GEOINFO 2020

XXI Brazilian Symposium on GeoInformatics

November 30th to December 03rd, São José dos Campos, SP, Brazil

Proceedings

Tiago Garcia de Senna Carneiro and Carlos Alberto Felgueiras

Dados Internacionais de Catalogação na Publicação

SI57a Simpósio Brasileiro de Geoinformática (12.: 2020: São José dos Campos, SP)

Anais do 21o. Simpósio Brasileiro de Geoinformática, São José dos Campos, SP, 30 de novembro a 3 de dezembro de 2020. / editado por Tiago Garcia de Senna Carneiro (UFOP), Carlos Alberto Felgueiras (INPE) – São José dos Campos, SP:
MCTIC/INPE, 2020.
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. Carneiro, T. G. S. II. Felgueiras, C. A. III. Título.

CDU:681.3.06

Preface

GEOINFO 2020

This volume of proceedings contains the papers presented at the XXI Brazilian Symposium on Geoinformatics, GEOINFO 2020. Again, it was an honor and privilege for the Brazilian National Institute for Space Research (INPE) and the General Coordination of Earth Sciences (CGCDT) to host the GEOINFO event for the second consecutive year. For the first time, the GEOINFO was carried out entirely online because of the dangers of agglomeration of the COVID-19 pandemic. It was a new challenge to adapt the entire event from face-to-face to remote. Still, the important thing at this difficult time is to avoid illness and maintain the efforts made over the last 20 years to have our international symposium as a reference for the Brazilian Geoinformatics Education and Research Community.

In this edition, there were 52 high-quality submissions, of which 37 papers were accepted after being analyzed by our national and international academic and research reviewers. Fifteen full papers, 20 short papers, and 2 demonstrations were accepted. All the accepted papers were assigned for oral presentation during the Symposium. Authors from many distinct Brazilian and international academic institutions and research centers were represented. Participants of the event also had the opportunity to carry out questions to the authors after each oral presentation. Moreover, this year the GeoInfo was pleased to have three special keynote speeches. The first one by Profa. Dra. Cláudia Torres Codeço from the Oswaldo Cruz Foundation, Fiocruz, Rio de Janeiro State, Brazil addressing the keynote (in Portuguese) "*Vulnerabilidade geográfica para doenças emergentes: o que aprendemos com a COVID-19* (In Portuguese)". The second by Prof. Dr. Alexey A. Voinov from the Center on Persuasive Systems for Wise Adaptive Living (PERSWADE), School of Information, Systems and Modelling, University of Technology Sydney, Australia, presenting the keynote "*Participatory Modeling for Socio-Environmental Decision Making*". The third by Prof. Gregory Giuliani from the Institute for Environmental Sciences, University of Geneva, Switzerland, addresses the keynote "*Big Data for Big Challenges: The Swiss Data Cube for Environmental Monitoring*". The abstract of these keynotes and the entire official program can be accessed on the official program web page of the event http://www.geoinfo.info/geoinfo2020/program.php.

We want to thank all Program Committee members and additional reviewers, that were essential to guarantee the quality of accepted papers. All the submitted papers were evaluated by at least three academic specialists from the geoinformatics area. We also thank Daniela Seki, Adriana Gonçalves, and Gislaine Faria from INPE to staff the symposium preparation and execution. Finally, we would like to thank the Society of Latin American Remote Sensing Specialists (SELPER) to support the XXI GeoInfo.

Tiago Garcia de Senna Carneiro, UFOP
Program Committee Chair

Carlos Alberto Felgueiras, INPE
General Chair

Conference Committee

General Chair

Carlos Alberto Felgueiras
National Institute for Space Research, INPE

Program Chair

Tiago Garcia de Senna Carneiro
Federal University of Ouro Preto, UFOP, Brazil

Local Organization

Daniela Seki
Adriana Gonçalves
Gislaine Faria
National Institute for Space Research, INPE

Organized by

UFOP - Federal University of Ouro Preto
INPE - National Institute for Space Research

Supported by

SELPER - Associação de Especialistas Latinoamericanos em Sensoriamento Remoto

UFOP

Universidade Federal
de Ouro Preto

INPE

SELPER

Program Committee

Afonso de Paula dos Santos, UFV, Brazil
Alan Salomão, UERJ, Brazil
Antonio Miguel Vieira Monteiro, INPE, Brazil
Armanda Rodrigues, Univ. Nova de Lisboa, Portugal
Cédric Grueau, Polytechnic Inst. of Setúbal, Portugal
Carlos Felgueiras, INPE, Brazil
Carolina Pinho, UFABC, Brazil
Claudia Robbi Sluter, UFRGS, Brazil
Claudio Baptista, UFCG, Brazil
Claudio Campelo, UFCG, Brazil
Clodoveu A. Davis, UFMG, Brazil
Cristina Ciferri, USP, Brazil
Fabiano Morelli, INPE, Brazil
Fabrício A. Silva, UFV, Brazil
Flávia Feitosa, UFABC, Brazil
Gilberto Queiroz, INPE, Brazil
Giovanni Ventorim Comarela, UFES, Brazil
João P. de Albuquerque, U. Warwick (ICMC-USP), UK
José Alberto Quintanilha, USP, Brazil
José Giovanni Guzmán-Lugo, IPN, México
Jugurta Lisboa Filho, UFV, Brazil
Julio D'Alge, INPE, Brazil
Karine R. Ferreira, INPE, Brazil
Karla A. V. Borges, Prodabel, Brazil
Laercio Namikawa, INPE, Brazil
Lubia Vinhas, INPE, Brazil
Mário J. Gaspar da Silva, Univ. de Lisboa, Portugal
Marconi de Arruda Pereira, UFSJ, Brazil
Marcus Vinicius A. Andrade, UFV, Brazil
Maria Isabel S. Escada, INPE, Brazil
Mariana Abrantes Giannotti, USP, Brazil
Maxwell Guimarães de Oliveira, UFCA, 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
Salles Viana Gomes de Magalhães, UFV, Brazil
Sergio D. Faria, UFMG, Brazil
Sergio Rosim, INPE, Brazil
Silvana Amaral, INPE, Brazil
Thales Sehn Körting, INPE, Brazil
Tiago G. S. Carneiro, UFOP, Brazil
Valéria C. Times, UFPE, Brazil
Vania Bogorny, UFSC, Brazil
W. Randolph Franklin, Rensselaer P. Inst., USA
Yuri Lacerda, IFCE, Brazil

Contents

Traffic Flow at Night: a custom algorithm for identifying basal nighttime radiance levels of roadways

Gabriel Da Rocha Bragion[1], Gabriel C. Gonçalves[1], Ana Paula Dal'Asta[1], Ana Carolina De Faria Santos[1], Lucas Maia De Oliveira[1], Antônio Miguel Vieira Monteiro[1], Silvana Amaral[1].

[1]Remote Sensing – National Institute for Space Research (INPE)
Av. Dos Astronautas – São José dos Campos – SP – Brazil

{gabriel.bragion,gabriel.goncalves,ana.dalasta,lucas.maia, miguel.monteiro, silvana.amaral}@inpe.br, anacarolina.fs@outlook.com

Abstract. *The recent COVID-19 outbreak drove the attention to methods for monitoring the flow between settlements, including traffic flow. Although the remote sensing of nighttime lights is a viable option to estimate traffic flow derived indicators, changes on radiance levels at night are not all associated with traffic. This paper presents the theoretical approach proposed on the development of an algorithm able to identify spectrally unbiased control samples for regions of interest (ROI), namely roadway sections. Firstly, an overview of the algorithm is presented, followed by an empirical estimation of its time complexity. The results showed that the algorithm has an O(n) time complexity and that control samples and ROIs can have similar time series features, indicating that an analysis without the use of control samples can lead to biased results.*

1.Introduction

Typifying and monitoring regional road traffic spatiotemporal patterns might be crucial to better understand the possibilities of COVID-19 spread between human settlements. The monitoring of phenomena associated with the road traffic via remotely sensed data is mostly restricted to very high-resolution sensors or on-road measurements, which are often neither accessible nor systematically distributed [Tuerner et al. 2013]. Some studies exploited images and composites from the Visible Infrared Imaging Radiometer Suite (VIIRS) – Day/Night Band (DNB) to successfully detect and monitor light sources at night in a sub-pixel level, such as boats, gas-flares, and biomass burning, but only a few approached the traffic of land vehicles [Elvidge et al. 2015a, Polivka et al. 2016, Elvidge et al. 2015b, Chang et al. 2019]. Road traffic lights can be relatively dim and arguably hard to resolve from space, given the anisotropic factor and oblique emission angles of auto headlights [Kyba et al. 2014]. Therefore, the characterization of spatiotemporal patterns associated with road traffic radiance at night would be more robust if supported by methods for identifying patterns strictly associated with environmental changes, rather than the road traffic itself.

Albeit diverse, on-road sensors are most usually fixed and used for monitoring and fining purposes. Despite registering the total number of vehicles along a roadway section in a very fine spatial and temporal resolution, this type of sensor is not to be found in many smaller roadways. Smaller roadways are often the only vehicle route available to less prominent towns and play an important role in the spreading of contagious diseases to areas that are closer to the base of the human settlement hierarchy [Fortaleza et al. 2020]. The use of satellite remote sensing methods comes in handy, for it can produce spectral information in a regular and extensive form, surpassing the drawbacks of the on-road driven methods.

Although daytime high-resolution imagery matches these criteria, studies of this sort are mostly focused on object-oriented techniques, limiting the recognition of features that have a similar spatial scale to a given sensor ground sample distance (GSD) [Batz and Schäpe 2000]. High-resolution imagery still has a high cost of acquisition, generally covering areas only on demand, and lacking the higher availability usually meat by moderate resolution sensors. However, sub-pixel target detection based on the reflectance factor analysis, through moderate resolution sensors, usually requires a higher spectral resolution [Change and Heinz 2000]. In this sense, the detection of targets at night is a suitable approach, for it does not depend on a higher spectral resolution, neither a finer GSD.

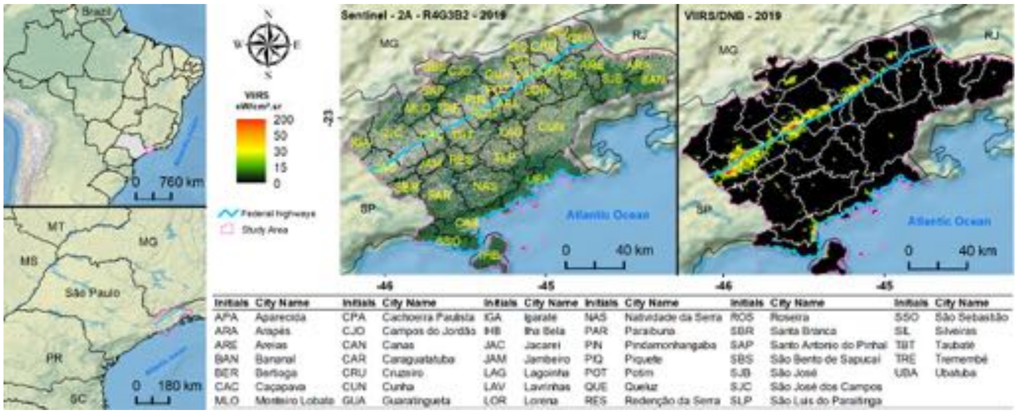
Target detection from nightly imagery is mostly based on the expected level and frequency of radiance associated with optical radiation sources [Elvidge et al. 2015]. The radiation amount from headlights measured by the DNB sensor can be lower than the amount expected from other typical artificial light sources, potentially lower than high albedo features and background areas near lit sites [Kyba et al. 2014]. Chang et al. (2019) presented a study analyzing the correlation between traffic flow and a DNB derived metrics from accumulated pixels overlapping freeways in China, and found out a broad correlation degree (R^2 ranging from 0.267 to 0.818), depended on the metric and vehicle type. Despite the outcome, factors like the higher density of roadhouses on higher flow freeways could lead to similar results, putting in check the assumption that these correlations are strictly due to the vehicle flow.

Previous studies showed that the monthly nighttime lights (NTL) average radiance is correlated to factors like the vegetation cover and changes in albedo and that some of these influences can be found even in the annual NTL composites [Levin 2017, Levin and Zhang 2017]. In order to identify different spatiotemporal patterns of road traffic from the monthly NTL composites, one must first investigate what is the contribution of other side parameters to the changes observed in the average radiance levels at night. Part of this problem could be analyzed by comparing the monthly average radiance associated with pixels lacking the presence of light sources to pixels overlapping roads of similar spectral response. This paper presents the theoretical approach applied to develop an algorithm able to identify spectrally unbiased unlit areas. These areas shall be used as control samples of the radiance levels from roadways sites, allowing a systematic estimation of the basal radiance level of a given roadway. An analysis of the empirically estimated time complexity is presented, given that early drafts of the code tended to present an exponential time complexity growth.

2. Material and methods

2.2. Study Area

The Metropolitan Region of the Paraíba Valley and North Coast (RMVPLN) is located in the State of São Paulo and, along the BR-116 highway, it connects the metropolitan regions of São Paulo (RMSP) and Rio de Janeiro (RMRJ) (Figure 1). The region comprises 39 municipalities divided into five subregions, holding a high variety of economic activities and a heterogeneous demographic distribution, most concentrated in the urban areas. [Gomes, Reschilian, and Uehara 2018]. In a fresh reading of the RMVPLN centered regional planning, Gomes, Reschilina, and Uehera (2018) pointed out that the strategic location of the RMVPLN seemed to led the political vision for development towards the exploiting of the distinct local advantages in a competitive way. Although logic, without the population, society, and political engagement, this historical approach for the development resulted in sub-regional inequalities, without the carrying for the urban-regional needs as well. Smaller town workers often adopt a daily routine of traveling across different municipalities in carpool systems or public transportation, increasing the probability of transmission and spreading of the COVID-19 from hub cities to smaller towns.



Initials	City Name	Initials	City Name	Initials	City Name	Initials	City Name	Initials	City Name	Initials	City Name
APA	Aparecida	CPA	Cachoeira Paulista	IGA	Igaratá	NAS	Natividade da Serra	ROS	Roseira	SSO	São Sebastião
ARA	Arapés	CJO	Campos do Jordão	IHB	Ilha Bela	PAR	Paraibuna	SBR	Santa Branca	SIL	Silveiras
ARE	Areias	CAN	Canas	JAC	Jacareí	PIN	Pindamonhangaba	SAP	Santo Antonio do Pinhal	TBT	Taubaté
BAN	Bananal	CAR	Caraguatatuba	JAM	Jambeiro	PIQ	Piquete	SBS	São Bento de Sapucaí	TRE	Tremembé
BER	Bertioga	CRU	Cruzeiro	LAG	Lagoinha	POT	Potim	SJB	São José	UBA	Ubatuba
CAC	Caçapava	CUN	Cunha	LAV	Lavrinhas	QUE	Queluz	SJC	São José dos Campos		
MLO	Monteiro Lobato	GUA	Guaratinguetá	LOR	Lorena	RES	Redenção da Serra	SLP	São Luís do Paraitinga		

Figure 1. Municipalities, Average Nighttime Radiance of the Metropolitan Region of the Paraíba Valley and North Coast and samples.

2.1. Data and algorithm procedural approach

Monthly cloud-free nightly composites ("vcm" version), which were the main input data of this work, are processed and made available by the Earth Observation Group (EOG), at Payne Institute for Public Policy website (https://eogdata.mines.edu/download_dnb_composites.html). The composites represent the monthly average radiance at the surface from daily cloud-free pixels belonging to images retrieved by the Day-Night Band (DNB) sensor. The Visible Infrared Imaging Radiometer Suite's (VIIRS) DNB sensor, onboard the joint NASA/NOAA Suomi National Polar-orbiting Partnership (Suomi NPP) satellite, retrieves daily radiance values at night, approximately at 1h:30min, local time. The instrument collects data on a constant 742x742 m footprint, but its monthly composites are binned to a global 15 arc-second geographic grid (~463m at the equator) [Elvidge et al. 2017]. Nightly sensed radiance values were needed for two different algorithm processes. Firstly, they are required to assess the presence of light sources under a pixel footprint. For this purpose, it was determined that every pixel with an average radiance value higher than 2ηW/cm².sr is not to be considered as background by the algorithm, a value considered higher than the average background radiance for latitudes between 10 and 50º [Elvidge et al. 2017].

Road network data was retrieved from the National Cartographic Base, made available by the Brazilian Institute for Geography and Statistics (IBGE) [IBGE 2019]. Although unlit roadways might be the closest available targets to be selected as control samples, a preliminary analysis showed that the magnitude of a cluster of pixels' radiances values overlapping a roadway can be as dim as completely unlit areas, in some cases. Moreover, there is no general optimal radiance value to specify if there are no light sources in a roadway, since traffic, noise and background radiance values are not stationary, both in space and time [Elvidge et al. 2017]. Therefore, the algorithm must automatically assume that a pixel overlapping a roadway is a non-background area.

Surface reflectance values were extracted from the MODIS MCD43A4 collection, band 1 (620-670nm), 2 (841-876nm), and 4 (545-565nm), a collection of images containing the best pixels of a 16-days-moving-window that have been modeled as if they were taken from a NADIR instantaneous field of view [Schaaf and Wang 2015]. The selected bands correspond to all the available bands in between the VIIRS/DNB spectral coverage (500 - 900nm). To increase the probability of high-quality pixels and proceed with the analysis with a more temporal compatible data, the quality assessment (QA) band of the MCD43A4 collection was consulted to produce a 30-day single composite for each month, ranging from January 2013 to January 2020. Finally, the processed MCD43A4 30-day images were reprojected to match the VIIRS/DNB grid.

VIIRS/DNB monthly NTL composites, MCD43A4 surface reflectance, and road network data are ingested into the algorithm (Figure 2a). Once the datasets are processed, the MCD43A4 data is associated to a region of interest (ROI) (Figure 2b). The goal of the algorithm is to find a cluster of 3x3 pixels, namely a control sample candidate, with the closest spectral response to a specific ROI sample, given a series of restrictions (Figure 2c and 2d). The proceedings illustrated by Figure 2 were implemented in a Python 3.5 environment.

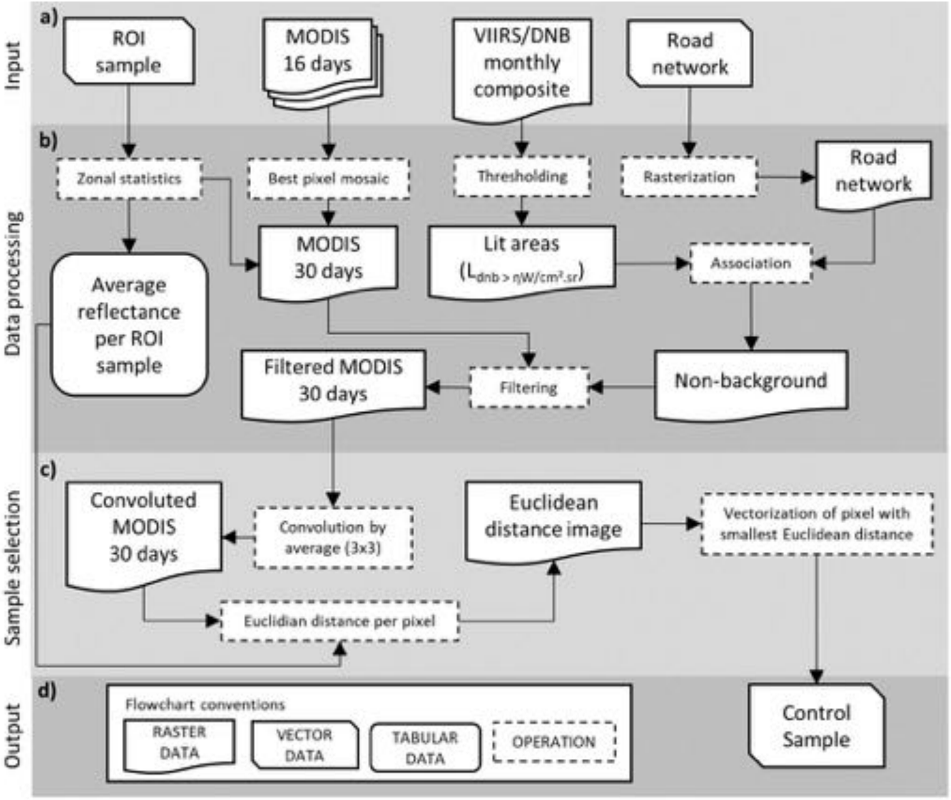


Figure 2. Flowchart of the proposed algorithm.

ROIs are clusters of pixels overlapping a roadway section. Currently, each ROI comprises nine pixels and can sum up to 5.9 km of roadway. A larger amount of pixels would result in a higher spectral mixture, difficulting the search and validity of a control sample. They were selected considering the functional role of the section, prioritizing roadways that give access to the municipalities but are not embedded in the lit urban area. The spectral response is represented by the average reflectance of a control sample candidate or ROI, based on the processed MCD43A4 bands, while its likehood is given by the Euclidean distance between those metrics (Equation 1).

$$\tilde{\lambda} = \sqrt{(\overline{B1}_{ROI} - \overline{B1}_{ctrl}) + (\overline{B2}_{ROI} - \overline{B2}_{ctrl}) + (\overline{B4}_{ROI} - \overline{B4}_{ctrl})} \qquad \textbf{(Equation 1)}$$

Where $\tilde{\lambda}$ is the spectral likehood, $\overline{Bn}_{ROI}$ is the ROI's average reflectance from the n'th band of the MCD43A4 product, and $\overline{Bn}_{ctrl}$ is the control sample's average reflectance from the n'th band of the MCD43A4 product.

Apart from the restrictions aforementioned, a control sample must not contain an invalid pixel. An invalid pixel is a pixel whose value has no true physical meaning, either due to instrument problems or cloudy atmosphere conditions during the acquirement of data. Based on these restrictions, the algorithm must test every sample candidate and then calculates their $\tilde{\lambda}$. Finally, the control sample candidate with the smallest spectral likehood is elected as a control sample for that specific ROI, given a specific month.

3. Algorithm time complexity and overview

A profiling of the algorithm identified the functions related to the calculation of average radiances and reflectances as the most time-consuming ones. Both functions increase the number of operations as the number of images or samples is increased. Those specific operations are dependent on third-party functions, making it difficult to determine the complexity of the algorithm in a theoretical approach. Therefore, we empirically tested the time demanded by the algorithm while increasing the number of images and ROI in 90 different combinations (Figure 3).

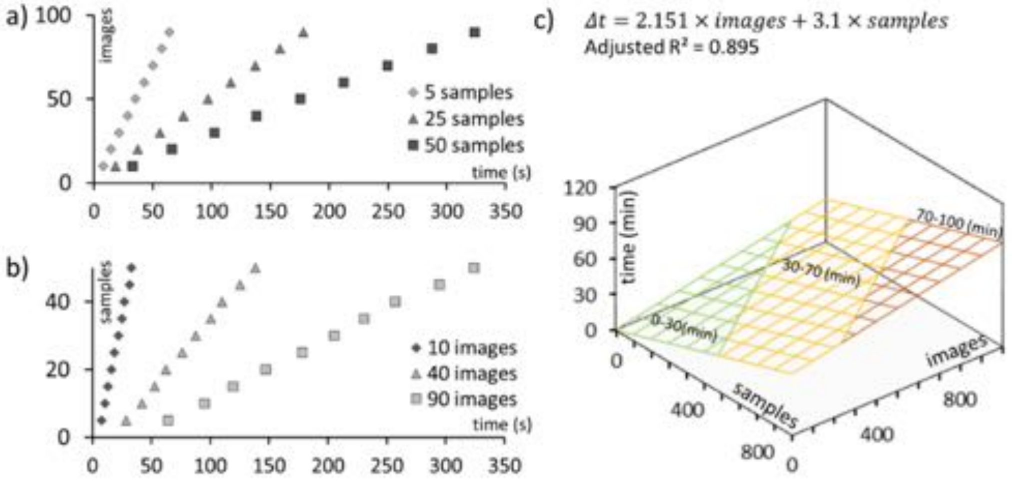


Figure 3. Algorithm time demand for different conditions regarding the number of images and samples, and fitted multivariate linear model.

In addition to the visual evidence, all the tested conditions presented a strong linear correlation ($0.996<R^2<0.999$; Figure 3.a and 3.b), suggesting that a linear multivariate regression model is appropriate to represent the time demand of the algorithm, regarding the number of images and samples (adjusted $R^2 = 0.895$; Figure 3.c). The ratio of the β_1 coefficient to the β_2 coefficient indicates that the number of images increases the time demand 44.12% slower than the number of samples. This is a positive outcome, since the number of images is the only dimension that grows indefinitely in a typical monitoring scenario. Another theoretical concern of the algorithm is the time complexity related to the sample selection method. Rather than testing the fitness of a given window as a potential control sample, and only then calculating the average reflectance, the method applied takes advantage of the algebraic implementation of the convolution filter [Tomilieri and Lu, 1997]. After reading all images as two-dimensional arrays, the algorithm defines all unelectable entries as non-numeric data. Therefore, all subsequentially operations result in a non-numeric entry, which is automatically excluded from the identification of the pixel with the smallest Euclidean distance, avoiding the need for multiple restriction tests.

Due to the association of restrictions criteria to non-numeric data, the algorithm is already able to filter off samples where there is a lack of good quality DNB or MODIS data, resulting in a gap in the time series. Moreover, the restrictions are all individually stored in arrays that can be retrieved based on the selected control sample coordinates. This allows the user to set quality flags to the output data, indicating what step has coerced the data to a non-numeric format, or even retain the values of the metrics needed for the processing methods and sample selection. This approach results in a series of metadata that can be used to further investigate the algorithm outcome and eventually investigate more precise alternatives to deal with problems concerning the input data quality. Currently, the output is a vector (or spatial table) comprising the average monthly radiance VIIRS/DNB values for both ROI and independent control samples, but the aforementioned metrics can be assigned to the table on demand.

Figure 4 displays examples of the algorithm's main outputs. Several relevant observations can be pointed out through a visual inspection of both averaged radiance time series (Figure a.2 and b.2). Regarding the averaged radiance level of the ROIs, it is clear that different roadways have distinct nominal radiances. While the seven-year time series of the BR-116 roadway shows values ranging from about 2.5 to 7.5 ηW/cm².sr, BR-353's has averaged radiance values barely higher than 0.4 ηW/cm².sr, the very same range observed in most of the control sample's time series. Likewise, Cao and Bai (2014) found averaged radiance values ranging from 2 to 4 ηW/cm².sr after analyzing daily DNB's radiances from a bridge section over the San Francisco Bay, California. These results indicate that it might be meaningless to define thresholds in order to separate lit from unlit areas, since there is a relatively wide range of mixture in radiance levels from background and dim or transient lit areas.

When it comes to heavy traffic roadways, such as the BR-116, the difference between the ROI's and the control samples' averaged radiance does not seem to change the time series' aspect (Figure 4a.2). Even though, after the subtraction, the resulting time series does present some relevant differences if compared with the original one, mainly expressed as shifts in the direction of the series in several pairs of months. The changes in the time series' aspect are clearer when observing roadways with a nominal dimmer averaged radiance. In Figure 4b.2, a major increase in the average monthly radiance is observed from 2017 onwards. Without taking account of the control samples' time series, it could be wrongly interpreted there was a relevant change in the vehicles' regime; or even a restructuring of the outdoor lights of nearby settlements.

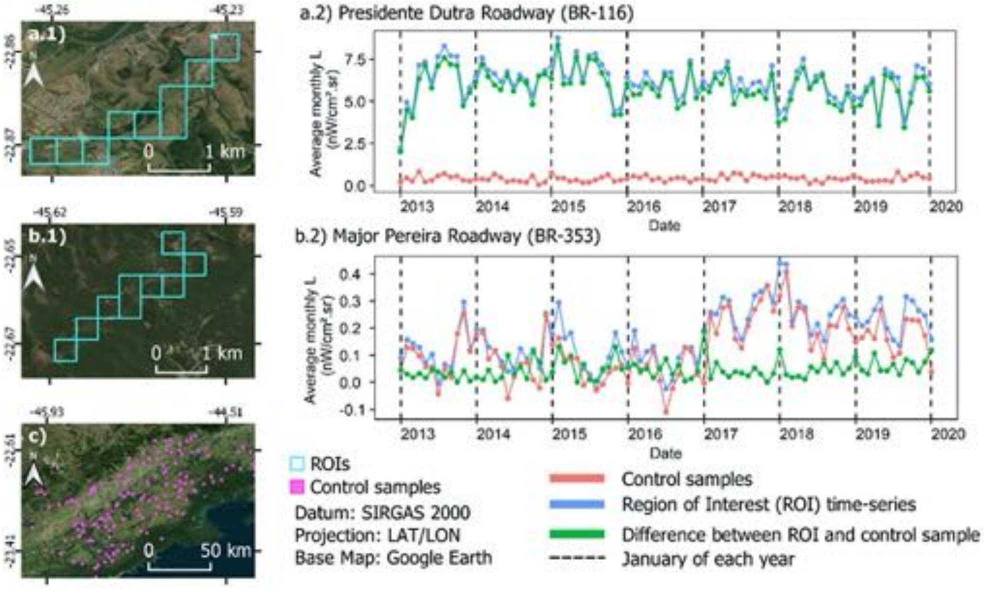


Figure 4. Location (a.1, b.1, c) and radiance level time series (a.2, b.2) of different ROIs and their respective control samples.

4. Conclusion

This paper presents the theoretical approach and the implementation strategy used on the elaboration of an algorithm able to access meaningful unlit control samples of monthly NTL composites based on its spectral response and restrictions criteria. An overview of the algorithm's empirically estimated time complexity showed that all the operations established by the code can be executed in a linear form. Currently, 85 monthly NTL composites are available to be ingested in a time series analysis, but this number will grow undefinedly. Moreover, daily processed nighttime images from the DNB sensor will be made available by NASA's Black Marble project soon (blackmarble.gsfc.nasa.gov/), stressing the need for algorithms that can be applied efficiently. In this instance, different datasets will certainly call for different likehood metrics and restriction criteria, nonetheless, they all can take advantage of the conceived structure of the presented algorithm.

By comparing the time series of roadway sections and spectrally-similar background areas identified by the proposed algorithm, it was confirmed that changes in radiance levels of roadways are not all associated with traffic flow at night. The results show that in the consideration of the VIIRS/DNB monthly composites as a dataset able to express quantitative information about the traffic of vehicles at night, the analysis of control samples is a necessary step. Whether variations in traffic flow can be detected by the VIIRS/DNB monthly composites, and what is the effect of the COVID-19 outbreak over the traffic flow at night are scientific questions that will be addressed in future works.

Acknowledgments

A Python 3.5 version of the algorithm discussed in this paper is available at the Laboratory for Investigations of Socio-Environmental Systems website (www.lissinpe.com.br/c%C3%B3digos). The authors are not aware of any conflict of interest. This study was financed in part by the Higher Education Improvement Coordination (CAPES) – Finance Code 001.

References

Baatz, M. and Schäpe. (2000). "Multiresolution Segmentation: an optimization approach for high quality multi-scale image segmentation". In: *Angewandte Geographische Informations-Verarbeitung, XII*, Karlsruhe, Germany, p. 12-23.

Cao, C. and Bai, Y. (2014). "Quantitative Analysis of VIIRS DNB Nightlight Point Source for Light Power Estimation and Stability Monitoring". *Remote Sensing*, v. 6, n. 12, p. 1 - 16.

Chang, Y.; Wang, S.; Zhou, Yi; Wang, L.; Wang, F. (2019). "A Novel Method of Evaluating Highway Traffic Prosperity Based on Nighttime Light Remote Sensing". *Remote Sensing*, v. 12, n. 1, p. 1-22.

Change, C., Heinz, D. C. (2000). "Constrained Subpixel Target Detection for Remotely Sensed Imagery". *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, n. 3, p.1144-1159.

Elvidge, C. D., Baugh, K., Zhizhin, M., Hsu, F. C., Ghosh, T. (2017). "VIIRS nighttimes lights". *International Journal of Remote Sensing*, v. 38, n. 21, p.5860-5879.

Elvidge, C. D., Zhinzhin, M., Baugh, K., Hsu, F. (2015a). "Automatic Boat Identification System for VIIRS Low Light Imaging Data". *Remote Sensing*, v. 7, p. 3020 – 3036.

Elvidge, C. D., Zhinzhin, M., Baugh, K., Hsu, F., Ghosh, T. (2015b). "Methods for Global Survey of Natural Gas Flaring". *Energies*. v. 9, n. 14, p. 1-15.

Fortaleza, C. M. C. B., Guimarães, R. B., Almeida, G. B., Pronunciate, M. and Ferreira, C. P. (2020). "Taking the inner route: spatial and demographic factors affecting vulnerability to COVID-19 among 604 cities from inner São Paulo State, Brazil". *Epidmiology and Infection*, v. 148, n. 118, p. 1-5.

Gomes, C., Reschilian, P. R., Uehara, A. Y. (2018). "Perspectives of the Regional Planning of Paraíba Valley and North Coast: Milestones and Institutionalization of the Metropolitan Region in the Action Plan of the Macro-metropolis of São Paulo". *Brazilian Journal of Urban Management*, v.10, n.1, p.154-171.

Instituto Brasileiro de Geografia e Estatística (IBGE). (2019). Base Cartográfica Contínua do Brasil, escala 1:250.000, version 2019. Accessed in: 28. Ago. 2020. Avaiable in: https://www.ibge.gov.br/geociencias/downloads-geociencias/cartas_e_mapas/bases_cartograficas_continuas/bc250.html.

Kyba, C., Garz, S., Kuechly, H., Miguel, A., Zamorano, J., Fischer, J., Hölker, F. (2014). "High-Resolution Imagery of Earth at Night: new sources, opportunities and challenges". *Remote Sensing*, v. 7, n. 1, p. 1-23.

Levin, N. (2017). "The impact of seasonal changes on observed nighttime brightness from 2014 to 2015 monthly VIIRS DNB composites". *Remote Sensing of Environment*, v. 193, p. 150-164.

Levin, N. and Zhang, Q. "A global analysis of factors controlling VIIRS nighttime light levels from densely populated areas". (2017). *Remote Sensing of Environment*, v.190, p.266-282.

Polivka, T. N., Wang, J., Ellison, L. T., Hyer, E. J., Ichoku, C. M. (2016). "Improving Nocturnal Fire Detection with the VIIRS Day–Night Band". *IEEE Transactions on Geoscience and Remote Sensing*, v. 54, n. 9, p. 5503-5519.

Schaaf, C., Z. Wang. MCD43A4 MODIS/Terra+Aqua BRDF/Albedo Nadir BRDF Adjusted Ref Daily L3 Global - 500m V006. 2015, distributed by NASA EOSDIS Land Processes DAAC, https://doi.org/10.5067/MODIS/MCD43A4.006. Accessed 2020-09-24.

Tolimieri, R., An, M. ., Lu, Chao. (1997). "Algorithms for discrete Fourier transform and convolution", New York, Springer.

Tuerner, S.; Kurz, F., Reinartz, P., Stilla, U. (2013). "Airborne Vehicle Detection in Dense Urban Areas Using HoG Features and Disparity Maps". IEEE Journal of Selected in Applied Observations and Remote Sensing, v. 6, n. 6.

QualiOSM: Improving Data Quality in the Collaborative Mapping Tool OpenStreetMap

Gabriel F. B. de Medeiros[1], **Lívia C. Degrossi**[2], **Maristela Holanda**[1],

[1]Departamento de Ciências da Computação – Universidade de Brasília (UnB)
Brasília – DF – Brasil

[2]Fundação Getúlio Vargas (FGV)
São Paulo – SP – Brasil

{gabriel.medeiros93,liviadegrossi}@gmail.com, mholanda@unb.br

Abstract. *The collaborative mapping tool OpenStreetMap (OSM) has a large database in which thousands of users are able to insert, edit and delete geographic data from the Earth's surface. As evidenced in multiple studies, collaborative tools tend to have a lack of data quality, since the information is often provided by inexperienced users. Due to its complexity, the quality of geographic data can be measured based on different aspects, which have been called quality dimensions in literature. In this context, this paper proposes the implementation of the QualiOSM tool in order to improve the quality dimension of attribute completeness within OpenStreetMap platform, increasing the address information associated with objects. The tool was tested in two different scenarios in Brazil: the city center of Brasilia, capital of the country, and part of the city of Rio Branco, in the state of Acre.*

1. Introduction

The activities of mapping and spatial data collection have undergone drastic changes in recent decades, due to factors such as the use of georeferencing, the emergence of devices with integrated GPS, the improvement of broadband internet and the development of high quality graphics. These new technologies have given rise to systems in which users are able to generate geographic information on a voluntary basis, thus the information contained in these types of systems has become popularly known as Volunteered Geographic Information (VGI) [Goodchild 2007], or more broadly, Crowdsourced Geographic Information (CGI) [See et al. 2016].

