. The main objective was to build an SDB to store all the information in a practical and secure way, based on web service access for information insertion, query and download, also accessed via geographic information systems.

2. Data

A database for fieldwork purposes should handle raster data, as satellite images, vector files, such as location points and trajectories, and also documentary data such as spreadsheets, videos and photographs. Figure 1 describes the present composition of fieldwork data according its data type. For the final database, the data amount should be reduced, like photographs, by selecting only the most significant ones.

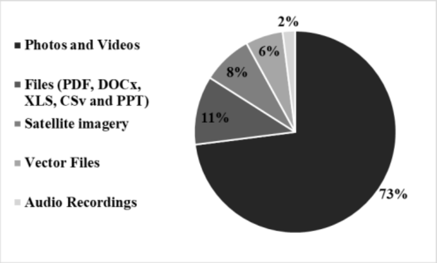
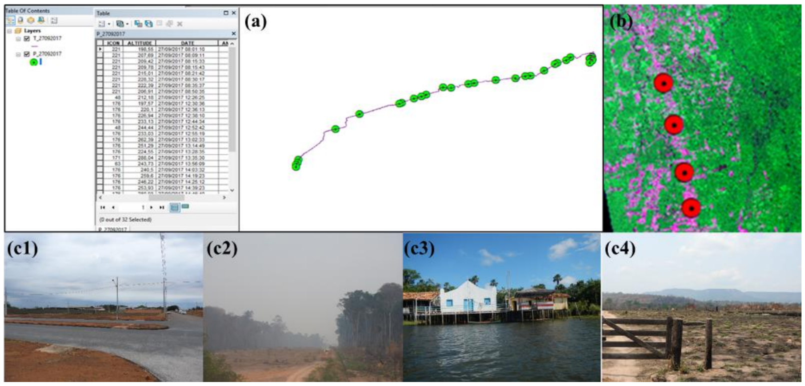


Figure 1.Type of fieldwork data available from 2008 to 2018.

The time and spatial reference of the data is central: information is referred to localities, towns, cities, institutions and trajectories, at different expeditions and dates. For each place visit on the field, indexed by a GPS coordinate, it is usual to have information from one or more questionnaire, with its audio records, and a physical description based on photographs and notes (Figure 2). After the fieldwork, all information should be organized in worksheets and standardized to compose the SDB.



P_27092017

	ICON	ALTITUDE	DATE	AN
▸	221	198,55	27/09/2017 08:01:10	
	221	207,69	27/09/2017 08:09:11	
	221	209,42	27/09/2017 08:15:33	
	221	209,78	27/09/2017 08:15:43	
	221	215,01	27/09/2017 08:21:42	
	221	228,32	27/09/2017 08:30:17	
	221	222,39	27/09/2017 08:35:37	
	221	206,91	27/09/2017 08:50:35	
	48	212,18	27/09/2017 12:26:25	
	176	197,57	27/09/2017 12:30:36	
	176	220,1	27/09/2017 12:36:13	
	176	226,94	27/09/2017 12:38:10	
	176	233,13	27/09/2017 12:44:34	
	48	244,44	27/09/2017 12:52:42	
	176	233,03	27/09/2017 12:55:19	
	176	262,39	27/09/2017 13:02:33	
	176	251,29	27/09/2017 13:14:49	
	176	224,55	27/09/2017 13:28:35	
	171	288,04	27/09/2017 13:35:30	
	63	243,73	27/09/2017 13:56:09	
	176	240,5	27/09/2017 14:03:32	
	176	259,6	27/09/2017 14:19:23	
	176	246,22	27/09/2017 14:25:12	
	176	253,93	27/09/2017 14:39:23	

(0 out of 32 Selected)

P_27092017

Figure 2. Fieldwork data examples: (a) worksheet with localities (points) information; (b) points to be checked over a reference Landsat image;(c) Land use and cover photos – urban area (c1), deforestation (c2), riverside community (c3) and pasture (c4).

3. Spatial Database

From fieldwork data and the needs presented by the users, a conceptual model for the SDB was proposed (Figure 3), considering as data categories: expedition (date), places visited (name and geographical coordinate), expedition raster (RS images), route (trajectory lines), expedition documents, questionnaires (applied to persons or institutions), questionnaire documents (authorization of rights of use of images, and any documents received from the interviewee), questions and answers from questionnaires, communities (name and geographical coordinate of each locality), images (photographs) and interviewed.

All information is organized by expedition, having a start and final date. Any answer for a question in the questionnaire is associated to a location, an expedition, and an interviewee. The photographs, information and documents obtained by the interviewee are also linked to the answer. This way, queries in the web platform can be performed consulting by date, expedition, locality, interviewee, type pf questionnaire, questions or answers.

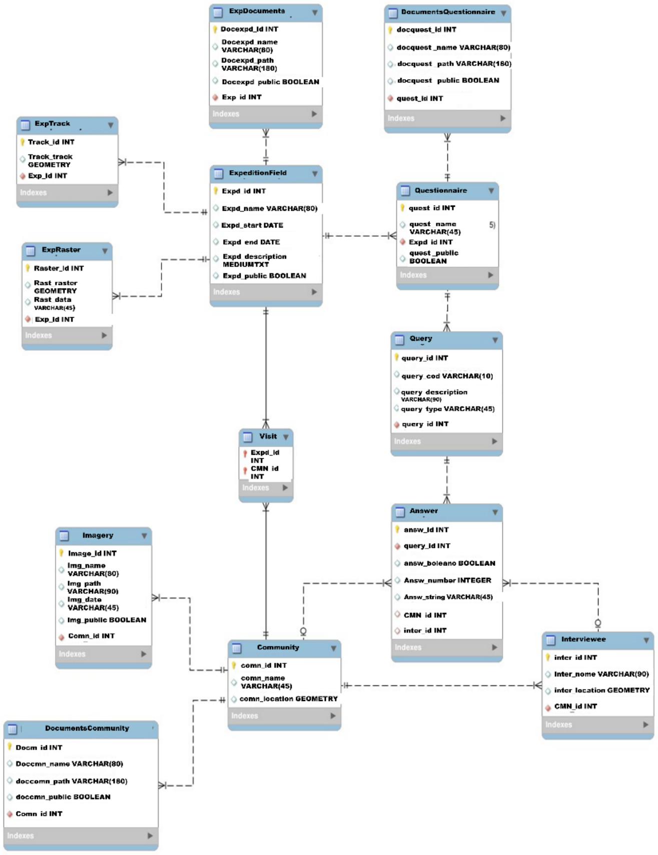


Figure 3. Conceptual model of the database.

From the SDB project specifications, the system should manage all the fieldwork data and provide additionally tools for analysis (spatial and attributes queries) and easy insertion of new data. First, the PostgresSQL was chosen as DBMS (Database Management System) because it is open source, stable, has great community support, and it contains the PostGIS extension, to manipulate geographic objects within an object-relational database. The extension supports from storing multiple geometries to advanced operations of intersections and polygon joins [QUEIROZ & FERREIRA, 2006].

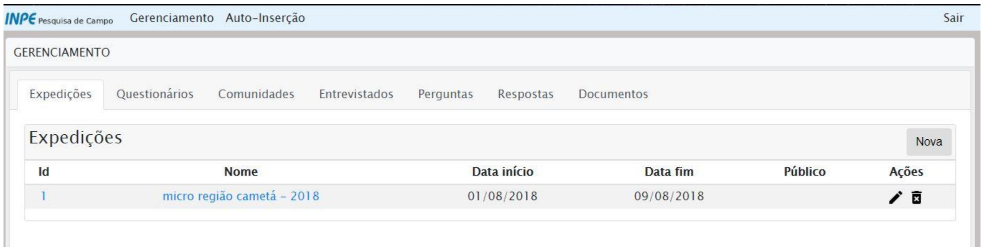
Then, a web system was proposed as user interface for the SDB. This interface was developed in Java language, using the Web Servlet Container. Such a system has a Restful API for data access for analysis, and a data management Web interface, which allows the CRUD (Create Read Update Delete) of objects to be made [PILGRIM, 2013].

Another feature of the system is easy-insertion of the data. The spreadsheets are loaded into the system, which extracts the originated entities, validates if the spreadsheet information is correct, and finally enters the database. All system persistence is managed by the Hibernate Object Relational Mapper (ORM). In it, it maps the objects to the respective tables in the database [IRELAND et al., 2009].

4. Web service platform and SDB management

In the web platform developed for the SDB management it is possible to insert, consult and visualize the fieldwork information. There is an initial user management interface to control and authorize the access by login and password. Once logged in, the page "management" enables the access to seven tabs: Expeditions, Questionnaires, Communities, Interviewees, Questions, Answers and Documents (Figure 4). These pages enable access to all the information in the SDB.

The "Expedition" tab enables insertion and consulting of expeditions, containing information about starting and ending date, and a brief description. The "Questionnaire" tab enables insertion and consulting of applied questionnaires, and associates each one to an expedition. The "Communities" tab enables insertion and consulting the locations in the database, including the name of the place, latitude, longitude and a brief description. The "Interviewed" tab contains some reference about the institution or person interviewed, like name, contact, latitude and longitude of the interview site and a brief description. The "Questions" tab contains the questions asked in the questionnaires. The "Answers" tab contains the answers, enabling consulting. And the "Documents" tab enables insertion and consulting of photographs, video, PDF files, MSOffice files, satellite images and vector files.

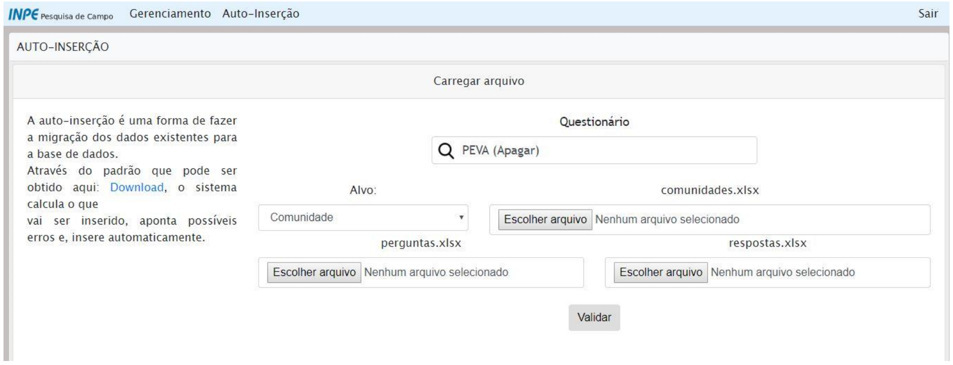


Expedições

Id	Nome	Data início	Data fim	Público	Ações
1	micro região cametá - 2018	01/08/2018	09/08/2018		

Figure 4. Overview of user interface of the Web Service access to the Spatial Data Base

An interface for automatic insertion (CRUD Wizard) was created to facilitate the data insertion into the database (Figure 5). For this option, data should be first sumarized on three files: geolocations, questions of the questionnaires, and answers. The first worksheet (Figure 6a) creates the communities and assigns a file ID to each community with its geographic coordinates. The second worksheet (Figure 6b) resumes the questions asked in that questionnaire with the code created for them, and the third worksheet (Figure 6c) contains the answers, with the question ID and the FID of the community. At CRUD Wizard page, a zip file containing the data template is available for download.



A auto-inserção é uma forma de fazer a migração dos dados existentes para a base de dados.
Através do padrão que pode ser obtido aqui: Download, o sistema calcula o que
vai ser inserido, aponta possíveis erros e, insere automaticamente.

Questionário: PEVA (Apagar)

Alvo: Comunidade

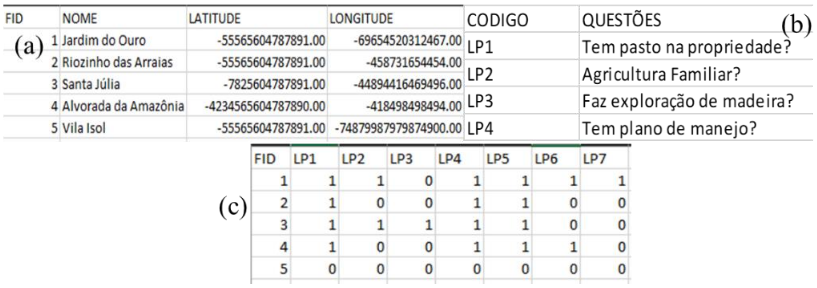
comunidades.xlsx: Nenhum arquivo selecionado

perguntas.xlsx: Nenhum arquivo selecionado

respostas.xlsx: Nenhum arquivo selecionado

Figure 5. Interface for automatic data insertion into the database (CRUD Wizard).

To use CRUD Wizard, the tabular data must be standardized as shown in Figure 6. Due to the limitation of characters in the header of DBF worksheets, file with the information in shapefiles, it is necessary to create codes for the questions. The questions are in the columns and the communities in the lines, so an FID is assigned for each location (a) and code for each question (b). In the response worksheet, there will be the question code and the community FID (c).



FID	NOME	LATITUDE	LONGITUDE
1	Jardim do Ouro	-55565604787891.00	-69654520312467.00
2	Riozinho das Arraias	-55565604787891.00	-458731654454.00
3	Santa Júlia	-7825604787891.00	-44894416469496.00
4	Alvorada da Amazônia	-4234565604787890.00	-418498498494.00
5	Vila Isol	-55565604787891.00	-74879987979874900.00

CODIGO	QUESTÕES
LP1	Tem pasto na propriedade?
LP2	Agricultura Familiar?
LP3	Faz exploração de madeira?
LP4	Tem plano de manejo?

FID	LP1	LP2	LP3	LP4	LP5	LP6	LP7
1	1	1	0	1	1	1	1
2	1	0	0	1	1	0	0
3	1	1	1	1	1	0	0
4	1	0	0	1	1	1	0
5	0	0	0	0	0	0	0

Figure 6. Examples of standardize spreadsheets to automatic insertion into the SDB (a) geolocations, (b) questions and (c) answers.

Currently the system is online, the web platform and SDB. There is still some functionality to implement such as Web Map Service (WMS) and Web Features Service (WFS). Also, the creation of a page to make data that can be made public remains. About 200 gb have already been cataloged and are being inserted into the system. Expeditions of 2013,2015 and 2016 have already been cataloged. Only 2015 is already in the database.

5. Final Remarks

This work presents the spatial geographic database created to store data from fieldwork expeditions in the Amazon, carried out by INPE researchers from the LiSS. The database used PostgresSQL management platform, which allows the use of PostGIS. To make database management easier, a web platform was created for inserting information and querying in the SDB. The variety of original tabular information led the project to propose spreadsheet standardization for data insertion into the system.

The logical and physical structure of the SDB is already accomplished. The web platform for management is online and at final test phase. From the 700 gigabytes of the original fieldwork data, 200 GB was already cataloged, organized and it is being gradually inserted in the SDB. The spatial analysis and consulting will be possible when the web map service (WMS) and web feature service (WFS) is completed, and then, data will be accessed straightforward from a GIS.

At the moment, the SDB finally organizes data collected from a large time of providing data storage and data access to the researches, in a safe and integrated environment. This is an essential tool for researches a great ease in accessing the information generated by the group, making the storage more practical and safer since all information is standardized and stored on a server. The SDB is an important achievement, considering the possibility of further make part of these fieldwork information available for public.

Acknowledgment

The authors thank National Institute of Space Research - INPE and MSA Project - BNDES (Number process: 1022114003005) - Sub-Project 5 - Trajectories of Patterns and Processes in the Characterization of New Dynamics of Deforestation in the Amazon for infrastructure and financial support.

References

AMARAL, S. ; BRIGATTI, N. ; DAL'ASTA, A. P. ; ESCADA, M. I. S. ; SOARES, F. R. . "Tem fofoca na currutela" Núcleos urbanizados e uso da terra de Alta Floresta (MT) ao Crepurizão (PA) na Transgarimpeira. São José dos Campos: INPE, 2012.

DAL'ASTA, A. P. ; ESCADA, M. I. S. ; BRIGATTI, N. ; AMARAL, S. . Núcleos de ocupação humana e usos da terra entre Santarém e Novo Progresso, ao longo da BR-163 (PA). São José dos Campos: INPE, 2011.

ESCADA, MARIA ISABEL SOBRAL ; DAL&APOS ; SOARES, F. R. ; ANDRADE, P. ; PINHO, C. M. D. ; MEDEIROS, L. C. C. ; CAMILOTTI, V. L. ; FERREIRA, V. C. ; AMARAL, S. . Infraestrutura, Serviços E Conectividade Das Comunidades Ribeirinhas Do Arapiuns, PA. São José dos Campos, SP: INPE, 2013.

IRELAND, CHRISTOPHER & BOWERS, DAVID & NEWTON, MIKE & WAUGH, KEVIN. Understanding object-relational mapping: A framework-based approach. Int J Adv Softw. 2. 2009.

PILGRIM, P. A. "The lifecycle of Java Servlets". Java EE 7 Developer Handbook. Professional expertise distilled. Packt Publishing Ltd. ISBN 9781849687959. Retrieved 2016-06-16. Java Servlets are governed by a web container (a Servlet container). 2013.