The increasing availability of CGI has drawn the attention of authors in the search for methods to assess data quality in collaborative activities [Degrossi et al. 2018], which were divided into three different categories: social media, collaborative mapping and crowd sensing [de Albuquerque et al. 2016]. In recent years, the proliferation of social computing practices has increased the amount of content generated by users online. This fact has brought positive and negative effects in relation to the study of geographic data and CGI [Meng et al. 2017]. On the one hand, the use of volunteers has enabled mapping of the most remote areas of the planet, where access is more difficult. On the other hand, collaborative data has brought difficulties regarding the degree of veracity of geographic information [Flanagin and Metzger 2008].

One of the challenges that researchers have in discussing, evaluating and measuring data quality is that it depends on different factors, like the characteristics of the volunteer and the type of information collected [Bordogna et al. 2016]. Thus, the concept of quality was divided into different aspects, which were called quality dimensions. In this way, some quality dimensions explored in the literature are accuracy, completeness, logical consistency and reliability [Firmani et al. 2016]. This work focuses on the dimension of completeness, represented as the proportion between the presence of metadata associated with a set of objects compared to the total number of objects in that set [Sehra et al. 2017].

In a CGI, the metadata is usually stored in the format of tags, which are treated as a key-value pair associated with the object in order to add new information. In most collaborative systems, users create or send content, make the notes they want using tags and share this information with other users, who can make any edits they deem necessary. The process of adding tags, also called tagging, has been described as one of the dilemmas associated with the behavior of users on Web 2.0, since incorrect tagging leads to unsatisfactory results in relation to the completeness of the information [Liu et al. 2011].

A successful example of CGI is the collaborative mapping tool OpenStreetMap (OSM), used in this paper as a case study. The data provided by the volunteers, as in OSM, requires special attention regarding the quality of the information, since users actively participate in the processes of editing, inclusion and exclusion of objects. One of the main reasons for the lack of data quality in these types of tools is the great heterogeneity observed in relation to its users, as they use different technologies and have different levels of knowledge [Senaratne et al. 2017].

In this context, this paper presents the QualiOSM tool, in order to improve the completeness of objects within the OpenStreetMap tool through the implementation of an automatic tag adder for adding address information to objects. The tests were carried out based on data collection in two different scenarios in the country of Brazil, taking into account the urban centers of the city of Brasilia, in the Federal District and Rio Branco, in the state of Acre.

The rest of this paper is structured as follows: Section 2 presents a set of works related to the theme of this research; Section 3 describes the implemented tool QualiOSM, as well as the methodology and architecture used for its development; Section 4 describes how data from Brazil was collected and later divided into the two test scenarios for using the tool; Section 5 presents the results obtained from the use of the tag adder implemented within the QualiOSM tool; finally, Section 6 presents the conclusion and future work.

2. Related Work

There are several studies in the literature that explored the process of adding tags in collaborative tools. For example, [Ames and Naaman 2007] explored the motivation for attributing tags to images on Flickr, concluding that most users tag objects to make information more accessible to the general public. In addition, [Kennedy et al. 2006] evaluated the performance of trained classifiers with photos from Flickr and their associated tags, demonstrating that tags provided by users contains a lot of misinformation.

In relation to the collaborative mapping tools, [Codescu et al. 2011] organized an ontology in order to standardize and facilitate the hierarchy of tags within the OpenStreetMap tool, but concluded that the use of an ontology is only efficient if users keep the tags constantly updated within OSM platform.

Still within OpenStreetMap, [Mooney and Corcoran 2012] carried out the analysis of more than 25,000 objects in the database of Ireland, United Kingdom, Germany and Austria. The results indicated that there are some problems arising from the way users assign tags to objects in OSM. The study also showed that these identified problems are a combination of the flexibility of the tagging process and the lack of a more rigid mechanism to verify the adherence to the OpenStreetMap ontology in relation to the tags added by its users.

Besides that, [Davidovic et al. 2016] used the recommendations provided on the "Map Features" page from the Wiki of the OpenStreetMap project[1] and analyzed the OSM database in forty cities around the world to see if contributors in these urban areas were using the guidelines in their tagging practices. The study concluded that compliance with the suggestions and guidelines is generally average or poor, since users in these areas do not always have the same level of knowledge.

Differently from the works mentioned above, this work proposes the implementation of the QualiOSM tool in order to improve the quality of geographic information within OpenStreetMap, especially with regard to the process of assigning address tags to objects. Thus, the intention of the tool is to contribute to the completeness of address information of objects in the OSM platform, assisting in automating the insertion of this information in the OSM platform.

3. QualiOSM

The QualiOSM tool was developed with the purpose of improving the completeness of address information associated with objects on the OpenStreetMap platform. Implemented as an extension (plugin) within the Java OpenStreetMap Editor (JOSM)[2], responsible for the largest number of object edits within the OSM platform, the application was written in Java programming language and can be downloaded from a public repository in Github[3].

Analyzing statistics present on the website TagInfo[4], it was observed that among the five most used tags for OpenStreetMap points, four are address tags ("addr:housenumber", "addr:street", "addr:city" and "addr:postcode"). It was also possible to observe that these four tags are included among the ten tags most used both for lines and for OpenStreetMap objects in general. In addition, the most used address tag, "addr:housenumber", was associated with more than 51 million points on March 1st, 2020, corresponding to more than a third of the total points contained in the OSM platform. In this context, the purpose of this paper is to implement the QualiOSM tool in order to generate the key-value pair for address tags within OSM, thus contributing to the improvement of information completeness in the OSM tool.

For the implementation of the tag adder within the QualiOSM application, the reverse geocoding technique was used, in which the extraction of textual information, such as name or address, is performed from a pair of geographical coordinates (latitude and longitude). This technique is common in many geographic application scenarios,

[1] https://wiki.openstreetmap.org/wiki/Map_Features [Accessed in May 2020.]

[2] https://josm.openstreetmap.de/ [Accessed in May 2020.]

[3] https://github.com/gmedeiros93/josm/tree/master/josm/plugins/Quali_OSM [Accessed in October 2020.]

[4] https://taginfo.openstreetmap.org/ [Access in May 2020.]

for example, free online mapping services [Kounadi et al. 2013]. In this work, the tool Nominatim[5] was used, looking for names and addresses in OSM data from a pair of geographic coordinates and generating the address data in Extensible Markup Language (XML) or Javascript Object Notation (JSON) format.

After verifying the presence of much incorrect information in relation to the tag "addr:postcode" for objects in Brazil within the Nominatim tool, it was decided to use the reverse geocoding tool CEP Aberto[6] in order to complement the postal code information of OSM - Brazil objects. Besides that, the list of postal codes in Brazil, called in Portuguese "*Código de Endereçamento Postal*" (CEP), which is presented in the database of Correios (Post Office service in Brazil), was downloaded in the form of a *.csv* file to check the accuracy of the postal code information entered in the platform.

Figure 1 presents the architecture used to implement the QualiOSM tool. As can be seen, the architecture was divided into three layers: the outermost layer is the Presentation Layer, responsible for providing the interface between the user and the JOSM data editor, in addition to providing the loading of aerial images; the QualiOSM plugin and the functionality of the tag adder were developed within the Application Layer, in which it is also possible to see the interaction with the OpenStreetMap tool API; finally, the Data Layer is responsible for providing data management in the OSM Database and interacting with the tools Nominatim, CEP Aberto and Correios Database.

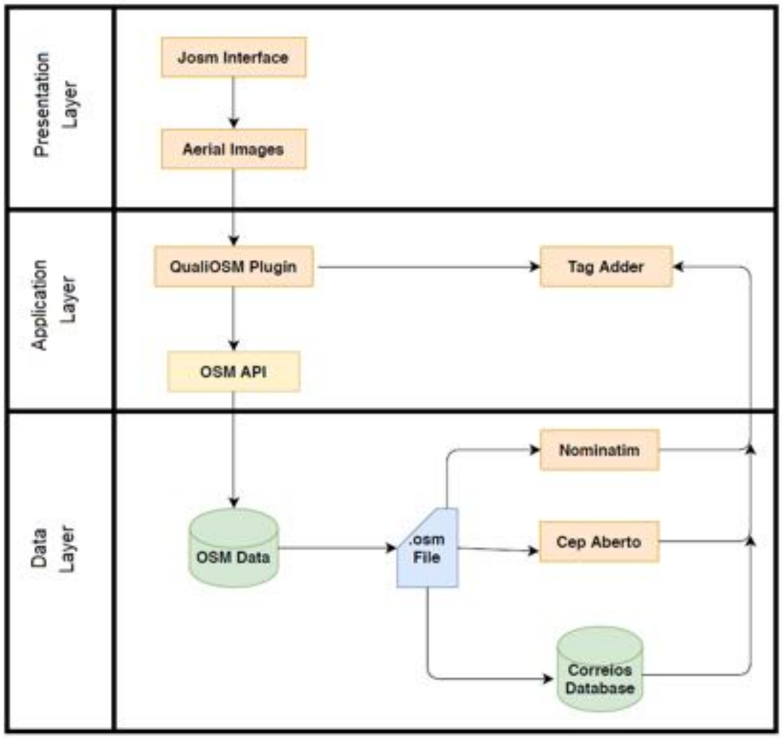


Figure 1. Architecture for implementing the QualiOSM application.

[5] https://nominatim.openstreetmap.org/ [Accessed in May 2020.]

[6] https://cepaberto.com/ [Accessed in August 2020].

The decision to implement the QualiOSM application within the JOSM data editor was reached for several reasons: (i) it is the data editor most widely used by OSM users [Ruta et al. 2012]; (ii) it is multiplatform, being written in the Java programming language; (iii) it offers a plugin mechanism to extend its main functionality. With an easily understandable user interface, the proposed tool can enable any OpenStreetMap user to enrich the map with address information, since no specific knowledge of semantic web languages or underlying formalisms is necessary.

After adding the plugin QualiOSM to the JOSM editor, the user can enjoy the functionality of the tag adder by loading the .osm file with the OpenStreetMap data to be edited on the map. Then, the user must select the objects and click on the "Add address tags" button. To insert the postal code information, the user can click on the options to use the tools Nominatim, CEP Aberto or Correios database. The interface of the QualiOSM tool can be seen in Figure 2.

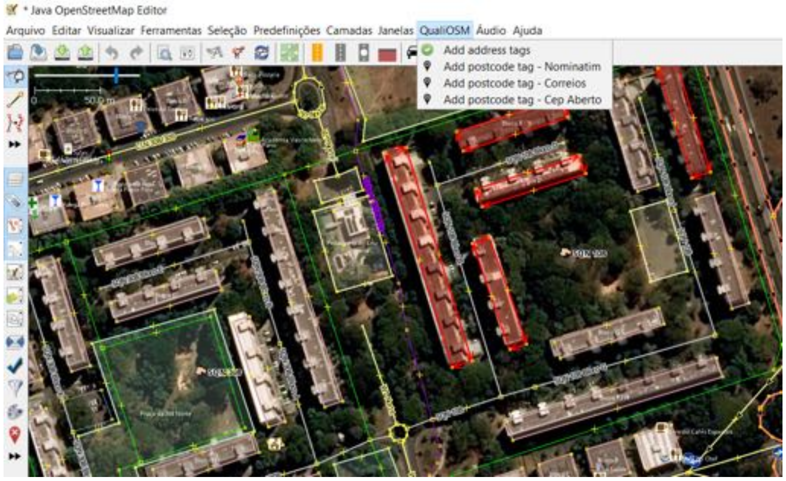

* Java OpenStreetMap Editor
Arquivo Editar Visualizar Ferramentas Seleção Predefinições Camadas Janelas QualiOSM Áudio Ajuda
Add address tags
Add postcode tag - Nominatim
Add postcode tag - Correios
Add postcode tag - Cep Aberto
50,0 m

Figure 2. QualiOSM tool interface.

4. Geographic Data in OSM - Brazil

In order to carry out some analysis in relation to address tags in the entire territory of Brazil, the data from OpenStreetMap - Brazil was downloaded following an architecture containing two main layers: a layer for collecting data and a layer for viewing and analyzing data. As can be seen in Figure 3, two files were used to collect OpenStreetMap data: the file FullHistory.osm [7], containing the history of OpenStreetMap tool data corresponding to the entire planet until October 31, 2019; and the Brazil.poly file, containing the outline of the Brazil region, made available on the Geofabrik project website[8]. Then, these two files were processed with the osmconvert tool[9] for the creation of the BrazilHistory.osm file, containing the history of OpenStreetMap data in Brazil. Next, the BrasilHistory.osm file was processed in the osm2pgsql tool [10] with the purpose of importing the

[7] https://planet.osm.org/planet/ full-history / [Accessed in May 2020.]

[8] https://download.geofabrik.de/south-america/brazil.html [Accessed in May 2020.]

[9] https://wiki.openstreetmap.org/wiki/Osmconvert [Accessed in May 2020.]

[10] https://wiki.openstreetmap.org/wiki/Osm2pgsql [Accessed in May 2020.]

data into the PostgreSQL database. In addition, PostGIS extensions were used to treat spatial data, and Hstore to capture tags of OpenStreetMap objects.

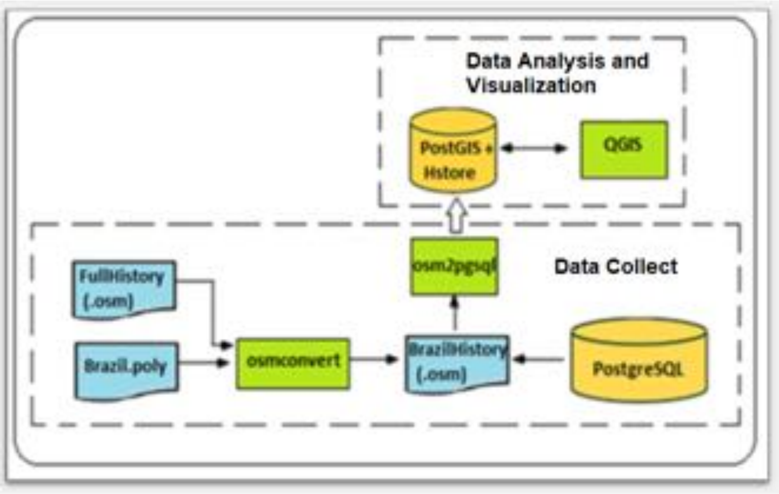


Figure 3. Architecture for collect and view data in OSM - Brazil.

The complete OpenStreetMap data in Brazil was downloaded so that analysis could be carried out in relation to the main labels used by mapping users in the country. The results showed that the most used tags were as follows: “addr:street”, “addr:city”, “addr:suburb” and “addr:postcode”. For this reason, these tags were chosen to integrate the tag adder implemented through the QualiOSM tool. From the OpenStreetMap data in Brazil, two different test scenarios for the QualiOSM application were considered:

- Scenario I - administrative region of Plano Piloto, in the city of Brasília. The center of the capital of Brazil is known for being a planned city, in which the buildings are arranged in an organized way and not very close to each other. Data were collected within the following bounding box: min latitude = -15.7929; max latitude = -15.7322; min longitude = -47.9093; max longitude = -47.8561.
- Scenario II - part of the city of Rio Branco, in the state of Acre (AC). This region was chosen based on the project “Mapping Flood Prone Urban Areas in Brazil”, available on the Hot Tasking Manager tool[11]. As can be seen, in this scenario houses are arranged much closer to each other, making the task of mapping the buildings more challenging. Data were collected within the following bounding box: min latitude = -9.9903; max latitude = -9.9733; min longitude = -67.8242; max longitude = -67.8021.

Since OpenStreetMap is a collaborative tool, it is natural that there is a great heterogeneity in the distribution of information mapping in relation to different regions, such as urban, rural and peripheral regions [Vargas-Muñoz et al. 2019]. Although mapping information on buildings and various other human constructions is widely available for urban areas, a significant number of buildings are not mapped in rural, peripheral regions or cities with less than 500,000 inhabitants, such as the city of Rio Branco.

[11] https://tasks.hotosm.org/projects/6124/ [Accessed in August 2020].

5. Results

Within OpenStreetMap, buildings are objects that often need associated address information, since users want to increase data about the location of points of interest, adding data such as the postal code, neighbourhood or building name. In this way, an analysis was carried out on the number of buildings that currently have address tags associated in Brazil and how these inclusions were made over time.

Thus, Figure 4 shows the evolution in relation to the inclusion of address tags in OpenStreetMap buildings in Brazil between the years 2009 and 2019. In this figure, it is observed that the inclusion of this type of tag has grown since 2015, but there is still a small number of buildings with associated address tags (in 2017, there were more than 860,000 buildings mapped, but only slightly more than 100,000 had associated address tags). In Figure 4 a peak of inclusion of these types of tags in 2017 is highlighted, mainly in relation to the tag "addr:street", corresponding to the street names. The predominance of this tag is because the OpenStreetMap tool has specialized in road mapping and information on names of roads near the buildings can facilitate routing mechanisms.

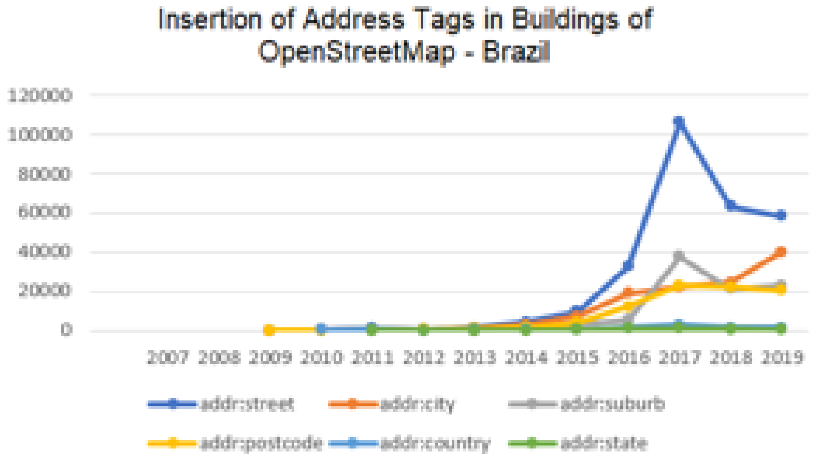


Figure 4. Insertion of address tags in buildings of OpenStreetMap - Brazil.

Within OpenStreetMap each user can also create their own labels to improve the map or to allow analysis of previously unmapped features. This feature can cause problems, such as the presence of misspelled tags which will be associated with a single object within the database. For the OpenStreetMap data in Brazil, 77 different address tags were identified, 27 of these tags (35%) were associated with only one object, and another 26 tags were associated with less than 10 objects.

Initially, the tests were performed with the addition of tags using the Nominatim reverse geocoding tool. Regarding the data from the first test scenario, a file in .osm format was downloaded containing the data of the administrative region of "Plano Piloto", a central district of the city of Brasília, corresponding to the data of July 31th, 2020. Regarding the second scenario, corresponding to the data in the city of Rio Branco in the state of Acre, data contained in the project "Mapping flood prone urban areas in Brazil" were used, through the Hot Tasking Manager tool. The tag adder was activated by selecting the preset "Human Construction/Edificaction" within the JOSM editor and then the adder was applied in the two different regions. After that, the tags associated with the selected objects were analyzed before and after the action of the tag adder.

The results obtained by adding tags in Scenario I can be seen in Table 1. In relation to the tag "addr:suburb", there was an increase from 1.76% to 41.83% of associated buildings; regarding the tag "addr:city", there was an increase from 2.12% to 45.46% of associated buildings; regarding the tag "addr:building", there was an increase from 0% to 10.21% of associated buildings. There was no change in relation to tags "addr:street"(5.28% of associated buildings) and "addr:housenumber" (1.59% of associated buildings) due to the lack of this information in the database of the Nominatim tool.

Table 1. Inclusion of address tags in Scenario I: Brasília - DF.

Tag	Before	After
addr:building	0%	10.21%
addr:city	2.12%	45.46%
addr:housenumber	1.59%	1.59%
addr:street	5.28%	5.28%
addr:suburb	1.76%	41.83%

Table 2 presents the results of applying the tag adder in the city of Rio Branco. As can be seen, the result was more satisfactory in relation to the inclusion of the tag "addr:city", in which there was a jump from 0.1% of associated buildings to 100% of associated buildings. However, there were no significant changes in relation to the other tags, "addr:building", "addr:housenumber", "addr:street" and "addr:suburb".

Table 2. Inclusion of address tags in Scenario II: Rio Branco - AC.

Tag	Before	After
addr:building	0%	0.40%
addr:city	0.1%	100%
addr:housenumber	0.03%	0.07%
addr:street	0.07%	0.07%
addr:suburb	0.03%	0.03%

When verifying the insertion of incorrect information in relation to the tag "addr:-postalcode", two more approaches were taken into account to include postal code information: using the reverse geocoding tool CEP Aberto and using the Correios database.

Cep Aberto acts similarly to the Nominatim tool, that is, from a geographic coordinate pair (latitude and longitude), it is able to search for information on that object and return this information in the form of a *.json* file. The Correios database, on the other hand, consists of a *.csv* file, with the coordinates of each object already associated. Thus, the postal code information was entered based on the coordinate closest to the center of the selected object in JOSM.

The distance was calculated according to the formula of the shortest distance between two points, expressed in the equation 1.

$$distance = \sqrt{(lat2 - lat1)^2 + (lon2 - lon1)^2} \tag{1}$$

Where (lat1, lon1) corresponds to the coordinates of the center of the selected object and (lat2, lon2) corresponds to the coordinates of the object within the Correios database. The algorithm finds the postal code of the selected object when this calculated distance is less than 10^{-4}.

An analysis was then carried out in relation to the addition of postal code tags in the tool, based on the reverse geocoding tools Nominatim and CEP Aberto, in addition to the use of the Correios database. The result for Scenario I (city of Brasilia) can be seen in Figure 5, in which it is observed that despite the fact that the Nominatim tool inserts postal code information for all selected objects, this tool adds lots of wrong information, having an error index of 96.15% and a hit rate of only 3.85%. The CEP Aberto tool obtained a hit rate of 17.31%, an error index of 26.92% and there was no addition of tags for 55.77% of the objects. Finally, the use of the Correios database led to a hit rate of 67.31% and did not add tags for 32.69% of the objects. One advantage of this approach is not adding wrong information to the OSM database.

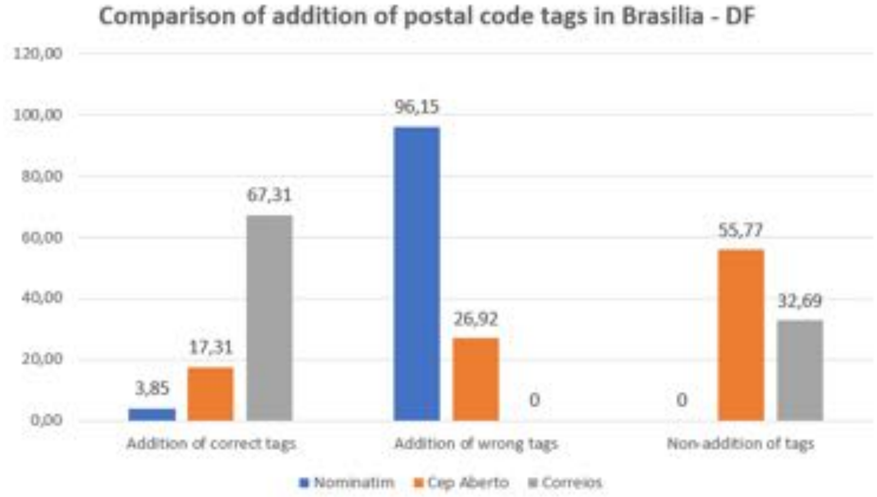


Figure 5. Comparison of addition of postal code tags in Scenario I.

Regarding Scenario II, in the city of Rio Branco, not much improvement was observed in relation to the completeness of the information using the CEP Aberto tool or the post office database. In addition, it should be noted that the Nominatim tool only inserted erroneous information in the QualiOSM tool, as can be seen in Figure 6.

An analysis was also carried out in relation to the time spent by the QualiOSM application for the inclusion of address tags in order to measure the performance of the tool. The tests were performed using a machine with 8.00 GB of RAM, Intel Core i7-9750H 2.60 GHz processor and Windows 10 operating system, 64 bits. For each selection of

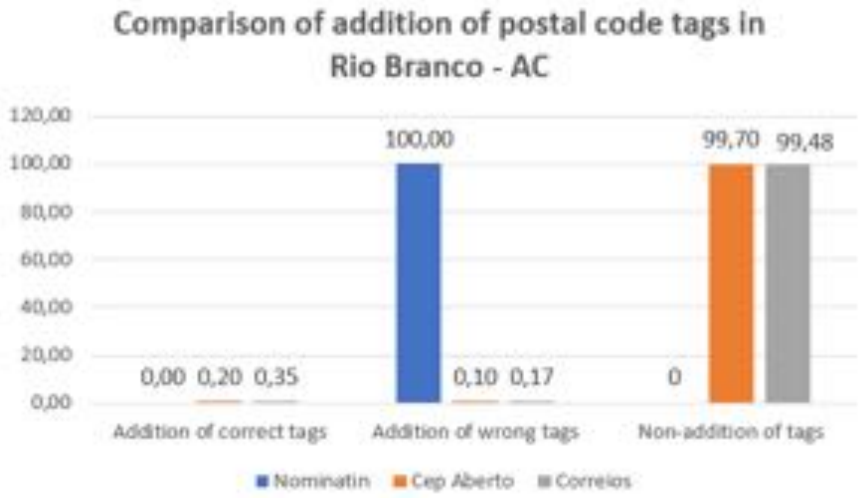


Figure 6. Comparison of addition of postal code tags in Scenario II.

objects, time was measured 10 times and the arithmetic mean was calculated. In this way, the results for Scenarios I and II are shown in Figure 7. Measuring the tool's execution time by adding ten more selected objects to each test, it can be seen that the time followed an approximately linear trend. Thus, the tool took an average of 500 milliseconds to include address tags per object.

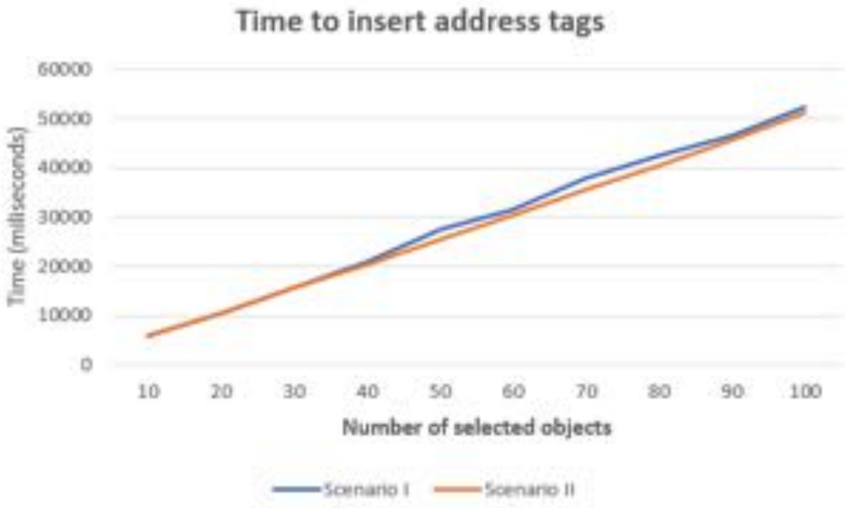


Figure 7. Time to insert address tags in the QualiOSM tool.

6. Conclusion

The tag adder implemented in this work has shown potential for improving the completeness dimension for object information within the collaborative OpenStreetMap tool. In large urban centers that are well mapped within OpenStreetMap, as is the case in the city of Brasilia, the developed tool QualiOSM improved by almost 70% the addition of postal code tags, which is an important tag for locating addresses, especially residential buildings.

It has been observed that the dimensions of completeness and accuracy often conflict with each other. The Correios database, despite having a good accuracy, still has many objects that are missing, especially when it comes to smaller urban centers, such as the city of Rio Branco, in Acre. On the other hand, with the Nominatim tool there was a greater increase in information, but there was also a greater increase in erroneous information, particularly, postal code information.

As a future work, we intend to explore other tags in addition to the address tags in this work, using other tools besides the Nominatim or CEP Aberto for finding information. It is also intended to test the tool in other scenarios and to evaluate other dimensions of quality in collaborative systems, such as logical consistency and accuracy.

References

Ames, M. and Naaman, M. (2007). Why we tag: Motivations for annotation in mobile and online media. *ACM SIGCHI Conf. Human Factors in Computing Systems*, page 971–980.

Bordogna, G., Carrara, P., Criscuolo, L., Pepe, M., and Rampini, A. (2016). On predicting and improving the quality of volunteer geographic information projects. *International Journal of Digital Earth*, 9(2):134–155.

Codescu, M., Horsinka, G., Kutz, O., Mossakowski, T., and Rau, R. (2011). Osmonto-an ontology of OpenStreetMap tags. *State of the map Europe (SOTM-EU)*, 2011.

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.

de Albuquerque, J. P., Eckle, M., Herfort, B., and Zipf, A. (2016). Crowdsourcing geographic information for disaster management and improving urban resilience: an overview of recent developments and lessons learned. *European handbook of crowdsourced geographic information*, pages 309–321.

Degrossi, L. C., Porto de Albuquerque, J., Santos Rocha, R. d., and Zipf, A. (2018). A taxonomy of quality assessment methods for volunteered and crowdsourced geographic information. *Transactions in GIS*, 22(2):542–560.

Firmani, D., Mecella, M., Scannapieco, M., and Batini, C. (2016). On the meaningfulness of “Big Data quality”. *Data Science and Engineering*, 1(1):6–20.

Flanagin, A. and Metzger, M. (2008). The credibility of volunteered geographic information. *GeoJournal*, 72:137–148.

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

Kennedy, L., Chang, S.-F., and Kozintsev, I. (2006). To search or to label? predicting the performance of search-based automatic image classifiers. *ACM Workshop Multimedia Information Retrieval*, page 249–258.

Kounadi, O., Lampoltshammer, T. J., Leitner, M., and Heistracher, T. (2013). Accuracy and privacy aspects in free online reverse geocoding services. *Cartography and Geographic Information Science*, 40(2):140–153.

Liu, D., Wang, M., Hua, X.-S., and Zhang, H.-J. (2011). Semi-automatic tagging of photo albums via exemplar selection and tag inference. *IEEE Transactions on Multimedia*, 13:82–91.

Meng, Y., Hou, D., and Xing, H. (2017). Rapid detection of land cover changes using crowdsourced geographic information: a case study of beijing, china. *Sustainability*, 9(9):1547.

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

Ruta, M., Scioscia, F., Ieva, S., Loseto, G., and Di Sciascio, E. (2012). Semantic annotation of OpenStreetMap points of interest for mobile discovery and navigation. In *2012 IEEE First International Conference on Mobile Services*, pages 33–39. IEEE.

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.

Sehra, S. S., Singh, J., and Rai, H. S. (2017). Assessing OpenStreetMap data using intrinsic quality indicators: an extension to the QGIS processing toolbox. *Future Internet*, 9(2):15.

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.

Vargas-Muñoz, J. E., Lobry, S., Falcão, A. X., and Tuia, D. (2019). Correcting rural building annotations in OpenStreetMap using convolutional neural networks. *ISPRS journal of photogrammetry and remote sensing*, 147:283–293.

Analysing the Tradeoff between Resource Consumption and Information Gain in the Gathering of Geolocation Data Using Smarphones

Thierry Silva Barros[1], **Claudio E. C. Campelo**[1]

[1]Systems and Computing Department
Federal University of Campina Grande – UFCG
Campina Grande – PB – Brazil

thierry.barros@ccc.ufcg.edu.br, campelo@dsc.ufcg.edu.br

Abstract. *Geolocation data have been widely used for the comprehension of various social phenomena. Nowadays, such data are produced on a large scale by people using their smartphones. However, the capture of this kind of data, in a mobile device, can be expensive, consuming device resources. On the other hand, the capture with low frequency may impair the quality and consistency of the information collected. In this context, we conducted a comparative study on the performance across different data collection frequencies to analyse the impact on resource consumption and data quality. Afterwards, an evaluation was performed in order to show the pros and cons of the different capture frequencies and estimate a frequency best suited for different usage scenarios.*

1. Introduction

Geolocation is the identification of a object's geographical location in the real world. This information commonly used to identify an electronic device's physical location. Geolocation data have been used to help comprehend various social phenomena. Nowadays, this kind of data is produced in large scale by people using their smartphones equipped with different types of sensors. Some examples of data generated by smartphones that may contain references to locations include photos, videos and posts in social networks. In addition, several applications retrieve user's location data with a certain frequency, in order to offer targeted products and services based on their locations.

Geolocation data are also used to conduct research with relevant social impact, such as those related to transport policies, public safety, traffic engineering and other topics related to urban planning. Moreover, this kind of data has been exploited to investigate people's trajectories, which express characteristics of human behaviour, enabling different kinds of studies, particularly in large urban centres [Kong et al. 2018]. Furthermore, location-based services has been using this type of data to predict the trajectory of users to recommend products and services based on their destinations or along their routes [Herder et al. 2014].

Although advances in mobile technology have increased the devices' capacity, both in terms of processing capabilities and battery life, these resources are still considered limited. In spite of this, application developers have been largely ignoring the cost for capturing geolocation data in terms of resource consumption. Very frequent geolocation data collection may cause a significant impact on resource consumption, reducing the devices' performance and providing bad experience for applications' users. On the other hand, low frequency captures may cause loss of crucial data. Consequently, information derived from these data may become inaccurate or uncertain. A deeper understanding of this tradeoff may help developers to minimise resource consumption while maintaining levels information granularity that do not impair their analysis.

For instance, a recommender system that captures geolocated data with 60-second frequency may lose crucial information about the user's locomotion occurred within this time interval, consequently offering services and information which may not be useful, frustrating the customers. On the other hand, an application with high collection frequency can considerably drain the battery of the user's device, or slow it down. Deciding the appropriate collection frequency is not a trivial task, since it is necessary to analyse the impact for the application's user and also the impact on the quality of information that may be derived from the data. Generally, arbitrary values are assigned by developers, without theoretical or experimental foundations.

Thus, in this work, we conducted a comparative study to observe the pros and cons of choosing different frequencies of data collection, in terms of effectiveness and efficiency. For this study, we developed an android application capable of collecting geolocation data along with other sensor data from smartphones. Ten volunteers were recruited to participate in the experiment, over a period of 4 weeks, using the application for 5 consecutive days (weekdays) for each collection frequency. At the end of each day, data about the consumption of smartphones' resources were collected from the volunteers. We then analysed different collection frequencies and compared them based on the consumption of smartphones' resources and the loss of information derived for different application scenarios. In this paper, we also discuss whether the choice of frequency has a significant impact on these two variables and whether specific frequencies appear to be more suitable for certain contexts.

The analysis of the tradeoff between information gain and collection efficiency was guided based on the use of data by geolocation analysis algorithms. The main algorithm used was the Dynamic Time Warping [Vaughan and Gabrys 2016], an algorithm used for comparison and alignment of two time series, which is commonly used in geolocation surveys to compare the similarity between trajectories. We also implemented a variation of the DBSCAN algorithm (Density-based Spatial Clustering of Applications with Noise) [Luo et al. 2017], targeted to identify stop regions in trajectory analysis problems.

The rest of this paper is structured as follows. In Section 2, we discuss the research methodology proposed in this work. Then, in Section 3, we present the obtained results. Finally, Section 4 concludes the paper and points to future work.

2. Methodology

This section describes the methodology adopted to analyse the tradeoff between different geolocalised data collection frequencies and information quality derived from these data.

2.1. Defined indicators

To enable the analysis of the proposed tradeoff, we first defined a set of indicators related to the resources consumption and the quality of produced information. The three indicators related to resource consumption are:

Battery Consumption: This indicator refers to the power consumption of the geolocation gathering application on the participant's device. Modern smartphones provide information on battery consumption by different applications, that is, the amount of battery spent by a certain application (not skewed by the use of other applications on the smartphone). The high battery consumption is one of the main resistance factors for using applications that capture geolocation data, thus we consider this a crucial indicator to be analysed.

RAM Memory Consumption: This indicator refers to the total memory allocated by location gathering application on the participant's device. High RAM consumption may impact the smartphone performance, decreasing the device's responsiveness and generating user discomfort. For this reason, we also consider this a very relevant aspect to be observed.

Amount of data transmitted: This indicator refers to the amount of data sent by the application to a cloud server over the network. High data transmission rates can directly impact the monetary amount paid by the users with internet packages (specially in non-developed countries), reducing their interest for the app. Hence, this is also a decisive indicator to be investigated.

Apart from the privacy concerns, we believe those indicators represent the main factors that make people avoid using apps that activate the GPS sensor very frequently. Moreover, relevant researches with the aim of tracking people's location have been conducted using considerably reduced databases [El Faouzi et al. 2011, Zheng 2015, Parent et al. 2013], and related issues have been reported regarding the volunteers' engagement.

We also defined two indicators to assess the quality of the information produced from the geolocation data: trajectories inferred from raw data (i.e., routes taken by the users); and stop regions, which are places within the users' trajectories where they stayed for a certain time (these places also include places of origin and destination). The latter is more related to the semantic aspects of the trajectory [Xiang et al. 2016] and is of high interest by both industry and academy, since it can be used to infer other relevant information, such as: types of places; human activities; points of interest; among others. This information, in turn, can be used to perform different kinds of analyses, such as those related to urban mobility patterns and trajectory prediction [Feng and Zhu 2016, Mazimpaka and Timpf 2016, Kong et al. 2018].

Additionally, we defined indicators based on the data collected from other smartphone's sensors: ambient lighting; proximity between the user and the smartphone; screen locked / unlocked; and audio status (i.e., normal, muted or in silence mode). From that data, we could estimate, for example, how accurate would be the information of whether the place the user is located is well lit. This kind of information can be considerably impacted by the granularity of geolocation data, since the light level normally changes as the users change their location.

2.2. Data capture strategy

To carry out this study, we used a mobile application developed in the research laboratory [Barros T. 2019], which can capture both geolocation data and smartphone sensors data. The research was divided into three stages. The first stage consisted in capturing the devices' resource consumption and geolocation data. To carry out these tasks, we count with the participation of 10 volunteers recruited by the researchers. The participants were subjected to a observational study where they installed the application on their device for data collection over a period of 20 days. The types of transportation used by the volunteers to move around the city were vehicles or buses.

Apart from the geolocation data, we collected data about the consumption of smartphones' resources at 4 different data collection frequencies: 15, 30, 60 and 120 seconds. Each frequency was observed for a period of 5 consecutive days, so that we estimate with greater precision the confidence interval of resource consumption of each collection frequency. The data about resource consumption was collected manually, since the application does not have functionalities for collecting consumption data in a automatic way. At the end of each day, the volunteers informed, through a messaging application, the data consumption of each resource.

Before starting the data acquisition phase, all volunteers were trained on how to obtain data from smartphone resources consumption and how to format the daily report. At the end of the first stage, four datasets were produced, one for each frequency collection, containing geolocation data at each different frequency, and 3 dataframes, one for each resource, containing data about the consumption of each resource by frequency.

2.3. Information generation

The second stage consisted in generating information from captured data. To generate information on resource consumption, confidence intervals for each resource are calculated by collection frequency. This estimate was used for verifying whether there is a significant difference in relation to resource consumption among the different collection frequencies. In order to calculate the confidence interval of each frequency, we first calculated the average values of the 5 days for each user; then, using this, the sample standard deviation could be calculated, and thus obtaining the confidence intervals.

In regard to the analysis of geolocation information, it should be noted that the contexts are different in each of the 4 datasets, that is, the participants visited different places and performed different trajectories in each of the 5-day periods. Thus, it is not possible to carry out an adequate comparative analysis between them. For example, we intend to carry out analysis in terms of the loss of geolocation information, such as the accuracy of the trajectory produced from a set of geographic coordinates. Thus, this requires to compare the same trajectory produced from coordinates collected at different frequencies, and consequently it becomes necessary to keep the same spatial context (i.e., to use spatial data produced in the same period of time).

To address this issue, we produced datasets of 30-, 45- and 60-second frequencies by temporally aggregating the data from the 15-second frequency dataset, which was built in the first 5 days of the data collection phase. By applying this methodology, we ensured that all 4 datasets have the same spatial context. The procedure consisted in removing values from the original dataset. For example, to generate a dataset of 30-second frequency from 15-second frequency dataset, we just removed half of the data points (alternately). After deriving such 3 "simulated" datasets, algorithms were applied on the data to generate geolocation information. The algorithms produced information about: trajectory identification performed by users, stop regions, information from the embedded sensors and the use of smartphone.

2.4. Metrics and information analysis

Finally, in the third stage, an analysis of the information generated in the previous stages was performed, in order to compare the tradeoff between the consumption of smartphone resources and the geolocation data quality. To perform the resource consumption analysis, we carried out a comparative study between the values of the confidence intervals for the consumption value of each resource by frequency collection, aiming at evaluating whether there is a significant difference between the values of each frequency. After analysing the consumption of each smartphone's resource by frequency, a comparative analysis was performed to quantify the impact that each frequency had in the loss of geolocation information, using the 3 derived datasets and the original one.

The metric used to compare geolocation data quality, in relation to the trajectories produced, was the level of dissimilarity, calculated though the Dynamic Time Warping algorithm [Lerato and Niesler 2019]. This algorithm provides a numerical value related to the measure of distance between trajectories, which can be interpreted as the level of dissimilarity, or loss of information, between the trajectories.

In order to identify stop regions, we made an ad-hoc implementation based on DBSCAN. In addition, to assess the quality of information on stop regions, the metric adopted was the similarity in the number of stop regions identified in each derived dataset and the original ones. A similar metric was adopted to calculate the similarity in relation to smartphone usage. For example, taking into account the original dataset compared with the 30-second derived dataset, to calculate the similarity between information on whether the smartphone screen is on or off, it is only necessary to check, in each pair, in sequence, of the original dataset (which was mapped to a single value of the derived set), the number of occasions when the screen was off and compare with the number of occasions when the screen was off in the derived dataset, since each pair in the original dataset value is mapped to a single value in the derived dataset.

The same strategy was also applied to calculate the loss of information in relation to the smartphone audio. In order to calculate similarity for the others information (i.e., ambient lighting, smartphone usage and identification of the audio mode), the metric used was the average of each sequence of values in the original dataset, which was mapped to the derived dataset, and calculate the difference, in percentage, in relation to the value of the derived dataset. Thus, the sum of all differences is the value of dissimilarity. The greater the value, the greater the loss of information. The next section presents the results obtained.

2.5. Devices (smartphones)

Strategy to select and prepare the devices: To carry out the experiments and measurements, the first step was to select volunteers who have Android devices, so that the data capture application could be installed. Most smarphones were Samsung Galaxy Note 3, Motorola Moto G5, Xiaomi Redmi Note 8, among other similar models. The Android versions were between 5.0 and 9.0. We required at least version 5.0 since it allows monitoring the resources consumed by applications individually (some older versions of Android only offered general device consumption). The next step was to install the *My Data Manager*[1] tool on the devices to monitor data consumption by application. Since Android supports the monitoring of CPU usage and energy consumption, no third party software was needed for obtaining this information.

Discussion on the use of different devices: The devices differ both in hardware configurations and in terms of Android versions. We decided to use a diversity of devices with the aim of obtaining a more accurate and generalised simulation of reality, that is, to obtain a greater representation of how the consumption of resources occurs in devices with different characteristics. The use of devices with different characteristics does not pose a risk to the validity of the obtained results, since the consumption information was captured individually by application, that is, the values analysed corresponds only to the resources consumed by the application used to conduct the research. In addition, we have ensured the same set of devices were used in different stages of this research (when different frequencies of collection were adopted). Thus, since the same diversity of devices are encountered in each group (each frequency), it possible to compare the average values obtained in each group, without posing a risk to the validity of the experiment.

3. Results and Discussions

By following the methodology discussed above, it was performed a comparative analysis of each resource consumption by collection frequency, which were calculated in terms of confidence intervals. In addition, it was also obtained the comparative analysis of the geolocalised data of the dataset with greater granularity (dataset with frequency time of 15 seconds), with the derived datasets (datasets of 30, 60 and 120 seconds). Finally, it was performed a comparative analysis for proximity sensors and ambient lighting, and for the indicators smartphone usage: whether the screen was locked or unlocked and if the audio was in normal, mute or silent mode.

In Figure 1, it is possible to notice that the data collection with a frequency of 15 seconds had the highest resources consumption. In contrast, the statistical test of the confidence intervals indicated that there were no significant difference in consumption of the resources, among the collection frequencies 30 and 60 seconds, because in all cases their confidence intervals had intersection of values. In other words, this result indicates that using the 60-second collection frequency does not present significant gains in the resources consumption in comparison to collection frequency of 30 seconds. Regarding the collection frequency of 120 seconds, it presented the best performance, that is, lower consumption, for data and battery resources, compared to all other frequencies analysed. However, there was no significant difference for the consumption of CPU compared to the frequencies of 30 and 60 seconds. Thus, it was possible to perceive that, for the consumption of resources, in certain cases, there is no advantages of using collection frequencies with lower collection granularity.

[1] My Data Manager - https://www.mydatamanagerapp.com

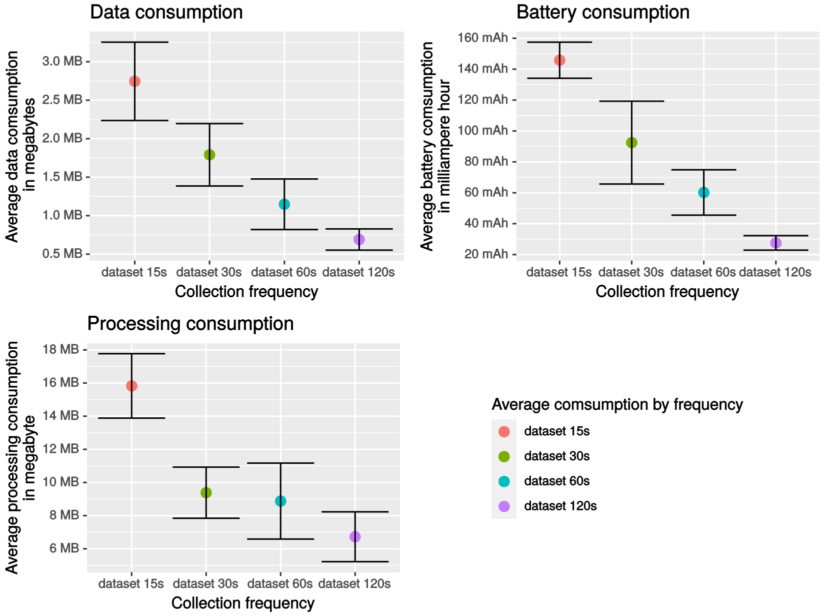


Figure 1. Charts of confidence intervals for each resource consumption by collection frequency

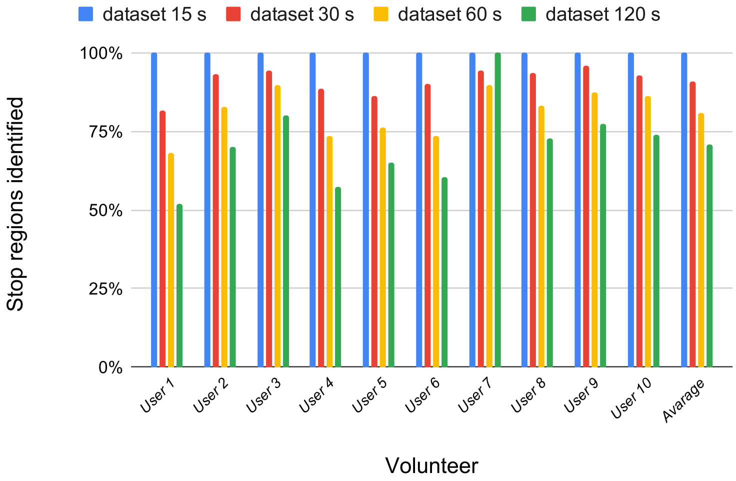


Figure 2. Chart of the number of stop regions identified by the original compared to derived datasets

In Figure 2, it can be seen the number of stop regions that were identified by each collection frequency. These stop regions took into account a radius of maximum distance of 50 meters, and a minimum time internal of 5 minutes. That is, for a stopping region to be identified, the user could not distance himself more than 50 meters from the region and stay at least 5 minutes in that region. The dataset with collection frequency of 30 seconds managed to capture an average of 90% of all stop regions, while the dataset with

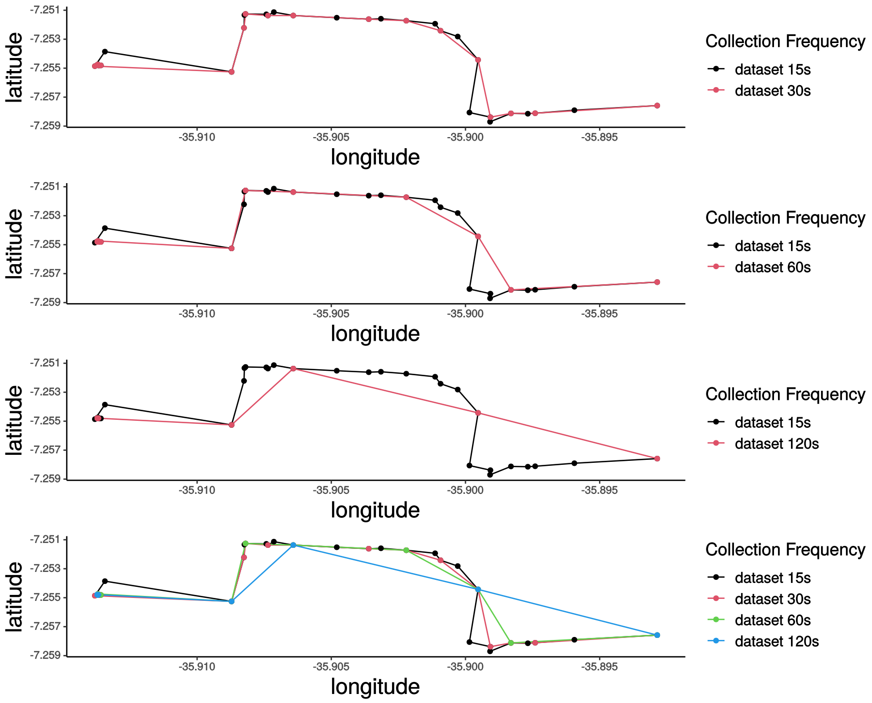


Figure 3. Graphs of a trajectory produced by dataset with a frequency of 15 seconds, compared to the same trajectory produced by derived datasets

a frequency of 60 seconds captured on average 80% and the dataset with a frequency of 120 captured only 70% of the stop regions. Hence, we can perceive a gradual increase in the number of stop regions that were not identified by the derived datasets. For the 30-second frequency, a loss of only 10% in the number of stop regions, for some usage scenarios, may not be so problematic. On the other hand, the adoption of a collection frequency of 120 seconds may significantly impact the application or research that relies on that information, given the average loss of 30% in the number of stop regions identified.

The distributions of the number of stop regions that were identified for each participating user are shown in the boxplots of Figure **??**. Depending on the movement pattern of each user, a different number of stop regions are identified. The variation was between 100 and 250, users who move more tend to have more stop regions in their trajectories, than users who stay stationary for a long period of time in the same region.

In Figure 3, it is possible to notice the loss of information of the trajectories produced by the derived datasets with collection frequencies of 30, 60 and 120 seconds, compared with the actual trajectory produced by the original dataset with collection frequency of 15 seconds. The loss of information on trajectories for the dataset with a frequency of 60 seconds, compared to the dataset with frequently of 30 seconds, grew by an average of 30%. For the dataset with frequency of 120 seconds, compared to the 30-second dataset, the loss of information grew by an average of 46%, indicating a significant increase in the loss of trajectory information produced by the derived datasets. Besides that, it is worth highlighting that this loss of information was obtained by taking into account only the first 5 days of study, where no volunteers have moved for a long period of time (such as a long

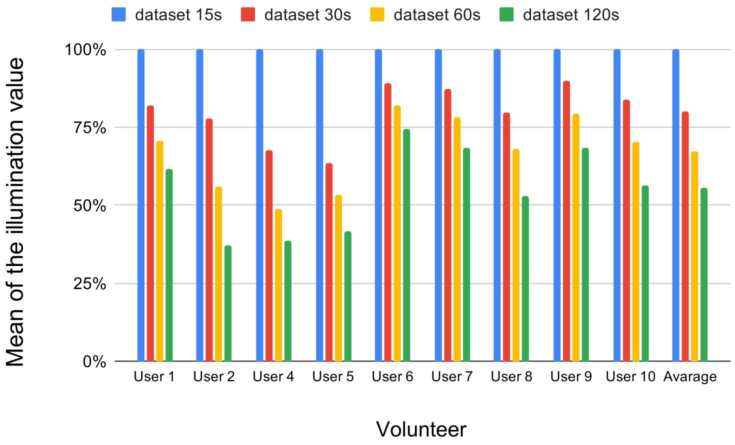


Figure 4. Average lighting values graph estimated by derived datasets compared to the actual values from the original dataset

road trip). If longer trajectories had been performed in the period studied, the observed loss of information could be even greater.

The averages values for information produced from the ambient lighting sensor are shown in Figure 4. As it is an indicator with a wide variation in its value over time, the loss of information was significantly large for the derived datasets, compared to the original dataset. This average value information was calculated using the average difference, in percentage, of the value estimated by the derived dataset compared to the actual value of the original dataset. The values obtained from the datasets with a frequency of 30, 60 and 120 seconds were, on average, 80%, 66%, 55% of the original value, respectively, indicating significant loss of information. Furthermore, if we remove the data in moments where the user was sleeping, on which the lighting was practically constant for a long period, the loss of information is even greater. For searches using this type of information, this difference may represent an significant estimate error in the survey. For example, a researcher/developer can try to determine whether the user was in a working environment or not, through the value of lighting, because the Brazilian Standard[2] determines that in offices and other working environments the ideal illuminance values should be from 500 to 1000 lux. Therefore, if data is captured from lighting with a hit rate of just 55% of the real value, the estimated value can easily be outside this range values indicating possibly misleading information, according to that the user would not be in a work environment.

The other sensor captured was the proximity sensor, which captures the distance that the front of the smartphone is from a particular object. In this case, the average inaccuracy of the values produced from the derived datasets was lower, when compared to the inaccuracy produced from the sensor lighting data. This is mainly due to the fact that it has less variation over time. The average of correct answers for collections with a frequency of 30, 60 and 120 seconds were 89%, 83% and 76%, respectively. This data can be important, for example, to determine situations on which the smartphone screen

[2] Standard NBR 5413 - Interior Illumination `http://ftp.demec.ufpr.br/disciplinas/TM802/NBR5413.pdf`

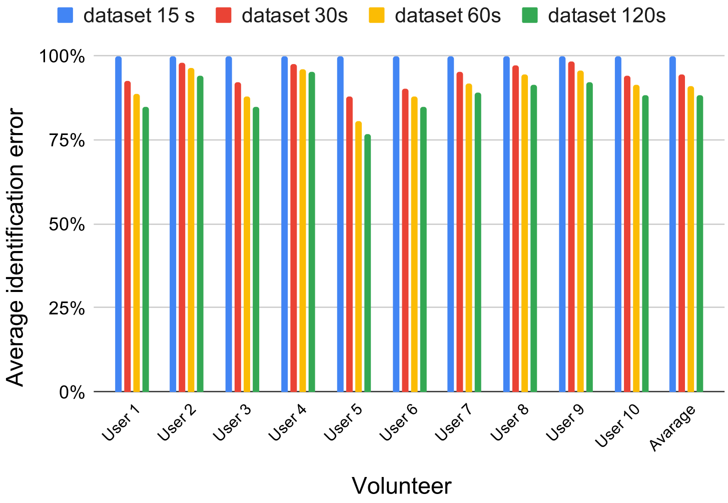


Figure 5. Graph of the average precision of the identification of the screen on or off

was close to the user's face, indicating that he could be in a call.

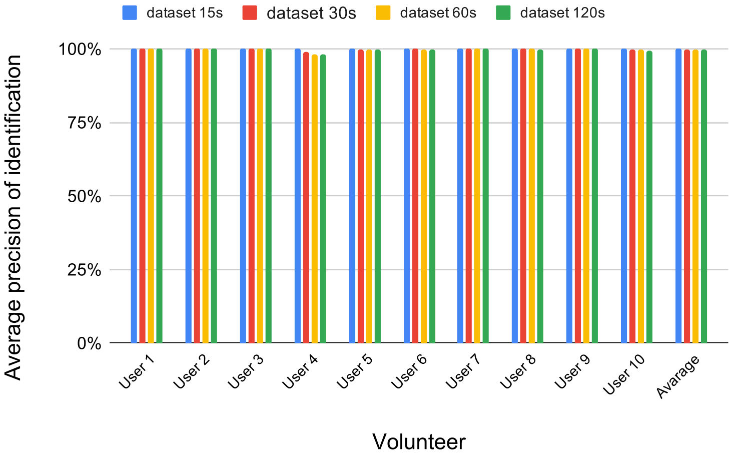


Figure 6. Graph of the average precision of the identification of the silent audio mode

Finally, in Figure 5, it is possible to observe the precision in relation to smartphone usage information, that is, if the screen was locked or unlocked. On average, the precision, was very high, for all the derived datasets, usually close to 90%. This is due to the fact that users, in general, leave the screen locked for a period of time longer than 1 or 2 minutes, or use the smartphone, with the screen unlocked, for a long period of time. The other data captured about the use of the smartphone was the audio mode (whether it was in normal, silent or silent mode). Similarly, as shown in Figure 6, the precision was also considerably high, averaging around 95% for all derived datasets. These results demonstrate that the frequency of collection has no significant impact on loss of information of smartphone usage.

4. Conclusion

The frequency collection of geolocation data can directly impact on the performance of smartphones and also on quality of applications that use these types of data to different needs. The result obtained by the comparative analisys presented in this article is an important artifact to support development teams and researchers in their decisions regarding the frequency collection of geolocation data, so that they maintain user satisfaction without compromise the quality of the captured data.

In this study, it was possible to observe the advantages and disadvantages that each collection frequency presented in comparison to the others. Regarding the consumption of resources, as expected, the frequency with greater granularity (collection frequency of 15 seconds) had the worst performance, that is, the highest cost in the consumption of resources. The 30-second and 60-second frequencies, in turn, showed no significant difference in resource consumption between them. These results indicate that there are no advantage in choosing any one of the two frequencies (in relation to the resource consumption).

The frequency with less granularity of collection (120 seconds) obtained the best performance in relation to battery and data consumption. Nonetheless, it did not show any performance gains in relation to CPU usage, compared to the frequencies of 30 and 60 seconds. From this results, it is possible to see that, regarding the smartphone's resources consumption, in certain situations, there are no advantages in changing between these frequency of collection.

In addition, in relation to the quality of the data produced, it was possible to perceive that, regarding spatial data, the loss of information was considerably large for the frequencies 60 and 120 seconds. For example, the collection frequency of 120 seconds only managed to capture 70% of all stop regions. Moreover, it was possible to notice that, in relation to the information obtained from smartphone's sensors data, such as the lighting sensor, the loss of information was even greater, with hits of just 55% of the reference value. Finally, in relation to smartphone usage data, the loss of information was considerably smaller, showing that there is no great difference in information loss depending on the collection frequency.

In this perspective, the comparative analysis showed that, depending on the situation, a collection frequency may be more indicated than another. For example, in scenarios where one needs spatial data or sensor data, but does not need a smartphone's low resource consumption, the use a frequency with a 15-second collection would be the most adequate to obtain the data with higher quality and better precision. On the other hand, in scenarios where just smartphone usage data are needed, the adoption of frequencies with less granularity would not significantly impact the quality of data, therefore being a good alternative for decrease the cost of resource consumption. In general, the 30-second collection frequency was the one that obtained the best tradeoff between the frequencies analysed, as it presented a reasonable performance in resource consumption (similar to consumption with a frequency of 60 seconds), while obtaining a considerably better performance in relation to the quality of the information produced from data (when compared to frequencies of 60 and above).

As future work, it is intended to replicate the experiment with a larger number of volunteers (at least 20), for a longer period of time (40 days). We also intend to analyse other factors, such as the activities performed by the users at specific times and other collection frequencies. Finally, other types of information derived from data should be considered, such as those related to quality of life metrics.

References

Barros T., Campelo, C. (2019). *Plataforma para fomentar a produção e a disponibilização de dados geolocalizados*. Federal University of Campina Grande, Campina Grande, PB, Brazil. Technical Report - Technological Initiation Program (CNPq-UFCG).

El Faouzi, N.-E., Leung, H., and Kurian, A. (2011). Data fusion in intelligent transportation systems: Progress and challenges – a survey. *Information Fusion*, 12:4–10.

Feng, Z. and Zhu, Y. (2016). A survey on trajectory data mining: Techniques and applications. *IEEE Access*, 4:1–1.

Herder, E., Siehndel, P., and Kawase, R. (2014). Predicting user locations and trajectories. volume 8538, pages 86–97.

Kong, X., Li, M., Ma, K., Tian, K., Wang, M., Ning, Z., and Xia, F. (2018). Big trajectory data: A survey of applications and services. *IEEE Access*, 6:58295–58306.

Lerato, L. and Niesler, T. (2019). Feature trajectory dynamic time warping for clustering of speech segments. *EURASIP Journal on Audio, Speech, and Music Processing*, 2019.

Luo, T., Zheng, X., Xu, G., Fu, K., and Ren, W. (2017). An improved dbscan algorithm to detect stops in individual trajectories. *ISPRS International Journal of Geo-Information*, 6:63.

Mazimpaka, J. D. and Timpf, S. (2016). Trajectory data mining: A review of methods and applications. *Journal of Spatial Information Science*, (13):61 – 99.

Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M. L., Gkoulalas-Divanis, A., Macedo, J., Pelekis, N., Theodoridis, Y., and Yan, Z. (2013). Semantic trajectories modeling and analysis. *ACM Comput. Surv.*, 45(4).

Vaughan, N. and Gabrys, B. (2016). Comparing and combining time series trajectories using dynamic time warping. *Procedia Computer Science*, 96:465–474.

Xiang, L., Gao, M., and Wu, T. (2016). Extracting stops from noisy trajectories: A sequence oriented clustering approach. *ISPRS International Journal of Geo-Information*, 5:29.

Zheng, Y. (2015). Trajectory data mining: An overview. *ACM Trans. Intell. Syst. Technol.*, 6(3).

Towards a Resilient Spatial Data Infrastructure

Helisson Luiz do Nascimento[1], **Cláudio de Souza Baptista**[1],
Fabio Gomes de Andrade[2], **Leanderson Coelho dos Santos**[2]

[1]Universidade Federal de Campina Grande (UFCG)
Caixa Postal 58109-970-Campina Grande-PB-Brazil

[2]Instituto Federal de Ciência de Tecnologia da Paraíba (IFPB)

helisson@copin.ufcg.edu.br, baptista@computacao.ufcg.edu.br

fabio@ifpb.edu.br

leanderson.coelho@academico.ifpb.edu.br

Abstract. *Spatial Data Infrastructures (SDI) has contributed expressively to the discovery and sharing of geospatial data. However, with the big number of geo-services available in SDI catalogs, and its sparse description, it is difficult to monitor the availability during the consumption of these resources. Aiming to overcome this limitation, this paper proposes an architecture that defines resilience layers in SDI context, with a Circuit Breaker pattern implementation for retrieving data even during instability. This architecture delivers a novel way of reliable access to resources and spatial data from the SDI catalogs.*

Resumo. *Infraestruturas de Dados Espaciais (IDE) têm contribuído de forma expressiva para a descoberta e o compartilhamento de dados. Porém, com o grande número de geo-serviços disponíveis nos catálogos das IDE, e sua descrição esparsa, é difícil monitorar a disponibilidade durante o consumo desses recursos. Visando superar essa limitação, este trabalho propõe uma arquitetura que define camadas de resiliência no contexto de IDE, com uma implementação do padrão Circuit Breaker para a recuperação de dados mesmo durante instabilidade. Esta arquitetura oferece uma nova forma de acesso confiável a recursos e dados espaciais dos catálogos de SDI.*

1. Introduction

The development of Spatial Data Infrastructures (SDI) has contributed expressively to the discovery and sharing of geospatial data. The implementation of these infrastructures has, among its primary issues, political and budgetary challenges [Grus et al. 2011]. Hence, issues related to their architectures have remained in the background for a long time. Over the years, most SDIs have been implemented based on the Service Oriented Architecture (SOA) standard, following the guidelines and standards defined by the Open Geospatial Consortium (OGC). The SOA architectural approach proposes that web systems must be broken into web services focused on business logic [Krafzig et al. 2005]. The increasing number of applications based on cloud SOA has enabled developers to refine concepts and establish architectural, technical, and organizational standards for the development of service-oriented applications. SDIs based on this architectural model have been conceived as monolithic structures [Assis et al. 2019], in which a set of applications run under a single process using a single module.

Nevertheless, SOA has remained a very broad concept, interpreted in different ways by different organizations. Usually, it has been related to a group of medium complexity services, which access the same database and communicate through an ESB (Enterprise Service Bus). This has inserted bottlenecks and points of failure in web applications.

The Microservice-Based Architecture has emerged as an alternative for dealing with the challenges identified over the maturing years of the concepts encompassed in SOA [Soldani et al. 2018]. Microservices can be defined as a group of small, autonomous services that work together [Newman 2015].

In a monolithic application, if an important functionality fails, the whole application stops working, causing an availability failure. In turn, microservice-based distributed architecture prevent applications from becoming completely unavailable. However, when an application is distributed in several microservices, many problems and possibilities of failures, which do not exist for monolithic architecture, require attention and control. With the development of applications with architectures based on microservices, large companies have used the resources and standards to structure their systems, as is the example of Netflix, which developed a framework for managing resilient architectures based on microservices, the Netflix Open Source Software [OSS 2020]. Features like Hystrix, Eureka, and Zuul make up the Netflix OSS framework and can implement patterns like Circuit Breaker, Service Discovery, and API Gateway that would empower distributed applications to achieve resilience and scalability.

The Circuit Breaker pattern [Montesi and Weber 2016] is one of the primary strategies to deal with the recovery of unavailable service requests. When a resource is unavailable, the Circuit Breaker acts as a proxy, similar to a tripped Circuit Breaker, throwing the exception immediately. This exception can be handled with a function that retrieves alternative resources. The Circuit Breaker can represent an interesting way to deal with the unavailability of services, by allowing to manage the effects of unavailability.

The amount of geo-services offered in SDI catalogs makes the task of monitoring the availability of these services very complex, requiring a method of dynamic integration of geo-services that provides a resilience layer. Microservice patterns can achieve some of these resilience requirements.

Recently, some authors have developed SDIs based on a microservice architecture [Assis et al. 2019, Mena et al. 2019, Li 2019]. In their works, they have extended the capabilities of the services provided by the infrastructure in terms of scalability, better use of cloud resources, and orchestration of container instances. However, there are still important issues that have not been addressed in the literature for the development of these infrastructures, such as resilience and fault tolerance.

In an SDI, it is expected that different services provide similar features about the same place or theme. Nevertheless, when a feature that is being used becomes unavailable for some reason, the client is in charge of searching the SDI's catalog to find a service that supplies a similar feature that could replace it. Since current infrastructures do not keep information about feature similarity, as well many services are poorly described in the catalog service, this task can be quite tedious and time-consuming. This problem can be especially critical in applications such as environmental monitoring and disaster management, in which real time decisions are due.

Aiming at solving these limitations, we propose an architecture that enables the implementation of resilient spatial data infrastructures. To validate our solution, we conducted a case study based on the Brazilian National Spatial Data Infrastructure (INDE). In this paper, we present a method for adding geo-services as vertices of a resilient architecture based on microservices. Using a Circuit Breaker implementation to manage unavailable resources, this architecture provide reliable access to SDI catalogs resources.

The remainder of this paper is structured as follows. Section 2 addresses related works. Section 3 presents our SDI architecture. Section 4 focuses on some use case scenarios. Finally, section 5 concludes the paper and points out further research to be undertaken.

2. Related Work

Over the years, several authors have analyzed the way in which SDIs are being implemented from an architectural point of view. Thus, some works proposed SDI implementations using the SOA architecture as an approach to the evolution of each service that makes up the infrastructure. Friis-Christensen et al. (2006) propose an SDI prototype aimed at assessing areas of damage caused by fires using SOA. Oliveira et al. (2008) proposed the application of SOA concepts by evolving a municipality GIS to a local SDI. However, the authors concluded that the implemented model did not perform satisfactorily. Basanow et al., (2008) used SOA principles to develop an SDI for 3D data. However, the analyzed services orchestration principles did not meet the complexity ranges that the system could reach. Likewise, Barik et al. (2016) also applied SOA principles to an SDI for the tourism sector in the city of Bhubaneswar, India. The authors detailed a methodology for building their own geospatial database. However, aspects of using data services from other SDI have not been addressed.

In general, based on initiatives led by the OGC [Friis-Christensen et al. 2006], many of the implemented SDI have sought to comply with at least some of the SOA principles. However, even complying with these principles, most applications do not reach minimum current requirements on scalability [Scholten et al. 2006], performance and availability [Soldani et al. 2018].

We observe that SOA has reached an important step forward in the systems complexity, as is the case of a cloud SDI. However, in a cloud environment, the microservice architecture achieves better performance than SOA. Several authors approach this topic as an architecture to enable SDI in the cloud. [Krämer 2018], for example, introduced a native cloud GIS proposal built on a microservice-based architecture, in order to process large volumes of distributed geospatial data. [Schäffer et al. 2010] carried out a study on the feasibility of implementing a cloud SDI. The authors identified some barriers to make this transition, including budget and legal difficulties.

[Li 2019] proposed a four-layer microservices architecture for public waterway information services. The proposed four layers are: data layer; microservices layer; application layer and client layer. Although they implemented a web application based on the microservice architecture, the authors did not explore availability and resilience issues.

Another application of microservices in SDI was proposed by [Assis et al. 2019] by presenting TerraBrasilis, an infrastructure for analyzing geospatial data on deforestation. In this proposal, the authors developed an SDI optimized for data analysis using a microservice-based architecture. The proposed solution performs real-time monitoring of system services in virtualized containers [Docker 2020], which facilitates scalability, enables the availability of resources and protects the system from potential external attacks. The system uses the agility of the microservice architecture to provide a geospatial data analysis platform regarding deforestation in the Brazilian cerrado, and, therefore, is focused on a particular domain. The authors deal with fault tolerance and availability for SDI using services available in virtualized computing environments in an IaaS (Infrastructure as a Service) platform. However, a more in-depth solution for handling unavailable services that goes beyond managing instances and service states is not addressed.

When implementing a cloud SDI using the microservice architecture, several possibilities arise for deployment automation, easy integration, as well as scalability. However, distributing an application into several microservices introduces some challenges such as the low reliability of the network. In particular concerning SDI, it is also necessary to take into account the possibility of unavailability of the data services that compose the SDI catalog. Therefore, in order to fully exploit the possibilities of cloud computing under microservice-based architecture, the use of resilience and fault tolerance patterns as Circuit Breakers is of fundamental importance.