QUEIROZ, G. R. FERREIRA, K. R. Tutorial sobre Bancos de Dados Geográficos. GeoBrasil, 2006.

WINSTANLEY, A. C. Spatial Database. International Encyclopedia of Human Geography. P 345-347. 2009.

Utilizando dados georeferenciados para o tratamento do problema *Indoor-Outdoor Detection*

Raissa P. P. M. Souza[1], Fabrício A. Silva[1],Thais Regina M. B. Silva[1]

[1]Universidade Federal de Viçosa - *Campus* Florestal

{raissa.papini,fabricio.asilva,thais.braga}@ufv.br

Resumo. *Com o crescimento do número de usuários de dispositivos móveis, os provedores de serviços móveis estão cada vez mais preocupados com a qualidade, visando atrair novos clientes e reter os atuais. No contexto de telecomunicações, uma informação relevante para ajudar na percepção da qualidade dos serviços é se o usuário está em ambiente aberto ou fechado. Este problema, conhecido como Indoor-Outdoor Detection, já vem sendo abordado na literatura com técnicas de aprendizado supervisionado, em que são necessários dados com rótulos sobre o tipo de ambiente para se treinar um modelo. Neste artigo, é proposta uma solução não-supervisionada que utiliza dados georeferenciados e o nível de sinal para inferir o tipo de ambiente de um usuário móvel. Os resultados preliminares mostram que a solução é promissora em termos de precisão, além de ser simples e de fácil implementação.*

1. Introdução

Nos últimos anos, o número de usuários de dispositivos móveis vem crescendo significativamente. Com isso, cresce também o número de provedores de serviços a esses usuários, sejam de comunicação, de conteúdo ou de entretenimento. Com diferentes possibilidades de serviços similares sendo oferecidos, os usuários estão cada vez mais exigentes com a qualidade dos mesmos.

Em particular para empresas de telecomunicações, saber se um usuário móvel se encontra em ambiente aberto ou fechado é muito relevante para a qualidade dos serviços. Com isso, as operadoras podem identificar e entender os motivos de usuários estarem sofrendo com má qualidade dos serviços, e assim investirem em melhorias de forma mais assertiva. Por exemplo, se vários usuários estão questionando a qualidade do serviço em uma região, e esses usuários estão em ambientes abertos, pode ser por algum problema nas antenas que atendem aquela região. Por outro lado, se esses usuários estão em ambientes fechados (e.g., um Shopping), pode ser necessário a instalação de outras antenas, ou o aumento de potência das existentes.

No entanto, o problema de identificar se um dispositivo está em ambiente aberto ou fechado (conhecido como *Indoor-Outdoor Detection*) não é trivial. Primeiramente, não é possível obter em larga-escala rastros com rótulos (*fingerprints*) para que sejam utilizados [Lakmali e Dias 2008]. Para isso, seria preciso que fossem coletados dados de latitude, longitude, nível de sinal, e o rótulo indicando se em ambiente fechado ou aberto. Considerando a extensão do território nacional, essa tarefa é inviável. Em segundo lugar, a mesma antena celular atende a ambientes fechados e abertos na maioria das vezes, dificultando a classificação de ambientes com base apenas na antena utilizada. Por fim, é inviável a implantação de transmissores estáticos em locais estratégicos de ambientes internos para indicar quando um usuário está em ambientes fechados.

Para resolver esse problema, este trabalho propõe uma solução não-supervisionada que utiliza dados georeferenciados não rotulados. Para isso, o algoritmo proposto recebe como entrada dados históricos de latitude, longitude, célula e nível de sinal, faz agrupamentos de registros próximos em uma mesma célula, e cria um modelo de classificação do tipo de ambiente com base no nível de sinal de cada agrupamento. A partir desse modelo, é possível classificar se um novo registro refere-se a um ambiente aberto ou fechado. A premissa do trabalho é que registros próximos geograficamente e utilizando a mesma célula possuem padrões de distribuição de nível de sinal diferentes para ambientes abertos e fechados. O algoritmo então separa essas distribuições em dois grupos, um para ambientes abertos (com valores de nível de sinal maiores) e outro para ambientes fechados (com valores de nível de sinal menores).

O restante deste trabalho está organizado da seguinte forma. A Seção 2 descreve os principais estudos relacionados encontrados na literatura. Os detalhes da solução e uma análise dos resultados são apresentados nas Seções 3 e 4, respectivamente. Por fim, o trabalho é concluído na Seção 5.

2. Trabalhos Relacionados

Alguns trabalhos utilizam os chamados *fingerprints*, que são medidas de nível de sinal coletadas de diferentes localizações em ambientes fechados. A proposta do trabalho [Lakmali e Dias 2008] utiliza uma base de dados previamente coletada com informações sobre a localização conhecida e o nível de sinal de diferentes antenas, para então estimar a localização de um objeto com base em seu nível de sinal. O trabalho descrito em [Gallagher et al. 2010] utiliza WiFi para localização em ambientes fechados, e GPS para ambientes abertos, em um campus universitário. Já o estudo publicado em [Kuo et al. 2010] propõe uma solução de localização em ambientes fechados que utiliza sensores e a tecnologia ZigBee, fazendo uma separação dos ambientes fechados em zonas.

Outros trabalhos não necessitam de dados detalhados de nível de sinal, mas utilizam a localização de pontos de monitoração fixos. O trabalho [Mizuno et al. 2007] utiliza GPS para ambientes abertos e nível de sinal de *Bluetooth* para fechados. O estudo publicado em [Luo et al. 2011] compara algoritmos de localização utilizando identificação por rádio frequência, que avalia o nível de sinal recebido. Essas soluções requerem a instalação de leitores fixos (i.e.,*beacons*) em locais estratégicos para o bom funcionamento.

Alguns trabalhos são mais elaborados e utilizam diferentes técnicas em conjunto para a localização. Os autores de [Pereira et al. 2011] descrevem uma solução flexível que faz uso de técnicas como registros de localização e nível de sinal, pontos de acesso com localizações conhecidas, e células de operadoras com localizações conhecidas. O trabalho descrito em [Reyero e Delisle 2008] propõe um modelo que visa identificar, a cada momento, qual a melhor alternativa para localizar um dispositivo móvel, com base em sinais de GPS e de pontos de acesso disponíveis na proximidade. Por outro lado, os autores de [Kohtake et al. 2011] utilizam o próprio *chipset* do GPS para a localização de objetos móveis tanto em ambientes abertos quanto fechados.

Outros trabalhos focam em detectar apenas em qual tipo de ambiente o usuário se encontra: fechado ou aberto. No trabalho [Gallagher et al. 2011], para o caso de mudança de ambiente aberto para fechado, é identificada uma redução brusca no sinal do GPS. Caso contrário, são utilizados pontos de transição entre um ambiente e outro (i.e., portas), e um temporizador que contabiliza o período de tempo sem nenhum sinal de localização interna em ambientes abertos. Já o estudo de [Li et al. 2015] utiliza sensores diversos, como de luz, magnetômetro, e sinal de torres celulares para fazer essa detecção de transição entre um ambiente e outro. Por fim, o trabalho apresentado em [Radu et al. 2014] descreve uma solução de aprendizado semi-supervisionado, que utiliza valores de intensidade da luz, hora do dia e nível de sinal como parâmetros do modelo.

Apesar dos bons resultados, essas últimas soluções apresentam alguns pontos fracos. Primeiro, requerem a coleta de dados de sensores menos usuais (e.g., luminosidade), consumindo recursos do dispositivo móvel. Além disso, a análise com base no nível de sinal não utiliza a informação de georeferenciamento, considerando somente a variação do nível para detectar uma troca de ambiente. Por fim, alguns trabalhos requerem que um conjunto de dados rotulados seja fornecido para treinamento, o que inviabiliza a sua utilização em escala. A solução proposta neste artigo também visa identificar se um usuário móvel está em um ambiente aberto ou fechado. Porém, faz uso apenas do nível de sinal e da geo-localização do dispositivo, criando um modelo não-supervisionado que não necessita de dados rotulados para funcionar.

3. Solução Não-supervisionada

Seja c_i=<lat,lng,$celula$,$sinal$> os dados contextuais do acesso i de um usuário móvel em uma localização definida pela sua latitude e longitude, as informações de acesso à rede celular definidas pelo identificador da célula de acesso, e a qualidade do sinal definido pela força do sinal. O objetivo é inferir se o usuário está em um ambiente fechado (I, do inglês *Indoor*) ou aberto (O, do inglês *Outdoor*).

A solução proposta neste artigo visa utilizar dados históricos não-rotulados (i.e., não requer a informação de qual ambiente se encontra para o treinamento de um modelo supervisionado) para resolver o problema. Em linhas gerais, a solução funciona da seguinte maneira:

- Para criar o modelo de aprendizado, é utilizado um histórico de dados R contendo vários registros r_i=<lat_i,lng_i,$celula_i$,$sinal_i$> de acesso com as informações de latitude, longitude, célula, e nível de sinal;
- Para reduzir a abrangência de um conjunto de acessos próximos, a precisão da latitude e longitude é reduzida para 3 casas decimais, fazendo com que os registros com mesmas localizações e células estejam a aproximadamente 100 metros de distância no máximo;
- Foram criados grupos G_k em que $r_i \in G_k$ se $r_i[lat, lng, celula]$=$r_j[lat, lng, celula]$ $\forall r_j \in G_k$. Para que o treinamento seja estatisticamente confiável, foram considerados somente os grupos G_k em que $|G_k|$>=30;
- Para cada grupo G_k formado por valores únicos de latitude, longitude e célula, é aplicado um algoritmo de agrupamento que separa os níveis de sinal em duas categorias distintas. A premissa básica desse passo é que, em uma mesma localidade (dentro de um raio de 100 metros alcançado pela redução da precisão da latitude

e longitude) e com a mesma célula, teremos duas categorias distintas de nível de sinal, sendo que o conjunto com menores valores tendem a indicar acessos em ambientes fechados, e o conjunto com valores maiores em ambientes abertos.

Para classificar um acesso, os dados de latitude, longitude e célula são utilizados para recuperar as duas categorias de agrupamento criadas na etapa de treinamento. Então, verifica-se, dentre as duas categorias (i.e., de ambientes abertos ou fechados) qual se aproxima mais do valor do nível de sinal do acesso de entrada. Nesse ponto, duas abordagens foram avaliadas:

1. *Predição Original*: utiliza a predição original do algoritmo de agrupamento, em que a categoria com centro mais próximo ao valor do nível de sinal de entrada é alocada a ele.
2. *Predição com Tratamento de Fronteira*: utiliza a predição original do algoritmo de agrupamento, mas considera como *Desconhecido* caso o registro esteja equidistante, com uma margem de erro, dos dois centros das categorias.

A solução *Predição Original* funciona muito bem quando as categorias são bem separadas, não ocorrendo nenhum registro de fronteira (i.e., valor do nível de sinal está praticamente equidistante dos dois centros). Porém, quando um valor que se aproxima das duas categorias é encontrado, a predição irá associa-lo ao mais próximo, mesmo que seja por uma diferença muito pequena. Esta estratégia cria um problema de precisão pois, apesar de possuir uma revocação de 100%, pode resultar em uma classificação errônea de registros de fronteira.

A solução *Predição com Tratamento de Fronteira*, por outro lado, visa resolver esse problema, atribuindo a esses registros a categoria de *Desconhecido*. Com isso, o objetivo é aumentar a precisão (i.e., os registros classificados serão mais corretos), a um preço de se reduzir a revocação (i.e., menos registros serão classificados). Essa solução apoia-se na premissa de que uma classificação incorreta tem um impacto mais negativo que uma não-classificação, pois ações podem ser tomadas de forma indevida. Em outras palavras, erros do tipo falso-positivo e falso-negativo são mais críticos do que simplesmente não classificar um acesso.

4. Resultados

Para avaliar a proposta, foi utilizado um conjunto de dados real de milhares de usuários de todo o Brasil, contendo 8.878.370 registros. Desses registros, foram encontrados 3.118.305 combinações diferentes de <latitude, longitude, célula>, considerando que a latitude e longitude tiveram suas precisões reduzidas para três casas decimais. Para essas combinações, descartamos aquelas que possuem menos de trinta registros, para que os resultados sejam estatisticamente válidos de acordo com o Teorema Central do Limite, resultando em 25.497 grupos para análise. Para cada uma dessas combinações restantes, foi utilizado o algoritmo *K-Means* para separar os níveis de sinal em duas categorias distintas. Após essa etapa, o modelo de aprendizado está criado.

Para avaliar a qualidade da solução, foram separados e rotulados manualmente 177 acessos aleatórios que não fizeram parte do conjunto original de treinamento. Para a *Predição Original*, o mesmo algoritmo *K-Means* foi utilizado para classificar os registros desconhecidos. Para a *Predição com Tratamento de Fronteira*, o algoritmo do *K-Means* foi adaptado para tratar os acessos de fronteira como *Desconhecidos*. Para isso, foram

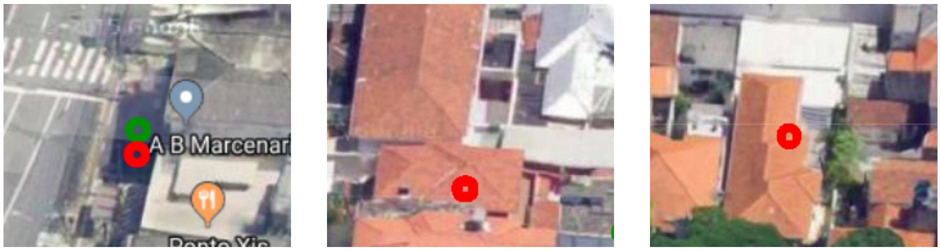


Figura 1. Em vermelho estão exemplos de registros classificados incorretamente pela Predição Original, mas que foram inferidos como *Desconhecidos* pela *Predição com Tratamento de Fronteira*

considerados de fronteira se a diferença da distância do nível de sinal para o centro das duas categorias (i.e., ambiente fechado e aberto) for menor que 20%. Foi preciso que a margem de erro fosse ajustada a este valor por haver locais com muitos prédios e árvores, o que poderia interferir na precisão da análise.

A Figura 1 ilustra alguns dos resultados avaliados. Nela são representados em verde pontos em que a *Predição Original* acertou na classificação, e em vermelho pontos em que a mesma solução errou na classificação. Nestes mesmos casos, a *Predição com Tratamento de Fronteira* classificou como *Desconhecidos* os registros que foram classificados de forma errônea pela *Predição Original*.

A partir dos resultados encontrados, foi possível construir uma matriz de confusão, que pode ser vista na Tabela 1. Observou-se então que, apesar de a quantidade de registros classificados pela *Predição com Tratamento de Fronteira* ter diminuído, isso somente fez com que se reduzisse o número de erros do tipo falso-positivo e falso-negativo, não interferindo muito nas classificações corretas da *Predição Original*.

Tabela 1. Matriz de Confusão da Classificação das Soluções

	Predição Original		***Predição com Tratamento de Fronteira***	
	Fechado	***Aberto***	***Fechado***	***Aberto***
Fechado	44	29	44	21
Aberto	36	68	28	65

Com base nos valores apresentados na Tabela 1, foi possível calcular a porcentagem de precisão e revocação de cada uma das soluções. Para a *Predição Original* observou-se um total de 100% de revocação e 63,27% de precisão. Já para a *Predição com Tratamento de Fronteira* foi obtido um total de 89,26% de revocação e 68,98% de precisão. Estes últimos dados confirmam a hipótese de aumento da precisão em virtude da não-classificação de alguns registros de fronteira.

5. Conclusão

Este trabalho apresentou uma solução não-supervisionada para o problema *Indoor-Outdoor Detection*, que identifica o tipo de ambiente, se aberto ou fechado, que um usuário móvel se encontra durante um acesso a algum serviço. A solução requer as informações de latitude, longitude, célula e nível de sinal, e com base em um modelo não-supervisionado treinado, infere o tipo de ambiente. Os resultados preliminares são promissores, sendo que foi alcançada uma boa precisão. Além disso, a solução é simples e de fácil implementação. Como trabalhos futuros, pretende-se avaliar a solução com um conjunto maior de dados. Também é importante tratar casos em que não seja possível separar os valores de nível de sinal em dois grupos, por serem muito similares. Por fim, a aplicação de outras técnicas de agrupamento devem ser avaliadas.

Referências

Gallagher, T., Li, B., Dempster, A. G., e Rizos, C. (2011). Power efficient indoor/outdoor positioning handover. In *Proceedings of the 2nd International Conference on Indoor Positioning and Indoor Navigation (IPIN11)*.

Gallagher, T. J., Li, B., Dempster, A. G., e Rizos, C. (2010). A sector-based campus-wide indoor positioning system. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pages 1–8. IEEE.

Kohtake, N., Morimoto, S., Kogure, S., e Manandhar, D. (2011). Indoor and outdoor seamless positioning using indoor messaging system and gps. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN2011), Guimarães, Portugal*, pages 21–23.