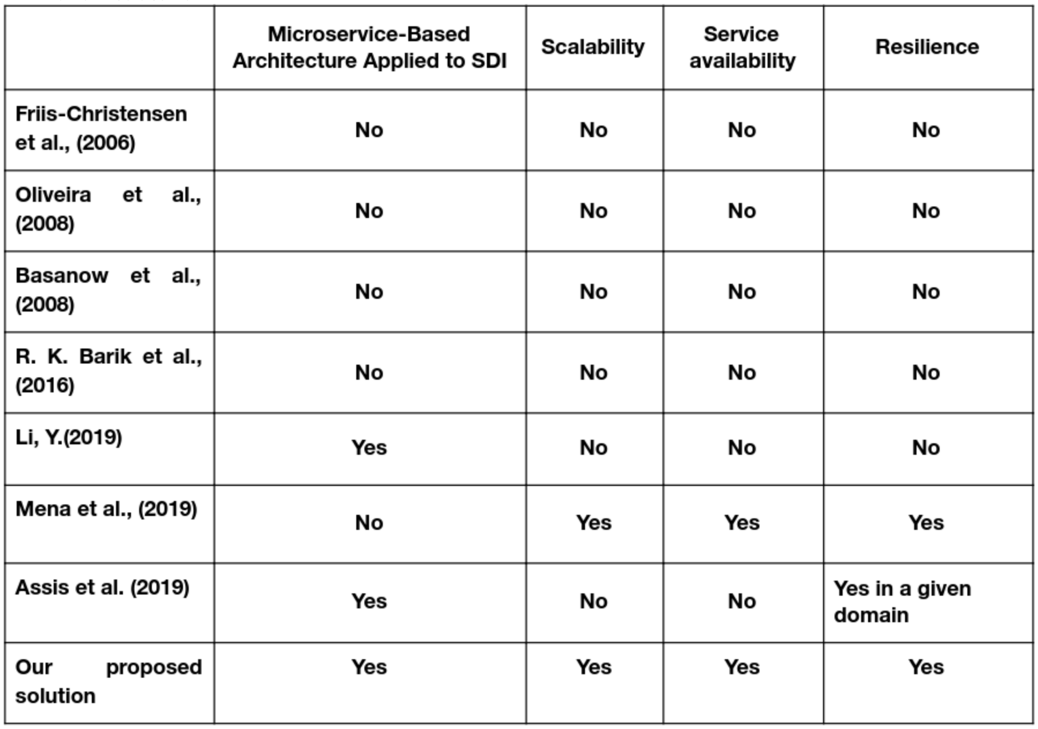
The handling of exceptions to redirect calls to services is something totally dependent on the business logic where the Circuit Breaker pattern is used [Nygard 2018]. In the context of applications such as SDI, it is important to prioritize data retrieval in order to improve the underlying decision making process.

Mena et al. (2019) apply microservice patterns to build a resilient application for geospatial data visualization. The application implements the Circuit Breaker pattern through the use of the Netflix OSS Hystrix framework [OSS 2020]. The authors isolated microservices in containers [Docker 2020], which are replicated and in case of unavailability of the microservice, the Ribbon [OSS 2020] is used as a resource to redirect calls to mirrored microservices. However, this work does not implement the OGC geo-services standards, as well the proposed architecture does not perform a scalable monitoring of geo-services availability. The work developed introduces a scalable geospatial data application, but not an SDI.

The solutions proposed in the aforementioned approaches aimed at cases of failure of a call or unavailability of a service that composes the internal architecture of the application. When dealing with the unavailability of external services, the alternative is a data return with low adaptability to maintain the user experience. Table 1 presents a comparison of the aforementioned research works.

The application of resilience patterns in the availability of spatial data resources is still an unexplored theme. The works that applied the microservices-based architecture paradigms to SDI did not address this aspect directly. Therefore, in this paper we introduce an architecture based on microservices that applies resilience standards in the

Table 1. Comparison between geospatial approaches that use service oriented architecture



	Microservice-Based Architecture Applied to SDI	Scalability	Service availability	Resilience
Friis-Christensen et al., (2006)	No	No	No	No
Oliveira et al., (2008)	No	No	No	No
Basanow et al., (2008)	No	No	No	No
R. K. Barik et al., (2016)	No	No	No	No
Li, Y.(2019)	Yes	No	No	No
Mena et al., (2019)	No	Yes	Yes	Yes
Assis et al. (2019)	Yes	No	No	Yes in a given domain
Our proposed solution	Yes	Yes	Yes	Yes

provision of geo-services.

3. The Proposed Architecture

To create a resilient application of geospatial data, it is necessary to build an infrastructure agile enough to solve situations of unavailability with low latency. Figure 1 depicts our high-level architectural model, developed from the concepts explored in the topics aforementioned.

As an SDI aggregates resources from different services, it must be aware of the availability of each of these sources. Dealing with each geo-service available in the SDI catalog as an architectural microservice enables the resilience and recovery patterns available for microservices-based architectures. Thus, we consider each geo-service that compose the SDI catalog as a point of failure within the architecture, seeking to establish strategies to mitigate possible downtime.

The features of an SDI are made available from geo-services, which are implemented based on the OGC standards. One of these standards is the Web Map Service (WMS), which renders and returns images of the available geographic resources [OGC 2020]. In our work, we focused on providing reliable access to WMS services because they are widely used in different applications and contexts. So, they serve as a main entry point for SDI spatial data consumption.

Next subsections detail the functions and the characteristics of each component of the proposed architecture.

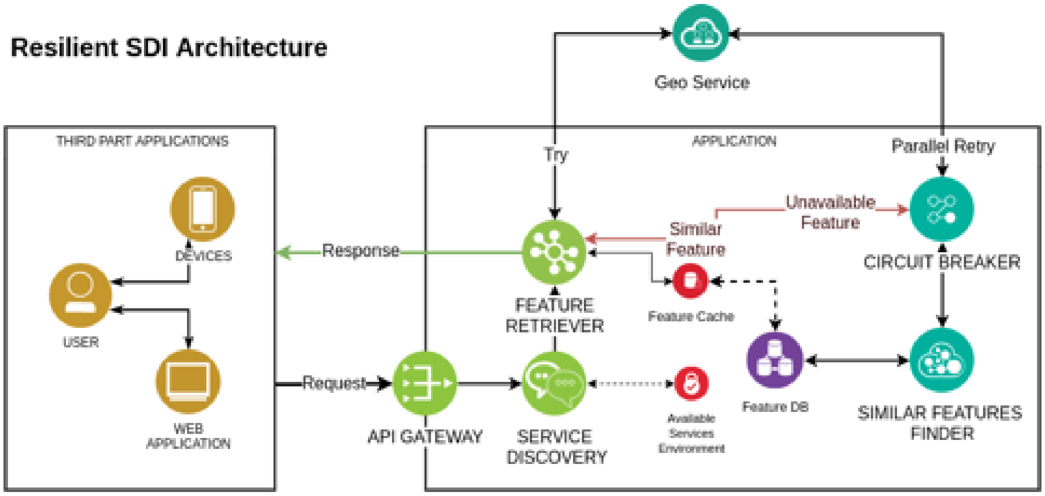


Figure 1. Our Proposed architecture.

3.1. The API Gateway

The *API gateway* performs the management of our application's REST entry-points available for public requests. This component intercepts the requests for services and redistributes them through the application, which processes the necessary actions. The importance of this resource is due to the monitoring, which are the most common paths taken by users and services that use the application. This data is useful to guide the evolution of the architecture, prioritizing the most used resources.

The *API Gateway* is also responsible for performing the load balancing, in which requests are redistributed among the available service instances. To accomplish this task, microservices are instantiated in Docker containers [Docker 2020]. Then, when the application is under a heavy load of requests, it is possible to instantiate several microservices in parallel to balance the load and distribute the requests.

Figure 2 shows the *API Gateway* flow for two different requests. The *API Gateway* checks if the service requested in the request is unavailable. The *API Gateway* accesses the *Available Services Environment* to check if the requested service is unavailable and has been replaced by another one. If applicable, the request is adjusted and forwarded to a *Feature Retriever* instance to get the data.

3.2. The Service Discovery

When an application is distributed on the network, several instability and latency problems may arise among the possibilities of failure. Then, it is always necessary to check which services are available. The primary function of a Service Discovery implementation is to listen and keep information about the microservices operating in the application. When a microservice is successfully instantiated, it is registered with service discovery to receive requests. When a service is unavailable, the Service Discovery is notified, and this service is added to the list of unavailable services until it registers again.

In our architecture, the *Service Discovery* registers the instances of available microservices and operates an *Available Services Environment*. This environment is used by

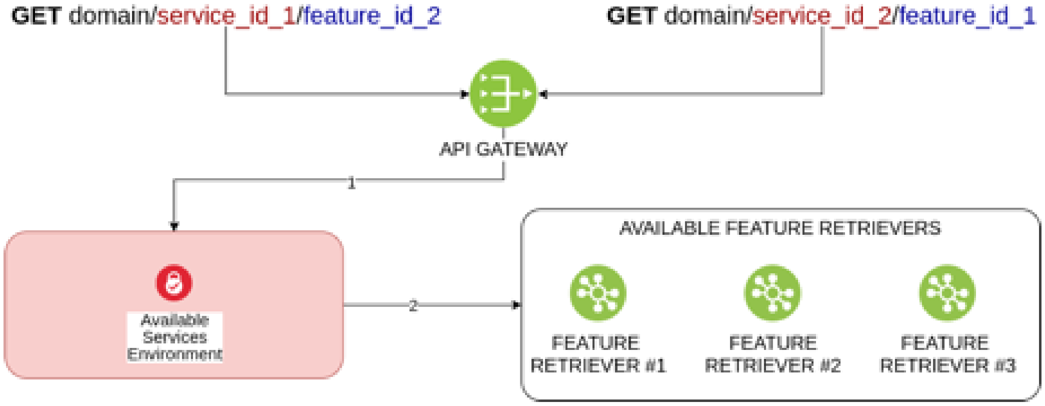

GET domain/service_id_1/feature_id_2
GET domain/service_id_2/feature_id_1
API GATEWAY
1
2
Available Services Environment
AVAILABLE FEATURE RETRIEVERS
FEATURE RETRIEVER #1
FEATURE RETRIEVER #2
FEATURE RETRIEVER #3

Figure 2. API Gateway process.

other fixed microservices of the application as an environment to check if the service that provides a requested resource can be replaced by an available service that has the same resource.

The *Available Services Environment* associates the identification of a service and feature with its similar one currently available. This strategy has been adopted because when dealing with third party services it is not possible to receive registration requests. Hence, only missing services and their available counterparts are registered.

3.3. The Feature Retriever

The *Feature Retriever* seeks to work with an interface that guarantees the safe recovery of data and features of a geo-service with maximum security. This service has only one HTTP GET route as an access point, as the goal is to simplify access to resources as much as possible and facilitate replication in different containers, since this is the main load point of the application.

In the *Feature Retriever* there is a cache layer under a database that contains information about the geo-services and feature types provided by the SDI, as it is shown in Figure 3.

The service poses a query to retrieve the requested feature formatted URL. When the *Feature Retriever* is replicated in several containers, its cache is also replicated, avoiding overloading the main database and maintaining the principle of decentralized access to data.

As is depicted in Figure 3, the *Feature Retriever* gets the access data to a geo-service and performs an attempt to retrieve the data, as requested by the user. If the request receives a successful response, the response is forwarded to the user. If an error occurs when requesting access to the geo-service, the *Feature Retriever* accesses the *Circuit Breaker* to retrieve an available geo-service that has a similar feature, similar data is returned to the user from the *Feature Retriever*.

3.4. The Circuit Breaker

The Circuit Breaker pattern is used as one of the main strategies for handling requests to unavailable services in microservices-based architectures. In this case, the *Circuit Breaker*

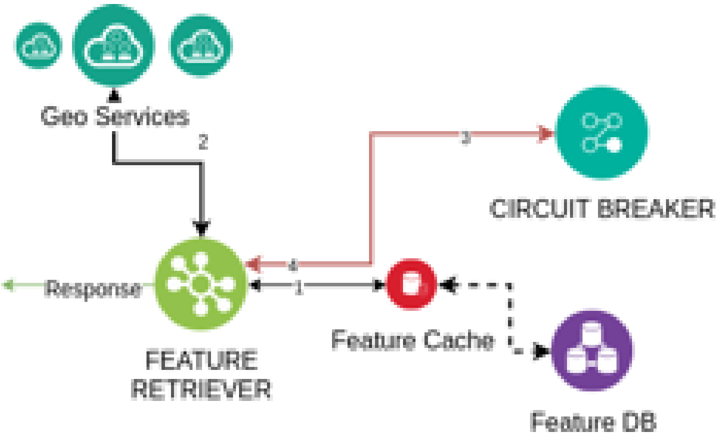


Figure 3. Feature Retriever process.

acts as a proxy, similar to an eletrical breaker that opens the circuit when it detects changes in voltage to avoid unwanted consequences.

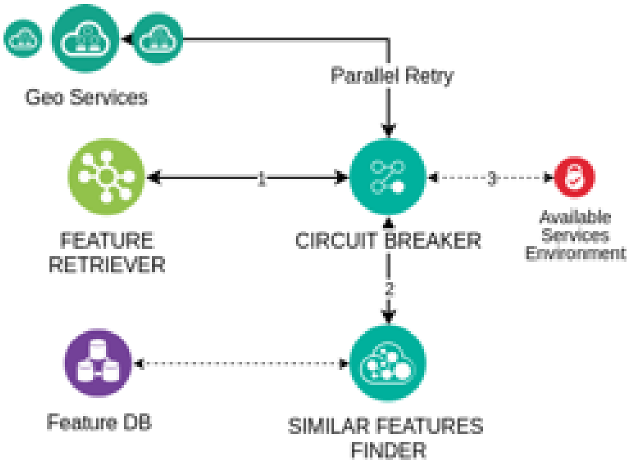


Figure 4. Circuit Breaker process.

In our work, the *Circuit Breaker* aims to keep the flow of spatial data constant for the user. Figure 4 depicts the flow of an unavailable service case. In step one, the *Feature Retriever* alerts the *Circuit Breaker* that a particular service has triggered an exception when trying to access a particular feature. In step 2, the *Circuit Breaker* asks the *Similar Feature Finder* to search for available services that have features similar to the one the user requested (the aim is to keep the user with useful information, even during the unavailability of the service that was requested). After this process, the service with the most similar and available feature is selected. In step 3, the *Circuit Breaker* inserts the available service and the feature of interest in the *Available Services Environment* associated with the service and feature of the original request. Then, alternative data is returned to the user.

If, in step 3, the *Similar Feature Finder* does not find any alternative data, the feature is labeled labeled as unavailable in the *Available Services Environment*. Then a response with a error is returned to the user.

After these steps, the *Circuit Breaker* registers in the *API Gateway* notification service, to be alerted when an user request for the unavailable service, is made again. When this next request is invoked, the circuit switches to Half-Open and a request attempt is made in parallel, without affecting the operation of the rest of the request, if the request is successful, the mention of the faulty service is removed from the *Available Services Environment*, and the circuit returns to Closed, otherwise it remains Open.

3.5. The Similar Features Finder

In our architecture, the *Similar Feature Finder* microservice is in charge of finding features that can replace a feature that became temporarily unavailable for some reason. During this process, it compares this feature to all the features provided by the infrastructure and returns the ones that have a similarity score higher than a predefined threshold. To perform this task, the service extracts three information about the unavailable feature type: the spatial extent, which is identified using its bounding-box, the temporal extent, in cases where temporal expressions can be found in its description, and theme, which is identified using its title.

An important characteristic that hindered the implementation of this microservice is that the catalog service provided by SDIs does not provide information at the level of the feature type. Hence, to overcome this limitation, we had to implement a module that extracts information from the catalog service. This process is performed in four stages. Firstly, it collects all the metadata records registered in the SDI's catalog service. In the second stage, it processes each one of these records and identifies the URL of the OGC web services (WMS and WFS) from which the data can be downloaded. Then, it accesses each one of these services to get information about the feature types they offer. Finally, a subset of the metadata describing the service and its respective feature types are stored in a local database, which is used as the source for finding similar features.

In order to find similar features, we implemented a search engine that uses a set of similarity metrics to estimate the similarity between the feature that is unavailable and each feature type stored in our database. The overall similarity between two feature types is based on three ranking values: spatial, temporal, and thematic. The spatial and temporal rankings are calculated using the approach proposed by Andrade et al. [Andrade et al. 2014]. To accomplish thematic ranking, we generated a document for each feature type containing information such as name, title, description, keywords, and some metadata about the service from which it is offered. These documents are indexed and retrieved using Apache Solr, which is a tool that provides scalable document retrieval. Whenever this tool executes a query, it returns a ranking value for each retrieved document. Then, we consider these values as the thematic ranking of the features related to these documents.

After all the ranking values are calculated, the features that got zero as the result for any of the rankings are discarded. For the remaining features, the similarity score is calculated using the average of the ranking values obtained for each dimension. Finally, the features with a similarity score higher than the threshold are selected, sorted, and returned by the microservice.

4. Case Study

To validate our approach, we implemented a case study based on the Brazilian National Spatial Data Infrastructure (INDE) [BRASIL 2008]. In this section, we demonstrate the application behavior during the open and closed states of the *Circuit Breaker*.

As shown in the top left section of Figure 5, the user makes a request to the application to get a feature of a specific service through its identifiers. In the case analyzed, the user requests a specific feature related to Public Health Equipment in Brazil.

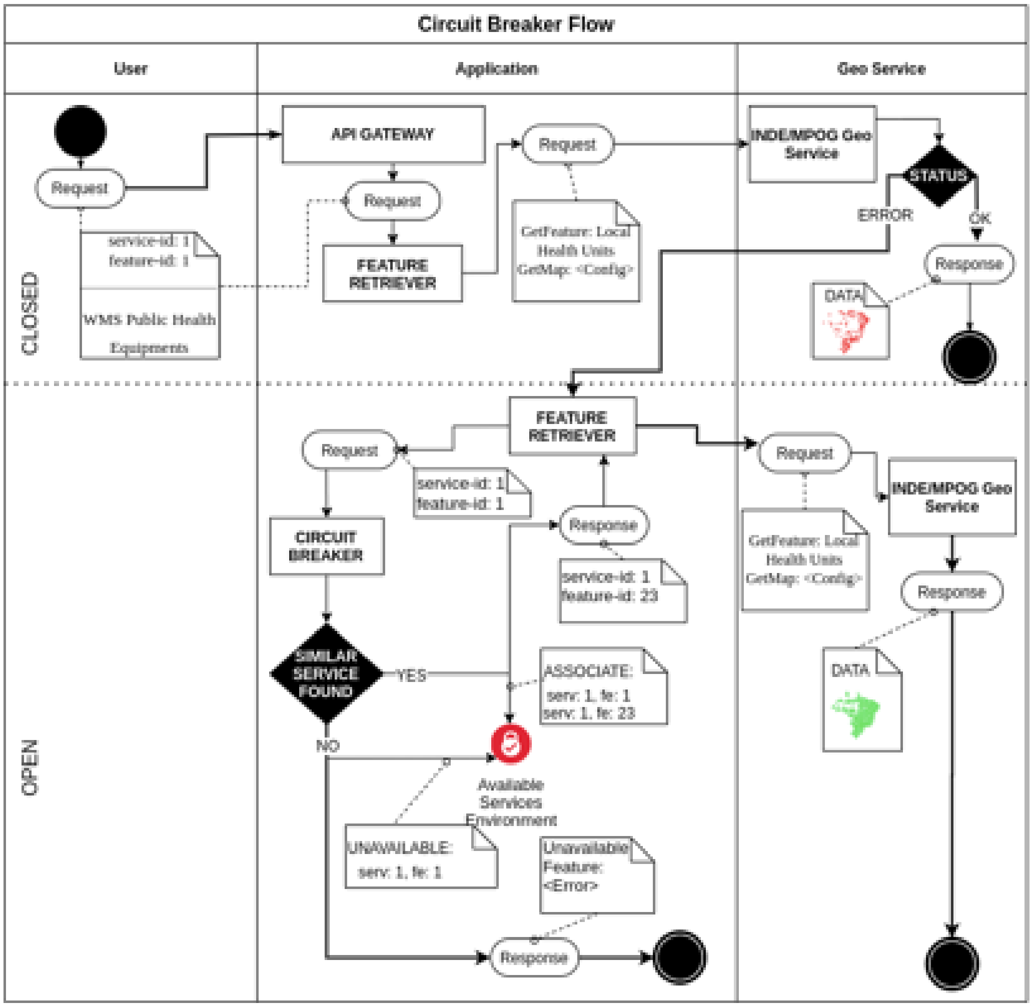


Figure 5. Request flow in our proposed fallback method.

The *API Gateway* collects the request and the filters are performed. Then, the request is forwarded to the microservice for features retrieval. The *Feature Retriever* retrieves the INDE's MPOG (Portuguese Acronym for Ministry of Planning, Budget and Management) geo-service data access, and the feature with public health equipment in Brazil. A request attempt is made, and if the response is obtained successfully, it is returned to the user.

If the *Feature Retriever* catches an error, it changes the *Circuit Breaker* state to Open. The *Feature Retriever Service* asks for a similar feature concerning Public Health Equipment in Brazil. In this case, the *Circuit Breaker* finds an available feature from the same MPOG geo-service. The founded feature has less equipment data, focusing in First Aid places, but it works well as an alternative to an error that can be exposed to the user or cause more cascading errors. The alternative feature is returned to the user. Meanwhile, the *Circuit Breaker* inserts the alternative feature into the *Available Services Environment*. Next attempts to access this feature will be redirected to the alternative feature.

If the *Circuit Breaker* does not find similar features, as shown in the lower center of the Figure 5, the feature is labeled in the *Available Services Environment* as unavailable. When any requested resource is labeled this way, the resilient SDI architecture returns an error response instantly. This reduces the response latency in the next attempts of this feature.

5. Conclusion and Future Work

This paper proposed a microservice based architecture that is able to to deal with the unavailability of geospatial data in SDI, maintaining the availability of applications that depend on this data. The proposed solution implements resilience patterns, providing a layer of reliability for users and applications that need high data availability. To make alternative data available, it relies on a microservice that evaluates the similarities between features.

As future work, we plan to implement a solution for detecting the semantic relationships between features using Natural Language Processing, including Named Entity Recognition. Such extensions would make our architecture able to perform deeper analysis of semantic relationship between features types, leading to better results in finding alternative data for fallback. We also intend to implement a new microservice for user administration to deal with user's settings for resources consumption. Moreover, we plan to develop a messaging service to provide notifications about unavailability of features consumed, as well as the feature substitutions performed by the system.

6. Acknowledgements

The authors would like to thank the Brazilian National Research Council (CNPq) for partially funding this research.

References

Andrade, F. G., de Souza Baptista, C., and Davis, C. A. (2014). Improving geographic information retrieval in spatial data infrastructures. *GeoInformatica*, 18(4):793–818.

Assis, L. F., Ferreira, K. R., Vinhas, L., Maurano, L., Almeida, C., Carvalho, A., Rodrigues, J., Maciel, A., and Camargo, C. (2019). Terrabrasilis: A spatial data analytics infrastructure for large-scale thematic mapping. *ISPRS International Journal of Geo-Information*, 8(11):513.

BRASIL (2008). Decreto no 6.666, de 27 de novembro 2008. `http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/decreto/d6666.htm`. Accessed on 2020-11-04.

Docker (2020). Official docker documentation. `https://docs.docker.com`. Accessed on 2020-08-27.

Friis-Christensen, A., Bernard, L., Kanellopoulos, I., Nogueras-Iso, J., Peedell, S., Schade, S., and Thorne, C. (2006). Building service oriented applications on top of a spatial data infrastructure–a forest fire assessment example. In *9th AGILE International Conference—Shaping the Future of Geographic Information Science in Europe*, pages 19–127.

Grus, Ł., Castelein, W., Crompvoets, J., Overduin, T., van Loenen, B., van Groenestijn, A., Rajabifard, A., and Bregt, A. K. (2011). An assessment view to evaluate whether spatial data infrastructures meet their goals. *Computers, Environment and Urban Systems*, 35(3):217–229.

Krafzig, D., Banke, K., and Slama, D. (2005). *Enterprise SOA: service-oriented architecture best practices*. Prentice Hall Professional.

Krämer, M. (2018). *A microservice architecture for the processing of large geospatial data in the Cloud*. PhD thesis, Technische Universität.

Li, Y. (2019). Research and application of micro-services framework for public information of waterway. In *2019 International Conference on Modeling, Simulation and Big Data Analysis (MSBDA 2019)*. Atlantis Press.

Mena, M., Corral, A., Iribarne, L., and Criado, J. (2019). A progressive web application based on microservices combining geospatial data and the internet of things. *IEEE Access*, 7:104577–104590.

Montesi, F. and Weber, J. (2016). Circuit breakers, discovery, and api gateways in microservices. *arXiv preprint arXiv:1609.05830*.

Newman, S. (2015). *Building microservices: designing fine-grained systems*. " O'Reilly Media, Inc.".

Nygard, M. T. (2018). *Release it!: design and deploy production-ready software*. Pragmatic Bookshelf.

OGC (2020). Web map service standard. `https://www.ogc.org/standards/wms`. Accessed on 2020-09-01.

OSS, N. (2020). Netflix open source software. `https://netflix.github.io/`. Accessed on 2020-08-28.

Schäffer, B., Baranski, B., and Foerster, T. (2010). Towards spatial data infrastructures in the clouds. In *Geospatial thinking*, pages 399–418. Springer.

Scholten, M., Klamma, R., and Kiehle, C. (2006). Evaluating performance in spatial data infrastructures for geoprocessing. *IEEE Internet Computing*, 10(5):34–41.

Soldani, J., Tamburri, D. A., and Van Den Heuvel, W.-J. (2018). The pains and gains of microservices: A systematic grey literature review. *Journal of Systems and Software*, 146:215–232.

An Efficient Solution to Generate Meta-features for Classification with Remote Sensing Time Series

Roberto U. Paiva[1,2], **Savio S. T. Oliveira**[1], **Luiz M. L. Pascoal**[1,2],
Leandro L. Parente[2], **Wellington S. Martins** [1]

[1]Institute of Informatics - Federal University of Goias (UFG)

[2]Image Processing and Geoprocessing Laboratory (LAPIG), UFG

urzedabr@ufg.br, wellington@inf.ufg.br,
{savioteles,luizmlpascoal,leal.parente}@gmail.com

Abstract. *Over the last years, the volume of Earth observation (EO) data increased significantly due to the large number of satellites orbiting the planet. These data are being used by automatic classification approaches to generate land-use and land-cover (LULC) products for different landscapes around the world. Dynamic Time Warping (DTW) is a classical method used to measure the similarity between two time series. In this context, DTW-based algorithms are an efficient approach to handle EO time series. These algorithms can be used to generate meta-features (i.e., new features automatically derived from the original features) to improve the performance of classification models. However, these algorithms have a long processing time and depends on large computational resources, making it difficult to use in large data volumes. Seeking to address this limitation, this work presents a full scalable parallel solution to optimize the construction of remote sensing meta-features. Additionally, a new classification strategy is presented, in which, the meta-features generated were used to train and evaluate a Random Forest model. Our results shows that both approaches leads to improvement in execution time and overall accuracy when compared to traditional methods.*

1. Introduction

The land-use and land-cover (LULC) presents several change dynamics that can be monitored through the analyzes of remote sensing time series [Foody 2002]. Due to the large number of satellites orbiting the planet, in general, these monitoring initiatives are using a huge volume of Earth observation (EO) data to produce mapping products for large areas of earth's surface, seeking to assist decisions related to food security, environmental conservation, sustainability, greenhouse gases emissions and deforestation [Nepstad et al. 2014, Bonan 2008, Bala et al. 2007, Dewan and Yamaguchi 2009].

The Dynamic Time Warping (DTW) [Sakoe and Chiba 1978], is a classic computer science algorithm introduced in the 1970s. It makes use of dynamic programming to measure the similarity between two time series. In the remote sensing context, some DTW-based algorithms were developed to map LULC changes consistently across the years [Guan et al. 2016, Romani et al. 2010, Maus 2016]. Among these algorithms, the Time-Weighted Dynamic Time Warping (TWDTW) [Maus et al. 2016] stands out as a method sensitive to seasonal climatic changes in the natural and cultivated vegetation. The TWDTW method is currently used jointly with the k-Nearest

Neighborhood (k-NN) algorithm, to generate meta-features for LULC classification [Dadi 2019, Oliveira et al. 2019, Manabe et al. 2018].

Although the TWDTW is effective in analyzing time series, its face some problems that impact on their performance, which makes it difficult do apply to large volumes of data. In addition, in some regions, the integration between k-NN and TWDTW presents low accuracy for LULC classifications [Dadi 2019]. More recent works are exploiting parallel processing to generate meta-features based in the TWDTW, seeking to obtain better performance when dealing with large datasets. These works also perform the classification using more sophisticated machine learning algorithms (e.g. Random Forest) to obtain improvements in the classification's accuracy [Oliveira et al. 2018, Oliveira et al. 2019, Paiva et al. 2020].

A parallel version of the TWDTW algorithm, called SP-TWDTW, obtained a speedup of 246 times in relation to the original version of TWDTW in R language and 11 times compared to the sequential version in C ++ [Oliveira et al. 2018]. However, this version presents a low thread usage and scalability limitations. SP-TWDTW is restricted to the size of the time series, which negatively impacts its scalability. Considering the technological advances for new space sensors and the emergence of satellites with high temporal resolution (e.g. PlanetScope), this limitation may undermine the usage of the SP-TWDTW algorithm in the future.

This work uses efficient parallelization strategies, using GPU/CUDA architecture, to propose a new DTW-based algorithm called Rapid-DTW, that computes part of the input data in windows of changeable size. The Rapid-DTW algorithm allows the workload of each thread to be variable according to the input data, thus reducing the idleness of each thread, decreasing the cost of synchronization, and removing the limitation regarding the size of the data input. The experiments carried out showed that the Rapid-DTW obtained better results in terms of performance when compared to the SP-TWDTW. Thus, it was possible to compare large sets of time series with various LULC patterns, in a short time, producing measures of similarity (meta-features). In addition, the result of a Random Forest model classification is presented using the meta-features generated by the Rapid-DTW. The Random Forest model showed accuracy improvements in relation to the classic k-NN learning algorithm used with SP-TWDTW in [Oliveira et al. 2019].

The remaining of this paper is organized as follows. Section 2 presents a brief discussion of remote sensing time series and related work. Section 3 discusses the classification of LULC using meta-features. Section 4 presents the proposed strategy, the Rapid-DTW, for computing the dynamic programming matrix in parallel. Section 5 presents the results of the tests and experiments carried out. Finally, in section 6 we present the conclusions and future work.

2. Time Series Processing: TWDTW and SP-TWDTW

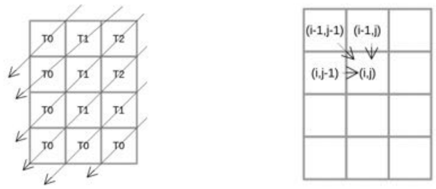
The TWDTW is an extension of the DTW algorithm, and it was designed to work with remote sensing time series. The TWDTW presents a logistical function capable of dealing with different seasonality in data, for example, the seasonality during a crop cultivation. This function is responsible for creating a penalty score in similarities between time series and recognized patterns that are displaced in time. The TWDTW computes two matrices: a matrix of weights Ψ and a dynamic programming matrix, called D matrix.

The weight matrix Ψ is computed through a logistic function based on the difference (in days) between the identified patterns and their time series. From the result of the weight matrix Ψ, the algorithm calculates the D matrix, using a recursive sum of the minimum dynamics, according to Equation 1. Then the algorithm uses the matrix D to find the path with the lowest cost, thus generating the measure of similarity between the pattern and the time series.

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

With the increase in the volume of remote sensing time series data, the computational demand for TWDTW has also increased as its original version was designed to work sequentially. This makes it difficult to analyze large areas, given the greater volume of data to be processed [Oliveira et al. 2018].

Due to the high computational cost to run the TWDTW algorithm, a parallel solution (SP-TWDTW) was proposed in [Oliveira et al. 2018]. This solution is based on the traditional strategy of computing elements in diagonals in parallel wavefront (Figure 1(a)). Each diagonal is processed in parallel, given the dependency on previous elements. In this matrix, the computation of each (i, j) element depends on the $(i - 1, j)$, $(i, j - 1)$ and $(i - 1, j - 1)$ elements previously calculated, as illustrated in Figure 1(b).



(a) Diagonal strategy.

(b) Data dependence.

Figure 1. SP-TWDTW [Oliveira et al. 2018]

SP-TWDTW target a highly multi-threaded GPU, and calculates one element per thread at each step when computing the D matrix. To ensure correctness, the number of threads in each block is determined as the size of the main diagonal, which is equal to the minimum value of the size of the pattern and the size of the time series. Since this operation is very simple and presents a very low workload for each thread (i.e., each thread computes only one element of the matrix per step of the algorithm), it causes a large amount of processing idleness during its execution, thus decreasing performance. Also, the GPU/CUDA programming architecture is currently limited to 1024 threads for each block [NVIDIA 2018]. Given this limitation, the SP-TWDTW is not able to work for values of similarity measures between patterns and time series when both are greater than 1024.

3. Classification Based on Meta-features

The use of machine learning algorithms for classification of LULC is essential to create models that enable automated and recurrent mapping. The DTW-based algorithms can generates similarity measures that can be used as meta-features to carry out classification of LULC. These meta-features are agnostic to the classification model and have been used with some well known classification algorithms.

In most studies, these meta-features are traditionally used as input to the k-NN [Belgiu and Csillik 2018] algorithm. However, some studies have reported that the algorithm presents low accuracy when applied in regions with higher data variability within each class [Dadi 2019]. Recent approaches presents the use of meta-features in more sophisticated machine learning algorithms, such as Random Forest and Support Vector Machines (SVM), and have obtained satisfactory results in terms of improving accuracy [Oliveira et al. 2019, Paiva et al. 2020]. In recent years, the Random Forest [Breiman 2001] and SVM [Vapnik 1995] algorithms have become a reference for good remote sensing classifiers [Rodriguez-Galiano et al. 2012, Belgiu and Drăguţ 2016].

In the field of remote sensing, the Random Forest has become the most used algorithm for classification of LULC, as it presents a simple configuration and obtains good accuracy [Pal 2005, Gislason et al. 2006, Belgiu and Drăguţ 2016]. According to [Pal 2005] Random Forest's popularity occurs due to its ability to achieve an accuracy similar to SVM, along side the ease of usage, with few parameters to be configured by the user and trivial adaptation to remote sensing data. Some recent works have used Random Forest for mapping large areas, showing its effectiveness when working with a large volume of data [Ayala-Izurieta et al. 2017, Parente and Ferreira 2018, Tsai et al. 2018].

4. Rapid-DTW

The Rapid-DTW, a new DTW-based algorithm, computes the elements of the dynamic programming matrix using windows of changeable size. The idea is to choose the workload performed by the threads at each step of the algorithm, allowing the method to experiment with different window sizes to obtain an ideal configuration for a specific instance of the problem. This decreases the idleness of threads and the cost of synchronization, which improves the use of GPU resources, enabling better performance of the algorithm.

The Rapid-DTW changes the flow in which the D matrix is computed. Rather than performing the computation of the elements in a diagonal flow, as the SP-TWDTW does [Oliveira et al. 2018], the window strategy performs the computation in elements within a given window. This results in the computation flow to be carried out vertically, allowing the dependence of data internally in each window in a sequential manner while guaranteeing the dependence of data between the windows of elements in parallel. Figure 2 illustrates the computation flow given the new strategy.

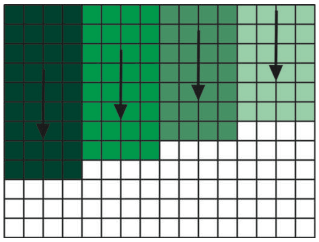


Figure 2. Computation flow of the D matrix with Rapid-DTW.

It is easy to notice that, the larger the window is set, the greater is the workload of each thread, which implies fewer idle threads in each step of the algorithm. As illustrated in Figure 3, in each window the thread runs sequentially, ensuring data dependency, while windows with the same color can be processed in parallel. The window size variability allows adaptation of the number of threads to be aligned to the size of the input problem. This variation enables to calculate similarity measures between patterns and time series for any given size of data entry.

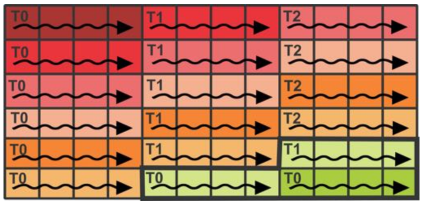


Figure 3. Behavior of threads inside the window.

Due to data dependency inherent to the dynamic programming algorithm, the number of synchronizations to maintain the correctness of the algorithm is very high. Therefore, the Rapid-DTW also works with smaller number of threads, thus significantly reducing the number of synchronizations.

The strategy for computing the programming matrix D is described in Algorithm 1. In lines 1-4, constant variables are defined. In line 2, the window size is calculated according to the number of threads. Two auxiliary variables, $auxj$ in line 7 and aux in line 23, are used to locate the elements of each window, while the $base$ variable, declared in line 6, is responsible for keeping each thread within the specified window size.

The for loop in lines 5 to 18 computes the upper part of the matrix, and similarly the for loop in lines 21 to 35 computes the lower part. The conditionals in lines 8 and 25 control the number of threads working in parallel, while lines 17 and 34 perform the synchronization. The inner for loops in lines 10 to 15 and 27 to 32 are used to locate where the elements of the windows are computed in parallel. Finally, an auxiliary function, $update\_element(i, j)$, is used to update each element according to the presented Equation 1.

5. Experiments and Results

In this section, we will detail the conditions for carrying out the experiments, as well as the results obtained. In this paper we defined two experiments. The first experiment was designed to evaluate the performance of Rapid-DTW when compared to SP-TWDTW and TWDTW in C++. In the second experiment, we applied the generated meta-features to train and evaluate a Random Forest model for LULC classification. All experiments were performed on a computer with an Intel Core i7-9700 processor (3.2 GHz and 8 MB Cache), 16GB DDR4 RAM, and NVIDIA GeForce GTX 1660 Ti video card with 6 GB GDDR6 of memory with Turing architecture, 1536 CUDA cores, 1770 MHz of frequency.

In the first experiment, the implementations for the Rapid-DTW, SP-TWDTW, and TWDTW algorithms were executed 10 times in each scenario and the average computed. Moreover, significance tests (paired t-tests) were run to test for significant differences, with 95% confidence. The TWDTW and SP-TWDTW codes were obtained directly from the work published in [Oliveira et al. 2018]. The three algorithms are used to generate the meta-features, therefore, this experiment was designed to assess the impact in terms of execution time when different number of observations patterns and time series are applied. Thus, in order to compare the execution time of the algorithms and carry out the experiments with time series of different sizes, input data was generated synthetically from the MODIS13Q1 database presented in [Maus et al. 2016].

```
Algoritmo 1: Rapid-DTW - Computation of $D$ matrix

Data: $\Psi$ Weight matrix
$y$: number of lines
$x$: number of columns
$num\_threads$: number of threads
Result: $D$ Matrix

$tid \leftarrow$ thread id
$windowSize \leftarrow x/num\_threads$
$tidWindow \leftarrow tid * windowSize$
$tidWindowaux \leftarrow tid * (windowSize - 1)$
for $(si = 0; si < y; si++)$ do
    $base \leftarrow tidWindow + (si - tid) * x$
    $auxj \leftarrow tidWindowaux$
    if $(tid \leq min(si, x - 1))$ then
        All threads in paralalel do:
        for $(index = base; index < base + windowSize; index++)$ do
            $i \leftarrow si - tid$
            $j \leftarrow tid + auxj$
            $update\_element(i, j)$
            $auxj \leftarrow auxj + 1$
        end
    end
    $sync\_barrier$
end
$si \leftarrow (y - 1 - tid) * x$
$auxj \leftarrow 0$
for $sj \leftarrow ((x/windowSize) - 2; sj \geq 0; sj - -)$ do
    $base \leftarrow tidWindow + si + windowSize + auxj$
    $aux \leftarrow 0$
    $auxj \leftarrow auxj + windowSize$
    if $(tid \leq min(sj, y - 1))$ then
        All threads in paralalel do:
        for $(index = base; index < base + windowSize; index++)$ do
            $i \leftarrow y - tid - 1$
            $j \leftarrow x - (windowSize * sj) - windowSize + aux + tidWindow$
            $update\_element(i, j)$
            $aux \leftarrow aux + 1$
        end
    end
    $sync\_barrier$
end
```

Initially, tests were performed to evaluate the performance of the Rapid-DTW on datasets with sizes commonly used in the literature for classification using the MODIS13Q1. In this test, 50 patterns and 1000 time series were generated, with 24 observations for the patterns and 50 observations for the time series. Then, to carry out the experiment on larger time series, with the objective to demonstrate our method's ability to process time series for the next generation of satellites, another set of 50 patterns, and 1000 time series with sizes ranging from 48 to 768 observations for patterns and 100 to 1600 observations for the time series was generated. A last data set was created to test patterns and time series simulating spatial sensors with really high temporal resolution. In this set, 5 patterns and 10 time series were generated, with sizes ranging from 1536 to 12288 observations for patterns and 3200 to 24800 observations for the time series. The SP-TWDTW was not used in this last dataset, due to its scalability limitation.

In order to generate the meta-features, the database presented in [Picoli et al. 2018] was used. This database presents MODIS13Q1 data extracted from the region of Mato Grosso - Brazil, to classify nine different classes of LULC. The samples are distributed as follows: Cerrado (400), Cotton-fallow (34), Forest (138), Pasture (370), Soy-corn (398), Soy-cotton (399), Soy-fallow (88), Soy-millet (235) and Soy-sunflower (53), totaling 2115 samples with 23 observations of NIR, MIR, EVI and NVDVI bands each sample. Te meta-features were generated for each point within the samples using the Rapid-DTW. The meta-features present the distances for each class (i.e., nine values for each point) bearing in mind that each class represents a pattern.

The Random Forest model was trained using 500 trees, as there was no improvement in the results with the increase in the number of trees. To estimate accuracy, all experiments were performed using the K-Fold cross-validation technique, with K = 5. Finally, a matrix of confusion was generated with the producer accuracy (PA), the user accuracy (UA), and overall accuracy (OA) from the results obtained. The Kappa coefficient was also calculated in order to analyze a baseline classification result.

5.1. Performance Analysis of Rapid-DTW

Using patterns of size 24 and time series of size 45 all multiple values of the main diagonal, which in this case has size 24, were tested. Figure 4 presents the results of this experiment, in which the Rapid-DTW presented the lower response time overall, during the computation of the D matrix, specifically 2.4 times faster than SP-TWDTW and 5.3 times faster than TWDTW. The best results were obtained when a 2 elements window size is applied, which is the smallest multiple of 24. This occurs because the number of threads per block in this input set is less than 32, which is favored since the CUDA architecture makes use of warps, with minimum sets of 32 threads that share instructions and memory. Therefore, for this scenario, reducing drastically the number of threads ends up harming more than the gain obtained with the decrease of idleness and synchronization cost, causing a negative result in performance.

The next test was designed to present the execution time of the three aforementioned algorithms, performing all the necessary steps to calculate the similarity measure. The results are illustrated in Figure 5(a). Next, Figure 5(b) presents a new set of tests with pattern and time series sizes over 1024, in which the SP-TWDTW was unable to perform due to limitation of threads per block in CUDA architecture. We omit confidence intervals in both graphs for the sake of clarity.

Overall, the performance of Rapid-DTW is better when the sizes of patterns and time series increase. With the results obtained in the experiments, it was possible to compare Rapid-DTW versus SP-TWDTW, in which the best performance was obtained

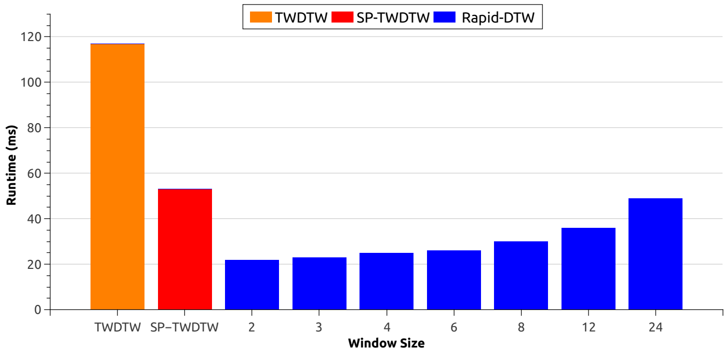
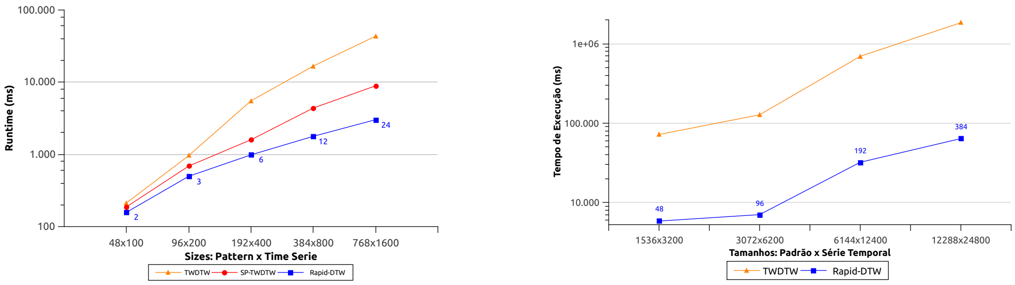


Figure 4. Execution time of the D matrix by TWDTW, SP-TWDTW and Rapid-DTW with patterns size 24 and time series size 50.

using the largest data set ($768x1600$), with an improvement in runtime of 2.9 times faster. Compared to the sequential version, Rapid-DTW achieved a speedup of up to 28.8 times on the largest data set ($12288x24800$). This improvement is due to the computation time of the dynamic programming matrix, which takes significantly longer than the rest of the steps, totaling 87% of the runtime.



(a) TWDTW x P-TWDTW x Rapid-DTW with window size ranging from 2 to 24.

(b) TWDTW x Rapid-DTW with window sizes ranging from 48 to 384.

Figure 5. Runtime of all steps of the algorithms. The blue numbers correspond to the window size used for the input set. Axis y in logarithmic scale.

5.2. Meta-features classification

The second experiment aims to evaluate the classification accuracy using the meta-features generated by the Rapid-DTW. In this scenario, the meta-features were applied as input to the Random Forest algorithm. Each point in the sample presents 9 meta-features according to the classes to be classified. The Table 1 shows an example of the algorithm entry for three pixels.

Pixel	**Cerrado**	**Fallow_Cotton**	**Forest**	**Pasture**	**Soy_Corn**	**Soy_Cotton**	**Soy_Fallow**	**Soy_Millet**	**Soy_Sunflower**
1	3,53	4,03	4,00	2,67	3,62	4,45	4,34	3,29	4,07
2	2,62	4,27	2,91	2,22	3,70	4,83	4,59	3,33	4,19
3	1,70	4,08	3,34	1,72	3,31	4,71	4,15	2,70	3,79

Table 1. Example of the meta-features generated by Rapid-DTW as an input vector for Random Forest.

Table 2 presents the confusion matrix with the results of this experiment. The Random Forest model showed an overall accuracy of 84.02% using the 9 meta-features (Kappa 0.81). The same experiment on the same data set was carried out in the work of [Oliveira et al. 2019], in which was proposed a combination of SP-TWDTW and k-NN that obtained an overall accuracy of 78%. Random Forest was able to obtain a better result, in terms of accuracy, of 6.2% in relation to k-NN, with no addition of new features.

	1	2	3	4	5	6	7	8	9	**UA(%)**
1 Cerrado	**375**	0	7	16	0	0	0	0	0	94,25%
2 Fallow_Cotton	0	**13**	0	0	0	5	0	0	0	85,29%
3 Forest	11	0	**126**	4	0	0	0	0	0	89,13%
4 Pasture	14	0	5	**343**	2	0	0	7	0	92,43%
5 Soy_Corn	0	2	0	1	**326**	48	0	44	35	67,34%
6 Soy_Cotton	0	17	0	1	22	**333**	0	6	2	87,97%
7 Soy_Fallow	0	1	0	0	0	1	**81**	7	0	89,77%
8 Soy_Millet	0	1	0	5	46	12	7	**171**	7	66,81%
9 Soy_Sunflower	0	0	0	0	2	0	0	0	**9**	96,23%
PA(%)	93,75%	38,24%	91,30%	92,70%	81,91%	83,46%	92,05%	72,77%	16,98%	**OA = 84,02%**

Table 2. Confusing matrix of the application of meta-features to Random Forest.

Still in Table 2, it can be seen that according to the producer's accuracy the Random Forest obtained good results (PA over 90%) with Cerrado, Forest, Pasture and Soy_Fallow classes. The agriculture-related classes present similar behavior in their time series, thus hampering the classifier in differentiating them and consequently obtaining good accuracy results [Picoli et al. 2018, Paiva et al. 2020].

Finally, given the similarities between the agriculture classes, a new experiment was conducted by merging the classes Fallow_Cotton, Soy_Corn, Soy_Cotton, Soy_Fallow, Soy_Sunflower into a single class called Agriculture. In this experiment a better result is observed using only 9 meta-features, in which the classifier managed to obtain an OA of 96.50% (Kappa 0.94). Regarding the new class of Agriculture, a PA of 99.59% was obtained. In scenarios where there is no need to typify the classification of agriculture, the use of this simplification is encouraged given its significant results. Table 3 shows the confusion matrix for this experiment.

	1	2	3	4	**UA (%)**
1 Cerrado	**375**	7	16	0	94,25%
2 Forest	11	**127**	3	0	89,86%
3 Pasture	14	4	**337**	5	93,78%
4 Agriculture	0	0	14	**1202**	98,84%
PA(%)	93,75%	92,03%	91,08%	99,59%	**OA = 96,50%**

Table 3. Confusion matrix of the application of meta-features simplifying the agriculture classes.

6. Conclusions and Future Work

Given the large number of satellites being constantly launched into Earth's orbit and their increasingly powerful sensors, it is expected that the sizes and quantities of remote sensing time series continue to increase in the near future. However, the computational cost to process this volume of data should also increase proportionally to the size and quantity of time series. The future of Remote Sensing may depends on the exploitation of high performance computing for the efficient processing of this huge volume of data [Houborg and McCabe 2018, Hansen and Loveland 2012, Plaza 2008, Camara et al. 2016]

In this work, the Rapid-DTW algorithm was presented, exploiting parallel processing for dealing with remote sensing data and applying the generated meta-features into the classification of land use and cover, demonstrating the potential of high performance computing techniques in the area of remote sensing. The conducted experiments shows that the Rapid-DTW presents significant improvement on runtime over traditional methods regarding the generation of meta-features.

Additionally, this work also proposed the application of the generated meta-features into the Random Forest, a state-of-the-art classifier. The experiments reported an improvement of 6.2% in overall accuracy when compared to the traditional k-NN. The results also encouraged a new experiment in which the agriculture related classes were merged into a single class, which lead to a significant overall accuraccy of 96.5% using the nine generated meta-features.

The use of Rapid-DTW in conjunction with the Random Forest algorithm allowed the analysis of a classification methodology capable of generating good accuracy results with few characteristics. Rapid-DTW is the first step towards creating an application capable of generating meta-features for various data from satellite time series. Once meta-features are generated for a given set of data, they can be used in various classification analyzes. These meta-features can also be used in several other classification methodologies, being incorporated into the input vectors of the machine learning algorithms.

However, there are some limitations to this work. This methodology does not deal directly with the satellite images. Therefore, all inputted data must be pre-processed in order to create a table with the time series data and their patterns. In future work, we propose the application of our method in different regions in order to verify its performance in runtime and accuracy. For that, we intend to create a method to directly process the satellite images and input its results to Rapid-DTW into a more complete methodology.

References

Ayala-Izurieta, J. E., Márquez, C. O., García, V. J., Recalde-Moreno, C. G., Rodríguez-Llerena, M. V., and Damián-Carrión, D. A. (2017). Land cover classification in an ecuadorian mountain geosystem using a random forest classifier, spectral vegetation indices, and ancillary geographic data. *Geosciences*, 7(2):34.

Bala, G., Caldeira, K., Wickett, M., Phillips, T., Lobell, D., Delire, C., and Mirin, A. (2007). Combined climate and carbon-cycle effects of large-scale deforestation. *Proceedings of the National Academy of Sciences*, 104(16):6550–6555.

Belgiu, M. and Csillik, O. (2018). Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis. *Remote sensing of environment*, 204:509–523.

Belgiu, M. and Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:24–31.

Bonan, G. B. (2008). Forests and climate change: forcings, feedbacks, and the climate benefits of forests. *science*, 320(5882):1444–1449.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Camara, G., Assis, L. F., Ribeiro, G., Ferreira, K. R., Llapa, E., and Vinhas, L. (2016). Big earth observation data analytics. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data - BigSpatial 16*. ACM Press.

Dadi, M. M. (2019). Assessing the transferability of random forest and time-weighted dynamic time warping for agriculture mapping. Master's thesis, University of Twente, Enschede.

Dewan, A. M. and Yamaguchi, Y. (2009). Land use and land cover change in greater dhaka, bangladesh: Using remote sensing to promote sustainable urbanization. *Applied geography*, 29(3):390–401.

Foody, G. M. (2002). Status of land cover classification accuracy assessment. *Remote Sensing of Environment*, 80(1):185–201.

Gislason, P. O., Benediktsson, J. A., and Sveinsson, J. R. (2006). Random forests for land cover classification. *Pattern Recognition Letters*, 27(4):294–300.

Guan, X., Huang, C., Liu, G., Meng, X., and Liu, Q. (2016). Mapping rice cropping systems in vietnam using an ndvi-based time-series similarity measurement based on dtw distance. *Remote Sensing*, 8(1):19.

Hansen, M. C. and Loveland, T. R. (2012). A review of large area monitoring of land cover change using landsat data. *Remote Sensing of Environment*, 122:66–74.

Houborg, R. and McCabe, M. F. (2018). A cubesat enabled spatio-temporal enhancement method (CESTEM) utilizing planet, landsat and MODIS data. *Remote Sensing of Environment*, 209:211–226.

Manabe, V. D., Melo, M. R., and Rocha, J. V. (2018). Framework for mapping integrated crop-livestock systems in mato grosso, brazil. *Remote Sensing*, 10(9):1322.

Maus, V. (2016). *Land Use and Land Cover Monitoring Using Remote Sensing Image Time Series*. PhD thesis, PhD thesis, Instituto Nacional de Pesquisas Espaciais, Sao José dos Campos.

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.

Nepstad, D., McGrath, D., Stickler, C., Alencar, A., Azevedo, A., Swette, B., Bezerra, T., DiGiano, M., Shimada, J., da Motta, R. S., et al. (2014). Slowing amazon deforestation through public policy and interventions in beef and soy supply chains. *science*, 344(6188):1118–1123.

NVIDIA, C. (2018). Cuda c programming guide, version 9.1. *NVIDIA Corp.*

Oliveira, S. S., Cardoso, M. d. C., Bueno, E., Rodrigues, V. J., and Martins, W. S. (2019). Exploiting parallelism to generate meta-features for land use and land cover classification with remote sensing time series. *Brazilian Symposium on Geoinformatics (GeoInfo)*, pages 135–146.

Oliveira, S. S., Pascoal, L. M., Ferreira, L., Cardoso, M. d. C., Bueno, E., Rodrigues, V. J., and Martins, W. S. (2018). Sp-twdtw: A new parallel algorithm for spatio-temporal analysis of remote sensing images. *Brazilian Symposium on Geoinformatics (GeoInfo)*, pages 46–57.

Paiva, R., Oliveira, S., Martins, W., and Parente, L. (2020). Análise de metacaracterísticas para classificação de uso e cobertura do solo utilizando random forest. In *Anais do XI Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais*, pages 71–80. SBC.

Pal, M. (2005). Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222.

Parente, L. and Ferreira, L. (2018). Assessing the spatial and occupation dynamics of the brazilian pasturelands based on the automated classification of modis images from 2000 to 2016. *Remote Sensing*, 10(4):606.

Picoli, M. C. A., Camara, G., Sanches, I., Simões, R., Carvalho, A., Maciel, A., Coutinho, A., Esquerdo, J., Antunes, J., Begotti, R. A., et al. (2018). Big earth observation time series analysis for monitoring brazilian agriculture. *ISPRS journal of photogrammetry and remote sensing*, 145:328–339.

Plaza, A. (2008). *High performance computing in remote sensing*. Chapman & Hall/CRC, Boca Raton, FL.

Rodriguez-Galiano, V. F., Ghimire, B., Rogan, J., Chica-Olmo, M., and Rigol-Sanchez, J. P. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67:93–104.

Romani, L. A., Goncalves, R., Zullo, J., Traina, C., and Traina, A. J. (2010). New dtw-based method to similarity search in sugar cane regions represented by climate and remote sensing time series. In *2010 IEEE International Geoscience and Remote Sensing Symposium*, pages 355–358. IEEE.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49.

Tsai, Y. H., Stow, D., Chen, H. L., Lewison, R., An, L., and Shi, L. (2018). Mapping vegetation and land use types in fanjingshan national nature reserve using google earth engine. *Remote Sensing*, 10(6):927.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer New York.

A Meta-Learning Framework for Imputing Missing Values in Weather Time Series

Vinícius H. A. Alves[1], **Marconi A. Pereira**[1]

[1]Departamento de Tecnologias em Eng. Civil, Computação e Humanidades – DTECH
Universidade Federal de Sao João del-Rei - Campus Alto Paraopeba
MG 443, KM 7 – Ouro Branco – MG – Brazil

viniciushaalves97@gmail.com, marconi@ufsj.edu.br

Abstract. *This paper describes an application of a meta-learning framework based on bagged trees. The proposed tool is used to estimate missing weather values in time series. The framework combines 8 different models of bagged trees that were optimized by a meta-learning algorithm. One of those 8 models was trained using only the date and each one of the remaining seven was calibrated with one weather parameter (max. temperature, min. temperature, insolation, among others), in addition to the respective date. The results show improvements in accuracy of the predicted values, achieving values such as $R^2 = 0.94$.*

1. Introduction

Climatic forecasting is very relevant, for instance, in agriculture planning, energy generation, natural disaster alerts, among others. Thus, it is necessary to learn from the past, considering the historical information, what is possible through stored data. When it comes to the elaboration of a study, it is important to verify the availability of data. Thus, using a complete and reliable database it is possible to generate studies with fewer errors [Bayma and Pereira 2017, Bayma and Pereira 2018]. Inconsistencies and unsatisfactory volume of data generate a limited or even a false representation of the real picture [García et al. 2009]. Rates of 1-5% of missingness are considered manageable. However, dealing with rates of 5-15% of missing values requires advanced methods, and over 15% may lead to significant interpretation losses [Acuna and Rodriguez 2004].

Despite having a large reservoir of climate data in Brazil, relevant institutions, such as the data division of CPTEC/INPE, do not have continuous information for all regions of the country [Barbosa and Carvalho 2015]. There are some periods of time without registration, for different reasons, which can lead to the problems mentioned above.

Predictive modeling is used to develop models capable of predicting missing values with great accuracy, but that is not an easy task. Thus, it has motivated several researches in the area [Yang et al. 2007]. When it comes to modeling the behavior of weather parameters, it is noticeable that trying to build a model using only a single imputation approach (e.g. linear regression) becomes difficult and sometimes ineffective. Therefore, finding different processes that best describe the problem or even conceiving multiple ways of dealing with it becomes a more appropriate measure, bringing with it greater precision [Solomatine and Ostfeld 2008].

This paper presents as a contribution a proposal of a framework for imputing missing data, using meta-learning algorithms. The tool uses bagged trees as both base learners and meta-learner. The base learners suggest the values to be imputed in the gaps, and then it is used a meta-learner to combine the previously suggested values to generate the most suitable outputs to fill the data gaps. This proposal increases the accuracy of the outputs when compared with other related works.

The framework is applied in 10 databases, each one composed of weather time series. These databases hold weather information from cities located in regions with different climatic configurations, distributed throughout the Brazilian territory. This approach aims to bring robustness to the framework, showing that it can deal with climatic diversity.

This text is organized as follows. Section 2 presents the literature review. Section 3 describes the data acquisition and preprocessing analysis. The Section 4 presents the regression method and the meta-learning layers. Section 5 describes the proposed framework. Section 6 details the framework validation. Section 7 presents the results and their analysis. Section 8 presents the conclusions.

2. Literature Review

A lot of studies propose approaches to fill the missing data values in time series. The most recent ones, generally, apply some computational intelligence tool. The most relevant works will be presented below, which served as a basis for what was developed in this study.

[Olcese et al. 2015] presented a method that uses artificial neural networks (ANNs) to predict missing aerosol optical depth (AOD) values at an AERONET station. ANNs with different topologies were trained with historical AOD values at two stations and air mass trajectories passing through both of them, generating 18 different datasets that were individually used to train 56 ANNs. It was used the coefficient of determination R^2 to compare measured and calculated AOD values to choose the best ones to be used to calculate the missing values. The model created was capable of imputing missing values with the average relative error equals to 25% (with 45% of the values having a relative error of less than 10%) and R^2 between 0.67 and 0.86 for the Iberian Peninsula and Eastern US, respectively.

[Bayma and Pereira 2018] compared the effectiveness of four imputation methods, used to fill data gaps in the historical time series from databases of the Brazilian Institute of Meteorology (INMET)[1]. The used methods were linear regression, ANNs, support vector machines and regression bagged trees. To compare the performance of each model, a part of the data was artificially removed so that the imputation methods could identify the missing values. In order to emphasize the importance of data imputation, the study also performed prediction of future data, considering the bases with and without the imputed data. A total of 20 models were generated by combining the four regression models and five different inputs that represented one scenario without imputation and four scenarios that represent the imputation of each method. In addition, the k-folds cross-validation method was implemented for all machine learning techniques to perform

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

a statistical test. The study concludes that, when the database was filled with the imputed data, there was an improvement in the forecast of new climatic values. This improvement was more significant with the use of bagged trees, both for imputation and for forecasting future data.

Another relevant approach was proposed in [Assis 2019]. In this work, a framework based on meta-learning methods was presented to identify price trends for the stock market assets. The implemented tool was based on the WEKA[2] API through which 7 regressors were combined to predict values and trends: ANNs, support vector machines, decision trees, random forest, Bayesian networks, minimum sequential optimization and genetic programming. The results showed an accuracy with up to 57% and financial results with gains of up to 100% of the capital value initially invested. The proposed framework can be used both to identify future values as well as to perform imputation to past values.

Meta-learning is a relatively new methodology, but its application is becoming more recurrent. The present study applies the concept of meta-learning to improve regular learning algorithms in the imputation values task.

3. Data Acquisition and Preprocessing Analysis

3.1. Data Acquisition

The Brazilian Institute of Meteorology (INMET) has more than 400 meteorological stations spread across the country and provides hourly, daily and monthly data on its website, in addition to several other resources that go beyond the interests of this work. The data acquisition for each city studied was made through the INMET website. In this research, daily data from 10 different meteorological stations were used. The parameters used were: date, rainfall, maximum temperature, minimum temperature, insolation, evaporation rate, average relative humidity, average compensated temperature, and average wind speed time-series.

Table 1 shows a summary of the used time series data. The second and third columns present the start date and end date of each analyzed city. The last column presents the total number of days used from each database.

Cities	Start Date	End Date	Number of days
Barreiras	01/01/1961	31/12/2019	21548
Belo Horizonte	01/01/1961	31/12/2019	21548
Cruz Alta	01/01/1961	31/12/2019	21548
Cuiabá	01/01/1961	31/12/2019	21548
Curitiba	01/01/1961	31/12/2019	21548
Diamantino	01/01/1961	31/12/2019	21548
Ouricuri	01/10/1975	31/12/2019	16162
Rio Branco	01/06/1969	31/12/2019	18475
São Felix do Xingu	01/09/1972	31/12/2019	17287
São Paulo	01/01/1961	31/12/2019	21548

Table 1. Analyzed periods and total number of days.

[2] https://www.cs.waikato.ac.nz/ml/weka/

3.2. Data Preprocessing Analysis

Pearson product-moment correlation, “R”, and the p-value represent dimensionless measures of the covariance between two variables, which is a scale that ranges from –1 to +1 [Wackerly et al. 2014]. The closer to those limits the correlation value is, the stronger is the association between the variables compared (their linear dependence). It is 0 whether there is no correlation between them. Moreover, it is possible to evaluate that relationship through the p-value, which the closer to 0 the p-value is, the stronger is the correlation between the variables compared.

[Bayma and Pereira 2017] applied the Pearson correlation method [Pearson 1900] to analyze the relationship between date and maximum temperature. They found out that the p-value of the variables month and year are less than the significance level of 0.05, which means that they are strong correlated. Then, they created an approach that considers just the day, month and year in the imputation process, but the month and year have a greater relevance in models than the day.

Aiming to characterize the correlation among the weather parameters, the correlation coefficients test was performed on the 10 cities' databases. Table 2 shows the average of the results. The main diagonal is set to 1, since it means the correlation between the parameter with itself. The other cells represent the p-value among the variables identified in each row and column. The p-values that are less than 0.05, indicate that the couple of variables has a statistically significant correlation [Bolboaca and Jäntschi 2006].

	R	**MaT**	**MiT**	**I**	**ER**	**ACT**	**ARH**	**AWS**
R	1	0.00	0.09	0.00	0.00	0.02	0.00	0.17
MaT	0.00	1	0.00	0.00	0.00	0.00	0.00	0.09
MiT	0.09	0.00	1	0.05	0.03	0.00	0.00	0.10
I	0.00	0.00	0.05	1	0.00	0.00	0.00	0.09
ER	0.00	0.00	0.03	0.00	1	0.00	0.00	0.07
ACT	0.02	0.00	0.00	0.00	0.00	1	0.00	0.02
ARH	0.00	0.00	0.00	0.00	0.00	0.00	1	0.00
AWS	0.17	0.09	0.10	0.09	0.07	0.02	0.00	1

R - Rainfall – MaT - Maximum Temperature – MiT - Minimum Temperature –
I - Insolation – ER - Evaporation Rate – ACT - Average Compensated Temperature –
ARH - Average Relative Humidity – AWS - Average Wind Speed

Table 2. Average p-values of the 10 cities.

The Table 3 shows the distribution of the gaps, detailing the percentages by database parameters. In the first column, it is presented the cities studied in this work. The remaining columns indicate the percentage of records in which 0, 1, 2, 3, 4, 5, 6, 7 or 8 parameters are available. The second column (“0”) indicates the percentage of records in which there is a lack of values for the all 8 parameters of the database, i.e. there is not any weather value in the record. The third column (“1”) indicates the percentage of records where 1 of the 8 weather parameters is available, and so on. The last column indicates the percentage of complete records, that is, none of the eight parameters is missing.

This study presents a better performance in the cases in which there are from 1 to 7 weather parameters, since the framework seeks to take advantage of the existing parameters to infer the others, especially those that have a high correlation value with the existing parameters in the record.

Cities	**Quantity of weather parameters**								
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Barreiras	12.572%	0.009%	0.074%	1.615%	2.409%	12.711%	2.418%	21.362%	46.830%
Belo Horizonte	7.444%	0.005%	0.005%	0.023%	0.088%	0.469%	0.334%	3.021%	88.611%
Cruz Alta	12.256%	0.065%	0.028%	0.715%	5.694%	1.638%	1.063%	18.401%	60.140%
Cuiabá	3.007%	0.023%	0.107%	0.691%	1.703%	11.259%	11.880%	17.241%	54.089%
Curitiba	0.650%	0.005%	0.014%	0.975%	2.836%	0.464%	0.575%	8.869%	85.614%
Diamantino	8.275%	0.023%	0.037%	0.733%	2.497%	4.799%	20.763%	39.748%	23.125%
Ouricuri	22.658%	0.037%	0.012%	0.099%	0.526%	12.096%	2.128%	7.945%	54.498%
Rio Branco	0.698%	2.425%	0.081%	4.141%	0.774%	1.846%	6.490%	23.648%	59.897%
São Felix do Xingu	19.194%	2.493%	0.289%	0.978%	0.445%	6.207%	5.733%	17.181%	47.481%
São Paulo	1.402%	0.181%	0.065%	0.111%	0.051%	0.900%	0.367%	25.752%	71.171%

Table 3. Percentage between the days with missingness and the total number of days of each city's database.

3.3. The Construction of Train and Test Datasets

As presented in [Bayma and Pereira 2017, Bayma and Pereira 2018], the learning methods present a better performance using a window of 5 years of data from the time series. For instance, to fill a gap in a month, a maximum of the last 5 years of data should be used to train the learning methods. Therefore, different intervals of 5 years were selected to apply the framework. A limit of 5% of missing values was admitted to provide more intervals without much distortion of the real picture to the study.

The data collected by each one of the 10 selected meteorological stations was splitted into three datasets: learners training set, meta-learning training set and validation set. It was ensured that no data used in the training was also part of the validation amount. Around 40% of the data was used to train the base learners, 40% was used to train the meta-learner and 20% for validating the final outputs.

4. Theoretical Foundations

4.1. Machine Learning

[Bishop 2006] describes the machine learning algorithm as being the task to represent a database as belonging to a function $Y(\bar{x})$, where a vector of independent variables $\bar{x}$ is taken as input and generates the output Y as a function of $\bar{x}$. The function $Y(\bar{x})$ is determined during the training stage, also called the learning phase, which uses a part of the available data for the calculation. Once trained, new entries can have their set of images determined through this type of algorithm. This ability to determine correct outputs for unprecedented input values is called "generalization".

In the present work, the machine learning method used to correlate the weather parameters and to infer the missing values is the bagged trees [Witten et al. 2005]. This method is presented below with its respective configuration.

4.1.1. Regression Tree and Bagged Trees

According to [Witten et al. 2005], decision trees employ the "divide to conquer" approach. The name "tree" comes from the relationship of learning nodes with the branches and leaves of a real tree. Each node represents a test of the attributes for decision making. Typically, the test consists of comparing the attribute with a constant or a range of values. Each leaf node represents an average value among all the values of the training set to which the leaf applies to.

The difference between classification trees and regression trees refers to the content of their results. While the first one seeks to find classes among the data, the second one seeks numerical results according to the training set. In this work, the interest is in finding numerical values for weather parameters imputation, so the research turns to the regression trees. As the attributes are numeric, the test usually consists of determining whether a given value is less than or greater than a predefined constant, which generates each a binary division or whether this value is below, within or above a range, which generates a division into three nodes. The test is applied successively with different constants or intervals. Figure 1 represents a regression tree.

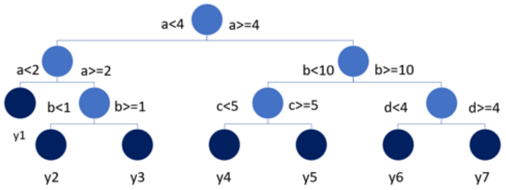


Figure 1. Scheme of a generic regression tree.

This work uses the concept of "Bootstrap Aggregating", sometimes known by the acronym "Bagging", so that the grouping of regression trees occurs, which tends to minimize the effects of overfitting [Witten et al. 2005]. The generated models had a maximum number of divisions of the branch node equals to 3 per tree, which characterizes trees that are not very deep.

4.2. Meta-Learning

The multi-classifiers may be described as a knowledge's combination of an ensemble of classifiers seeking for more accurate decisions [Kuncheva 2014]. Some multi-classifiers are: voting, ranking, mixture of experts and meta-classifiers. The last one is based on learning about the base classifiers to obtain a knowledge about which one may be the most efficiently applied [Brazdil et al. 2008]. In the context of this paper it was used regressors instead of classifiers, hence, they are given the term meta-learners.

[Kuncheva 2014] emphasizes that the meta-learning process implies in an increase in complexity. However, the authors still mention that combining an ensemble of base learners with less complex approaches becomes more straightforward than finding parameters' combination that best describes the problem's complexity.

5. The Meta-Learning Framework to Fill Missing Values

The proposed framework consists of 8 base learners (level 0), which suggest the values to be imputed in the gaps, and then a meta-learner (level 1) combines the previously suggested values to generate the most suitable outputs to fill the data gaps.

Figure 2 represents a scheme of the layers of the meta-learning, where the left hand side shows the N base learners in which is applied N different inputs, being one input for each learner, and the right hand side shows a meta-learner that receives as input the outputs of the previous learners generating the optimized output.

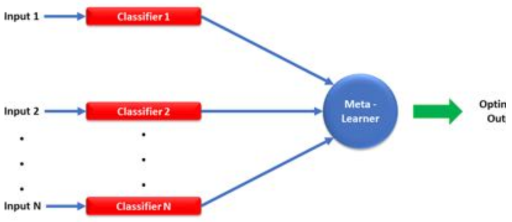


Figure 2. A scheme of the generated framework.