Kuo, W.-H., Chen, Y.-S., Jen, G.-T., e Lu, T.-W. (2010). An intelligent positioning approach: Rssi-based indoor and outdoor localization scheme in zigbee networks. In *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, volume 6, pages 2754–2759. IEEE.

Lakmali, B. S. e Dias, D. (2008). Database correlation for gsm location in outdoor & indoor environments. In *Information and Automation for Sustainability, 2008. ICIAFS 2008. 4th International Conference on*, pages 42–47. IEEE.

Li, M., Zhou, P., Zheng, Y., Li, Z., e Shen, G. (2015). Iodetector: A generic service for indoor/outdoor detection. *ACM Transactions on Sensor Networks (TOSN)*, 11(2):28.

Luo, X., O'Brien, W. J., e Julien, C. L. (2011). Comparative evaluation of received signal-strength index (rssi) based indoor localization techniques for construction jobsites. *Advanced Engineering Informatics*, 25(2):355–363.

Mizuno, H., Sasaki, K., e Hosaka, H. (2007). Indoor-outdoor positioning and lifelog experiment with mobile phones. In *Proceedings of the 2007 workshop on Multimodal interfaces in semantic interaction*, pages 55–57. ACM.

Pereira, C., Guenda, L., e Carvalho, N. B. (2011). A smart-phone indoor/outdoor localization system. In *International conference on indoor positioning and indoor navigation (IPIN)*, pages 21–23.

Radu, V., Katsikouli, P., Sarkar, R., e Marina, M. K. (2014). A semi-supervised learning approach for robust indoor-outdoor detection with smartphones. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 280–294. ACM.

Reyero, L. e Delisle, G. Y. (2008). A pervasive indoor-outdoor positioning system. *JNW*, 3(8):70–83.

Metodologia para classificação subpixel de imagens MODIS com base em classificação de imagem de maior resolução

Noeli A. P. Moreira[1], **Thales S. Körting**[1], **Luciano V. Dutra**[1], **Emiliano Castejon**[1], **Egidio Arai**[2]

[1]Divisão de Processamento de Imagens – Instituto Nacional de Pesquisas Espaciais
Caixa Postal 12227-010 – São José dos Campos – SP – Brazil

[2]Divisão de Sensoriamento Remoto – Instituto Nacional de Pesquisas Espaciais
Caixa Postal 12227-010 – São José dos Campos – SP – Brazil

noeli.geo@gmail.com, thales.korting@inpe.br, dutra@dpi.inpe.br, emiliano.castejon@inpe.br, egidio@dsr.inpe.br

Abstract. *This paper presents preliminary results of a subpixel classification methodology, applied to a MODIS scene, for land cover classification of extended areas. Class proportions are calculated inside cells determined by the MODIS pixel grid placed over a much higher resolution image, which is initially classified with the desired number of land classes. Clustering is applied to the class proportions synthetic image to estimate typical proportions of classes. The typical proportions map areas are used as region of interest (ROI's) over the MODIS image and then classified by minimum euclidean distance and maximum likelihood rule. The resulting large area map of typical proportions showed consistent result and good agreement with test areas.*

Resumo. *Este trabalho apresenta resultados preliminares de uma metodologia de classificação subpixel, aplicada a uma cena MODIS, para classificação de cobertura vegetal em áreas estendidas. As proporções de classe são calculadas dentro de células determinadas pela grade de pixels do MODIS, colocada sobre uma imagem de resolução muito maior, que é inicialmente classificada com classes básicas de interesse. Clustering é aplicado à imagem sintética de proporções para estimar algumas proporções típicas. O resultante mapa de proporções típicas é, então, usado como um conjunto de regiões de interesse (ROI) sobre a imagem MODIS que é classificada pelas regras de mínima distância euclidiana e máxima verossimilhança. O mapa de grandes áreas obtido com essas proporções típicas mostrou consistência e boa concordância com áreas de teste.*

1. Introdução

Mudanças da cobertura do solo vêm ocorrendo de forma cada vez mais acelerada e nem sempre de forma planejada. Monitorar tais alterações, em nível regional e em grandes áreas, vem sendo um desafio na área do sensoriamento remoto.

O sensor MODIS (*Moderate Resolution Imaging Spectroradiometer*), a bordo das plataformas TERRA e AQUA, faz parte do Sistema de Observação Terrestre (EOS) da NASA e tem como finalidade contribuir com monitoramento global [Justice et al. 2002]. Os produtos são disponibilizados gratuitamente, com alta resolução temporal (revisita diária), entretanto a sua moderada resolução espacial faz com que a mistura espectral de diferentes classes de cobertura dentro de um mesmo pixel dificulte o monitoramento. Técnicas de classificações de subpixel são consideradas apropriadas para estimar proporções de área de cada classe de cobertura [Foody e Cox 1994; Zhang e Foody 1998] e tem sido cada vez mais aplicadas em dados de moderada resolução espacial por representar com maior precisão a mistura dentro de um pixel [Lu, Moran e Hetrick 2011]. Neste sentido, o objetivo desta pesquisa vem sendo desenvolver processos metodológicos para mapeamento subpixel, construindo-se uma imagem com proporções de classes de cobertura em quadrantes relativos à imagem de resolução moderada. O estudo pretende possibilitar a construção de mapas de cobertura com menores índices de incertezas resultantes da mistura espectral e contribuir diretamente, por exemplo, com a criação de mapas com maior detalhamento e frequência temporal para grandes regiões visando auxiliar no monitoramento ambiental.

2. Materiais e Métodos

2.1 Área de Estudo

A área de estudo está localizada entre os municípios de Belterra e Santarém, estado do Pará. Envolve parte da Unidade de Conservação Federal Floresta Nacional do Tapajós, nas proximidades do rio Tapajós, como pode ser visualizada na Figura 1.

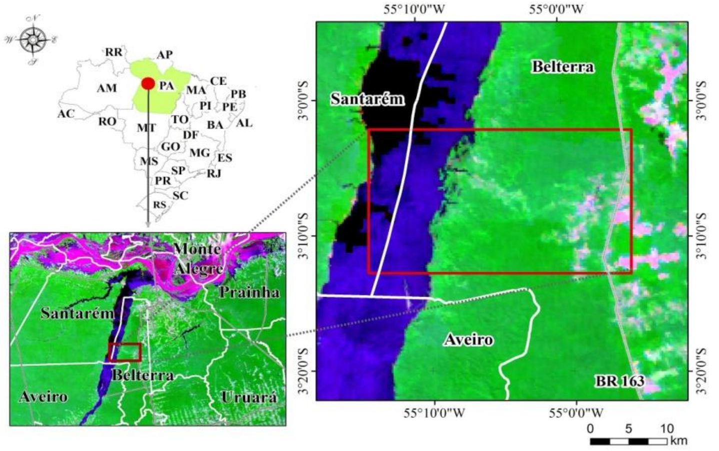


Figura 1. Mapa de localização da área de estudo.

2.2 Base de dados

Foi adquirida para a data 01 de agosto de 2012, uma imagem do sensor ResourceSat-1/LISS, com 23m de resolução espacial no catálogo de imagens gratuito CDRS do INPE. Foi adquirida também, uma imagem para a data 04 de agosto de 2012, do sensor MODIS (MOD09GQ e MOD09GA), com 231m e 462m de resolução espacial respectivamente, ambas do site *NASA's Land Processes Distributed*, referenciadas pelo tile h12v09. O produto MOD09GQ disponibiliza as bandas Red e Nir, enquanto o produto MOD09GA as bandas Blue e Mir. Estas duas bandas foram reamostradas para 231m de resolução espacial, a mesma das bandas Red e Nir.

2.3 Classificação supervisionada sobre imagem de maior resolução

A primeira etapa envolveu o ajuste posicional da imagem ResourceSat-1/LISS sobre a imagem MODIS e sua posterior classificação supervisionada, indicada no fluxograma metodológico na Figura 2. Neste processo foi utilizado o algoritmo de máxima verossimilhança no software ENVI 4.7. Na classificação foram definidas 4 classes de cobertura: floresta, água, solo exposto e pasto e/ou agricultura. As regiões de treinamento foram delimitadas com base em visitas realizadas em campo.

3. Metodologia

3.1 Aplicação do Programa Subpixel para construção da imagem de proporções

Para a construção da imagem de proporções foi elaborado o programa Subpixel, construído na biblioteca TerraLib [Câmara 2000]. Este programa constrói uma grade ao utilizar os parâmetros de resolução espacial e posicional referentes à imagem de menor resolução, neste caso MODIS. Após a construção da grade, é feita a sobreposição da classificação supervisionada feita sobre a imagem de maior resolução, neste caso ResourceSat-1/LISS, para inserir tais proporções de cobertura dentro de cada quadrante de 231m. Este programa permitiu intersectar as duas imagens e criar uma imagem sintética contendo resolução espacial da imagem de menor resolução e porcentagens de cobertura do mapa classificado descrito no tópico 2.3. Esta etapa metodológica está ilustrada na Figura 2.

3.2 Classificação não supervisionada sobre imagem de proporções

Estudos iniciais desta pesquisa demonstraram que a análise direta entre proporções de cobertura e valores espectrais da imagem MODIS, relacionadas por modelo de regressão linear múltipla, não são eficientes. Sendo assim, buscou-se aplicar métodos de agrupamento de classes típicas de proporções para relacionar novamente aos valores espectrais. A maioria dos softwares realizam a classificação não-supervisionada baseada em métodos de agrupamentos, como o K-médias e o Isodata - *Iterative Self Organizing Data Analysis* [Ball and Hall 1967]. Foram realizadas 9 testes envolvendo classificações não supervisionadas, sobre a mesma imagem de proporções, utilizando os dois algoritmos, no software ENVI 4.7. As 9 classificações foram executadas com parâmetros de entrada diferenciados (número mínimo e máximo de clusters e número de iterações distintos) para testar qual dos mapas resultantes representava mais corretamente a área de estudo. Outros testes aumentando tais parâmetros foram aplicados, porém a detecção de clusters típicos foi estabilizada em 11 agrupamentos. Os testes de 1 a 3 foram processados utilizando o algoritmo K-médias e de 4 a 9, utilizou-se o Isodata. Dentre os nove mapas classificados foi escolhido o mapa resultante do teste nº 9 (executado pelo algoritmo Isodata com parâmetros de entrada: 10 a 20 clusters e 10 iterações), por apresentar melhores resultados comparados à realidade da área de estudo. Além disso, separou proporções acima de 90% para as 4 classes de cobertura e englobou diferentes combinações de proporções. Portanto criou o maior nº de clusters (11 no total) caracterizando melhor a heterogeneidade do local. Desta forma, se optou em escolher o mapa resultante do método nº 9, para definição de atributos estatísticos explicados na seção 3.3.

3.3 Definições de parâmetros estatísticos como treinamento para classificador

O mapa contendo 11 clusters típicos de proporções foi sobreposto à imagem MODIS para possibilitar o cálculo dos parâmetros média e matriz de covariância para cada um dos clusters e para cada banda da imagem (Red, Nir, Blue e Mir). A identificação destes parâmetros foram necessários para serem utilizados como indicadores de treinamento para a reclassificação supervisionada de uma região de maior abrangência.

3.4 Reclassificação supervisionada sobre imagem MODIS

Foram construídos dois programas para classificação supervisionada, um utilizando o algoritmo mínima Distância Euclidiana e outro Máxima Verossimilhança, ambos na biblioteca TerraLib. A construção destes programas permitiu realizar o processo de reclassificação supervisionada sobre a imagem MODIS utilizando como treinamento os parâmetros estatísticos definidos anteriormente para cada um dos 11 clusters. Neste processo foram gerados dois mapas reclassificados sobre uma área de maior abrangência, contendo em cada um deles 11 classes de cobertura. Os mapas reclassificados podem visualizados na Figura 4.

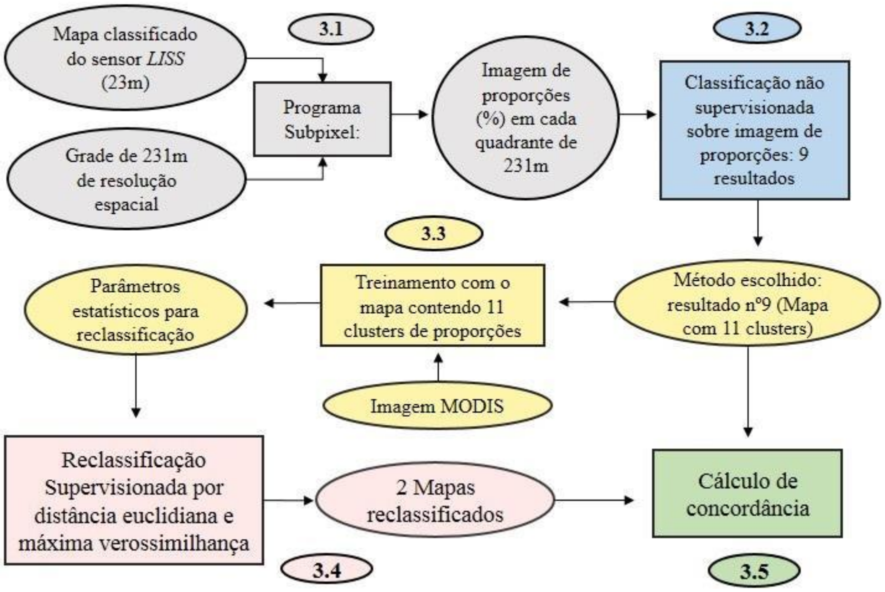

Mapa classificado do sensor *LISS* (23m)
3.1
Programa Subpixel:
Grade de 231m de resolução espacial
Imagem de proporções (%) em cada quadrante de 231m
3.2
Classificação não supervisionada sobre imagem de proporções: 9 resultados
3.3
Parâmetros estatísticos para reclassificação
Treinamento com o mapa contendo 11 clusters de proporções
Método escolhido: resultado nº9 (Mapa com 11 clusters)
Imagem MODIS
Reclassificação Supervisionada por distância euclidiana e máxima verossimilhança
2 Mapas reclassificados
Cálculo de concordância
3.4
3.5

Figura 2. Fluxograma metodológico com as subseções assinaladas.

3.5 Cálculo de Concordância

Para estimar a concordância dos resultados foi utilizado como referência o mapa resultante da classificação não supervisionada para comparação dos mapas reclassificados. Para este processo foi utilizado álgebra de mapas, para realizar a subtração numérica entre cada pixel (valores de 1 a 11) entre os mapas. Este processo permitiu a contagem do total de pixels com valor zero, que representa que os dois pixels eram de clusters do mesmo número (iguais). Os processos metodológicos estão sintetizados pelo fluxograma de execução na Figura 2.

4. Resultados e discussão

4.1 Mapa resultante da classificação não supervisionada

O mapa resultante do teste nº 9 identificou 11 clusters típicos presentes sobre a imagem de proporções e podem ser identificados pela Figura 3. Destes 11 clusters, 4 tiveram proporções acima de 90% (clusters nº 5, 7, 8 e 10) que compreendem respectivamente as classes floresta, água, solo exposto e pasto e/ou agricultura. Além destas 4 classes, foram gerados 7 clusters típicos de proporções (nº 4, 9, 6, 3, 11, 1 e 2) correspondendo à misturas de mais de uma classe em pixels de 231m. A ordem numérica dos 11 clusters foram classificados em ordem decrescente para classe Floresta.

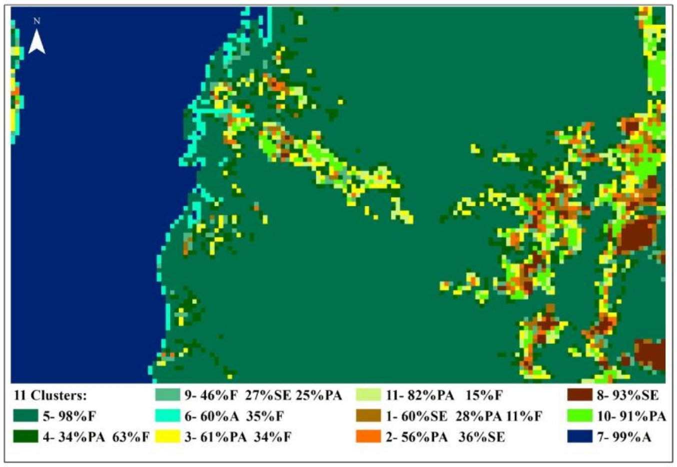


Figura 3. Mapa resultantes do método 9 contendo 11 clusters típicos, sendo 4 clusters de uma única classe e 7 clusters com proporções de mais de uma classe em pixels de 231m. Legenda: A: Água, F: Floresta, SE: Solo exposto, PA: Pasto e/ou Agricultura.

4.2 Reclassificação da imagem MODIS

Após a geração de parâmetros estatísticos resultantes da sobreposição do mapa de classificação não supervisionada do método 9 sobre a imagem MODIS, foi possível reclassificar a imagem identificando as 11 classes de proporções. Os mapas reclassificados e de área com maior abrangência podem ser observados na Figura 4.