The framework is divided into two stages, where each one is represented by a block in the scheme in Figure 3. Those stages are: the learning stage, the meta-learning stage.

In the first stage, level 0, the base learners are trained using the base learners training set. The base learners are in charge of generating models capable of calculating the missing value from a given day based on the date and one of the weather parameters of the same day. Each model generated by each one of the 8 bagged trees must be fed with the inputs used in the learning stage. This ensemble allows that, for a given day, there are 8 different predictions available for the same missing parameter.

There are 8 base learners (level 0) that match each input. To generate those inputs, the framework removes the parameter that represents the one that is being imputed along the iteration, remaining 8 out of 9 types of inputs: (1) date, (2) date + rainfall, (3) date + maximum temperature, (4) date + minimum temperature, (5) date + insolation, (6) date + evaporation rate, (7) date + average relative humidity, (8) date + average compensated temperature and (9) date + average wind speed. Both inputs and the output represent records of the same day. In each iteration, a different weather parameter is imputed.

Subsequently, in the second stage, level 1, the meta-learner is trained using the meta-learner training set. That set of inputs are applied to the models generated in the learning stage to generate level 0 imputation. The level 0 outputs of the 8 models, in addition to the date of the day they refer to, become the inputs to feed the meta-learner. It seeks to learn from the group of base learners' knowledge to generate a model capable of combining those level 0 outputs to calculate an optimized one that is more accurate.

In the end, there are 9 trained models, being one of them in charge of giving the very best output. The trained ensemble is, then, validated.

6. Validation Methods

Aiming to measure the quality of the imputed data, the coefficient of determination R^2 was used to determine how well the models can reproduce the actual outputs [Homma and Saltelli 1996]. This coefficient compares the difference between the calculated value and the actual value, weighting the result with the difference between the average and the actual value. The closer to 1 the coefficient of determination R^2 is, the better the model calculates the dependent variable.

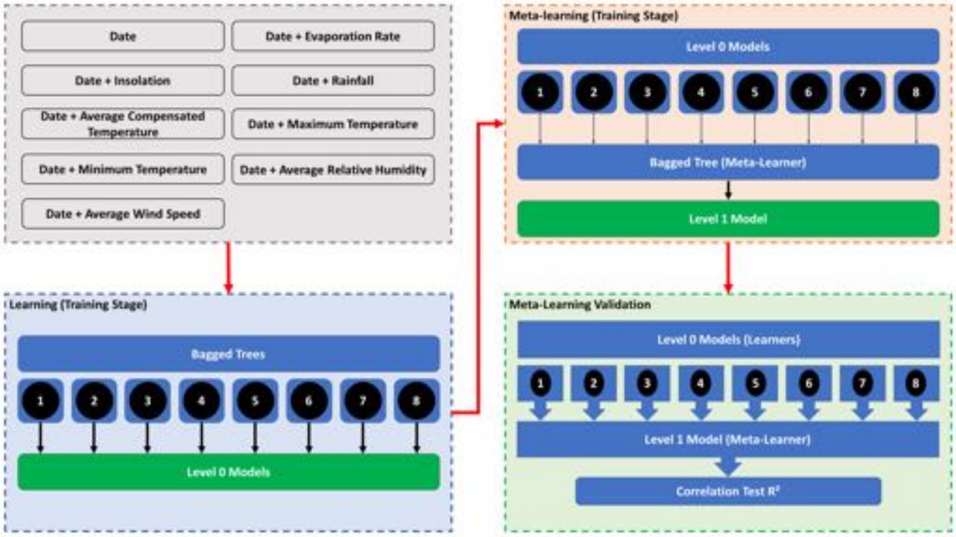

Date
Date + Evaporation Rate
Date + Insolation
Date + Rainfall
Date + Average Compensated Temperature
Date + Maximum Temperature
Date + Minimum Temperature
Date + Average Relative Humidity
Date + Average Wind Speed
Meta-learning (Training Stage)
Level 0 Models
1
2
3
4
5
6
7
8
Bagged Tree (Meta-Learner)
Level 1 Model
Learning (Training Stage)
Bagged Trees
1
2
3
4
5
6
7
8
Level 0 Models
Meta-Learning Validation
Level 0 Models (Learners)
1
2
3
4
5
6
7
8
Level 1 Model (Meta-Learner)
Correlation Test R²

Figure 3. Scheme of the framework.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N}(\overline{y} - y_i)^2} \tag{1}$$

where y_i is the i-th actual value, $\hat{y}_i$ is the i-th calculated value and $\overline{y}$ is the average of the N actual values.

To perform the validation test, artificial gaps were created in the dataset. In order to simulate the real scenario where the lack of data occurs randomly, a total of 20% of the database was randomly chosen to validate the trained models. The predicted outputs were compared to the actual values using the coefficient determination R^2 test.

To simulate different scenarios with different combinations of parameters missingness, artificial gaps in the inputs were created by removing some parameter in the inputs. It creates 8 different scenarios: no weather parameter available; one weather parameter available; two weather parameters available; three weather parameters available; four weather parameters available; five weather parameters available; six weather parameters available; seven weather parameters available. The gaps were replaced by a constant which is a value that is completely out of the bounds of all variable used in this study to simulate $-\infty$, as suggested by [Han et al. 2011]. It was chosen the constant -9999.

Working as a second validation method, the algorithm was applied in databases from cities with different climatic characteristics, being each couple of meteorological stations located in each one of the 5 Brazilian regions: north, northeast, midwest, south, southeast. For each database, the methodology adopted was performed 30 times to generate sufficient material to make statistic analysis.

7. Results And Analysis

Due to space restrictions, since there are plenty of results to be analyzed in this study, as the framework is applied to 10 cities using 8 different inputs, all the results presented below refer only to the Belo Horizonte station. The results from the other stations will be summarized and available as appendix at the following address: `https://ufsj.edu.br/marconi/geoinfo2020_-_paper_1.php`. As soon as this article is published, the source code will also be made available at that same address.

Figure 4 shows the coefficient of determination R^2 between the measured and the calculated for the base learner (BL) that considers fewer variables and the meta-learning in 8 distinct scenarios: (0) when there is not any weather parameter and only the date is used to generate all the outputs of the base learners; (1) when there is only one parameter available to increase the calculation; (2) when there are two parameters available to increase the calculation; (3) when there are three parameters available to increase the calculation; (4) when there are four parameters available to increase the calculation; (5) when there are five parameters available to increase the calculation; (6) when there are six parameters available to increase the calculation; (7) when there are seven parameters available to increase the calculation. Except for the average wind speed and rain fall, the average of the coefficient of determination of the parameters increases, demonstrating the effectiveness of considering more variables than only the date when trying to calculate the missing values. There are some parameters that present improvement of more than 30% when it is the only information that is missing, for instance, insolation (49%), average compensated temperature (36%), maximum temperature (53%) and average relative humidity (38%).

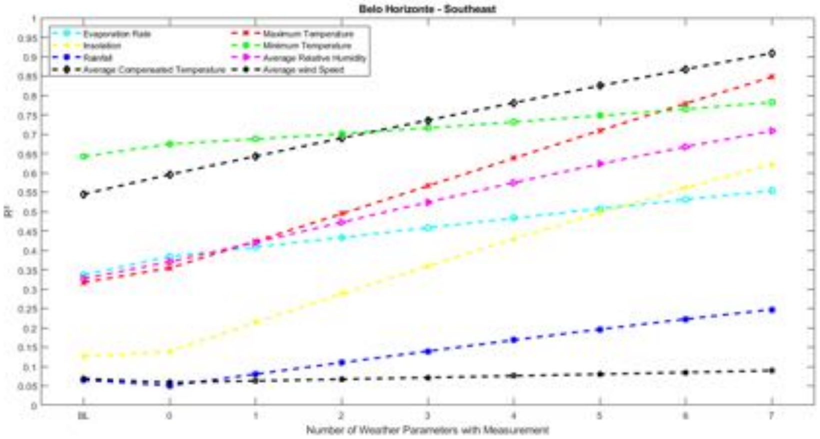


Figure 4. The coefficient of determination R^2 of the base learner that considers fewer variables and the meta-lerning applied in 8 different scenarios when imputing each weather parameter - Belo Horizonte station.

Figure 5 shows how the coefficients of determination of the base learner (BL) that only uses the date and the meta-learning in the different scenarios are distributed when calculating the average compensated temperature. Note that the boxplots with big areas (2, 3, 4 and 5) occurs because there is no differentiation among which weather parameters were available to generate the outputs, in other words, when parameters with lower correlation are used to estimate the missing data, it may lead to damages to the calculation and when the ones with high correlation are used, it may increase the results.

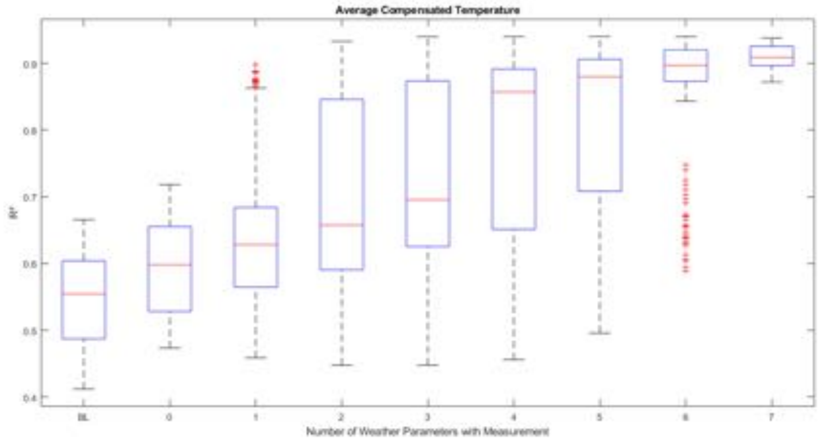


Figure 5. The coefficient of determination R^2 of average compensated temperature - Belo Horizonte station.

Figure 6 shows, in percentage, the comparison between the meta-learning and the base learner that only uses the date as input. It represents the probability that the result generated by the meta-learner is better than the approach that considers fewer variables, as presented by [Bayma and Pereira 2017, Bayma and Pereira 2018, Assis 2019]. It is possible to see that the meta-learning is affected by the different weather scenarios analyzed and the presence of parameters with low correlation. However, except for the average wind speed and rain fall, the meta-learning shows better results than the base learner with fewer inputs in at least seventy percent of the executions. In the best scenario, the meta-learning reaches better results 100% of the time, except for the average wind speed.

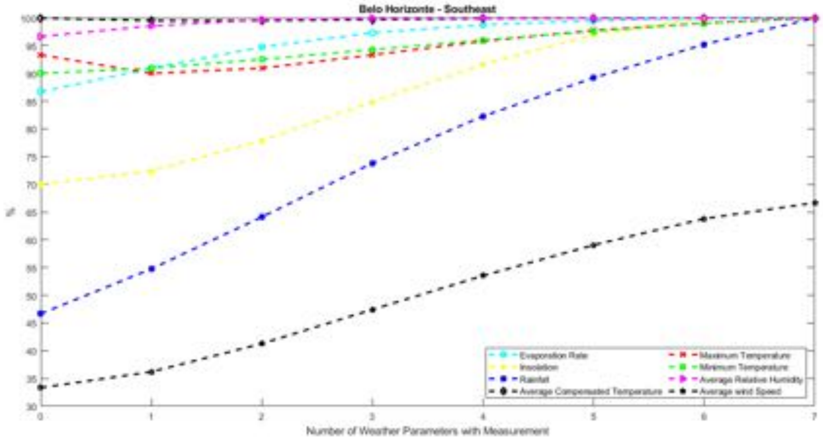


Figure 6. Comparison between the meta-learning (proposed approach) and learning process [Bayma and Pereira 2017]. The x axis present the number of weather parameters used in the meta-learning. The y axis presents the percentage of improvement of meta-learning compared to leaning process - Belo Horizonte station.

Through Figure 6 it is possible to conclude that, for example: except for average wind speed, whenever there are 7 weather parameters available, the meta-learner's output is better than the base learner's prediction; and except for average wind speed, when there are 3 weather parameters available, for more than 70% of the inputs the meta-learner's predictions are better than the base learner's predictions.

8. Conclusions

Computational resources, such as machine learning, plays an important role in modeling physical phenomena through less complex analysis that consider reduced numbers of variables that affects the system. Due to that, this resource can be used in meteorology to look for meteorological events models.

In this work, it is demonstrated through the analysis of coefficient of determination R^2, that meta-learning can increases the accuracy in imputing missing values in weather time series. Even though the meta-learner's output may not be better than the best level 0 model's output for any type of input, it diminishes or get rid of the possibility of choosing an inadequate single model.

It is noticeable that the more information is available, the better the results will be. Nevertheless, the results demonstrate that the meta-learner can recognize which parameters or which combinations of inputs can generate the most suitable values to fill the data gaps.

The low values of R^2 characteristic of rainfall and average wind speed may be due to their complex behavior and the low correlation with the other parameter (in the case of the average wind speed).

Despite the complexity of the climatic dynamics of the different regions impacts the recognition of patterns of different parameters, this approach takes advantage of the available information, which provides a better representation of the real picture of a given date and, consequently, of certain parameters. Moreover, the present paper modeled the complex weather parameters behaviors through less complex approaches, getting rid of the hard work of finding out the very best combination of independent variables to infer the dependent variable missing values.

Acknowledgments

The authors thank FAPEMIG for financial support and the INMET and database sector of CPTEC/INPE[3] for the availability of data and technical report. We are especially grateful to Carlos Alberto da Silva Assis (*in memoriam*) for all the tips and guidance.

References

Acuna, E. and Rodriguez, C. (2004). The treatment of missing values and its effect on classifier accuracy. In *Classification, clustering, and data mining applications*, pages 639–647. Springer.

Assis, C. A. S. (2019). *Predição de Tendências em Séries Financeiras Utilizando Meta-Classificadores*. PhD thesis, Centro Federal de Educação Tecnológica de Minas Gerais.

[3] http://www.inpe.br/

Barbosa, M. and Carvalho, M. (2015). Sistemas de armazenamento de dados observados do cptec/inpe. *Instituto Nacional de Pesquisas Espaciais*.

Bayma, L. O. and Pereira, M. A. (2018). Identifying finest machine learning algorithm for climate data imputation in the state of minas gerais, brazil. *Journal of Information and Data Management*, 9(3):259–259.

Bayma, L. O. and Pereira, M. d. A. (2017). Comparison of machine learning techniques for the estimation of climate missing mata in the state of minas gerais, brazil. In *GEOINFO*, pages 283–294.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Bolboaca, S.-D. and Jäntschi, L. (2006). Pearson versus spearman, kendall's tau correlation analysis on structure-activity relationships of biologic active compounds. *Leonardo Journal of Sciences*, 5(9):179–200.

Brazdil, P., Carrier, C. G., Soares, C., and Vilalta, R. (2008). *Metalearning: Applications to data mining*. Springer Science & Business Media.

García, S., Fernández, A., Luengo, J., and Herrera, F. (2009). A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. *Soft Computing*, 13(10):959.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Homma, T. and Saltelli, A. (1996). Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety*, 52(1):1–17.

Kuncheva, L. I. (2014). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

Olcese, L. E., Palancar, G. G., and Toselli, B. M. (2015). A method to estimate missing aeronet aod values based on artificial neural networks. *Atmospheric Environment*, 113:140–150.

Pearson, K. (1900). X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175.

Solomatine, D. P. and Ostfeld, A. (2008). Data-driven modelling: some past experiences and new approaches. *Journal of hydroinformatics*, 10(1):3–22.

Wackerly, D., Mendenhall, W., and Scheaffer, R. L. (2014). *Mathematical statistics with applications*. Cengage Learning.

Witten, I. H., Frank, E., and Hall, M. A. (2005). *Practical machine learning tools and techniques*. Elsevier.

Yang, Y., Lin, H., Guo, Z., and Jiang, J. (2007). A data mining approach for heavy rainfall forecasting based on satellite image sequence analysis. *Computers & geosciences*, 33(1):20–30.

Towards the Identification of Semantic Points in Trajectories of Moving Objects with Weighted Averages

Jarbas Nunes Vidal-Filho[1,2], Valéria Cesário Times[1], Jugurta Lisboa-Filho[3]

[1] Informatics Center (CIn), Federal University of Pernambuco (UFPE) – Recife – Brazil

[2] LAPIS – Federal Institute of Ceara (IFCE) - Tabuleiro do Norte – Brazil

[3] Informatics Department (DPI), Federal University of Viçosa (UFV) – Viçosa – Brazil

jarbas.vidal@ifce.edu.br, vct@cin.ufpe.br, jugurta@ufv.br

Abstract. *In this paper, we explore a technique of clustering GPS points to: (1) extract a single point from each candidate cluster for stop point, named semantic point for this research, and (2) analyze similarities of semantic points identified in trajectories using three different algorithms. We propose a new algorithm based on a weighted average for the identification of the semantic point in the cluster - that which is the simplest, most efficient, and possesses the least computational cost when compared to other state-of-the-art solutions. We identified 1050 semantic points in trajectories of the Geolife project and compared the distances between them from the semantic points. The algorithm proposed was compared to the central point and K-Medoid algorithms. From the results, we concluded that the semantic points are at an acceptable distance from one another as defined by the literature.*

1. Introduction

The discovery of stop points in raw trajectories of moving objects has been an important research topic for data mining [Lehmann et al. 2019; Fu et al. 2016]. Due to great heterogeneity, lack of accuracy, and differing sampling levels in the collection of trajectories, one encounters some uncertainty in the detection of stop points (that is to say, the place visited by the user) in raw trajectories [Lehmann et al. 2019; Furtado et al. 2018]. Uncertainty in the identification of stop points causes difficulties in the association of semantic information (i.e., information about the place visited) with semantic points, inference of activities performed by the object in movement, discovery of patterns, among others. In this work, we define the semantic point as a spatial-temporal point representing the physical stop point in the cluster (i.e., the clustering of points seen as a candidate for the stop point). The semantic point is identified after the formation of clusters candidate stop points in trajectories.

The mobile individual has their localization (longitude/latitude) registered over time, represented by a sequence of spatial-temporal points. This is also known as the raw trajectories of moving objects [Furtado et al. 2018]. From the raw trajectory, it is possible to extract diverse parameters such as, for example, velocity and direction of the moving object (i.e., semantic information). With the use of these parameters, it is possible to identify the similarity between semantic points by way of diverse techniques such as, for example, classification, clustering, and matching patterns. For [Furletti et al. 2013], even though trajectories are the representation of stop points and movements (i.e., the sequence of spatial-temporal points between stop points), it is important to identify stop points as the place visited by the individual.

Stop points are normally represented by semantic points or by a geographical region. The literature reports few works that approach the identification of semantic points in clusters, principally due to the uncertainties that exist in current approaches. Let us imagine that a determined method of semantic annotation uses the approach of associating the closest place to visit concerning the semantic point. For the same cluster, the location of a semantic point returned by different methods will hardly be the same, resulting in uncertainties in the definition of the place visited. Therefore, the uncertainties of current methods for returning semantic points can give occasion to flaws in the inference of the place visited, in the inference of activities performed by the moving object, amongst other things.

The main goal of this paper is to propose a new algorithm based on weighted average to infer semantic points. Besides this, the paper seeks to discuss, analyze, and compare identification methods for semantic points in raw trajectories. The motivation to investigate new approaches for identification of semantic points took place because of disadvantages in the traditional methods, highlighted by the literature - for example, the computational cost of iterations and the necessity for defining values of parameters of entry for the choice the semantic point. The approaches found in the literature initially require the definition of the number of clusters, the number of interactions, and the number of medoids (among other parameters) to realize the processing of algorithms. Besides this, the results depend on a good calibration of these parameters.

The approaches available identify the semantic point based on the central point method and the K-Medoid and K-Means algorithms. The problems with these approaches are: (1) the central point will always be a point which does not belong to the clustering; (2) the centroids returned by K-Means will hardly belong to the cluster; (3) these algorithms require entry parameters such as, for example, number of clusters and probable random semantic points; and (4) K-Medoid and K-Means have computational cost with iterations and return semantic points closer to the center. According to [Steinbach et al. 2005], the results of the K-Medoid and K-means algorithms depend on the calibration of initialization parameters. However, undue calibration causes more uncertainty in the definition of the semantic point.

We propose an algorithm based on a weighted average that seeks to infer the semantic points closest to the points of lowest velocity, has a lower computational cost, and always chooses a semantic point belonging to a candidate cluster the stop point. We compared the similarity based on the distance between the semantic points returned by the weighted average, central point and K-Medoid algorithms, in order to identify the proximity between the semantic points returned by these approaches. By way of this comparative analysis, we identify that the choice of the semantic point also affects the services of the inference of activities.

The rest of this paper is organized into four sections. Section 2 describes the works related to the problem of identification of stop points. Section 3 presents the contributions of the paper. Section 4 exhibits a comparative evaluation between the method proposed and the central point and K-Medoid methods. In Section 5, conclusions and possible future studies are brought up to discussion.

2. Theoretical Foundation

Before delineating the objectives of this paper, we present the most important basic concepts to understand this work and some works about the identification of stop points.

2.1. Basic concepts

The process of semantic enrichment has the finality of semantically annotating raw trajectories with information on the place visited or the activity that the user performs during movement [Fu et al. 2016]. The moving object can be represented by a single localization in the region that is a candidate for the stop point (i.e., semantic point), which would then be used to label the place visited by the user. Besides the definition of the process of semantic enrichment, the definitions of raw trajectory, stop point and clusters are important for understanding the work proposed in this paper, and as such, they are listed here.

Definition 1 (*Raw trajectory*): is a sequence $<p_1, p_2, ..., p_n>$ of points p = $((x, y), t)$, where $(\underline{x}, \underline{y})$ is the localization of the object and $\underline{t}$ is the moment of collection.

Definition 2 (*Stop point*): is the place visited by a user during time interval T = $(t_{(start),} t_{(end)})$, represented by a sequence of GPS points belonging to time interval T.

Definition 3 (*Clusters*): is a subset of the raw trajectory formed by points that possess similarity with each other based on a certain trajectory collection parameter.

2.2. Related work

K-Means is a clustering algorithm that receives a predefined number of clusters to be formed and randomly selects centroids iteratively to be grouped into clusters [Zhou et al. 2007]. The algorithm iterates by way of the centroid and the rest of the points in the cluster, calculating the distance between all the points, computing the centroid and attributing each point to the centroid of least distance. Finally, the centroid of each cluster is found by way of the average of all the instances associated with the cluster. The K-Medoid is similar to K-Means, however, instead of choosing the centroid that never corresponds to a true data point, it randomly selects medoids (i.e., points of clusters with the best computational cost) [Steinbach et al. 2005]. The K-Medoid receives, as parameters, the number of clusters to be formed, the number of medoids and the instances of the cluster to calculate the cost function. The problem of the K-Medoid is found in the definitions of values for the entry parameters which can generate less optimized medoids and computational cost with iterations. The K-Medoid has greater relevance for choosing a medoid point that belongs to the cluster and is more robust in the face of noise and outliers.

[Kang et al. 2005] defined a clustering approach based on time where the user stays stopped in one place. If the distance between the starting point and the finishing point of the cluster is less than a determined threshold, new clusters are formed. Smaller clusters, having less time in their formation, are discarded. Finally, if the centroid of the cluster is at a certain distance from an existing POI, this approach merges the centroid with the POI to represent the semantic point.

[Fu et al. 2016] proposed an approach to clustering based on two stages, considering the similarity of values for the parameters of time and distance. The algorithm initially groups points based on the time of stay. Following this, it verifies the distance between the points to reduce problems with the loss of signal. Finally, the algorithm does a scan to identify the points' peaks of density and extract stop points. [Zhou et al. 2017] presented a genetic clustering algorithm based on density and K-Means variation that does not need to inform the number of clusters. Finally, this variation in the K-Means uses quality indicators to reduce the size of the clusters generated.

The related works focus on improvements in the existing clustering techniques, taking time, speed, direction, density and distance thresholds into consideration to minimize the problems with the collection of trajectories or loss of GPS signal. We perceive that few works focus on the identification of semantic points in clusters, and those that do so are based on centroids or medoids. Therefore, little is yet discussed about semantic points, being valid the investigation and implementation of new approaches that use semantic points to improve the performance of localization applications.

3. The Weighted Average Algorithm

GPS data represented by movements and stop points with semantic information are called semantic trajectories. Recently, the literature has been expanding upon the concept of semantic trajectories to multiple aspect trajectories. According to [Petry et al. 2019], this new type of trajectory has as its objective to receive semantic annotations from different sources and formats such as, for example, places visited by the user [Furletti et al. 2014] and data from sensors and social networks. The objective is semantically enriching the raw trajectory gathered in real-time. We identified that semantic and multiple aspects trajectories constantly use real-time resources from different Application Programming Interface (API). APIs supply information on places, health, climate, social networks, and other Web services. In this context, the quality of semantic addition to the stop point is related to the identification of the semantic point.

The algorithm proposed in this work has as its objective the contribution to the identification of semantic points in clusters using weighted average as a way of improving the inference of activities and semantic annotations of trajectories. Initially, to understand the proposal in a better manner, we formalize the concept of semantic point as the contribution of our study. Following this, we present the weighted average algorithm as the main contribution of this work.

Definition 4 (*Semantic point*): is a tuple ((*x*, *y*), *t*, *as*), where ($\underline{x}$, $\underline{y}$) are the geographical coordinates, $\underline{t}$ is the time register and $\underline{as}$ the semantic annotation, representing the localization of a single point within a candidate cluster for a stop point and its semantic information. The semantic point can be obtained from the algorithm proposed for this paper, Algorithm 1.

Algorithm 1 receives as its entry parameter a list of clusters that can be defined automatically by way of clustering algorithms and based on some kind of similarity criteria. From this point on, the central idea of the algorithm is to prioritize points belonging to the cluster that have low speed. Therefore, the algorithm will check the instantaneous speed of each point belonging to the cluster. We understand that, the user using a means of transport tends to reduce his speed during the stopping process, consequently, being able to arrive until reaching a complete halt - approximately, velocity null.

Algorithm 1. Identification of semantic point based on the weighted average

```
List<Stop> StopsDiscoveringByWeightedAverage (clusters)

Input: clusters: clusters list
Output: SemanticPoint
1: stops = new List <Stop>;
2: FOR EACH cluster IN clusters DO
//sort trackpoints by instantaneous velocity
3:              ordered = new List<TrackPoint>
4:              Collections.sort(ordered);
// weighted average calculation of the latitude and longitude of each cluster
5: FOR (i = 0, weight = ordered.size(); i < ordered.size(); i++, weight--) DO
6:               totalWeight += weight;
7:               totalLatitude += weight*ordered.get(i).getCoordinate().getLat();
8:               totalLongitude += weight*ordered.get(i).getCoordinate().getLng();
9:              totalTime += weight*ordered.get(i).getInstant().getTime();
10:              stopLatitude = totalLatitude/totalWeight;
11:              stopLongitude = totalLongitude/totalWeight;
// the value totalTime/totalWeight corresponds to the instant in millis
12:              instant = new Date((totalTime/totalWeight));
13:              stopCoordinate = new Coordinate (stopLatitude, stopLongitude);
// stop definition
14:              stop = new Stop (stopCoordinate, instant, cluster);
15:              stops.add(stop);
16:              cluster.setStop(stop);
17: return stops;
```

To give priority to the points of low velocity, we use the concept weighted arithmetic average to attribute weights to the points belonging to the clusters. In this way, the points of low velocity receive larger weightings while high-velocity points receive lower weightings. After receiving a list of clusters, the algorithm orders the points based on the velocity of each candidate cluster for a stop point (lines 2 – 4). Having done this, the algorithm calculates the weighted average of latitude, longitude, time, and instant of the points of each cluster (lines 5 – 12). The weights used are defined based on the number of points that exist in the cluster. For example, if the cluster possesses 60 points the weights attributed are numbered 0 to 59. In this way, points at low velocity will be given priority in the generation of the semantic point. Finally, the algorithm defines the localization of the semantic point as the coordinate of the stop point (line 13), instantiates and returns a new semantic point related to the cluster (lines 14 – 17). The localization of the returned semantic point is utilized as a parameter to access and seek API services, as well as semantic information, to enrichen trajectories.

4. Experiments and Evaluation

In this section, we present two types of similarity analysis between semantic points of trajectories. The first considers the similarity of the distance parameter between points and the second approaches the inference of activities between semantic points. For all algorithms used in the experimental analysis of this work, the CB-SMoT algorithm proposed by [Palma et al. 2008] was utilized for the formation of clusters, because it uses similarity based on time and speed to form clusters.

The simulations occurred from the catalog of services from SDI4Trajectory (www.sdi4trajectory.ifce.edu.br), and the parameters defined for CB-SMoT were *stop time = 300 s, Speed limit = 4 m/s,* and *Average speed = 3 m/s,* chosen for the definitions utilized by [Furletti et al. 2013]. The algorithms compared in this study are: weighted average proposed for this study, central point and K-Medoid [Steinbach et al. 2005]. The choice of algorithms is based on some reasons, such as: (1) The literature cites works using methods based on medoids and centroids [Zhou et al. 2017; Kang et al. 2005]; (2) The K-Medoid is frequently discussed in the literature because it selects semantic points belonging to the points of the clusters [Steinbach et al. 2005]; (3) The central point is a simple approach that does not select semantic points belonging to the cluster and always uses the center of the cluster to represent the semantic point. The definition of the number of clusters for K-Medoid is done after the execution of the CB-SMoT algorithm, not manually as that which frequently occurs.

4.1. Data Acquisition

To validate and verify the efficiency of the proposed algorithm, we used two sets of data. Geolife is a project proposed by Microsoft Research Asia that collected 17,621 raw trajectories by 178 users between 2008 and 2012. The data from this project are organized into folders, where each pasta represents a determined user. Some folders possess a text archive informing the period of collection of the trajectory and the transport method used. The data represent routine activities of users, for example, in the same folder there are trajectories of a user who used the train between 10 a.m. and 11 a.m., and then walked on foot between 11 a.m. and 11:30 a.m., on different days. These trajectories can correspond to the activity of commuting to work. Therefore, among the folders that have labels, we chose 15 to perform a comparative analysis of similarity based on the parameter of distance. Table 1 exhibits the distribution of data analyzed by means of transport, clusters and identified semantic points.

Table 1. Set of experimental data from the Geolife project

Geolife Dataset	Number of Trajectories	Number of Clusters	Number of Semantic Points
Car	55	95	95
Taxi	47	202	202
Subway	20	90	90
Train	33	200	200
Bike	201	345	345
Walking	53	118	118

We also analyzed the similarity between points based on the inference of activities. To do this, we used the second set of data collected by 7 volunteer users for 1 month. Of the 7 users, 5 were using a smartphone with the My Tracks application and 2 were using bicycles and carrying the TomTom watch for data collection. All users involved in the experiment were lawyers, teachers, and amateur cycling athletes. They collected the trajectories on foot, bicycle, and by car. Table 2 exhibits the distribution of the set of data gained by the volunteers.

Table 2. Set of experimental data collected by volunteers

Voluntary dataset	Number of Trajectories	Number of Users	Number of Clusters	Number of Semantic Points
Bike	2	2	2	2
Car	20	4	26	26
Walking	3	1	5	5

The quantitative clusters and semantic points presented in Tables 1 and 2 were the same for all the algorithms used in the analyses of similarity. In total, 409 trajectories of different users were analyzed and 1,050 semantic points for the set of Geolife data. For the voluntary data set, 33 semantic points were analyzed for the inference of activities. Even though being a limited set of data, the idea using the second set of data is to investigate if the inferences of activities that were returned can be related to the identification algorithm of the chosen semantic point and the means of transport used by the moving object. In this way, in exploring the manual semantic annotations and inferences of activities, it is possible to gain results that aid in deciding upon the methods of identification of semantic points to be used, based on the means of travel.

4.2. Analysis of similarity between semantic points based on the parameter of distance

According to [Furletti et al. 2013], a distance covered by a user between the stop point and the place visited can be flexible up to 500m. For a candidate cluster for a stop point being formed within the region of a shopping center, a semantic point can be identified in the parking lot or the establishment visited by the user. In the face of this, we consider that 500m is an acceptable distance to indicate similarity between semantic points. For [Smith and Butcher 2008], in the different types of traveled environments by a user, the distance between stop points can vary between 300m to 500m. In this section, we initially analyze the distance between semantic points returned by the weighted average, central point, and K-medoid algorithms.

Besides this, we assume that the K-Medoid algorithm should be used as the baseline in the analysis of similarity. This is because K-Medoid has already undergone many experiments and is well defined in the literature [Velmurugan et al. 2010; Arora et al. 2016], becoming the object of study through a diversity of works that explore clustering algorithms [Steinbach et al. 2005]. Therefore, the closer a semantic point identified by another method is to the semantic point returned by K-Medoid, it is considered that there is a similarity between the methods. Table 3 shows the number of semantic points - returned by weighted average and central point algorithms – that were closest to the semantic points returned by K-Medoid.

Table 3. Quantity of semantic points closest to K-Medoid

Method/Transport	CAR	TAXI	SUBWAY	TRAIN	BIKE	WALKING
Weighted average	43	82	44	72	148	56
Central point	51	117	46	111	182	60

We used the Google Earth tool to measure the distance between semantic points *in loco*. Of the 1050 semantic points analyzed from the Geolife data, we identified that 3.62% (38 semantic points) are practically in the same localization, so we assumed that they were at the same point. Of the other 96.38% (1012 semantic points), it was verified that the central point algorithm presented 54% (567 semantic points) closer to K-Medoid, while the weighted average algorithm presented only 42.38% (445 semantic points). The *in loco* analysis verified that the K-Medoid identifies medoids closer to the center of the clusters which can justify a greater quantity of semantic points identified by the central point. This is probably associated with its similarity to the K-Means, which can identify a medium point in the cluster.

Besides this, we explore in greater detail the distances between the semantic points returned by the three algorithms under study. Firstly, we compared the distance between the semantic points of the K-Medoid and the central point. The results can be seen in Table 4, which presents the number of semantic points by ranges of distance.

The idea is to be able to understand how these points by ranges of distance are distributed, seeing that the literature defines parameters of distance that an object can travel after a stop. For [Smith and Butcher 2008], climate and time also influence the activity of a mobile object. We consider the means of transport and the distance between semantic points as parameters that can influence the inference of activities, due to the variations in localizations of semantic points.

Table 4. Quantity of semantic points by ranges of distances (Central Point vs K-Medoid)

Distance/ Transport	CAR	TAXI	SUBWAY	TRAIN	BIKE	WALKING
0 – 10 m	10	72	29	84	117	30
10 – 50 m	37	65	27	55	115	34
50 – 100 m	15	31	15	31	52	32
100 – 500 m	23	31	17	24	59	19
above 500 m	6	3	2	6	2	3

We know that shopping centers possess various physical locations with distances that exceed 500m. Therefore, if the user went to the shopping center, whatever semantic point in the region of the shopping center is considered a hit in the process of identification of stop points. In this context, we believe that the distance between semantic points can vary up to 500m. Therefore, the closer they are, the more similar they can be considered. In Table 4 only 22 semantic points exist which are at a distance greater than 500m. Approximately 97.6% of the semantic points within an acceptable distance, defined as the distance between the stop point and the place visited by the user. The problem with the central point is the fact of not selecting the points of clusters. This causes uncertainties in the definition of activities performed at the stopping point of the moving object, due to the points tend to be distant from the points of interest.

In Table 5, we also compare distances between semantic points returned by the weighted average and K-Medoid algorithms. The idea is also to analyze how the semantic points are distributed by ranges of distance. Table 5 exhibits the quantitative view of semantic points by ranges of distance concerning the K-Medoid. One can verify that only 20 semantic points are at a distance superior to 500m. Therefore, 98.1% of the semantic points are at an acceptable distance that consists of a distance between the stop point and the place visited by the user. This means saying that the algorithms possess similarity concerning the definition of semantic points and considering the parameter of distance. The weighted average algorithm becomes important in the process of identification of stop points represented by semantic points. The algorithm prioritizes points of low velocity and that belongs to the candidate cluster for the stop point and possesses similarity to the K-Medoid.