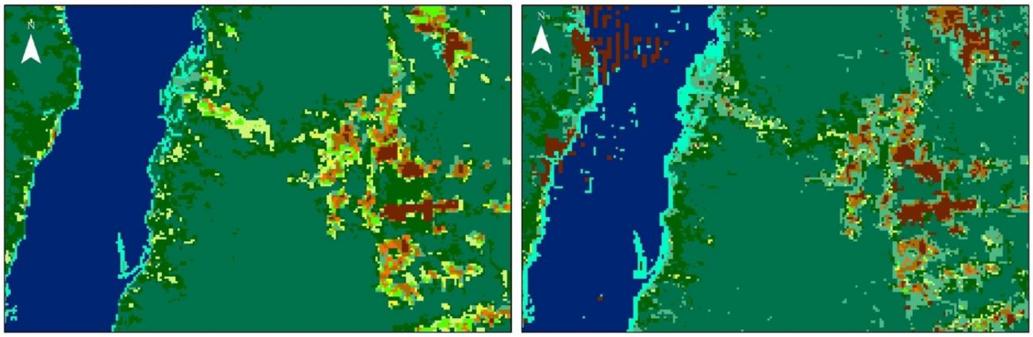


Figura 4. Mapas reclassificados em área de maior abrangência utilizando classificador por distância euclidiana (esquerda) e máxima verossimilhança (direita). Para interpretação das classes de proporções utilizar legenda da Figura 3.

No processo de análise da confiabilidade dos resultados foi calculada a porcentagem relativa de concordância entre o mapa resultante da classificação não supervisionada e os dois mapas reclassificados. Foi estimado um total de 73 % de concordância para o mapa da classificação por distância euclidiana e 72 % para o da classificação por máxima verossimilhança. Os mapas utilizados para cálculo de concordância podem ser visualizados pelas Figuras 3 e 4.

5. Conclusões

O modelo de classificação proposto nesta pesquisa possibilitou reclassificar a área de estudo com proporções subpixel com mapas contendo 4 classes de pixels puros e 7 classes de proporções de cobertura em um mesmo pixel. Os resultados de concordância comparativa entre os mapas demonstram bom desempenho da metodologia, porém, é preciso analisar as regiões de concordância e não concordância.

A metodologia desenvolvida permite classificar grandes áreas, com pixels em resolução moderada e com maior resolução temporal, em percentagens de ocupação de classes, no lugar do método padrão que classifica apenas uma classe por pixel. A metodologia ainda continuará a ser avaliada e depois usada para detecção de mudanças em pequenos intervalos de tempo.

Referências

Ball, G. and Hall, D. (1967) "A Clustering Technique for Summarizing Multivariate Data", *Behavior Science*, v. 12, p. 153-155.

Câmara, G., De Souza, R.C.M., Pedrosa, B. M., Vinhas, L., Monteiro, A.M.V., Paiva, J.A., De Carvalho, M.T. and Gattass, M. (2000) Terralib: Technology in Support of GIS Innovation. In:: Proceedings of the II Brazilian Symposium on GeoInformatics, GeoInfo, São Paulo, Brazil.

Foody G.M., Cox D.P. (1994) - Sub-pixel Land Cover Composition Estimation Using a Linear Mixture Model and Fuzzy Membership Functions. International Journal of Remote Sensing, 15: 619-631. doi: http://dx.doi.org/10.1080/01431169408954100.

Justice, C. O.; Townshend, J.R.G.; Vermote, E.F., Masuoka, E., Wolfe, R.E., Saleous, N., Roy, D.P., Morisette, J.T. (2002) "An overview of MODIS Land data processing and product status. Remote Sensing of Environment", v. 83, n.1-2, Nov. p 3 –15.

Lu, D., Moran, E. and Hetrick, S. (2011) "Detection of Impervious Surface Change with Multitemporal Landsat Images in an Urban-Rural Frontier." ISPRS Journal of Photogrammetry and Remote Sensing : Official Publication of the International Society for Photogrammetry and Remote Sensing (ISPRS) 66 (3): 298–306. doi:10.1016/j.isprsjprs.2010.10.010.

Zhang J., Foody G.M. (1998) - A Fuzzy Classification of Sub-urban Land Cover from Remotely Sensed Imagery. International Journal of Remote Sensing, 19: 2721-2738. doi: http://dx.doi.org/10.1080/014311698214479.

Uso de um Dashboard Geoespacial como ferramenta de suporte para o diagnóstico socioeconômico e ambiental da Reserva Biológica Bom Jesus - Litoral do Paraná

Josemar Pereira da Silva[1], Henrique Guarneri[2], Flávia Cristina Arenas[2], Eduardo Vedor de Paula[3], Silvana Phillipi Camboim[1]

[1] Programa de Pós-Graduação em Ciências Geodésicas
[2]ITTI - Instituto Tecnológico de Transportes e Infraestrutura
[3] Departamento de Geografia
Universidade Federal do Paraná (UFPR) – Curitiba, PR - Brazil

{josemarps,henriqueguarneri,flaviaarenas,silvanacamboim}@gmail.com,
eduardovedordepaula@yahoo.com.br

Abstract. *This paper presents the application of a Geospatial Dashboard for socioeconomic data visualization combined with spatial data of environmental variables organized in graphics and dynamic maps through a Web interface based on the Django Python framework. In this paper, it is presented the results of the application of this tool to support field data collection which was realized in protect area Biological Reserve of Bom Jesus. Using this tool, it was possible the visualization of statistical and spatial indicators that resume the information of the area in a single display panel based on spatial analysis.*

Resumo. *O presente trabalho apresenta a aplicação de uma Dashboard Geoespacial para visualização de dados socioeconômicos combinados a informações ambientais espacializadas dispostos em gráficos e mapas dinâmicos, através de uma interface Web baseada no framework Python Django. Neste trabalho são apresentados os resultados da aplicação desta ferramenta para subsidiar o levantamento de dados de campo realizados na Unidade de Conservação Reserva Biológica Bom Jesus. Com o emprego desta ferramenta foi possível a visualização de indicadores estatísticos e espaciais que simplificam as informações da área num único painel de visualização apoiado em análises espaciais.*

1. Introdução

As Unidades de Conservação (UC) são espaços territoriais, incluindo seus recursos ambientais com características naturais relevantes, de modo a preservar o patrimônio biológico existente, bem como assegurar às populações tradicionais o uso sustentável dos recursos naturais além de proporcionar às comunidades do entorno o desenvolvimento de atividades sustentáveis (BRASIL, 2000).

As UC estão sujeitas a normas e regras especiais, de acordo com a categoria, sendo as mais restritivas definidas com Unidades de Proteção Integral, que permitem apenas o uso indireto dos recursos naturais, e Unidades de Uso Sustentável, que conciliam conservação com uso sustentável dos recursos naturais (BRASIL, 2000).

Conforme SNUC (BRASIL, 2000), as UC de Proteção Integral devem ter uma área de entorno onde as atividades humanas estejam sujeitas a normas e restrições específicas, com o propósito de minimizar os impactos negativos sobre a UC. Esta área é comumente denominada de Zona de Amortecimento e deve ser delimitada com base em critérios técnicos e científicos coerentes com as características da própria UC.

Em 2015, a Universidade Federal do Paraná (UFPR – Departamento de Geografia) e o Instituto Chico Mendes de Conservação da Biodiversidade (ICMBio) assinaram um termo de cooperação com o objetivo de elaborar ferramentas de planejamento e gestão do território para subsidiar a atuação dos analistas ambientais e conselheiros das UC, situadas no litoral norte do estado do Paraná. Como um dos primeiros resultados desta parceria, entre os anos de 2015 e 2016 foi criado e sistematizado um Banco de Dados Geográficos Ambiental (BDG-AMB) da APA Federal de Guaraqueçaba (PAULA *et al.* (2017).

No estágio atual estão sendo implementadas ferramentas de análise espacial para integração dos dados produzidos com a Infraestrutura de Dados Espaciais (IDE), denominada IDE-AMB, que é um repositório de dados acadêmicos com temática ambiental. Essa solução foi implementada utilizando a plataforma de código aberta Geonode, objetivando o intercâmbio e compartilhamento de dados entre UFPR e ICMBio através de um geoportal e geoserviços.

Geonode, que é uma ferramenta de código aberto para implantação de Infraestrutura. No contexto do termo de cooperação é denominada como IDE-AMB, visto que possui características acadêmicas com enfoque temático ambiental, Deste modo, o presente trabalho apresenta os resultados da aplicação de uma de *dashboard* geoespacial desenvolvida em ambiente *Web* para a visualização em tempo real de um conjunto de informações geoespaciais dispostas em gráficos e mapas dinâmicos. Neste trabalho aplicação dashboard é utilizada como instrumento de planejamento e suporte ao diagnóstico socioeconômico das comunidades que vivem na área de entorno e dentro da UC Reserva Biológica (REBIO) Bom Jesus.

2. Área de Estudo e contexto de aplicação

Situada no bioma Mata Atlântica, a REBIO Bom Jesus localiza-se entre limites dos municípios de Antonina, Guaraqueçaba e Paranaguá no Litoral do Paraná. Com uma área de 34.179,74 hectares, a UC apresenta uma grande diversidade biológica, e dezenas de comunidades com culturas e práticas distintas em seu entorno. Conforme SNUC (2000), as Reservas Biológicas têm como objetivo a preservação integral da biota e demais atributos naturais existentes em seus limites, sem interferência humana direta ou modificações ambientais, através de ações de manejo necessárias para recuperar e preservar o equilíbrio natural, a diversidade biológica e os processos ecológicos.

3. Mapeamento preliminar e critérios espaciais para as entrevistas

Com objetivo de subsidiar a regularização fundiária, formação do conselho gestor e zoneamento da REBIO Bom Jesus, foram elaboradas 37 questões objetivas pela equipe de pesquisadores da UFPR em concordância com os analistas ambientais do ICMBio, para um levantamento de dados socioeconômicos e ambientais das comunidades do entorno desta UC.

Para aplicação das entrevistas, inicialmente foi realizado um mapeamento preliminar das comunidades do entorno da REBIO Bom Jesus. Para tanto, foram utilizadas bases digitais (Google e Bing) disponíveis na extensão *Quick Map Services* do software QGIS 3.2. As imagens disponíveis por estes softwares são articulações digitas dos anos de 2014 a 2017. No total, foram identificaram-se 1.050 edificações dentro e no entorno da UC, com exceção da porção sul, a qual não possui acesso terrestre. Ainda em gabinete a partir análises espaciais foram selecionadas e classificadas as edificações que atendiam os seguintes critérios em ordem de prioridade: 1) as edificações que estão dentro da área da REBIO; 2) as edificações internas aos polígonos do Cadastro Ambiental Rural (CAR) e que estão dentro de um *buffer* de 400 metros da REBIO; e 3) edificações dentro dos polígonos do CAR, cujas áreas se sobrepõem a REBIO.

A segunda etapa compreendeu trabalho de campo de reconhecimento, cujo objetivo foi identificar e classificar as edificações que servem de moradia. A validação dos pontos foi realizada por um grupo de 22 entrevistadores distribuídos pelas 20 localidades do entorno da REBIO Deste modo reduziu-se para 138 o número de edificações potenciais para serem aplicadas as entrevistas.

A aplicação das entrevistas ocorreu no âmbito da disciplina de Prática em Planejamento e Gestão Ambiental (GB130), ofertada pelo Departamento de Geografia da UFPR. Inicialmente as entrevistas foram aplicadas em papel e depois lançadas no sistema através de tablets, smartphones e notebooks através de uma intranet montada na sede de operações localizada na porção central da REBIO. O trabalho de campo ocorreu no período de 31 de agosto a 3 de setembro de 2018.

4. Dashboard Geoespacial

Um Dashboard é instrumento de gestão que consiste em um painel de controle que apresenta de forma visual informações gerais através de gráficos, tabelas e mapas cujos dados podem ser oriundos de diferentes bancos de dados.

Para o acompanhamento dos indicadores socioeconômicos das comunidades de entorno da REBIO em tempo real, optou-se neste trabalho em utilizar uma *dashboard* geoespacial *Web* para visualizar, através análises espaciais dispostas em gráficos e mapas, o cruzamento dos dados primários levantados com as informações ambientais espacializadas descritas em PAULA et al. (2017).

Para isso, foi desenvolvida uma plataforma Web integrada para a inserção, armazenamento e recuperação de dados com funções espaciais . Neste sentido, optou-se pelo *framework* Python Django 2.1, que utiliza padrão *model-template-view* (MTV) para o desenvolvimento da interface. O sistema foi modelado a partir da técnica *Object-relacional mapping* (ORM) disponível no Framework Django. Neste experimento foi adotado o banco de dados SQLite com extensão *SpatiaLite*. As consultas espaciais foram implementadas utilizando o módulo *contrib* geodjango e. gerenciamento dos dados pela interface administrativa pelo módulo *django.contrib.admin*.

Foram utilizadas as bibliotecas Javascript: D3.js para controlar os dados e construir gráficos; crossfilters para filtrar e agregar os conjuntos de dados; DC.js para visualização dinâmica e interativa dos dados e integração da D3 com a crossfilters; leaflet para a visualização das análises espaciais em mapas interativos, além de suas extensões leaflet markercluster e leaflet heat,. As bibliotecas *axios* e *queue* proveram suporte para a comunicação do Dashboard com o Django. Todas as bibliotecas foram operacionalizadas através do *framework* Javascript Vue. Todas as bibliotecas utilizadas são livres e estão disponíveis no repositório GitHub.com e p

5. Resultados

A Figura 1 apresenta a localização e os limites da REBIO Bom Jesus no litoral paranaense. Nesta Figura, os polígonos em preto (hachuras) delimitam as porções do território onde se localizam as principais comunidades que habitam o entorno da reserva biológica. Deste modo, a aplicação dos questionários socioeconômicos concentrou-se nessas áreas. A porção sul da REBIO será objeto de uma próxima campanha de entrevistas, devido principalmente a dificuldade de acesso.

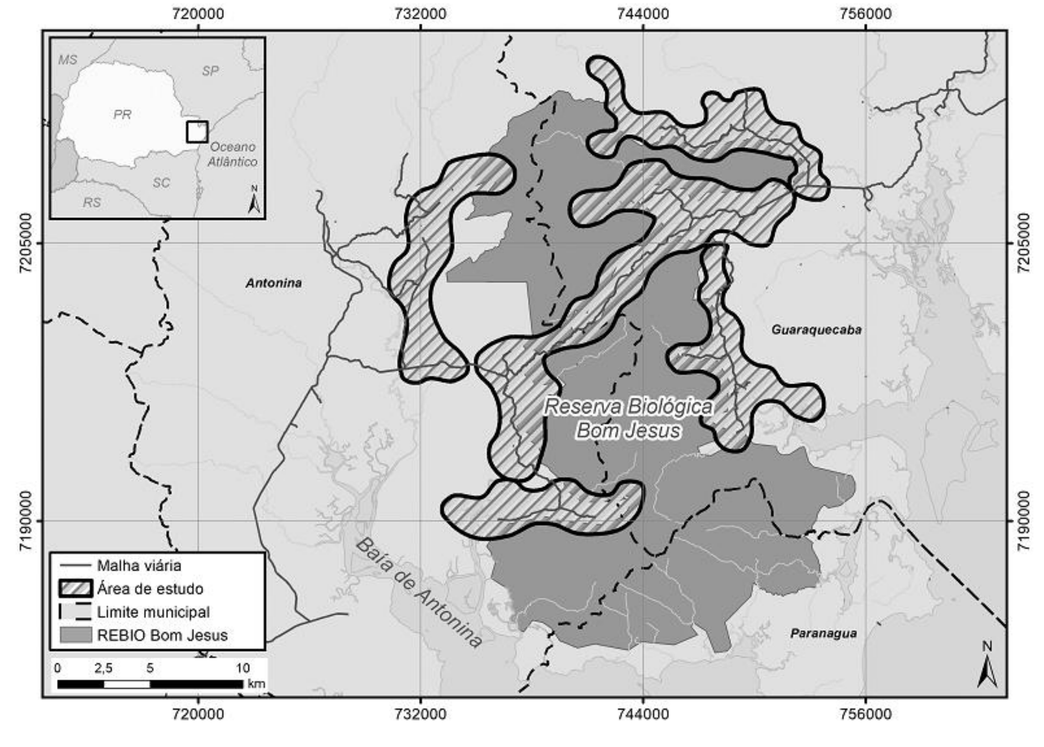
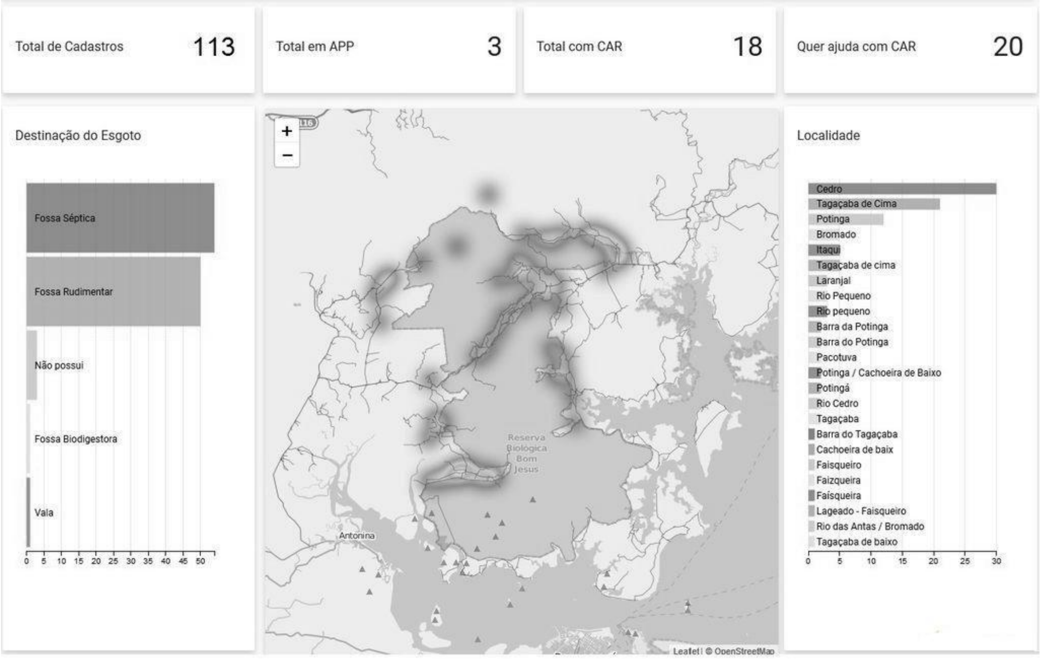


Figura 1. Localização da REBIO Bom Jesus e definição das áreas onde foram aplicadas as entrevistas

No final de cada dia de entrevistas, os dados coletados em campo foram lançados na interface *Web* de cadastro, alimentando desta forma gradativamente a base de dados. Através de rotinas automatizadas as informações e consultas espaciais são atualizadas em tempo real no painel de visualização.