Table 5. Quantity of semantic points by ranges of distances (Weighted Average vs K-Medoid)

Distance/ Transport	CAR	TAXI	SUBWAY	TRAIN	BIKE	WALKING
0 – 10 m	9	47	16	64	96	21
10 – 50 m	31	76	35	75	114	43
50 – 100 m	27	41	14	33	68	25
100 – 500 m	22	35	23	22	65	28
above 500 m	6	3	2	6	2	1

Finally, we consider the distances between the semantic points returned by the proposed algorithm and the central point, since Table 3 presented a greater quantity of semantic points extracted by the central point method and that are closer to K-Medoid. The medoids tend to be closer to the center of the clusters, principally due to similarity to K-Means. However, we use the weighted average algorithms and the central point to identify the similarity between semantic points based on the parameter of distance. The idea is to construct ranges of distance to investigate how the semantic points of Table 3 are distributed.

For example, based on Table 3 and considering the Train and Bike means of transport, the central point method presented, respectively, 39 and 34 semantic points more than the proposed algorithm and listed semantic points closer to K-Medoid. In Table 6 shows the number of semantic points by distance ranges for the proposed method and central point. However, for the Train as a mode of transport, only 5 semantic points are at a distance greater to 500m, while the Bike as a mode of transport did not present any semantic point with a distance superior to 500m. In Table 6 one perceives that approximately 99% of the semantic points returned by the weighted average algorithm are at a distance of up to 500m from the central point, in conformity with the acceptable distance between semantic points adopted by this work.

Table 6. Quantity of semantic points by ranges of distances (Weighted Average vs Central Point)

Distance/ Transport	CAR	TAXI	SUBWAY	TRAIN	BIKE	WALKING
0 – 10 m	21	60	17	89	103	34
10 – 50 m	36	90	41	67	177	44
50 – 100 m	18	30	8	24	44	27
100 – 500 m	18	20	22	15	21	13
above 500 m	2	2	2	5	0	0

From Tables 4, 5 and 6 we verify that more than half of the semantic points are at a distance of 0 – 50m between each other. Increasing the distance buffer between points, we identify that 99% of the semantic points analyzed are at a distance of up to 500m. [Yang et al. 2012; Smith and Butcher 2008] affirm that the distance covered on foot is related to activity, use of transport, recreation, and other things. The authors affirm that the acceptable measures of distance acceptable for a determined user who needs to stop and then walk to the place to be visited can vary from 300m to 1.5km, depending on the objective. [Millward et al. 2013] defined time and distance values for users who go on foot. Thus, we identified that 500m is an acceptable value for a user to go on foot.

The proposed algorithm possesses similarity when compared to the approaches defined in the literature when the parameter of the distance between semantic points is taken into consideration. The analyses based on ranges in distance help in the validation of similarity between methods. The proposal presented uses the concept of weighted average. It is based on the parameter of velocity to give priority to points in the cluster, is easy to implement, it always chooses low-velocity points from the cluster, and the semantic points that are returned tend to be closer to the places visited.

4.3. Analysis of similarity between semantic points based on the inference of activities

In this section, we use the data set of volunteers. The maximum distance covered after a stop was defined by basing ourselves on the work of [Yang et al. 2012; Smith and Butcher 2008]: 500m for users who are in movement by foot or by bicycle, and for those using a car, 400m. We used the K-Medoid algorithm as the baseline.

For the set of data utilized, each user manually noted down the set of places visited during the collection of the trajectory. Table 7 presents for each one of the methods discussed, the ID Track and the number of semantic points identified by trajectory, followed by the activity and the probability of the activity occurring. We instantiated the definition of the gravitational model utilized by [Furletti et al. 2013], which uses the concept of attraction of bodies. This model returns the probability of the occurrence of a determined activity associated with the stop point.

As discussed in the previous section, 96.38% of the semantic points analyzed from the Geolife project presented variations in localizations. Under these circumstances, the objective of investigating similarity based on the inference of activities is due to the algorithms discussed returning semantic points with different localizations, causing uncertainties in the inference of activities performed by the moving object. Therefore, we explore the similarity between semantic points based on the inference of activities, also seeking to identify the correlation between the methods of identification of semantic points and the means of transport used by the moving object. In a general way, for the model used by [Furletti et al. 2013], if a stop point is closer to a set with a greater number of places visited from the same category, the probability that the activity is the same in this set is greater.

Table 7. Comparing similarities between semantic points based on the inference of activities.

ID TRACK	Semantic Point	Weighted Average Activity (%)	Central Point Activity (%)	K-Medoid Activity (%)
1	1	SERVICES: 62.49	SERVICES: 61.18	SERVICES: 54.60
	2	OTHERS: 88.36	SHOPPING: 99.98	SERVICES: 77.49
2	1	SHOPPING: 40.19	SHOPPING: 28.02	SHOPPING: 39.32
3	1	FOOD: 18.23	FOOD: 25.75	FOOD: 40.00
	2	OTHERS: 26.28	OTHERS: 54.33	OTHERS: 52.08
4	1	LEISURE: 72.50	LEISURE: 72.12	LEISURE: 72.50
5	1	OTHERS: 86.82	OTHERS: 42.99	SHOPPING: 75.32
6	1	SERVICES: 18.54	SERVICES: 15.43	SERVICES: 20.08
	2	SERVICES: 31.67	OTHERS: 30.34	SERVICES: 32.84
	3	SERVICES: 70.50	SHOPPING: 43.35	SERVICES: 89.42
7	1	OTHERS: 62.79	OTHERS: 56.59	OTHERS: 57.44
8	1	OTHERS: 59.05	OTHERS: 58.62	OTHERS: 55.17
9	1	SERVICES: 49.58	OTHERS: 44.98	OTHERS: 30.97
10	1	OTHERS: 62.53	OTHERS: 68.35	OTHERS: 59.42
11	1	SERVICES: 49.43	SERVICES: 51.25	SERVICES: 49.61
	2	OTHERS: 40.57	OTHERS: 39.56	OTHERS: 40.27
12	1	OTHERS: 45.13	OTHERS: 45.13	OTHERS: 47.98

continues on the next page

13	1	OTHERS: 31.84	OTHERS: 41.9	OTHERS: 28.89
	2	SERVICES: 3.40	OTHERS: 10.02	OTHERS: 33.98
14	1	SERVICES: 58.53	SERVICES: 82.37	SERVICES: 41.19
15	1	SHOPPING: 52.44	SHOPPING: 55.86	SHOPPING: 47.76
16	1	SHOPPING: 42.13	SHOPPING: 64.73	SHOPPING: 64.03
17	1	SERVICES: 42.77	SERVICES: 28.04	SERVICES: 62.13
18	1	OTHERS": 73.77	OTHERS": 58.41	OTHERS": 95.75
19	1	SERVICES: 99.91	SERVICES: 49.42	SERVICES: 17.45
20	1	OTHERS: 58.08	OTHERS: 45.20	OTHERS: 45.20
21	1	SERVICES: 29.65	SERVICES: 23.10	SERVICES: 14.38
	2	FOOD": 26.38	FOOD": 28.26	FOOD": 28.35
22	1	SHOPPING: 40.60	SERVICES: 39.30	SHOPPING: 72.76
	2	FOOD: 49.25	FOOD: 48.76	FOOD: 49.22
23	1	OTHERS: 50.40	SERVICES: 70.34	OTHERS: 55.23
24	1	OTHERS: 27.47	SERVICES: 97.32	SERVICES: 64.17
25	1	SERVICES: 36.87	SERVICES: 24.36	SERVICES: 51.46

In Table 7, of the 25 trajectories analyzed, 33 semantic points were identified. These were compared to each other concerning inference and the probability of the activity occurring. From the semantic points analyzed, only for the trajectory of Id 12 was there inference of the same activity and the same probability for the central point and weighted average. Another 4 semantic points analyzed had different activities for the three methods. For example, the semantic point for Id 02 (Id 01 track) presented a different activity for each method discussed.

As a way to improve the inferences of our results, we consider the category of the place visited by the user and the manual semantic annotation. The objective is to know the activity executed by the user and then compare it to the return of the activity done in the methods. Of the 4 semantic points analyzed with different activities, the K-medoid algorithm hit the activity of a semantic point. Analyzing the 28 remaining semantic points, 64.29% of the points presented greater similarity for inference and probability of activities between the weighted average and K-Medoid, while only 35.71% presented better similarity between the central point and K-Medoid. This leads us to define that the weighted average algorithm presents itself as a viable solution for the context of identification of semantic points, for the second set of examined data.

5. Conclusions

The comparisons between the proposed algorithm and the central point and K-Medoid methods present similarity based on the parameter of distance for the semantic points analyzed. Approximately 85% of the semantic points are at an acceptable distance within 500m. One also perceives that the inference of activities for returned semantic points by the proposed algorithm has better precision since it possesses the biggest hits concerning K-Medoid and better correspondence with manual semantic annotation. Therefore, the weighted average algorithm becomes an interesting approach for the identification of semantic points in trajectories.

It is also visible that the inferences of activities are sensitive to variations in localization of semantic points. These often tend to localize themselves closer to a set of Points of Interest with an activity that is different from that noted down manually. The variations in the inference of activity bring to the fore the importance of identifying semantic points. We identify the better similarity between the proposed method and K-Medoid, mainly for when the car is transport. However, it requires better investigations.

For future works, it is planned to amplify the analysis of the similarity between semantic points based on the inference of activities, utilizing more voluminous data such as those returned by Geolife and exploring new modes of transport used.

References

Arora, P.; Varshney, S. (2016). Analysis of k-means and k-medoids algorithm for big data. *Procedia Computer Science*, 78, 507-512.

Fu, Z., et al. (2016). A two-step clustering approach to extract locations from individual GPS trajectory data. *ISPRS International Journal of Geo-Information*, *5*(10), 166.

Furletti, B., et al. (2013). Inferring human activities from GPS tracks. In *Proceedings of the 2nd ACM SIGKDD international workshop on urban computing* (pp. 1-8).

Furtado, A. S., et al. (2018). Unveiling movement uncertainty for robust trajectory similarity analysis. *Int. Journal of Geographical Information Science*, *32*(1),140-168.

Kang, J. H., et al. (2005). Extracting places from traces of locations. *ACM SIGMOBILE Mobile Computing and Communications Review*, *9*(3), 58-68.

Lehmann, A. L., et al. (2019). SMSM: a similarity measure for trajectory stops and moves. *Int. Journal of Geographical Information Science*, *33*(9), 1847-1872.

Millward, H.; Spinney, J.; Scott, D. (2013). Active-transport walking behavior: destinations, durations, distances. *Journal of Transport Geography*, *28*, 101-110.

Palma, A. T., et al. (2008). A clustering-based approach for discovering interesting places in trajectories. In *Proceedings of the ACM Symposium on Applied Computing*, pages 863-868.

Petry, L. M., et al. (2019). Towards semantic-aware multiple-aspect trajectory similarity measuring. *Transactions in GIS*, *23*(5), 960-975.

Smith, M. S., et al. (2008). How far should parkers have to walk?. *Parking*, *47*(4).

Steinbach, M.; Kumar, V.; Tan, P. (2005). Cluster analysis: basic concepts and algorithms. *Introduction to data mining, 1st ed. Pearson Addison Wesley*.

Velmurugan, T.; Santhanam, T. (2010). Computational complexity between K-means and K-medoids clustering algorithms for normal and uniform distributions of data points. *Journal of Computer Science*, 6(3), 363-368.

Yang, Y.; Diez-Roux, A. V. (2012). Walking distance by trip purpose and population subgroups. *American Journal of Preventive Medicine*, *43*(1), 11-19.

Zhou, C., et al. (2007). Discovering personally meaningful places: An interactive clustering approach. *ACM Transactions on Information Systems*, *25*(3), 12.

Zhou, X., et al. (2017). An automatic K-Means clustering algorithm of GPS data combining a novel niche genetic algorithm with noise and density. *ISPRS Int. Journal of Geo-Information*, *6*(12), 392.

Human Spatial Reasoning in Everyday Language: Inferring Regions that Describe Spatial Relations

Lucas Freitas[1], Claudio E. C. Campelo[1]

[1]Federal University of Campina Grande (UFCG)
Systems and Computing Department
Campina Grande – PB – Brazil

joselucas@copin.ufcg.edu.br, campelo@dsc.ufcg.edu.br

Abstract. *The study of natural spatial language used by humans can foster the development of geographically aware systems that are able to assist us in our daily lives. In this specific type of language, one of the most important features is the usage of spatial relations. Used to describe the position of an object in relation to another, spatial relations play a crucial role in the proper understanding of spatial communication. The computation of these relations in the application level is important to locate objects in the space or even interpret a location description provided by a human. There are many types of spatial relations such as metric and topological, and in fact implementations of these relations are even available in common spatial database query languages. However, these relations are not easily translated into the ones that people use in daily communication, for which there is still a lack of approaches to estimating regions described by them. In this paper, a set of algorithms that derive polygons that match spatial relations used in daily communication are proposed. The algorithms were evaluated by comparing their outputs with drawings produced by humans. The results indicate that although this is a very difficult task, the proposed procedures produce satisfactory spatial region extents that almost always intersect the ones drawn by the subjects.*

1. Introduction

One piece of crucial information often used by people when describing the location of an object in an environment are spatial relations. They describe how an object is located in a given scenario, in relation to another reference object. In the sentence *The place is near the school, right next to that big old church*, **near** and **next to** are the parts that play the role of spatial relations, as they describe the location of the place in relation to the school and the big old church respectively.

The emergence of models that are capable of interpreting the spatial relations is of utmost importance for the development of geographic aware applications. Even though methods for generating spatial relations do exist, they tend to be focused on a different set of relations than the ones that are present in the language used by people more often. To address this issue, the main goal of this paper is to propose a set of algorithms that project in a 2D map, some of the spatial relations that are most often used by humans in daily communication. They take as input the geometry of the object that serves as reference in the spatial relation and return the polygon that corresponds to the real-world region described by it.

An experiment was carried out to evaluate the precision of these algorithms, where volunteers were asked to read phrases containing references to landmarks and spatial relationships and then to draw on a map the polygon(s) they thought that best describes the region referred to in the text. Apart from validating the algorithms, the collected data was analyzed with the aim of answering some research questions on mental representations of spatial relations. The dataset of geometries drawn by participants is made publicly available, hoping that it can contribute to further research in the area.

The emergence of algorithms that project the relations that people use the most in daily dialogue, as well as the better understanding of human spatial reasoning, has the potential of enabling the development of many applications, such as geographically aware chatbots, different search interfaces to be used in map services or even improving the operation of driverless vehicles.

This paper is structured in the following manner. The next section explores previous works that can be related to this line of research. After that, Section 3 describes the methods of developing the research as well as the materials used. Section 4 presents the proposed algorithms. Using the data collected in the experiment, section 5 tries to answer a relevant question about the human interpretation of spatial relations. The experiment data is used to evaluate the precision of the algorithms in section 6. Lastly, section 7 presents some final thoughts on the research.

2. Related Work

Spatial Relations have been studied for decades and have been classified into topological, projective and metric [Bucher et al. 2012]. While metric relations are important, humans seem to have a qualitative reasoning of space [Cohn and Renz 2008]. Topological relations describe the positioning of objects in terms of the intersections of their interiors, boundaries and exteriors. They have been extensively studied [Egenhofer and Franzosa 1991, Mark and Egenhofer 1994, Clementini et al. 1994], and in fact are even supported by spatial query languages. Most of the relations explored in this work fall in the directional category.

Directional relations are a common subcategory of projective relations that include daily expressions used in natural language such as “right of”, “in front of” and “between”. Directional relations are ambiguous and need additional contextual information such as Frames of Reference [Clementini 2013]. In his work, Clementini defines a taxonomy of frames of reference, mapping relations to the 5-Intersection model of projective relations [Clementini and Billen 2006], this gives the additional geometric definitions needed to compute relations. [Clementini and Bellizzi 2019] build on top of this mapping and present a Java application framework that implements the directional relations given the assumption that the relations are being interpreted in a few of the frames of reference.

In the present work, a different approach is presented in computing directional relations. A set of algorithms that generate regions that correspond to the relations by computing intersections between buffers around landmarks and nearby streets is proposed. The idea is that these procedures could be used in an application after a stage of entity extraction from natural language, where landmarks and spatial relations are collected, to generate possible projections of spatial relations. Despite most of the relations explored being directional, the set of relations covered by the proposed algorithms is not intended to be a comprehensive list of all relations in this class. In fact, the main focus of the study is to explore a subset of relations, that seem to be among the ones that are most often used when people describe places. This subset includes common expressions that although are widely used by people in conversation, to the best of our knowledge have neither been categorized as directional nor explored before such as Next-To, Near and At-Street.

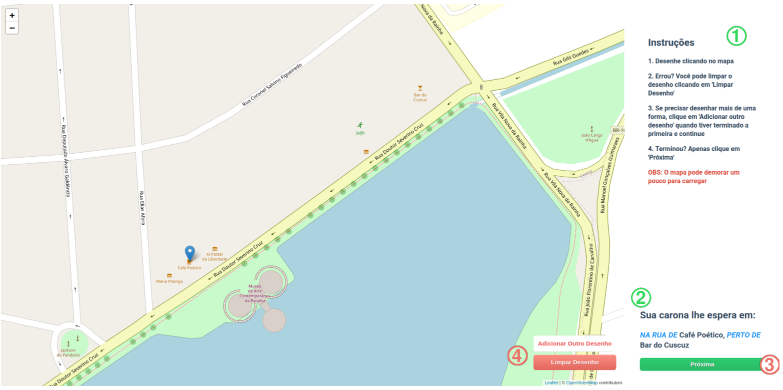
3. Materials and Methods

In a preliminary study, a group of 57 participants were presented with a point in a map and asked to describe its location. The descriptions were then studied and a list of the most frequently mentioned spatial relations was compiled.

These relations were implemented in a database spatial query language and the algorithms designed. In order to evaluate them and also better understand the way people reason about spatial relations, another experiment was carried out. Through the usage of a web app, another group of 20 participants, none of whom participated in the preliminary study, were told to picture the following scenario:

"Imagine that a friend will give you a ride and tell you over the phone where the car stopped and is waiting. Based on the description he gave you, we ask you to draw on the map the area where you think the car might be."

The participants of this experiment form a diverse group of people from different backgrounds. However, most of them are students (undergrad and grad) aged between 20 and 35.



Instruções

1. Desenhe clicando no mapa

2. Errou? Você pode limpar o desenho clicando em 'Limpar Desenho'

3. Se precisar desenhar mais de uma forma, clique em 'Adicionar outro desenho' quando tiver terminado a primeira e continue

4. Terminou? Apenas clique em 'Próxima'

OBS: O mapa pode demorar um pouco para carregar

Sua carona lhe espera em:

NA RUA DE Café Poético, *PERTO DE* Bar do Cuscuz

Figure 1. 1 - Drawing Instructions. 2 - Sentence Describing Location. 3 - Next Button. 4 - Drawing Controls

The web app then shows up a map with a highlighted landmark and a sentence that describes the location of the car. Figure 1 shows the screen that the participants see when they are supposed to start drawing. The sentence in (2) means *Your ride awaits you at: AT Café Poético's STREET, NEXT TO Bar do Cuscuz*. The blue capitalized words represent spatial relations while the black ones represent spatial landmarks. Participants drew the regions by clicking on the map and creating points and lines. It is also possible to draw multiple disconnected geometries, to support scenarios where a participant wishes to draw on more than one place. Each person had to draw five relations (Table 1) for each of the four landmarks.

Table 1. Spatial Relations Names

Brazilian Portuguese Relation Name	English Translation
NA FRENTE DE	In front of
NA RUA - PERTO DE	At Street - Near
ENTRE	Between
AO LADO DE	Next to
À DIREITA DE	Right of

A street might extend itself for kilometers, and this was a concern when designing the experiment, since participants could get tired of drawing really large areas. For this reason, relations At-Street and Next-To were combined so that participants were supposed to draw a polygon on only a smaller portion of the street.

The drawings were then stored in the GeoJSON format and a CSV of the data is available at GitHub [1].

4. Spatial Relations Algorithms

This section presents the algorithms designed to infer spatial extents of regions, vaguely described in terms of different spatial relations to certain landmarks. The algorithms are presented as functions named as spatial relations.

These algorithms must deal with some level of uncertainty when there is insufficient information about the spatial features referred to in the descriptions. For example, for a building located at a street corner, defining its facade may be considerably challenging or even impractical using traditional mapping data, posing even more challenges for modeling some relations, such as In-Front-Of, as the buildings' facade may be extended around the corner.

The lack of geographic data in the appropriate format may also impact the efficacy of this kind of algorithm. For example, in traditional mapping datasets, many spatial extents of landmarks are available, however, many others are represented as single geographic coordinates (points). The algorithms proposed here are able to better infer the spatial extents of regions for polygon inputs. In the absence of this format of data they are also capable of working with point inputs. However, we believe the availability of this kind of data as polygons tends to increase considerably in the next years, contributing directly to the accuracy of systems that will incorporate those algorithms.

All proposed functions take as input a geometry, representing the spatial extent of a landmark, and return a generated polygon, representing the spatial extent of a region that best describes the relation with respect to that landmark. These regions are called here **acceptance regions**. An important observation is that the algorithms work in the scope of streets. This way, when generating the acceptance region to the Right-Of relation for instance, one should expect that the algorithm will produce a region that encompasses the portions of street that are to the right of the landmark.

[1] https://github.com/jslucassf/geoinfo-spatial-relations

4.1. In Front of

Algorithm 1 implements the relation In-Front-Of. Line 2 tests whether the input geometry is of type point or polygon. For point input geometries a buffer around the input is computed (Line 3), the intersection between this buffer and the nearby streets (Line 4) represents the candidates to be included in the acceptance region. For each candidate, the algorithm tests if there is another object between the input landmark and the candidate and includes the street in the final result, if it does not meet these conditions (Lines 5 to 10).

Algorithm 1 In Front of

```
 1: function InFrontOf(landmark geometry)
 2:     if landmark is of type point then
 3:         Compute a buffer around landmark
 4:         intStreets = the intersection between the buffer and all streets that intersect it
 5:         for each street in intStreets do
 6:             testLine = a line from landmark to street
 7:             if testLine do not crosses another landmark or street in intStreets then
 8:                 finalFront = Union of street and finalFront
 9:             end if
10:         end for
11:     else
12:         for each side in the landmark polygon do
13:             Compute a one-sided buffer in the line representing the side of the polygon
14:             streetFront = the union of all streets that intersect the one-sided buffer
15:             Compute a buffer between the landmark and streetFront
16:             if There are no other objects inside this buffer then
17:                 finalFront = Union of streetFront and finalFront
18:             end if
19:         end for
20:     end if
21:     return finalFront
22: end function
```

For polygon input landmarks, the procedure is almost the same, with the exception that the buffer used to select the candidate streets as well as the tests that check if a candidate street is really in front of the landmark (Figure 2), can be computed for each of the lines representing sides of the polygon (Lines 12 to 19), this allows the generation of an acceptance region that is much more accurate, for it really represents the full extension of region that is in front of each particular side of landmark, as opposed to an estimate of such region, which is the case for point input landmarks. Line 12 returns the acceptance region produced by the union of street candidates for the appropriate input format (Line 8 for points and Line 17 for polygons).

4.2. At Street

An example sentence that uses this relation is: *"The car is at the university's street"*. When the university is a well known landmark in the area, the street in which it is located becomes a common landmark. This relation produces an acceptance region that includes the whole extension of the street.

Algorithm 2 At Street

```
1: function ATSTREET(landmark geometry)
2:     Compute the front of the landmark
3:     for Each street that intersects the landmark's front do
4:         if Area of intersection between the street and the front is big then
5:             finalStreet = Union of intersection and finalStreet
6:         end if
7:     end for
8:     return finalStreet
9: end function
```

Algorithm 2 computes the front region of the landmark by making use of Algorithm 1 (Line 2). It includes in the acceptance region all streets that intersect the front (Lines 3 to 7). For this relation it is important to filter out the parts of streets whose areas are small enough (line 4), as sometimes the crossing between streets is included in the front area (mostly for points) but only one of the streets is really in front of the landmark.

If the data includes the addresses of the objects, the projection of the acceptance region could be thought as straightforward, however we also consider that people might think that streets that are not the official address of some building but that are adjacent to one of its sides could also be seen as "the building's street".

4.3. Near

The Near relation is implemented in Algorithm 3. It is quite simple and is the same for points and polygons. A buffer around the landmark represents the region that is *near* it (Line 2).

Algorithm 3 Near

```
1: function NEAR(landmark geometry, distance float)
2:     return a buffer with a distance-sized radius around landmark
3: end function
```

The distance parameter should be tuned, and probably varies depending on the context (e.g. people who live in smaller cities might consider as near, a distance that is different from people that live in bigger cities). Future experiments could try to quantify this value, by averaging the distances that people consider as being Near some landmark.

4.4. Between

Between is the only ternary relation in this list. It defines the position of one object, with respect to two others as in *"The car is between the university and the bookstore"*. For this reason, Algorithm 4 takes as input two geometry parameters.

The function draws a line between the two input geometries (Line 2). If a buffer around the line between both input landmarks is returned, the result will include regions

Algorithm 4 Between

```
1: function BETWEEN(landmark1 geometry, landmark2 geometry)
2:     Draw a line between a point in the surface of each of the two geometries
3:     Get two points in the line that are at a distance $d$ from each end
4:     Draw a new line between the two points
5:     return a $d$-radius buffer around the new line
6: end function
```

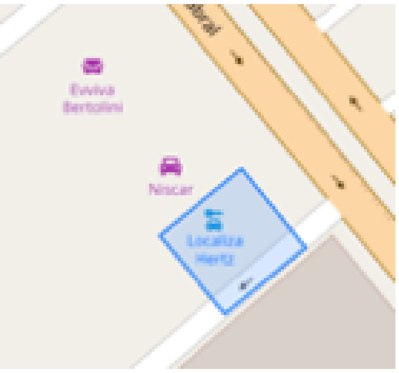


Figure 2. The street is in front of *Localiza Hertz*, not the input landmark (*Niscar*).

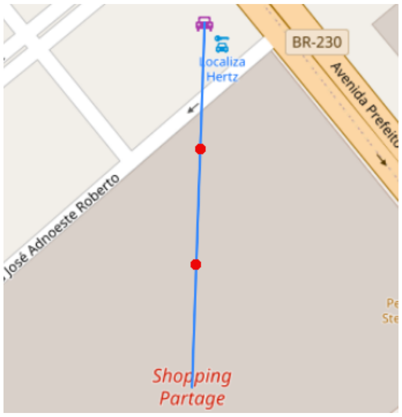


Figure 3. Points that are equidistant.

that are actually outside the desired relation. To fix this issue, the procedure finds two points along the line that are located at the same distance d to each of the lines ends (Line 3, Figure 3), in PostGIS, this could be done using ST_LineInterpolatePoint. A new line between these two points is drawn (Line 4) and a buffer of radius d around it is returned (Line 5). Since d is the exact distance between each input geometry and the new line, a buffer with radius d will not cover any region that is not between the two landmarks.

4.5. Next To

In the Next-To relation, regions that are immediately next to the landmark but not in front of it are included. This relation is basically a smaller Near minus the front. For this reason, Algorithm 5 starts by computing the landmark front (Line 2) using Algorithm 1. The street relation is also used, so Line 3 uses Algorithm 2 to compute the street relation. A buffer around the landmark is generated (line 4) and intersected with the region that correspond to the street of the landmark (Line 5). For each street in this intersection (Lines 6 to 18) a line is drawn starting from the landmark (Line 7, Figure 4), if this line crosses the difference between intersections and the street (Figure 5), this means that the intersection includes a street that is closer to the landmark, so the one that is farther away is removed (lines 8 to 10). A special case is when the input geometry is actually in point format, for the front relation for points can include large regions. In this scenario, Lines 13 to 17 find for each street in the resulting area, the point that is closest to the input landmark. Buffers around these points serve as the front relation for the input landmark. After this, the difference between the resulting area and the front is returned (Line 19),

Algorithm 5 Next To

```
function NEXTTO(landmark geometry)
    Compute landmark front
    Compute landmark street
    Compute a buffer around landmark
    nextInt = the intersection between the buffer and landmark street
    for Each partOfStreet that intersects nextInt do
        Draw a line from landmark to partOfStreet
        if Line do not cross the difference between nextInt and partOfStreet then
            nextFinal = union between nextFinal and partOfStreet
        end if
    end for
    if Landmark is of type point then
        for Each partOfStreet that intersects nextFinal do
            Get the point in partOfStreet that is closest to landmark
            nextFinal = nextFinal minus buffer around the closest point
        end for
        return nextFinal
    end if
    return Difference between nextFinal and the landmark front
end function
```

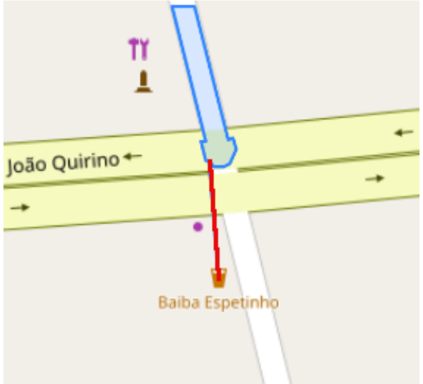


Figure 4. A line from landmark to a street in the Near relation.

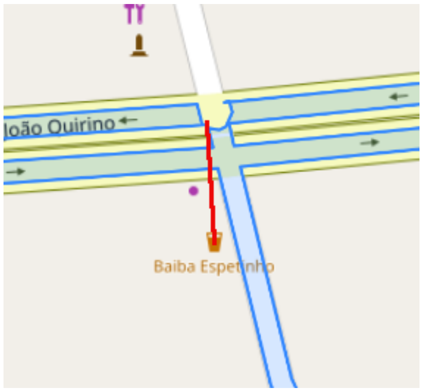


Figure 5. It crosses the difference between the Near relation and the street.

4.6. Right of (Left of)

When someone say "The car is to the right of the university", the message can be interpreted in different ways according to one's spatial mental reasoning. If directions are defined based on the observer's point of view, the region defined by the relation may assume a completely opposite position than if directions were defined by the position of the reference object itself. Contextual information such as a clear frame of reference is then needed to disambiguate the sentence. According to [Retz-Schmidt 1988] frames of reference can be classified as **intrinsic** where an intrinsic property of the reference object (such as its front) defines orientation, **extrinsic** where orientation is defined by another external landmark and **deictic** that defines orientation based on an observer's point of view.

The Right-Of Algorithm 6 receives a string of text as second argument, representing the type of frame of reference that should be used to define the relation. It can assume the values of two of the three aforementioned types, intrinsic (defines right, based on the landmark front) and deictic (defines right based on the point of view of an observer positioned in front of landmark and looking towards it, as in Figure 6).

Algorithm 6 Right of

```
 1: function RIGHTOF(landmark1 geometry, for text)
 2:     if for == "intrinsic" then bufferSide = "left"
 3:     else if for == "deictic" then bufferSide = "right"
 4:     end if
 5:     Compute landmark front
 6:     Compute landmark Next To relation
 7:     for Each partOfStreet that intersects landmark front do
 8:         Draw a line from landmark to the centroid of partOfStreet
 9:         Create a one-sided buffer that grows in the direction of the bufferSide variable
10:         for Each polygon in landmark Next To relation do
11:             if polygon intersects buffer then
12:                 finalRight = union between finalRight and polygon
13:             end if
14:         end for
15:     end for
16:     return finalRight
17: end function
```

This function computes the front relation using Algorithm 1. For each of the streets that intersect the front acceptance region, (Lines 7 to 15) it computes a one-sided buffer on a line that goes from the input landmark to the street (Lines 8 and 9). To determine in which side of the line the buffer is generated, the string representing the frame of reference is used (Lines 2 to 4). The relation Next-To includes regions that are positioned immediately to the left and to the right of the landmark. For this reason, it is computed (Line 6). If any of its containing polygons intersects the one-sided buffer, it is included in the final result. Line 16 return the acceptance region, formed by the polygons of the Next-To relation that intersect the one-sided buffer.

The algorithm for the Left-Of relation is almost the same as this one, the only difference is in lines 2 and 3, where the “left” and “right” values are swapped. For brevity reasons, it is not included here.

5. Evaluating Frames of Reference in the Spatial Reasoning of People

As already discussed, correctly interpreting the relation Right-Of can be challenging. One of the main interests during the experiments, was to try to understand the spatial reasoning behind decisions when interpreting this relation. The question of interest here is: Which type of frame of reference (FoR) best describes the reasoning behind the decisions of people, when faced with the task of locating a region said to be at the right side of some reference object?

To answer this question, the drawings in the experiment were analyzed. Particularly the ones made when participants were prompted with the sentence that included the Right-Of relation. The translated sentence was: “Your ride is to the right of <landmark>”.

The terminology in the analysis assumes a **deictic** FoR, therefore, here, when a drawing is said to be to the right of the landmark, this means that it is located to the right side in the perspective of an observer that looks towards the landmark (Figure 6).

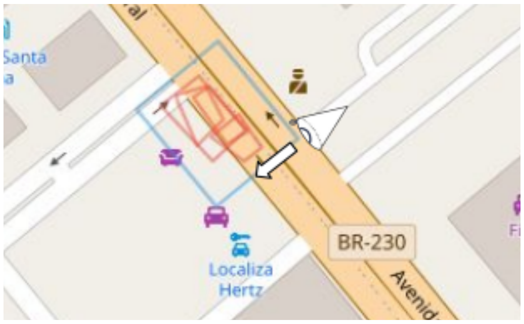


Figure 6. Drawings to the right of the observer’s point of view

The drawings were classified using buffers for each of the sides (the blue polygon on Figure 6). Results are shown in Figure 7. In most of the cases, drawings intersected only with the left buffer, which seems to indicate that participants consider the intrinsic properties of the landmark when interpreting the relation, i.e. the reasoning behind the **intrinsic** frame of reference. However, many participants positioned their regions in the right side, in fact for one of the landmarks, this was the case more often than the left side. This raises a few questions. Given the high number of drawings on the right side for landmarks *Café Poético* and *Maria Pitanga*, might this result be affected by the geographical direction towards which the landmarks are facing (these landmarks are neighbors and face the same direction)? What happens if we factor in people’s problems in telling left from right or even map reading difficulties? A future work, would be to repeat this experiment, but showing participant’s the actual images of landmarks facades.

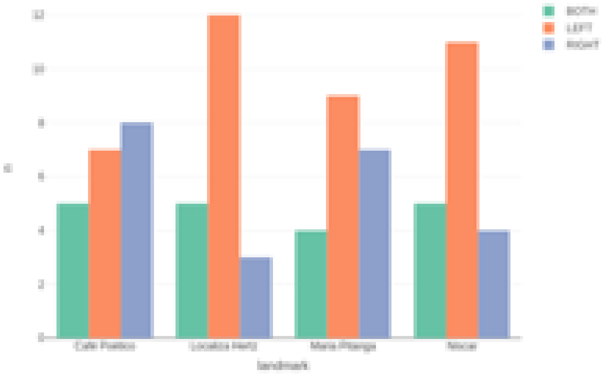


Figure 7. Location of drawings in the Right-Of relation

Another interesting finding is that many participants chose to draw on both sides of the landmark. In another future work, the reason behind this choice could be investigated by identifying and questioning the individuals.

6. Evaluating the Precision of the Proposed Algorithms

In order to evaluate the precision of the algorithms, they were implemented using PostGIS and executed for each of the four landmarks used in the experiments. The regions produced by them were then compared against the collected drawings. One issue with the drawings was that although the experiment defined that participants should imagine the location of a car, some drawings do not intersect streets at all. This might be due to not so clear instructions and a future experiment can try to address this issue. However, as the algorithms function in the scope of streets (the regions produced by them are mostly located on the streets), the drawings that do not intersect streets at all were not considered.

6.1. Intersection of Areas

The chart presented in Figure 8 shows that for almost all relations, the algorithms produce regions that intersect the majority of drawings made by participants of the experiment. The relation Right-Of got the lowest results however this could be explained by the uncertainty in this relation, explored in Section 5.