Pelo painel de visualização (Figura 2) o número total de 113 de entrevistas realizadas é apresentado de forma quantitativa. A distribuição destas entrevistas são espacializadas no mapa de calor no centro da tela. Pelo gradiente térmico desta ferramenta é possível visualizar a concentração das áreas com maior quantidade de entrevistas aplicadas. Os painéis "Total com CAR" e "Quer ajuda com o CAR" dispostos no topo da tela do sistema referem-se as respostas totais obtidas nas entrevistas aplicadas. O gráfico "Descrição do Esgoto" quantifica o destino do esgoto. Ao clicar neste gráfico o mapa da área de estudo no centro tela é automaticamente atualizado as com as informações de temáticas.



Total de Cadastros 113

Total em APP 3

Total com CAR 18

Quer ajuda com CAR 20

Figura 2. Interface do Painel de visualização do sistema (Dashboard Geoespacial)

Também pela Figura 2 são apresentadas duas análises espaciais. A primeira análise "Total em APP" disponível no topo da tela, apresenta 3 edificações das 113 pesquisadas, conflitam com Áreas de Preservação de Permanente (APP). A consulta espacial verifica se as coordenadas de cada edificação (ponto) pesquisada se sobrepõem as áreas definidas como APP de rios e nascentes no banco de dados geográfico. A sobreposição entre as edificações e APP certamente é maior, porém não foi possível validar em campo as 1050 edificações devido a limites de logística. A segunda análise espacial "Localidade" apresentada o percentual entrevistas realizadas por comunidade. Todas as análises espaciais foram implementadas no código-fonte do sistema.

A visualização dos dados pela *dashboard* permitiu o acompanhamento de resultados parciais da aplicação das entrevistas auxiliando deste modo a equipe de técnicos e pesquisadores na solução de problemas relacionados à atividade de campo.

De modo geral, o emprego desta tecnologia integrada possibilitou a otimização na distribuição espacial das entrevistas pela área de estudo resultando na economia de tempo em deslocamento das equipes de entrevistadores resultando em melhor aplicação dos recursos financeiros, técnicos e humanos.

6. Conclusões e trabalhos futuros

Neste trabalho foram apresentados os resultados parciais da estruturação de um ambiente integrado *Web* para análise da aplicação de um diagnóstico socioeconômico e ambiental de uma Unidade de Conservação, a partir de critérios prioritários espaciais visualizados em tempo real através de uma *dashboard* geoespacial.

A solução desenvolvida para armazenar os dados obtidos das entrevistas integrado a *dashboard* geoespacial, apresentou resultados satisfatórias, uma vez que permite através de analises espaciais automatizadas a detecção de conflitos ambientais e de ocupação irregular, validação amostral dos pontos em campo e controle da distribuição dos entrevistadores pela área, reduzindo custos técnico, com pessoal e financeiros. Pretende-se adicionar novas funcionalidades ao sistema em trabalhos futuros, como o desenvolvimento de uma aplicação mobile que permita a sincronização de dados sem a necessidade de manuseio de papel, bem como a inclusão de novas consultas espaciais automatizadas e integração com serviços WMS e WFS.

Os resultados obtidos neste trabalho motivam a continuidade do desenvolvimento desta plataforma utilizando código livre, bem como a aplicação em outros levantamentos de campo. Pretende-se disponibilizar o código-fonte em plataformas de controle de versão para receber contribuições de toda a comunidade.

7. Referências

BRASIL. Lei n.º 9.985, de 18 de julho de 2000. Cria o Sistema Nacional de Unidades de Conservação da Natureza, 2000. Disponível em: http://www.planalto.gov.br/ccivil_03/leis/L9985.htm. Acesso em: 8 set. 2018.

ICMBIO – Instituto Chico Mendes de Biodiversidade. Painel Dinâmico de informações. http://qv.icmbio.gov.br/QvAJAXZfc/opendoc2.htm?document=painel_corporativo_6476.qvw&host=Local&anonymous=true. Acesso em: 8 set. 2018.

Paula, E. V.; Paz, O. L. S.; Silva, J. P.(2017) Elaboración de Bases Geográficas para Planificación y Gestión de Áreas Protegidas. In: VI Seminario Internacional de Ordenamiento Territorial, 2017, Mendoza, Argentina. Anales do IV Seminario Internacional de Ordenamiento Territorial. Mendoza: UNCuyo, v. 1. p. 1-12.

Paula, E. V.; Pigosso, A. M. B.; Wroblewski, C. A. (2018). Unidades de Conservação no Litoral do Paraná: Evolução Territorial e Grau de Implementação. In: Mayra Taiza Sulzbach, Daniela Resende Archanjo, Juliana Quadros. (Org.). Litoral do Paraná: território e perspectivas. 1ed.Rio de Janeiro: Autografia, v. 3, p. 41-92.

Medeiros, R.; Araújo, F. F. S. (2011) Dez anos do Sistema Nacional de Unidades de Conservação da Natureza: lições do passado, realizações presentes e perspectivas para o futuro. Brasília: MMA. 220p.

Ferramenta para recuperação de informação utilizando indexação espacial e textual

Mairon Q. Castro[1], **Clodoveu A. Davis Jr.**[2]

[1,2] Departamento de Ciencia da Computação – Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brasil

{mairon.castro,clodoveu}@dcc.ufmg.br

Abstract. *Search engines usually focus on keyword-based search, and thus require searches related to places to be resolved using place names among the keywords, with mixed results. This paper describes the structure and development of an information retrieval engine that allows the user to search by terms and by geographic limits. Searching combines a term index and a geographic index, allowing results to reflect a combination of both interests. We demonstrate the effectiveness of the approach using a dataset of 1.7 million georeferenced tweets, collected during the 2014 World Cup.*

Resumo. *Máquinas de busca em geral focam a busca baseada em palavras-chave, e portanto exigem que buscas relacionadas a lugares sejam realizadas com base em nomes geográficos, obtendo resultados de qualidade variável. Este artigo apresenta a estrutura e implementação de um mecanismo de recuperação de informação capaz de combinar termos e limites geográficos na entrada, obtendo resultados que refletem a combinação das estratégias. A eficiência do enfoque proposto é demonstrada em buscas sobre um conjunto de 1.7 milhão de tweets georreferenciados, coletados ao longo da Copa do Mundo de 2014.*

1. Introdução

Máquinas de busca são recursos bastante comuns e presentes na vida cotidiana das pessoas. A partir de um conjunto de palavras-chave, máquinas de busca localizam e classificam fontes de conteúdo online referentes ao que lhes parece ser a intenção de busca do usuário.

No entanto, quando existem referências a lugares entre as palavras-chave, muitas vezes a máquina de busca não consegue capturar a intenção de restringir a busca ou relacionar os resultados obtidos a determinado local. Seria interessante, portanto, que o usuário pudesse expressar sua intenção de busca através de referências geoespaciais mais diretas, apontando um lugar em um mapa ou delimitando uma região de busca, além de informar palavras-chave que pudessem definir tematicamente o interesse de busca.

Para isso, seria interessante contar tanto com uma interface de consulta quanto com mecanismos de indexação que tratassem tanto da parte geoespacial quanto da parte temática de uma consulta. Além disso, a classificação ou ranqueamento dos resultados poderia combinar aspectos dos componentes geoespacial e temático, expressando uma noção de preferência do usuário nos resultados obtidos.

Este artigo apresenta um método de indexação espacial e textual de dados coletados na Web, e descreve um protótipo de ferramenta que permite fazer buscas híbridas, espaciais-textuais. São exploradas diferentes modalidades de consulta e de apresentação de resultados, tendo em vista um potencial uso das propostas aqui apresentadas na indexação espacial-textual de conjuntos de documentos de diferentes naturezas, tais como documentos científicos ou notícias cotidianas.

2. Trabalhos Relacionados

O projeto SPIRIT (Spatially-Aware Information Retrieval on the Internet) [Jones et al. 2002] explora conceitos de busca espacial e textual. Ao longo de sua existência, o projeto SPIRIT desenvolveu uma máquina de busca completa para localizar documentos e conjuntos de dados relacionados a lugares ou regiões referenciadas em uma busca. Aspectos do projeto abordaram desde a identificação de referências a lugares nos documentos e reconhecimento de referências espaciais entre as palavras-chave da busca, além da criação de índices e de funções de ranqueamento.

Outro projeto relacionado ao tema deste artigo é o Somewherenear [Banahan et al. 2000]. Trata-se de uma ferramenta de busca geográfica que permite que usuários localizem itens de interesse em proximidade a outros itens, com base na distância. O intuito principal é servir como fonte de informação para viajantes a lazer ou a negócios, em busca de lugares para visitar, acomodação, alimentação e outros serviços.

3. Componentes da ferramenta

3.1. Índice Espacial

Um índice geográfico ou espacial é responsável por recuperar informação sobre objetos espaciais, cuja forma geométrica e localização apresentam mais de uma dimensão. Assim, o índice deve ser capaz de, dado um par de coordenadas, ou a delimitação geográfica de uma região, retornar objetos que estão contidos neste limite. Sistemas de bancos de dados tradicionais usam índices unidimensionais (ou seja, baseados em um atributo ou chave), como árvore B e hash, para resolver consultas de forma eficiente. Índices convencionais não são suficientes para dados geográficos, pois consultas espaciais exigem a recuperação eficiente de dados considerando mais de uma dimensão, e também considerando proximidade, topologia, dimensões e outras características dos objetos geográficos, tipicamente codificados segundo pontos, linhas e polígonos.

Uma das estruturas de indexação espacial mais utilizadas em bancos de dados geográficos é a R-tree[Guttman 1984]. A R-tree utiliza o retângulo envolvente mínimo como representação simplificada da geometria dos objetos, e indexa retângulos. Cada nó da árvore representa um retângulo que contém todos os retângulos que descendem dele. Em um mesmo nível da R-Tree é possível que os retângulos de nós irmãos apresentem superposições, o que gera a necessidade de o algoritmo de indexação lidar com o problema de agregação e subdivisão de retângulos, buscando aumentar a eficiência do índice. Numerosas propostas de variação da política de agregação e subdivisão de retângulos foram apresentadas na literatura, gerando variações da R-Tree. Mesmo assim, a versão original é a mais usualmente empregada pelos gerenciadores de bancos de dados geográficos.

3.2. Índice Textual

No contexto de uma máquina de busca tradicional, que opera por palavras chave, é criado um índice ou arquivo invertido, em que palavras e expressões são relacionadas aos documentos onde foram encontradas. Quando o usuário fornece uma lista de palavras para sua busca, a máquina de busca recupera as listas de referências a documentos presentes no índice e obtém a interseção entre essas listas, ou seja, documentos relacionados a todas as palavras da busca. O índice invertido é, assim, responsável por organizar dados que possibilitem que a busca seja feita a partir de palavras, sem que haja acesso direto à coleção de documentos. Uma busca sequencial, sem qualquer tratamento dos dados inviabilizaria a priorização do conjunto resposta.

A utilização do arquivo invertido aumenta a eficiência de pesquisa em várias ordens de magnitude, característica importante para aplicações que utilizam grandes volumes de documentos constituídos de texto. O custo para se conseguir essa eficiência é a necessidade de armazenar uma estrutura de dados que pode ocupar tanto espaço quanto o texto original, dependendo da quantidade de informação armazenada no índice [Manning et al. 2008a].

Existem várias estratégias para buscar a redução do custo de manutenção do índice. *Stopwords* são palavras muito comuns em uma linguagem, como artigos e preposições. O objetivo é não indexar *stopwords*, pois não trazem muita informação sobre um documento. Um termo indexável é uma palavra que não seja *stopword* presente em um documento. Como a finalidade é recuperar informações de qualquer documento, um indexador deve percorrer a coleção, recuperar e armazenar todos os seus termos indexáveis, bem como informações sobre eles. O conjunto de todos os termos indexáveis únicos de uma coleção é chamado de *vocabulário*.

Com o vocabulário em mãos, é possível obter todos os pares termo-documento. Ordenando esses pares por termo e depois por documento, pode-se organizar todas as informações sobre um termo de forma contígua no dispositivo de armazenamento, otimizando a busca por este termo. Este conjunto ordenado de pares é o arquivo invertido de uma coleção. É de se notar que a ordenação deve ser realizada por alguma técnica de ordenação em memória secundária, devido ao grande tamanho do índice.

Portanto, a construção do vocabulário de uma coleção se baseia em percorrer todos os documentos e analisar palavra por palavra. Para cada documento, deve-se inserir sua chave na coleção. Após a identificação dos termos indexáveis, o indexador percorre todos e verifica se está no vocabulário ou não e insere caso não esteja. Se já estiver, apenas a frequência é atualizada. O mesmo é feito para o índice, verificando se o termo atual já possui entrada no índice para este documento.

O tamanho do índice cresce linearmente com o tamanho da coleção, fato que impossibilita armazenamento em memória primária de grandes coleções. Ao analisar a estrutura de cada entrada do índice, que é na forma da tupla (termo, documento, frequência), pode-se observar que após a leitura completa de um documento, entradas no índice relativas a este documento não serão mais atualizadas, pois consideramos que os documentos são únicos.

Existem estratégias para lidar com índices dinâmicos, ou seja, índices capazes de se adaptarem a mudanças em documentos, permitindo a reindexação dos mesmos[Manning et al. 2008a]. Este tipo de índice é útil para páginas que são atualizadas com frequência, como por exemplo páginas iniciais de portais de notícias. O índice dinâmico não é abordado neste trabalho.

3.3. Ranqueamento

Com o índice pronto, para as pesquisas terem resultados relevantes, funções de ranqueamento devem ser aplicadas, ou seja, é preciso calcular uma pontuação numérica para a associação entre consultas e documentos. No caso deste trabalho, isso precisa ser feito tanto para o texto, quanto para a representação geográfica associada ao documento.

3.3.1. Ranqueamento textual

Na busca textual, o primeiro passo é definir um modelo que dê pesos para pares termo-documento, ou seja, que descreva a importância de um termo para um documento. Com o modelo definido, é possível modelar as pesquisas e os documentos como vetores, nos quais cada posição é dada pelo valor do peso de cada termo único, tanto para a coleção quanto para o documento. O modelo $tf - idf_{t,d}$(*Term Frequency - Inverse Document Frequency*, onde t é um termo e d um documento) para atribuição de pesos é muito popular em recuperação de informação [Manning et al. 2008b].

O modelo tf se baseia no fato de que o peso de um termo é proporcional à sua frequência em um documento, ou seja, quanto mais frequente um termo é em um documento, maior é o seu peso. Isso se fundamenta na observação de que termos de alta frequência são importantes para descrever documentos.

Já $idf_{t,d}$ é importante porque, se um termo é muito frequente na coleção, e está em boa parte dos documentos, a probabilidade de ser um termo específico é baixa. Para isso, o modelo estabelece uma punição a termos muito frequentes, diminuindo o peso deles. Isto é, quanto menos frequente na coleção um termo da consulta for, maior deve ser o seu peso. Por exemplo, a consulta ‘refrigerante de guaraná‘ deve dar um maior peso ao termo ‘guaraná‘, porque é uma palavra que é mais específica do que ’refrigerante’ no vocabulário português.

O $tf - idf_{t,d}$ é um modelo que alia os dois pontos, no intuito de chegar a uma forma coerente de pesos. O cálculo é dado pela seguinte expressão:

$$tf - idf_{t,d} \ = \ tf_{t,d} \times log(\frac{N}{df_t}) \tag{1}$$

Onde $tf_{t,d}$ é a frequência do termo no documento, N é o tamanho da coleção e df_t é a quantidade de documentos em que o termo aparece. Representando as consultas e os documentos em um modelo de espaço vetorial [Manning et al. 2008b], onde os vetores são os pesos $tf_{t,d} \times idf_t$ entre a consulta e os documentos, a função de ranqueamento é dada pela similaridade de cosseno entre estes vetores.

$$sim(q,d) \ = \ v(\vec{q}).v(\vec{d}) \ = \ \frac{V(\vec{q}).V(\vec{d})}{\|V(\vec{q})\|.\|V(\vec{d})\|} \tag{2}$$

O ranqueamento é calculado através da ordenação dos documentos por ordem decrescente de pontuação, pois quanto maior o cosseno entre dois vetores, mais próximo os seus unitários estão. Isto é, mais próxima um documento está de uma consulta.

3.3.2. Ranqueamento geográfico

No caso da estratégia de ranqueamento para a busca por delimitação geográfica, não há um método que seja unanimidade, já que diversos fatores podem estar envolvidos [Kumar 2011]. Os limites geográficos da base de dados podem variar muito, o objetivo com a busca também. Geralmente as estratégias consideram medidas de similaridade espacial, como sobreposição, forma, contorno, etc.

Uma possibilidade para ranqueamento geográfico é o uso de alguma função de distância. Por exemplo, pode-se usar uma ordenação por proximidade a algum ponto de referência citado na consulta, ou a distância ao centro de uma região indicada para a consulta.

A interface da ferramenta desenvolvida possibilita ao usuário fazer uma delimitação geográfica através do desenho de um retângulo no mapa. Como os dados indexados estão representados apenas como pontos, a alternativa de ranqueamento implementada foi através da distância euclidiana do documento ao centro do retângulo delimitado na consulta, de forma que quanto mais perto do centro, maior é a relevância do documento. Assim sendo, se a consulta q for o retângulo delimitado por $(lng1, lng2, lat1, lat2)$ e o ponto de um documento d for (lng, lat), temos:

$$lngCenter = \frac{(lng1 + lng2)}{2} \tag{3}$$

$$latCenter = \frac{(lat1 + lat2)}{2} \tag{4}$$

$$sim(q,d) = \sqrt{(lng - lngCenter)^2 - (lat - latCenter)^2} \tag{5}$$

4. Ferramenta e coleção utilizada

A Figura 1 apresenta a estrutura da ferramenta implementada. O usuário apresenta termos de busca e/ou delimita uma região geográfica de seu interesse, utilizando um retângulo sobre um mapa. São realizadas consultas aos dois índices (textual e geográfico), e os resultados são combinados e ordenados, de acordo com a estratégia de ranqueamento descrita. Ao final, os resultados são apresentados sobre o mapa, e podem ser consultados individualmente.

A ferramenta desenvolvida[1] utiliza o gerenciador de bancos de dados PostgreSQL para armazenar o vocabulário, e emprega a extensão geográfica PostGIS para gerenciar objetos espaciais e geográficos e indexá-los, bem como para realizar operações com estes objetos. A estrutura de indexação usada pelo PostGIS é baseada na R-tree, já citada.

Já o arquivo invertido é armazenado no disco como um arquivo ordenado por termo e depois por documento, e contém a frequência do termo em cada documento. A ordenação é feita através do algoritmo de intercalação [Greene 1991].

A coleção de documentos utilizada para este trabalho, de modo a testar a ferramenta implementada, foi um conjunto de 1.715.167 tweets coletados ao longo da Copa do Mundo de 2014. Cada tweet está associado a uma posição (latitude e longitude). A coleta foi realizada por pesquisadores

[1] http://greenwich.lbd.dcc.ufmg.br/termgeo/

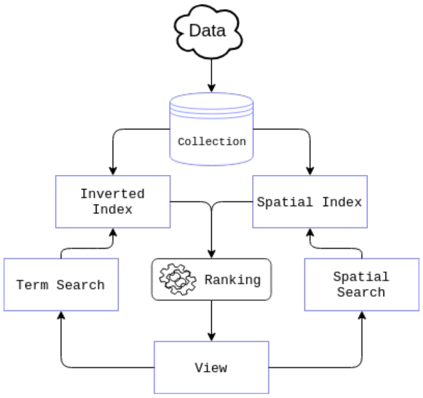


Figura 1. Estrutura do projeto

do Departamento de Ciência da Computação da Universidade Federal de Minas Gerais. O conjunto original de tweets continha cerca de 50 vezes mais elementos, mas aqueles sem geolocalização foram descartados: apenas cerca de 2% dos tweets coletados estavam associados a um par de coordenadas.

```
Index I = {}
Collection C = {}
Vocabulary V = {}
for each document d do
    read document;
    insert d coordinates and key in C
    for each valid term t do
        if t is not in V then
            insert with frequency 1
        else
            update frequency
        end
        if tuple(t, d) is not in I then
            insert with frequency 1
        else
            update frequency
        end
    end
    if memory is full then
        save I, C, V on disk
        I = {}
    end
end
order I by (t, d)
```

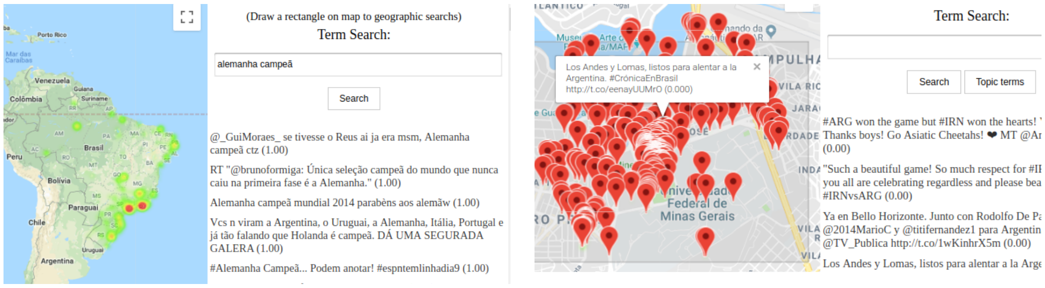
Algorithm 1: Construção dos índices

O algoritmo 1 resume a construção dos índices da ferramenta. As estratégias de ranqueamento já citadas são utilizadas para cada consulta.

A Figura 2 apresenta imagens da interface da ferramenta durante a realização das diferentes buscas. Ao passar o cursor do mouse sobre os documentos do conjunto resposta, a ferramenta evidencia no mapa, o local exato do documento em foco. É possível também dar zoom e acessar cada um dos Tweets, clicando no documento.

5. Conclusões e trabalhos futuros

Analisando a ferramenta e seus resultados, é interessante notar que uma simples consulta por termos destaca justamente as regiões mais desenvolvidas do país no mapa. Deve-se ao fato da facilidade de



@_GuiMoraes_ se tivesse o Reus ai ja era msm, Alemanha campeã ctz (1.00)

RT "@brunoformiga: Única seleção campeã do mundo que nunca caiu na primeira fase é a Alemanha." (1.00)

Alemanha campeã mundial 2014 parabèns aos alemãw (1.00)

Vcs n viram a Argentina, o Uruguai, a Alemanha, Itália, Portugal e já tão falando que Holanda é campeã. DÁ UMA SEGURADA GALERA (1.00)

#Alemanha Campeã... Podem anotar! #espntemlinhadia9 (1.00)

Los Andes y Lomas, listos para alentar a la Argentina. #CrónicaEnBrasil http://t.co/eenayUUMrO (0.000)

#ARG won the game but #IRN won the hearts! Thanks boys! Go Asiatic Cheetahs! ❤ MT @Ar (0.00)

"Such a beautiful game! So much respect for #II you all are celebrating regardless and please bea #IRNvsARG (0.00)

Ya en Bello Horizonte. Junto con Rodolfo De Pa @2014MarioC y @titifernandez1 para Argentin @TV_Publica http://t.co/1wKinhrX5m (0.00)

Los Andes y Lomas, listos para alentar a la Arge

Figura 2. À esquerda, busca por termos - Pesquisa por ”alemanha campeã”. À direita, busca por delimitação geográfica no entorno do estádio Mineirão, em Belo Horizonte. Esse tipo de busca ativa o botão ”Topic terms”, que possibilita a visualização dos principais termos na região delimitada.

acesso à internet, e consequentemente, um maior uso de redes sociais como o Twitter. Também é possível notar que busca por delimitação geográfica destaca termos relacionados à Copa do Mundo, o que é esperado dada a delimitação feita e o período de coleta dos dados. Uma ferramenta desse tipo pode possibilitar diversas análises interessantes, como por exemplo popularidade de um político ou um time de futebol, ou até mesmo identificar focos de doenças, analisando a localização de mensagens contento de citações a elas. É possível também analisar o que está sendo falado em determinado local e o porquê. É de se ressaltar que a ferramenta é genérica e pode funcionar para qualquer conjunto de dados, desde que geolocalizados.

Tomando o $idf_{t,d}$ como exemplo, é possível também desenvolver uma estratégia de ranqueamento espacial semelhante, que leve em consideração a especificidade de uma determinada região, assim como o $idf_{t,d}$ leva à especificidade de um termo. Isto é, se uma determinada região possui poucos documentos, uma busca por delimitação geográfica que inclua esta e outras regiões com mais documentos deverá dar um maior peso para esta região. Isso pode ser interessante para destacar documentos de áreas desfavorecidas.

Outra proposta de trabalho seria utilizar a ferramenta para análises de buscas, no intuito de tirar conclusões sobre determinado assunto e/ou região.

Referências

Banahan, M., Fisher, D., Greenwood, A., Riach, J., and Willis, L. (2000). Somewherenear is the uk's leading geographic search engine. [Online; accessed 29-August-2018].

Greene, W. A. (1991). k-way merging and k-ary sorts. In *[Proceedings] 1991 Symposium on Applied Computing*, pages 197–.

Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. *SIGMOD Rec.*, 14(2):47–57.

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. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 387–388, New York, NY, USA. ACM.

Kumar, C. (2011). Relevance and ranking in geographic information retrieval. In *Proceedings of the Fourth BCS-IRSG Conference on Future Directions in Information Access*, FDIA'11, pages 2–7, Swindon, UK. BCS Learning & Development Ltd.

Manning, C. D., Raghavan, P., and Schütze, H. (2008a). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Manning, C. D., Raghavan, P., and Schütze, H. (2008b). *Scoring, term weighting, and the vector space model*, page 100–123. Cambridge University Press.

Identificação de Bancos de Areia Situados na Região da Baía do Guajará Mediante Redes Neurais Convolucionais Profundas e Imagens de Radar de Abertura Sintética (SAR)

José A. S. Sá[1], Fábio F. Gama[2], Gilberto R. Queiroz[2],
Lúbia Vinhas[2], Brígida R. P. Rocha[3]

[1]Departamento de Ciências Sociais Aplicadas – Universidade do Estado do Pará (UEPA)
Tv. Doutor Enéas Pinheiro, 2626 – 66095-015 – Belém – PA – Brasil

[2]Instituto Nacional de Pesquisas Espaciais (INPE)
Av. dos Astronautas, 1758 – 12227-010 – São José dos Campos – SP – Brasil

[3]Centro Gestor e Operacional do Sistema de Proteção da Amazônia (CENSIPAM)
Av. Júlio César, 7060 – 66617-420 – Belém – PA – Brasil

josealbertosa@uepa.br,
{fabio.furlan,gilberto.queiroz,lubia.vinhas}@inpe.br,
rocha.brigida@sipam.gov.br

Abstract. *Sandbanks represent a danger to river and sea navigation due to the significant damage they can cause to ships, crews and passengers. This work aimed to develop a methodology for the automatic identification of sandbanks located in an Amazonian estuarine region (Baía do Guajará) using synthetic aperture radar (SAR) image and deep convolutional neural networks (Deep Learning) in order to assist the obstacles monitoring in the local navigation. The results demonstrated significant differences in the use of the SAR polarization for the monitoring of the sandbanks, as well as elevated accuracy (99,6%) for automatic identification through the convolutional neural network.*

Resumo. *Bancos de areia representam um perigo à navegação fluvial e marítima devido aos danos significativos que podem gerar para as embarcações, tripulações e passageiros. Este trabalho buscou desenvolver uma metodologia para a identificação automática de bancos de areia situados em uma região estuarina Amazônica (Baía do Guajará) mediante imagem de radar de abertura sintética (SAR) e redes neurais convolucionais profundas (Deep Learning) no intuito de auxiliar o monitoramento da evolução destes obstáculos à navegação local. Os resultados demostraram diferenças relevantes no uso das polarizações SAR para o monitoramento de bancos de areia, além de acurácia igual a 99,6% para a identificação automática.*

1. Introdução

Um banco de areia consiste no acúmulo de sedimentos (areia e cascalho) depositados no leito de um rio ou ao longo da costa marítima, constituindo-se em um obstáculo ao escoamento e à navegação. Nos rios, os bancos são formados pelos depósitos de aluvião e nas praias eles podem se formados pelo fluxo e refluxo do mar ou pela ação das ondas. Os bancos de areia constituem-se em um perigo à navegação (fluvial e marítima), podendo gerar danos e naufrágios as embarcações.

Na região amazônica, a malha fluvial constitui-se em um dos principais meios de acesso aos municípios e comunidades, sendo os bancos de areia um problema significativo para quem trafega nos rios da Amazônia. No intuito de auxiliar a gestão da qualidade da navegação desta região este trabalho teve por objetivo desenvolver uma metodologia para auxiliar a identificação da formação de bancos de areia e as alterações das delimitações das linhas de costa, no intuito de colaborar com a atualização de cartas náuticas que possuem um importante papel para a navegação fluvial e marítima.

O levantamento *in loco* das áreas que possuem bancos de areia é geralmente difícil (risco às embarcações) e dispendioso (custos logísticos associados à realização das batimetrias), resultando em uma quantidade limitada de dados que pode ser obtida pelas técnicas tradicionais. Alternativamente, as técnicas de sensoriamento remoto fornecem ferramentas eficazes e seguras para a realização de inferências e indicações auxiliares aos processos de medição de grandes dimensões espaciais. O radar de abertura sintética (*Synthetic Aperture Radar* - SAR), em particular, é um meio promissor para o monitoramento da evolução temporal de bancos de areia, principalmente devido às suas vantagens na capacidade operacional, de gerar produtos significativos independentemente de tempo e clima, além de permitir a vigilância para uma ampla área [Cheng et al. 2013], [Yang et al. 2008].

O Programa *Copernicus* da ESA (*European Space Agency*) pretende produzir, a longo prazo, dados SAR através das Missões Sentinel para subsidiar diferentes aplicações baseadas em séries temporais [EUROPEAN SPACE AGENCY 2018a 2018b 2018c]. O acesso aberto aos dados da Missão Sentinel-1, formada por dois satélites de órbita polar, que geram imagens SAR na banda C, configura-se, atualmente, em uma significativa fonte de dados para o monitoramento de bancos de areia na região Amazônica. A Missão Sentinel-1 foi projetada para fornecer alta resolução espaço-temporal para os serviços operacionais e aplicativos que exigem longas séries de tempo de dados terrestres, com estimativa que cada satélite da constelação Sentinel-1 transmita dados de observação da Terra por pelo menos 7 anos (com recursos de energia para 12 anos), sendo que para a região Amazônica o tempo de revisita é de 12 dias.

Aliado ao sensoriamento remoto, o monitoramento em tela também pode usufruir dos benefícios das técnicas de inteligência computacional para o reconhecimento automático de padrões evolutivos dos bancos de areia. Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNN) podem ser utilizadas para a identificação de mudanças em imagens orbitais (alterações nas áreas monitoradas). As CNN ou *ConvNets* são aplicadas no processamento e análise de imagens digitais. Em sua Modalidade Profunda caracterizam-se pela utilização de várias camadas (*Deep Learning*). Esta Aprendizagem Profunda é parte integrante dos métodos de Aprendizado de Máquina (*Machine Learning*). Uma das promessas da aprendizagem profunda é a substituição da obtenção de características feitas de forma manual por algoritmos eficientes capazes de extrair características de forma autômata [Zhu et al. 2017].

Assim, diante o exposto, este trabalho de nível exploratório-descritivo teve por objetivo propor uma metodologia de identificação automática de bancos de areia situados em uma baía Amazônica mediante a classificação de imagens de radar de abertura sintética (SAR) usando uma rede neural convolucional profunda (*Deep Learning*) denominada VGG-19, para extração de características das imagens SAR, e a Árvore de Decisão (*Decision Tree*) para determinação da base de regras de classificação.

2. Materiais e Métodos

2.1. Área de estudo: Baía do Guajará

A área de estudo escolhida foi a Baía do Guajará. Ela é formada pelo encontro da foz dos rios Guamá e Acará, que banha os municípios paraenses de Barcarena e Belém, capital do estado do Pará.

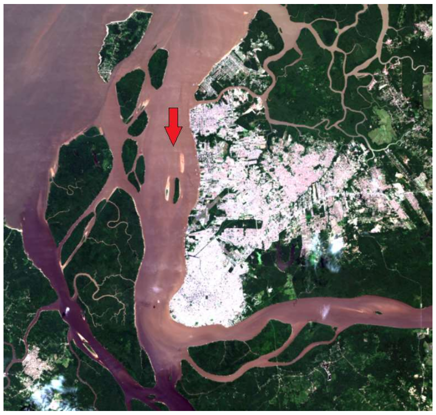


Figura 1. Área de Estudo: Baía do Guajará

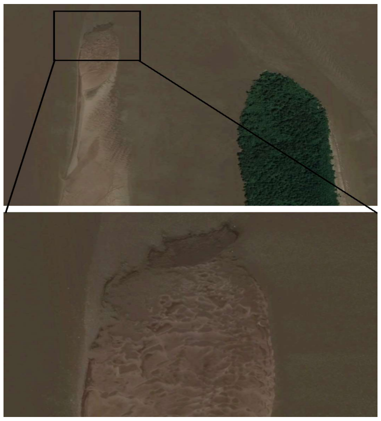


Figura 2. Detalhe de Banco de Areia

A Figura 1 mostra um recorte da cena do Satélite Landsat 8, Órbita/Ponto: 223/061, Sensor OLI, Composição RGB (Cor Natural), com data de aquisição em 07/06/2018, no momento em que a altura de maré era de 1,3 m acima do nível de redução (Porto de Belém). Para esta altura de maré já é possível a identificação de bancos de areia próximos a cidade de Belém (seta vermelha). A Figura 2 mostra detalhes de um banco de areia, na área de estudo, contíguo ao arquipélago denominado "Ilha da Barra", ambos as proximidades do Aeroporto Internacional de Belém.

2.2. Coleta de dados SAR

A imagem SAR foi obtida gratuitamente no "*Copernicus Open Access Hub*" da ESA, um produto da Missão Sentinel-1; Plataforma Orbital: S1A; Banda: C; Tipo: *Ground Range Detected* (GRD); Polarizações: VV e VH; Modo Sensor: *Interferometric Wide Swath Mode* (IW), sendo este considerado o modo principal de aquisição de dados terrestres devido atender a maior parte dos serviços demandados.

2.3. Altura de maré

Optou-se por escolher uma imagem SAR sincronizada com uma pequena altura de maré, para maior exposição dos bancos de areia (porções emersas). Utilizou-se as "Tábuas de Marés" (publicadas pela Diretoria de Hidrografia e Navegação - MB) que informam, diariamente, os horários das Preamares (PM) e Baixa-mares (BM). Entretanto, não é comum o horário da visita do Satélite Sentinel-1 coincidir com o horário das Baixa-mares (BM) na região em estudo, sendo então necessária a utilização das "Tabelas de Correção" para se determinar a altura da maré no momento das visitas do satélite. Após a análise de 56 produtos SAR disponíveis na base de dados da ESA, optou-se pela imagem SAR, com data de aquisição em 30/03/2018, no momento em que a altura de maré era de 0,2 m acima do nível de redução (Porto de Belém).

2.4. Pré-processamento das imagens SAR

Utilizou-se o Software "*Sentinel Application Platform*", mais conhecido como SNAP para realizar o pré-processamento da imagem SAR. Inicialmente, para a cena escolhida, conforme a etapa descrita no item 2.3, foi realizado um pré-processamento baseado em [Foumelis 2018], composto da seguinte sequência: (a) "*Apply Orbit File*"; (b) "*Calibrate*"; (c) "*Speckle Filtering*"; (d) "*Geocoding*"; e (e) "*Subset*". A Figura 3 expõe a visualização de parte da área de estudo para Sigma Zero nas polarizações VH e VV, sendo possível perceber uma diferença significativa de brilho para os bancos de areia (setas vermelhas).

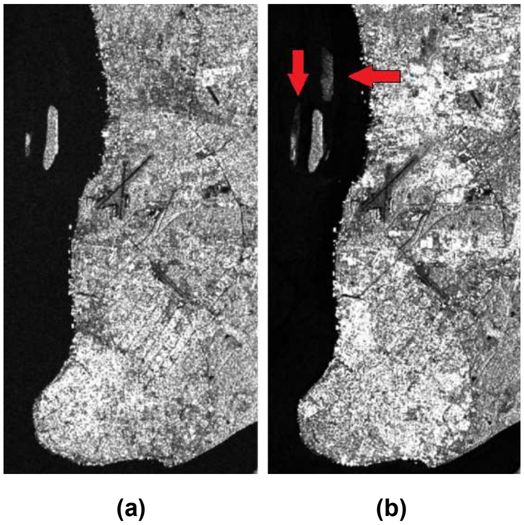


Figura 3. Visualização para: (a) Sigma Zero VH e (b) Sigma Zero VV

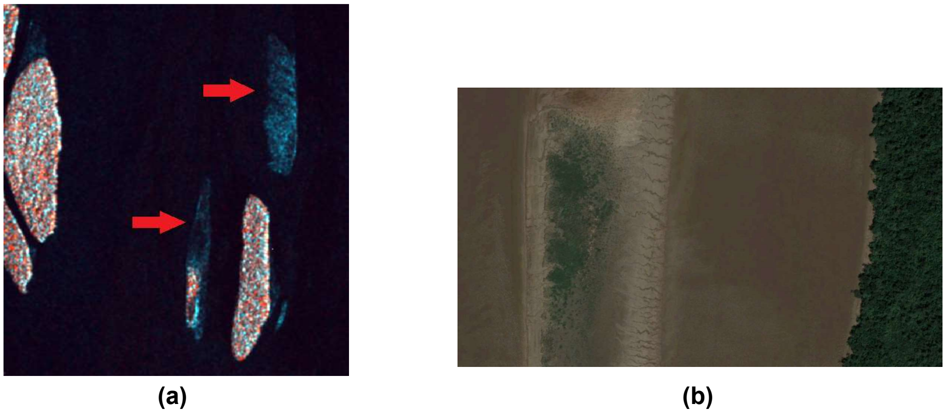


Figura 4. (a) Destaque dos Bancos de Areia e (b) Detalhes de Vegetação

2.5. Efeito *Bragg* e imagem RGB

O imageamento SAR diferencial, observado neste estudo, decorre do chamado Efeito *Bragg* (Ressonância *Bragg*) que representa uma maior intensidade da energia retroespalhada pelas "Ondas de *Bragg*" para o imageamento SAR na polarização VV e uma menor intensidade da energia retroespalhada para o imageamento SAR na polarização VH [Violante-Carvalho 2010]. Considerando esta diferença de brilho observada, buscou-se criar uma máscara polarimétrica (destaque) para facilitar a identificação de bancos de areia. Utilizou-se uma composição colorida RGB: Red: Sigma Zero VH; Green: Sigma Zero VV; Blue: Sigma Zero VV - Sigma Zero VH. A Figura 4(a), mostra o resultado da composição RGB, onde são destacados os Bancos de Areia (setas vermelhas). Ainda na Figura 4(a), percebe-se que parte de um banco de areia apresenta semelhança da composição colorida das ilhas contíguas. A Figura 4(b) expõe detalhes de uma vegetação sobre a porção em questão, o que explica o retroespalhamento e consequentemente o produto de composição semelhante.

2.6. Classificação de bancos de areia utilizando Rede Neural Convolucional

Após a aplicação da máscara polarimétrica (composição RGB), realizou-se amostragem aleatória de 600 recortes para banco de areia (BA), com dimensões constantes e iguais a 15x15 pixels e 600 recortes para não-banco de areia (NBA) (ilhas fluviais situadas na Baía do Guajará), também com dimensões constantes e iguais a 15x15 pixels, o que resultou em 1200 recortes. Deste total foram selecionados, aleatoriamente, 960 recortes para compor um conjunto de dados de treinamento (80% do total dos dados), sendo 480 recortes de BA e 480 recortes de NBA. Os dados remanescentes, portanto 240 recortes, foram utilizados para a composição do conjunto de dados de teste (20% do total dos dados), sendo 120 recortes de BA e 120 recortes de NBA. Desta forma, utilizou-se massas de dados balanceadas. A Figura 5(a) expõe exemplos de recortes BA e a Figura 5(b) exemplos de recortes NBA. Os recortes foram obtidos dentro de um retângulo envolvente com: Limite de Latitude Norte: 01° 19' 36" S; Limite de Longitude Oeste: 48° 31' 18" O; Limite de Latitude Sul: 01° 22' 38" S; e Limite de Longitude Leste: 48° 29' 16" O. O próximo passo da pesquisa foi desenvolver uma metodologia para a identificação automática dos bancos de areia. Inicialmente, utilizou-se uma arquitetura convolucional denominada VGG-19 (19 camadas ocultas), vencedora do Prêmio "*ImageNet Large Scale Visual Recognition Challenge 2014* (ILSVRC2014)" [VISUAL GEOMETRY GROUP 2018], que encontra-se disponível no Software Livre (*Free Software License*) denominado "*Orange Data Mining*" [ORANGE VISUAL PROGRAMMING 2018] para extrair características (*features*) dos recortes (4096 *features* por imagem). Posteriormente, foi utilizado o algoritmo denominado Árvore de Decisão (*Decision Tree*) com os parâmetros *default* do *Orange* - Versão 3.15, para a determinação da base de regras de classificação.

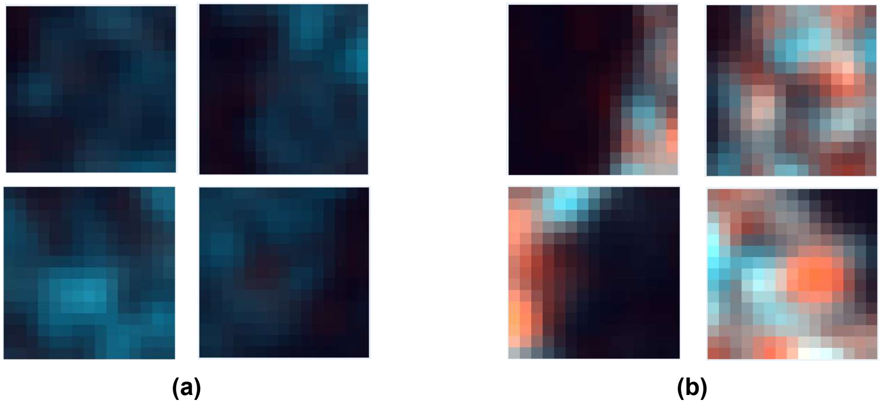


Figura 5. (a) Recortes BA e (b) Recortes NBA

3. Resultados da classificação

A Rede Neural Convolucional VGG-19 realizou a extração de 4096 características para cada amostra do conjunto de dados de treinamento e teste, sendo obtida a partir desta massa de dados, com a Árvore de Decisão (*Decision Tree*), a base de regras para a classificação. A base de regras gerada proporcionou uma acurácia preditiva igual a 99,6% na identificação automática dos bancos de areia do conjunto de dados de teste.

4. Conclusões

O estudo permitiu obter as seguintes conclusões: (a) De forma individual, as imagens SAR da Missão Sentinel-1, para Sigma Zero na Polarização VV, apresentaram melhor detecção dos bancos de areia quando comparadas as imagens SAR, para Sigma Zero na Polarização VH (Efeito *Bragg*); (b) A forma combinada de polarizações, pela máscara polarimétrica (composição colorida RGB), utilizada neste trabalho, permitiu um destaque para os bancos de areia na área de estudo; (c) Com a aplicação da CNN VGG-19 foi possível extrair 4096 características de cada amostra da massa de treinamento e da massa de teste. Após o uso do algoritmo *Decision Tree*, obteve-se acurácia preditiva igual a 99,6% para a identificação de bancos de areia; e (d) A Rede Neural Convolucional utilizada (VGG-19) demostrou-se útil para a extração automática de características (*features*) e formação de massas de dados robustas para posterior classificação, entretanto, pelo fato do estudo ter se limitado a uma única baía Amazônica, novos estudos tornam-se necessários para a consolidação desta técnica autômata.

5. Referências

Cheng, T. et al. (2013) "Sandbank and Oyster Farm Monitoring with Multi-Temporal Polarimetric SAR Data Using Four-Component Scattering Power Decomposition", . IEICE TRANS. COMMUN. p. 2573-2579.

EUROPEAN SPACE AGENCY (2018a) "Copernicus: Overview", http://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Overview3.

EUROPEAN SPACE AGENCY (2018b) "Sentinel Missions", https://sentinel.esa.int/web/sentinel/missions.

EUROPEAN SPACE AGENCY (2018c) "Acquisition Modes", https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-1-sar.

Foumelis, M. (2018) "Multi-temporal Analysis of Sentinel-1 SAR Backscattered Intensity", French Geological Survey – BRGM.

ORANGE VISUAL PROGRAMMING (2018) "Documentation - Release 3", https://orange.biolab.si/.

Violante-Carvalho, N. (2010) "Sobre os mecanismos de imageamento do radar de abertura sintética SAR para a estimação do espectro direcional de ondas geradas pelo vento", Revista Brasileira de Geofísica. p. 593-607.

VISUAL GEOMETRY GROUP (2018) Department of Engineering Science, University of Oxford. http://www.robots.ox.ac.uk/~vgg/research/very_deep/.

Yang, J. et al. (2008) "Review of the study on the underwater topography detection with sar imagery in sino-european dragon cooperation programme", ESA-SP Vol. 655, ISBN: 98-92-9221-219-3., p.10.

Zhu, X. X. et al. (2017) "Deep Learning in Remote Sensing: A review", IEEE GEOSCIENCE AND REMOTE SENSING MAGAZINE.

Utilização e estudo de dados de saúde georreferenciados para desenvolvimento de aplicação móvel

Juliana L. S. B. Cavalcante[1], **Mozart S. Neto**[1], **Nádia P. Kozievitch**

[1]Departamento de Informática - Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba – PR – Brazil

{julianacavalcante,neto}@alunos.utfpr.edu.br, nadiap@utfpr.edu.br

Abstract. *This research describes a prototype application for mobile devices dedicated to healthcare. Considering the state of the art of this subject, this proposal focuses at points not yet contemplated in existing applications, such as the centralization of data on health units, both public and private, focusing on the search for medical specialties. In this work we used the concepts from database and GIS, algorithms and programming for mobile devices, among others. To reach the proposed objectives, we sought to better understand the public involved, applying a questionnaire with the community from Curitiba.*

Resumo. *Esta pesquisa descreve um protótipo de aplicação para dispositivos móveis dedicado a área da saúde. Tendo conhecimento sobre o estado da arte desta temática, esta proposta visou pontos ainda não contemplados em aplicações existentes, como a centralização de dados sobre unidades de saúde tanto da rede pública quanto privada, com enfoque na busca por especialidades médicas. Neste trabalho foram utilizados os conhecimentos de banco de dados e GIS, algoritmos e programação para dispositivos móveis, entre outros. Para atingir os objetivos propostos, buscou-se compreender melhor o público envolvido, aplicando um questionário com a comunidade curitibana.*

1. Introdução e Objetivos

De acordo com [Werneck 2013], analisar a grande quantidade de dados gerados pelas novas tecnologias, pode ser benéfico para a área da saúde. Para isso, torna-se necessário a busca por informações relevantes, contidas nesses dados, que gerem conhecimento para auxiliar pacientes, médicos, hospitais e administradores públicos.

A falta de base confiável e de fácil acesso para a população consultar os hospitais próximos, é um exemplo de problema que pode ser solucionado analisando dados existentes. Isso pode beneficiar pacientes que precisem de atendimento médico e não conheçam unidades de saúde próximas. Outro ponto que pode ser prejudicial aos usuários é a busca por informações em ferramentas que não garantem veracidade ou dados atualizados.

O artefato proposto visa justamente trabalhar com os dados de saúde com um objetivo de melhoria: centralizar informações a respeito das unidades de saúde da cidade de Curitiba e de todo o Paraná, e dos médicos especializados disponíveis para atendimento do cidadão, tanto com relação ao atendimento gratuito (via Sistema Único de Saúde) quanto para atendimentos particulares. Além disto, os benefícios da aplicação podem ser expandidos para outra vertente: auxiliar a Prefeitura de Curitiba, uma vez que os dados aqui compilados podem ser utilizados no âmbito de planejamento de saúde da cidade.

O presente trabalho está organizado da seguinte forma: primeiramente uma seção sobre os trabalhos relacionados, seguindo para a seção sobre a metodologia da pesquisa. Em seguida, temos a parte de análise do público. Por fim, há o relato sobre o desenvolvimento do projeto e as considerações finais.

2. Trabalhos Relacionados

No que se refere a pesquisa por trabalhos relacionados, foram estudadas primeiramente algumas plataformas e aplicações direcionadas à saúde e que atendem a cidade de Curitiba. O Hospital Nossa Senhora das Graças [1], por exemplo, possui um aplicativo exclusivo para a seus serviços, visto na Figura 1(a) que fornece informações como: tempo de espera para atendimento pediátrico, lista de convênios e a localização de suas unidades.

O Doctoralia [2], por sua vez, possui uma interface web e um aplicativo móvel, este último sendo mostrado na Figura 1(b). A plataforma disponibiliza informações sobre unidades de saúde particulares de Curitiba, com localização, médicos cadastrados, convênios que o local aceita e as avaliações de outros usuários.

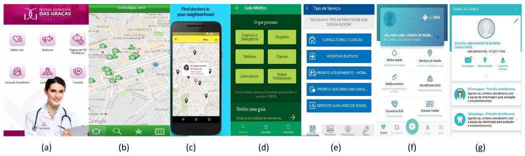


Figura 1. Tela inicial das aplicações: (a) Hospital Nossa Senhora das Graças; (b) Doctoralia; (c) ZocDoc; (f) digiSUS e (g) Saúde Já - Curitiba. Tela de busca por unidades das aplicações: (d) UNIMED Curitiba e (e) AMIL.

Referências externas também foram encontradas, é o caso do ZocDoc [3] na Figura 1(c), um aplicativo de busca por médicos, agendamento online de consultas e registro de avaliação de atendimentos, que atende algumas cidades nos Estados Unidos da América.

No que se refere à aplicativos disponibilizados pelos planos de saúde, temos na Figura 1 dois exemplos: (d) UNIMED [4] e (e) AMIL[5]. A característica do plano de saúde UNIMED é que ele apresenta diversos aplicativos, divididos por região ou por uma única cidade, além de também ter uma aplicação nomeada UNIMED Com Você que aparentemente é de âmbito nacional. Esse grande número de aplicações existentes para um mesmo prestador de serviços pode se mostrar como uma dificuldade para o usuário. O plano AMIL, por sua vez, apresenta apenas uma aplicação disponível para seus usuários, com dados de toda a rede. Entretanto, no caso de ambos os planos de saúde, o aplicativo só contempla unidades e médicos da rede conveniada e destinada apenas a seus usuários.

Sobre aplicações que contemplem a rede publica de saúde foram estudados dois casos: o aplicativo digiSUS [6], Figura 1 (f) disponibilizado pelo Ministério da Saúde e o aplicativo Saúde Já - Curitiba [7] (g), criado pela Prefeitura de Curitiba. O digiSUS

[1] http://www.hnsg.org.br/institucional/index.html. Acesso em: 24/05/2018

[2] https://www.doctoralia.com.br/. Acesso em: 23/05/2018.

[3] https://www.zocdoc.com/about/. Acessado em: 27/05/2018

[4] https://www.unimed.coop.br/ Acesso em: 25/10/2018

[5] https://amil.com.br/portal/web/institucional Acesso em: 25/10/2018

[6] http://portalms.saude.gov.br/acoes-e-programas/digisus Acesso em: 25/10/2018

[7] http://www.saudeja.curitiba.pr.gov.br/ Acesso em: 25/10/2018

apresenta recursos interessantes como possibilidade de armazenar o histórico médico do usuário e de seus atendimentos, além de também permitir a busca por unidades de saúde públicas mais próximas. O Saúde Já - Curitiba não realiza buscas por unidades, apenas mostra as informações e o endereço da Unidade Básica de Saúde em que o cidadão está registrado. Também é possível agendar algumas consultas na sua unidade através do aplicativo, mas o agendamento de odontologia, por exemplo, ainda não está disponível.

Em 2017, [Santos et al. 2017] realizaram um revisão sistemática da implementação de sistemas informatizados na área da saúde. Dentre os resultados, podemos destacar a dificuldade no uso dos dados por falta de uma terminologia padrão, a inconsistência nos dados disponibilizados, além da falta de identificação entre os indicadores usados num sistema e os profissionais de saúde que o utilizam. O trabalho de [Rocha et al. 2017] também realizou uma revisão integrativa de produções científicas especificamente com relação à aplicativos móveis voltados para saúde. Entretanto, não foram encontradas soluções que centralizasse os dados de todas as unidades de saúde da cidade, públicas e privadas. Além disto, o teste de algumas dessas aplicações revelou outros problemas, como informações desatualizadas ou faltantes.

Em [Oliveira et al. 2018] temos um estudo sobre os dados abertos de saúde disponibilizados pelo Paraguai e um comparativo dos mesmos com dados de saúde da cidade de Curitiba. Nesse estudo, foram relatadas algumas informações importantes como: Curitiba possui mais unidades de saúde, mais hospitais, além de apresentar uma categoria diferente de Unidade de Saúde, a CAPS (Centro de Atenção Psicossocial), que funciona para atendimento de saúde mental e dá atenção ao consumo de álcool e drogas. Compreender o panorama da saúde na cidade estudada é de vital importância para o melhor desenvolvimento deste trabalho.

3. Metodologia

A metodologia da aplicação proposta foi dividida em 3 etapas, sendo elas: realização da revisão bibliográfica, análise e projeto do sistema, e etapa de desenvolvimento e testes.

Quanto a revisão bibliográfica, os estudos realizados foram divididos em quatro tópicos principais: programação de dispositivos móveis, banco de dados/GIS, interação humano-computador e por fim, algoritmos. A etapa de análise e projeto abrange a aplicação de um questionário com uma amostra de moradores de Curitiba, sobre o tema da saúde e a possibilidade de uma solução tecnológica. Além disto, esta etapa inclui a definição de requisitos e a modelagem do software. Finalmente com relação ao desenvolvimento do projeto, serão realizadas as seguintes subfases: estruturação da base de dados, configuração e população do servidor, desenvolvimento da aplicação móvel e encerrando com testes do aplicativo.

4. Análise de público

Como informado anteriormente, um questionário foi aplicado para moradores de Curitiba durante o período de 06 de Abril à 25 de Maio de 2018, e obteve 85 respostas. O questionário contou com 8 perguntas e está disponível para visualização através do link[8].

Alguns dos pontos tratados na pesquisa terão seus resultados relatados a aqui. No gráfico ilustrado na Figura 2 identificamos primeiramente a distribuição de bairros em

[8] Questionário completo: https://goo.gl/HJeKPc

que residem os entrevistados. A pesquisa abrangeu respondentes alocados em 36, dos 75 bairros existentes na cidade de Curitiba. Além disso, 4% das pessoas registraram morar na região metropolitana. Ainda na Figura 2, o segundo gráfico se refere a faixa etária dos participantes da pesquisa: cerca de 15% encontra-se em idade inferior ou igual a 20 anos, 60% possui entre 21 e 30 anos, e cerca de 25% das pessoas está na faixa acima dos 31 anos.

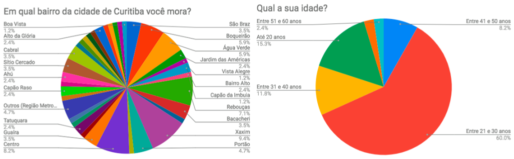


Figura 2. Primeiro gráfico referente a relação dos bairros de residência dos pesquisados, em seguida o gráfico com o intervalo de idade dos mesmos.

A respeito de funcionalidades que os participantes gostariam de ter acesso em um aplicativo de saúde:

- 65, dos 85 entrevistados selecionaram que gostariam de pesquisar por especialidades médicas;

- 65, dos 85 entrevistados selecionaram que gostariam de visualizar informações sobre os médicos, clínicas e hospitais (como telefone, endereço, horário de funcionamento)

- 63, dos 85 entrevistados selecionaram que gostariam de pesquisar por médicos, clínicas e hospitais do meu plano de saúde;

- 60, dos 85 entrevistados selecionaram que gostariam de pesquisar por atendimento médico mais próximo de si.

- 49, dos 85 entrevistados selecionaram que gostariam de pesquisar por médicos, clínicas e hospitais da rede pública de saúde.

Em suma, o questionário possibilitou um melhor entendimento sobre o público a qual se destina a aplicação, compreendendo o cenário atual e quais as necessidades a serem supridas para melhor satisfação dos usuários.

5. Desenvolvimento

Inicialmente, os dados de unidades de saúde, bem como suas especificações e localização foram selecionados da base fornecida pela Prefeitura Municipal de Curitiba[9] e pelo IPPUC[10]. Entretanto, esta base só apresentava registros de unidades de saúde públicas. Tendo o questionário retornado que, mais da metade do pesquisados utiliza atendimentos médicos via plano de saúde ou consultas particulares, não faria sentido focar o desenvolvimento da aplicação apenas no âmbito público. Foi então que, após contato do grupo com o Conselho Regional de Medicina do Estado do Paraná (CRM-PR)[11], o Conselho nos

[9] http://curitiba.pr.gov.br Acesso em: 23/10/2018

[10] http://www.ippuc.org.br/ Acesso em: 23/10/2018.

[11] http://www.crmpr.org.br/ Acesso em: 23/10/2018

cedeu acesso aos registros de todas as unidades de saúde cadastradas no banco de dados do mesmo.

As Figuras 3, fazem um comparativo geolocalizado dos registros encontrados nas duas fontes de dados aqui relatadas.

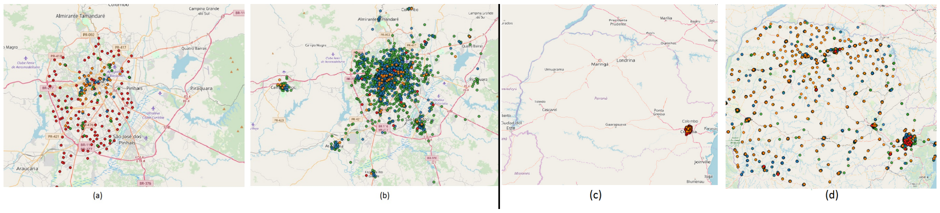
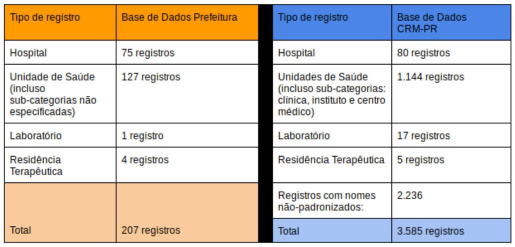


Figura 3. Imagens geradas de acordo com a localização das unidades de saúde cadastradas em (a) base da Prefeitura e IPPUC e (b) base de dados dos autores com dados fornecidos pelo CRM-PR com relação à cidade de Curitiba e região metropolitana. Imagens geradas de acordo com a localização das unidades de saúde cadastradas em (c) base da Prefeitura e IPPUC e (d) base de dados dos autores com dados fornecidos pelo CRM-PR com relação ao estado do Paraná.

Nota-se que a base de dados fornecida pelo CRM-PR apresenta uma quantidade mais expressiva de unidades de saúde registradas na cidade de Curitiba e região metropolitana, em detrimento da base de dados da Prefeitura. Os dados do CRM-PR também permitiu expandir o escopo da aplicação, uma vez que abrange informações sobre todo o estado do Paraná, ao invés de ficar limitado à região de Curitiba, como no caso dos dados da Prefeitura e do IPPUC. A Figura 4, por sua vez, demonstra quantitativamente a diferença entre as duas fontes de dados. Os registros do CRM-PR, apesar de não estarem perfeitamente padronizados, se mostram mais amplos, com conteúdo tanto da rede pública quanto privada, e fornecem dados importantes como: nome do local, contato, endereço e lista de especialidades médicas atendidas.



Tipo de registro	Base de Dados Prefeitura
Hospital	75 registros
Unidade de Saúde (incluso sub-categorias não especificadas)	127 registros
Laboratório	1 registro
Residência Terapêutica	4 registros
Total	207 registros

Tipo de registro	Base de Dados CRM-PR
Hospital	80 registros
Unidades de Saúde (incluso sub-categorias: clínica, instituto e centro médico)	1.144 registros
Laboratório	17 registros
Residência Terapêutica	5 registros
Registros com nomes não-padronizados:	2.236
Total	3.585 registros

Figura 4. Tabelas comparativas a respeito da quantidade de dados encontrados na base da Prefeitura de Curitiba e na base do CRM-PR

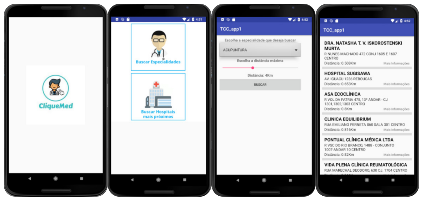
Através dos dados cedidos pelo CRM-PR, foi possível coletar informações capazes de geolocalizar cada estabelecimento do banco de dados. Para complementar os dados, com informações geográficas, foi utilizado a ferramenta *Google Geocode*[12]. Após geolocalizar cada local, utilizou-se a ferramenta *Google Directions API*[13], para estimar o

[12] https://developers.google.com/maps/documentation/geocoding/intro Acesso em: 25/10/2018

[13] https://developers.google.com/maps/documentation/directions/start?hl=pt-BR Acesso em: 25/10/2018

tempo de deslocamento entre a localização do usuário e a unidade de saúde que o mesmo selecionou.

As primeiras telas desenvolvidas para a aplicação móvel, nomeada como CliqueMed, podem ser vistas na Figura 5. Na figura temos, em ordem: tela de carregamento do aplicativo, com o símbolo criado pelos autores para o mesmo; tela inicial; tela de busca por especialidade; tela com um exemplo de lista de resultados para uma busca por especialidade.



Escolha a especialidade que deseja buscar

ACUPUNTURA

Escolha a distância máxima

Distância: 4Km

DRA. NATASHA T. V. ISKOROSTENSKI MURTA
R NUNES MACHADO 472 CONJ 1605 E 1607 CENTRO
Distância: 0.508Km Mais Informações

HOSPITAL SUGISAWA
AV. IGUACU 1236 REBOUCAS
Distância: 0.653Km Mais Informações

ASA ECOCLÍNICA
R VOL DA PATRIA 475, 13º ANDAR - CJ 1301,1302,1303 CENTRO
Distância: 0.8Km Mais Informações

CLINICA EQUILIBRIUM
RUA EMILIANO PERNETA 860 SALA 301 CENTRO
Distância: 0.816Km Mais Informações

PONTUAL CLÍNICA MÉDICA LTDA
R VSC DO RIO BRANCO, 1488 - CONJUNTO 1007-ANDAR 10 CENTRO
Distância: 0.82Km Mais Informações

VIDA PLENA CLÍNICA REUMATOLÓGICA
RUA MARECHAL DEODORO, 630 CJ. 1704 CENTRO

Figura 5. Imagens prévias de desenvolvimento do aplicativo CliqueMed.

6. Considerações finais

Este trabalho apresentou uma proposta de desenvolvimento de uma aplicação móvel que disponibilize ao seus usuários informações válidas a respeito das unidades de saúde do estado do Paraná e permitisse a busca por especialidades médicas, como cardiologia, oftalmologia, entre outros, que se encontram mais próximas do usuário.

Até o momento, não há nenhuma ferramenta que busca coletar, tratar e divulgar essas informações de maneira sucinta para a população. Não há também nenhuma ferramenta que forneça informações sobre a rede pública junto com os dados da rede privada de saúde. Pode-se concluir que a falta de base confiável, para consulta de informações sobre estabelecimentos de saúde, pode gerar perda de tempo no momento que o paciente necessita de atendimento médico. Essa perda de tempo, para encontrar tratamento, pode se tornar um grande problema em momentos de necessidade.

Agradecimentos: Nós gostaríamos de agradecer ao Município de Curitiba, IPPUC, CRM-PR e o projeto *EUBra-BIGSEA (MCTI/RNP 3rd Coordinated Call)*.

Referências

Oliveira, M., Kozievitch, N., Bim, S., and Legal-Ayala, H. (2018). **Caracterização dos Dados Públicos de Saúde do Paraguai**. pages 12–21. XIV Escola Regional de Banco de Dados — ERBD – 2018.

Rocha, F. S., Santana, E. B., Silva, E. S., Carvalho, J. S. M., and Carvalho, F. L. Q. (2017). **Uso de Apps para a promoção dos cuidados a saúde**. STAES2017 - III Seminário de Tecnologias Aplicadas em Educação e Saúde.

Santos, T. O., Pereira, L. P., and Silveira, D. T. (2017). **Implantação de sistemas informatizados na saúde: uma revisão sistemática**. Revista Eletrônica de Comunicação, Informação e Inovação em Saúde - 2017.

Werneck, V. (2013). **Informática médica: para uma vida melhor no século 21**. pages 61–67. Computação Brasil – Revista da Sociedade Brasileira de Computação - 2013.

Index of authors