6.2. Jaccard's Similarity Coefficient

A common metric used to access the similarity between sets is Jaccard's Similarity Coefficient. It expresses how similar two sets are in a scale of 0 to 1 and is computed by the Equation 1. This metric was used to evaluate how similar are the geometries produced by the algorithms and the drawings made by the participants.

$$Jaccard(A, B) = A \cap B / A \cup B \tag{1}$$

In order to assess the complexity of the task, a value to show how similar the drawings made by the participant's are with each other was also computed, here it was called the **inner jaccard**. For each drawing, the Jaccard's Similarity Index with all other drawings in the same category (landmark and spatial relation) is computed, the median result is the inner jaccard and it represents how similar is this drawing to all the others. Figure 9 displays the results of the analysis.

As can be seen in the low inner jaccard values, the drawings themselves are not very similar. This might indicate that people have different understandings of spatial relations. Considering this, the proposed algorithms had modest results comparing to such a diverse set of region polygons, the exception being the Between relation, this can be explained by the fact that the region produced by the algorithm is large, for this reason it also intersects all the drawings in the same relation. These results, when coupled with the high intersection percentages shown in Section 6.1, suggest that these algorithms are a good starting point for the implementation of some of the spatial relations that are most used in people's daily language.

7. Conclusion

This paper proposes algorithms to implement some of the spatial relations that are most used by people in natural conversation. It includes an experiment that evaluates how well

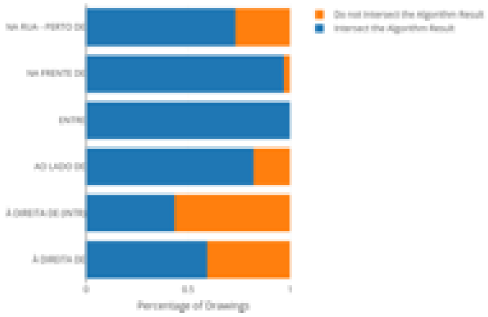


Figure 8. Percentage of drawings that intersect the region produced by the proposed algorithms

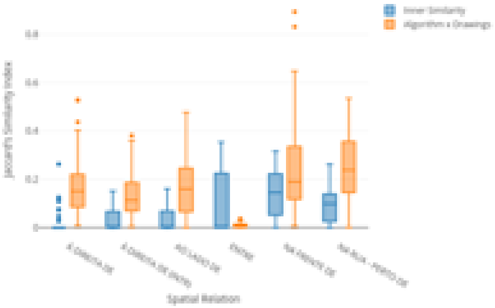


Figure 9. Jaccard's Similarity Coefficient between geometries

the output of the algorithms, match the mental representation of spatial relations in the minds of the participants. The analysis of the collected data shows that this is a difficult problem, however the proposed algorithms hold promissory results, intersecting most of the regions made by the participants and presenting some similarities to them. Another contribution to the field is making available a dataset of more than 400 drawings of spatial relations, allowing further studies on the interpretation of spatial relations by humans.

References

Bucher, B., Falquet, G., Clementini, E., and Sester, M. (2012). Towards a typology of spatial relations and properties for urban applications. *Usage, Usability, and Utility of 3D City Models–European COST Action TU0801*, page 02010.

Clementini, E. (2013). Directional relations and frames of reference. *GeoInformatica*, 17(2):235–255.

Clementini, E. and Bellizzi, G. (2019). A geospatial application framework for directional relations. *ISPRS International Journal of Geo-Information*, 8(1):33.

Clementini, E. and Billen, R. (2006). Modeling and computing ternary projective relations between regions. *IEEE Transactions on Knowledge and Data Engineering*, 18(6):799–814.

Clementini, E., Sharma, J., and Egenhofer, M. J. (1994). Modelling topological spatial relations: Strategies for query processing. *Computers and Graphics*, 18(6):815–822.

Cohn, A. G. and Renz, J. (2008). Qualitative spatial representation and reasoning. *Foundations of Artificial Intelligence*, 3:551–596.

Egenhofer, M. J. and Franzosa, R. D. (1991). Point-set topological spatial relations. *International Journal of Geographical Information System*, 5(2):161–174.

Mark, D. M. and Egenhofer, M. J. (1994). Modeling spatial relations between lines and regions: combining formal mathematical models and human subjects testing. *Cartography and geographic information systems*, 21(4):195–212.

Retz-Schmidt, G. (1988). Various views on spatial prepositions. *AI magazine*, 9(2):95–95.

Sugarcane canopy structure temporal analysis considering phenological stages and the temporal dynamics of NDVI values

João F. Gromboni[1], Luíz H. Pereira[1], Javier Pulido[1], Ana P.S.G.D. Toro[1], Mateus V. Ferreira[1]

[1]Department of Research and development– IDGeo – Agricultural intelligence
Piracicaba – SP – Brazil

joao.gromboni@idgeo.com.br,luiz.pereira@idgeo.com.br
javier.pulido@idgeo.com.br, ana.paola@idgeo.com.br,
mateus.vidotti@idgeo.com.br

Abstract. *This research aims to perform an exploratory analysis about the components of sugarcane canopy in each phenological stage. For this purpose, a methodology for field data sampling using an optical active sensor was proposed to obtain Normalized difference vegetation index (NDVI) values. Also, supervised classifications were conducted based on UAV images in order to measure the percentage of each component, in each phenological stage. As a result, the methodology had a satisfactory performance, further, the classification results demonstrated the temporal heterogeneity in sugarcane canopies.*

INTRODUCTION

Fundamental remote sensing of vegetation studies focusses on the characteristics and the dynamic of individual leaves reflectance, considering its causes and alterations. Individual properties of leaves reflectance are essential for the understanding of reflectance concerning the whole plants and the canopy. However, the measures made by remote sensors could not be actually explained exclusively for leaves characteristics, even with an interpretation directly associated to an ideal spectral behavior of an isolated leave (Goel and Strebel, 1984). Thus, it is not appropriated to extrapolate spectral data from an individual leave to a canopy without proper adjustment and comprehension. This, due to the fact that the canopies spectral behaviors are not composed exclusively by leaves (Bauer et al., 1981).

In this context, Meneses and Madeira Neto (2001) indicated that for canopies remote sensing, a complex interaction should be considered, by virtue of the many environmental parameters and factors described by Rosa (2009) and Jensen et al.(2009), that influence in the image(aerial or orbital) composition. The most cited parameters are atmospheric condition, illumination geometry and scene sight elements, sensor model, soils (substrate), leaf (shape, position, water content, pigmentation, internal structure), biomass (vegetation total density), canopy cover, canopy aspects in terms of vegetation/planting density and leaf area index (Ponzoni et al., 2012). Further, it should be weighed that all those cited parameters have spatial and temporal variation associated with vegetation type, development stage and crop conditions (Kimes and Kirchner, 1983).

Knipling (1970) analyzed vegetation interactions in remote sensing into aspects related to canopy, indicating that each canopy has a characteristic geometrical/structural, determined by plants size, shape and orientation, for both horizontal and vertical dimensions. Still, the same author emphasizes that crop management practices and growing environmental conditions also influence spectral behaviors, promote radiation attenuation (considering shadows and non-foliar background surfaces), also, it was observed that the reflectance in canopy is considerably lower than an individual leaf.

In terms of canopy structure, it could be mathematically described by physical parameters, such as plants disposal, leaf area index (LAI), leaves spatial distribution and inclination angles (Ponzoni et al. 2012). The same authors highlighted the clear distinctions made between complete agricultural canopies featured by homogeneity, complete soil cover and the incomplete ones, where plant lines and bare soil among lines could be clearly noted, this due to the incomplete canopy cover. Accordingly, Ranson et al. (1985) demonstrated spectral row crops complexity, since the scene obtained by the sensor is composed by vegetation and bare soil, in varying proportions as a consequence of crop cycle conditions. Likewise, the shadow produced by crops into bare soils rows should be considered.

Hence, as stated by Meneses and Madeira Netto (2001), the effectiveness of remote sensing of vegetation made by aerial and orbital platforms is directly depending on the capacity of relate the canopy spectral behaviors with its composition, this only could be achieved by the complete understanding of this relationship (field composition versus spectral reflectance) and its sources. The same authors also suggest the recognition of different spectral targets and their contributions for reflectance, based on repetitive field observations, in order to obtain the percentage of soil cover or LAI, shadows and vegetation. This comprehension could support many crop remote sensing studies, especially those related with crop yield for row crops, such as sugarcane.

Many sugarcane yield studies based on remote sensing use only values from the Normalized Vegetation Index (NDVI) (Mulianga et al., 2013). However, the use of NDVI for biomass and productivity estimation could cause inaccuracy, considering the composition of sugarcane canopies. The canopies in sugarcane lands are composed by a diverse vegetation condition with significant temporal dynamics (Bengué et al., 2010). Based on this framework, it is meaningful to figure out the sugarcane canopy particularities that contribute for spectral reflectance, especially for vegetation indexes such as NDVI and how they vary through time.

On account of this, the present work aimed the characterization (*in situ*) of sugarcane canopy and its dynamic change through phenological stages, based on the vegetative composition (green and dry leaves), shadow, substrate (soil and straw), correlated with NDVI values obtained by field sensor.

MATERIAL AND METHODS

This study was performed in partnership with a sugarcane mill, which provided data and environment for analyses, in Ribeirão Preto municipality, São Paulo state, Brazil.

Field data sampling

Twenty sampling points, in a ratio of 20 km, were defined over similar sugarcane varieties (considering architecture and canopy) and different development crop stages. The sampling occurred systematically during 12 months (2018-08-23 up to 2019-08-26) with intervals ranging from 30 up to 45 days.

For NDVI values collection the active optical sensor Trimble GreenSeeker Handhelt was used. This sensor emits a range of electromagnetic waves in two wavelengths, being, red (660nm) and near infrared (770nm), those that once achieve the target are reflected and collected by the sensor. Further, reflectance values are converted to NDVI (Tucker, 1979), accordingly to equation 1.

$$NDVI = \frac{(\mathrm{NIR}-\mathrm{RED})}{(\mathrm{NIR}+\mathrm{RED})} \qquad \text{equation 1}$$

Where NDVI is Normalized vegetation index, NIR is the target reflectance in the near infrared wavelength and RED is the reflectance in the red wavelength.

The data acquisition was made using a vertical distance ranging from 60 up to 90 centimeters from the target. Each sampling point was composed by four samplings (Figure 1), and for each one, five NDVI measurements were performed. Then, the final NDVI value for each one of the four samplings is composed by the mean value considering the five measurements. This procedure was implemented in order to obtain most representative NDVI values for each area, besides the sampling standardization.

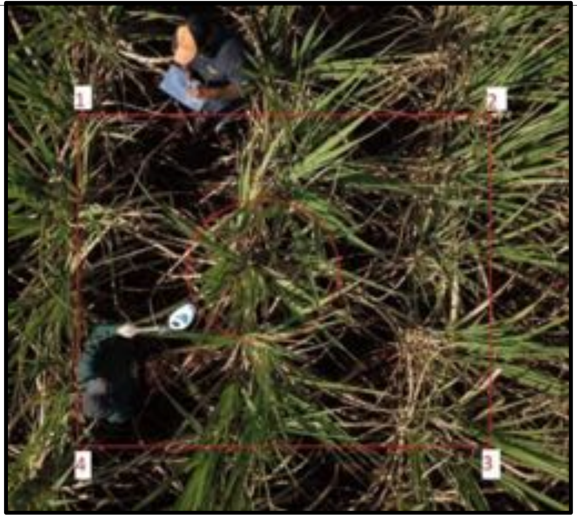


Figure 1.Schematization of field procedure performed to obtaining of NDVI values in Ribeirão Preto, São Paulo state, Brazil.

Moreover, to each sampling point, images from unmanned aerial vehicle (UAV) were obtained. A Mavic Pro from DJI was utilized, which is battery-powered and has a maximum flight autonomy of 27 min at 25 km/h or a maximum flight distance of thirteen kilometers. The UAV has a ½ 3" (CMOS) sensor and FOV 78.8 26mm lens. Four images were taken into each sampling point, in different altitudes, depending on sugarcane heights. Thus, for sugarcanes lower than 1 meter of heigh, the UAV imagery were obtained at flight altitudes of 3,10, 50 and 100 meters. In addition, for sugarcane plants taller than 1 meter, the flight altitudes were 5, 10, 50 and 100 meters. Those acquisitions were made aiming at broader imagery observation (DANDOIS et al., 2020).

Data classification

The three meters of altitude images were adopted to determine the percentage of green and dry leaves, shadow and soil, in order to characterize the canopy structure, sampling point conditions and NDVI values. The first step of this characterization was made in GIS environment using the software ArcGis 10.5 for image classification. The supervised classification was performed employing the Maximum likelihood algorithm (MAXVER), which is based on the Bayes optimization strategy, minimizing classification errors (Swain and Davis, 1978). This algorithm assumes that all the inputs attributes have a normal distribution, from this point, the probability of each pixel belong to a specific class is calculated.

The supervised classifiers are not usually recommended for high resolution images especially due to the fact that the classification could generate many isolated pixels. However, this limitation was attenuated increasing the number of samples for each class (50 samples) in addition to the methodology of sample collection, generating larger segments, thus, better representing a large variability of values for the same class.

Then, as a first step, five image classes were identified and designed as "green leaf", "dry leaf", "shadow", "straw" and "soil" as demonstrated in Figure 2. Further, for feature training, 50 samplings were taken for each class such as illustrated in Figure 2.

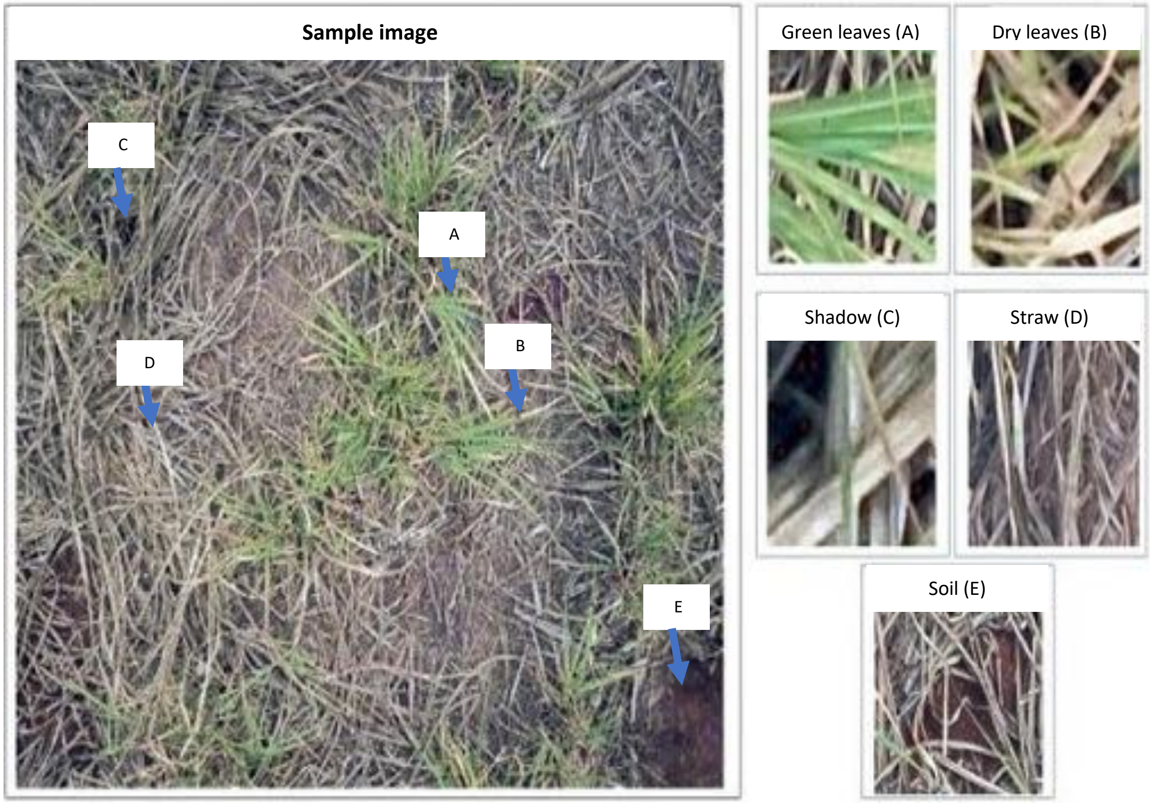


Figure 2. Example of targets adopted as classes in UAV images, in Ribeirão Preto, São Paulo state, Brazil.

The sugarcane mill provided the last harvest date for the sampled areas, thus it was possible to determine how many days were passed from the last harvest until the sampling day. Using this information, based on literature review, the phenological stages were determined as described into Table 2 (EMBRAPA, 2012).

Table 1. Description of standard values used to estimate phenological stages accordingly to days after harvest (DAH)/days after planting (DAP).

DAH	Phenological stages
0 - 30	Germination
30 - 120	Tillering
120 - 360	Vegetative
360 - 500	Maturing

Data analysis

For data analysis, the NDVI values obtained in field and the results generated by the classification were structured, filtered and organized based on an individual id (ID_TA) attributed to each plot plus the data collection date(date_camp). According to Dasu and Jhonson (2003) the dataset cleaning and structuration is essential in order to conduct consistent data analysis. Further, the attribute "DAH" was generated, indicating the days after the last harvest. From this attribute, the dataset was ordered as "DAH" equal 0 up to the last registered one (500), aiming at a temporal analysis.

In sequence, an exploratory analysis and a linear regression were performed among plots containing the same sugarcane variety "DAH" and the classification results (percentage of "Green leaf", "Dry leaf", "Shadow"," Straw" and "Soil"). Also, the same procedure was adopted for the comparison between NDVI values and classification results. All statistical analyses were conducted in python environment, using the libraries Pandas, Numpy, Statsmodels, Seaborn, Matplotlib and scikit_learn (Lemenkova, 2020).

RESULTS AND DISCUSSION

The results obtained by a supervised classification are illustrated in Figure 3, where it is possible to see how each pre-defined class was classified in each phenological stage. Based on Figure 3 observation, it is possible to argue that the pre-defined targets where identified in a satisfactory manner.

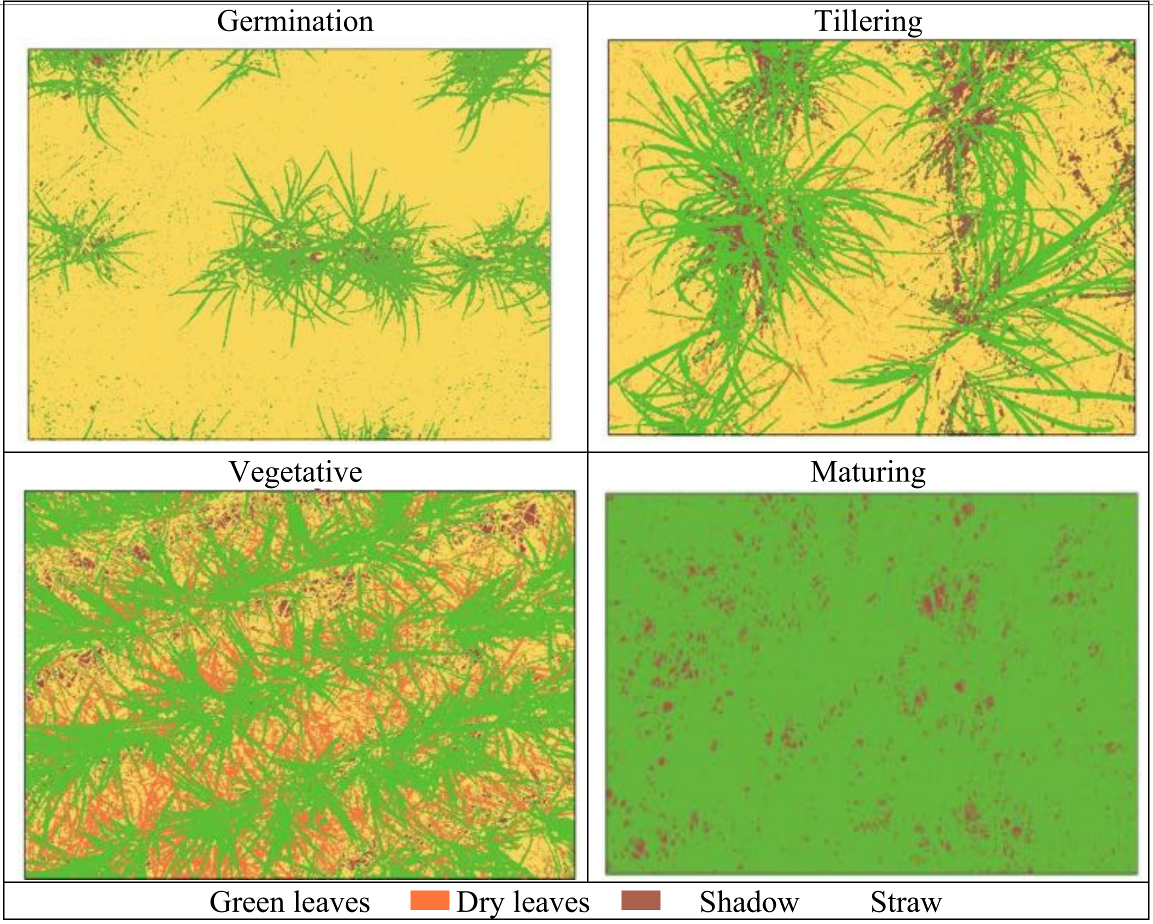


Figure 3. Example of classification results obtained based on UAV images for sugarcane canopy in each phenological stage, in Ribeirão Preto, São Paulo state, Brazil.

Further, in Figure 4 it is possible to see the classification results demonstrating the contribution (area - %) of each vegetative component (green and dry leaves), shadow and substrate (soil/straw), for all collected samples and how they evolve through each phenological stage.

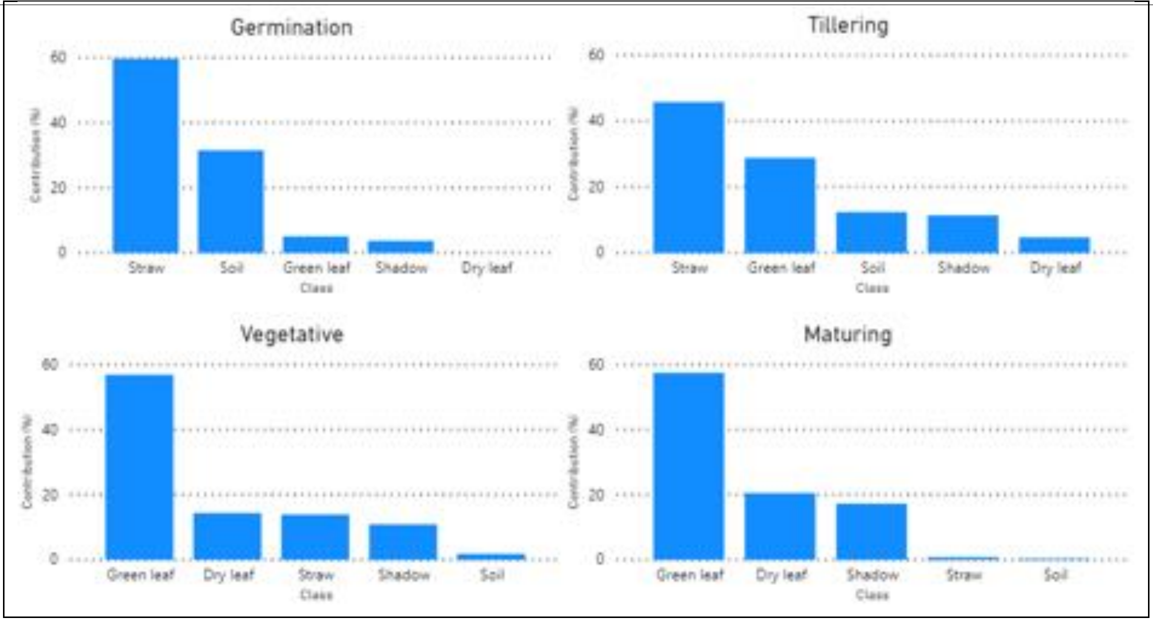


Figure 4. Distribution of vegetative components (green and dry leaves), soil, shadow and straw through phenological stages in Ribeirão Preto, São Paulo state, Brazil.

The first consideration about Figure 2 is the noticed high variability among classes even after grouped through phenological stages. This result reinforces the necessity of consider individual sugarcane management practices in order to perform precise analyses.

However, one constant could be observed, that is the high occurrence of straw in canopy cover until the tiller stage, followed by green leaves in vegetative and maturing stages (Figure 4). The dry leaves start to be more representative during vegetative and maturing stages. Other relevant observed result is the shadow occurrence that was constant through all phenological stages, with high occurrence in the maturing stage (17%) (Figure 4).

Further, as previously discussed, it is important to relate the canopy cover targets occurrence with NDVI values measured in field. Then, Figure 5 present the NDVI values variability, considering all the collected samples, through all phenological stages.

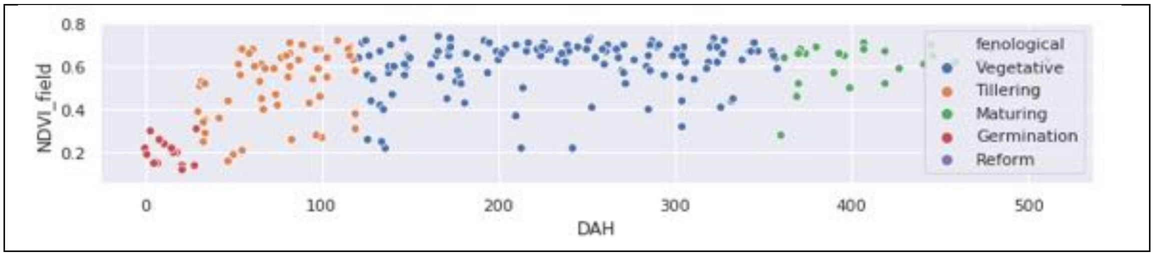


Figure 5. Distribution of NDVI values through phenological stages and days after harvest (DAH), in Ribeirão Preto, São Paulo state, Brazil.

In Figure 5, it is possible to see the increase of NDVI values through the DAH variable, thus if we consider the sugarcane development in field, low values of NDVI (<0.2) could be associated with bare soil, and the value increases following the sugarcane development until sugarcane maturity (NDVI>0.71). This temporal behavior is accordingly to related in literature (Fernandes et al., 2017; Gonçalves et al., 2012). The statistical values presenting the overall tendency for each target class, considering phenological stages and NDVI values are presented in Table 2. Those measures were obtained using the zonal statistics tool in GIS environment through the software ArcGis 10.5.

Table 2.Descriptive statistics for each target class, considering different phenological stages in Ribeirão Preto, São Paulo state, Brazil.

Target class occurrence (%)	Measure	Germination	Tillering	Vegetative	Maturing	Reform
Green_Leaf	**count**	16	55	134	26	9
	Mean	5.067	28.867	56.986	57.501	13.912
	Std	6.85	22.276	25.884	26.058	15.823
	Min	0	0	0	0	0
	Max	18.019	77.472	95.179	94.657	45.852
Dry_leaf	**Count**	16	55	134	26	9
	Mean	0	4.73	14.485	20.48	0
	Std	0	13.364	17.686	22.758	0
	Min	0	0	0	0	0
	Max	0	90.193	74.13	68.748	0
Shadow	**Count**	16	55	134	26	9
	Mean	3.698	11.482	10.999	17.297	4.049
	Std	5.343	10.214	10.29	12.296	3.857
	Min	0	0	0	0	0
	Max	16.795	47.971	67.138	42.201	10.792
Soil	**Count**	16	55	134	26	9
	Mean	31.514	12.492	1.829	0.05	54.345
	Std	42.237	24.855	7.493	0.253	32.21
	Min	0	0	0	0	1.007
	Max	100	100	52.488	1.29	100
Straw	**Count**	16	55	134	26	9
	Mean	59.721	45.887	13.924	0.825	27.695
	Std	38.822	33.794	23.991	4.209	26.374
	Min	0	0	0	0	0
	Max	100	100	88.768	21.46	72.213
NDVI	**Count**	16	55	133	26	9
	Mean	0.2	0.513	0.611	0.608	0.442
	Std	0.057	0.157	0.116	0.097	0.198
	Min	0.12	0.16	0.22	0.28	0.18
	Max	0.31	0.72	0.74	0.71	0.7

Based in Table 2 it is recognized that for the Germination stage soil and straw have larger mean values than other classes, this, due to the fact that in this phenological stage the sugarcane has a low number of leaves and it is starting its growing stage, further, for this stage the NDVI values are low, varying from 0.2 up to 0.31.

In sequence, in tillering stage, the cited mean values of soil and straw start to decrease, while green (28)/dry (4.7) leaves and shadow (11.4) begin to increase, as well as NDVI value. Furthermore, in this stage, the std values are high, many factors could cause this, such as sugarcane age, vegetative material, soil type, that were not considered in our evaluation, moreover this is a phenological stage where there is vertical and horizontal growing, which could increase the std values (Table 2).

In parallel with sugarcane development, in vegetative stage there is an increase of occurrence for green/dry leaves and shadow, as opposite of, the soil and straw occurrence, that decrease (Table 2).

Finally, in maturing stage, green leaves and NDVI achieve their maximum value of reflectance and occurrence, in the other hand, the increase in dry leaf and shadow is not so significant. This could be caused due to the canopy configuration of the selected varieties, that were developed to reduce the quantity of shadow between rows, in order to promote a better use of the area.

FINAL CONSIDERATIONS

The proposed methodology for collecting NDVI in field using an optical active sensor demonstrated to be effective, since the temporal NDVI values and patterns were similar to the ones found in literature.

The determined classes in order to represent the sugarcane canopy structure were effective. However, it was perceived that the analyses should be limited to agronomical specific groups, mainly by soil type (instead of production environment), specific varieties and ratoon number.

The high variance of NDVI during all the phenological stages, demonstrates the main difficulty in stablishing direct correlations among vegetation indexes with crop sanity, since the presence of dry leaves (as a vegetative component) and substrate (mainly due to straw presence) were observed in scene cover.

As a further step, we suggest the use of this methodology to verify the differences among sugarcane varieties. Also, we suggest a comparison between NDVI values obtained by the in-field sensor and satellite images.

ACKNOWLEDGEMENTS

The authors thanks to The São Paulo Research Foundation (FAPESP) for support the project named “high frequency remote monitoring system for quality management and prediction of agricultural productivity”, process Nº 2017/08449-8.

REFERENCES

BÉGUÉ. Agnes et al. Spatio-temporal variability of sugarcane fields and recommendations for yield forecast using NDVI. **International Journal of Remote Sensing**. v. 31. n. 20. p. 5391-5407. 2010.

BONNETT. Graham D. Developmental stages (phenology). **Sugarcane: physiology. biochemistry. and functional biology**. p. 35-53. 2013.

Bull. T.A. and Glasziou. K.T.. 1975. Sugarcane. In: L.T. Evans (Editor). Crop Physiology. Cambridge University Press. London. pp. 51-72.

DANDOIS. Jonathan P.; OLANO. Marc; ELLIS. Erle C. Optimal altitude. overlap. and weather conditions for computer vision UAV estimates of forest structure. **Remote Sensing**. v. 7. n. 10. p. 13895-13920. 2015.

Dasu. T.. & Johnson. T. (2003). *Exploratory data mining and data cleaning* (Vol. 479). John Wiley & Sons.

FERNANDES, Jeferson Lobato; EBECKEN, Nelson Francisco Favilla; ESQUERDO, Júlio César Dalla Mora. Sugarcane yield prediction in Brazil using NDVI time series and neural networks ensemble. **International Journal of Remote Sensing**, v. 38, n. 16, p. 4631-4644, 2017.

Goel NS. Strebel DE. Thompson RL. Simple beta distribution representation of leaf orientation in vegetation canopies. **Agronomy Journal**. V6. p.800-803. 1984.

GONÇALVES, Renata RV et al. Analysis of NDVI time series using cross-correlation and forecasting methods for monitoring sugarcane fields in Brazil. **International journal of remote sensing**, v. 33, n. 15, p. 4653-4672, 2012.

JENSEN. J. **Sensoriamento Remoto do Ambiente: Uma perspectiva em recursos terrestres** (tradução José Carlos Neves Epiphanio et. al.). São José dos Campos. SP. Parêntese. 2009.

LEMENKOVA, Polina. Python Libraries Matplotlib, Seaborn and Pandas for Visualization Geospatial Datasets Generated by QGIS. **Analele stiintifice ale Universitatii" Alexandru Ioan Cuza" din Iasi-seria Geografie**, v. 64, n. 1, p. 13-32, 2020.

KIMES. D.S.; KIRCHNER. J.A. Diurnal variations of vegetation canopy structure. **International Journal of Remote Sensing**. v. 4. n. 2. p. 257-71. 1983.

KNIPLING. E. B. Physical and physiological basis for the reflectance of visible and near infrared radiation form vegetation. **Remote Sensing of Environment**. New York. V11. P.155-159. 1970

MENESES. P.R. & MADEIRA NETTO. J.S.. orgs. **Sensoriamento remoto: Reflectância dos alvos naturais**. Brasília. UnB/Embrapa Cerrados. 2001. 262p.

MULIANGA. Betty et al. Forecasting regional sugarcane yield based on time integral and spatial aggregation of MODIS NDVI. **Remote Sensing**. v. 5. n. 5. p. 2184-2199. 2013.

PONZONI. F. J.; SHIMABUKURO. Y. E.; KUPLICH. T. M. **Sensoriamento Remoto da Vegetação**. Oficina de Textos. 2012. 176 p.

RANSON. K.J.. L.L. BIEHL AND M.E. BAUER. Variation in Spectral Response of Soybeans with Illumination. View. and Canopy Geometry. **International Journal of Remote Sensing**. Vol. 6. No. 12. pp 1827-1842. 1985.

ROSA. Roberto. **Introdução ao Sensoriamento Remoto**. Uberlândia. EDUFU. 7a ed. 2009.

SWAIN P.H. & DAVIS S.M. 1978. **Remote sensing: the quantitative approach**. New York. McGrawHill. 396 p.

TUCKER, Compton J. Red and photographic infrared linear combinations for monitoring vegetation. **Remote sensing of Environment**, v. 8, n. 2, p. 127-150, 1979.

Circular Hough Transform and Balanced Random Forest to Detect Center Pivots

Marcos L. Rodrigues[1], **Thales S. Körting**[1], **Gilberto R. De Queiroz**[1]

[1]National Institute for Space Research (INPE)
Caixa Postal 515 – 12.227-010 – São José dos Campos – SP – Brazil

{marcos.rodrigues,thales.korting,gilberto.queiroz}@inpe.br

Abstract. *Water management is a field related to the increased mechanization of agriculture, mainly through center pivot irrigation systems, therefore it is important to identify and quantify these systems. Currently, with 6.95 million hectares, Brazil is among the 10 largest countries in irrigation areas in the world. In this study, a combined Computer Vision and Machine Learning approach is proposed for the identification of center pivots in remote sensing images. The methodology is based on Circular Hough Transform (CHT) to target detection and Balanced Random Forest (BRF) classifier using vegetation indices NDVI and SAVI generated from Landsat 8 and CBERS 4 images, being able to detect up to 90.48% of center pivots mapped by the Brazilian National Water Agency (ANA).*

1. Introduction

The practice of irrigation is one of the oldest techniques in agricultural production, used mainly by civilizations that developed in arid regions such as Egypt and Mesopotamia [Britannica Escola Web 2020]. Although agriculture initially developed predominantly in regions where the amount, spatial, and temporal distribution of rainfall was able to supply the need for crops. In Brazil the irrigation started around 1900 for rice production in the Rio Grande do Sul state. However, from the 1970 and 1980 decades, there was a significant intensification of agricultural activity in other regions. Leading to the rise of new irrigation poles according to the Brazilian National Water Agency (ANA) [ANA 2017].

Irrigation is an agricultural practice that employs a set of equipment and techniques to supply the total or partial deficiency of water for cultivation. On the one hand, the use of irrigation has several advantages for agricultural production, such as for example, increased productivity in relation to rainfed cultivation[1]. On the other hand, this use changes the availability conditions of water, because the water consumed by the evapotranspiration of plants and soil, not return to water bodies. Currently, with 6.95 million hectares, Brazil is among the 10 largest countries in irrigation areas in the world. However, the country still has great potential to be explored, according to ANA until the year 2030 there will be a strong expansion of the irrigation activity, especially by center pivot systems (Figure 1). The center pivot system is a mechanism formed by a galvanized

[1] Comparative of the productivity of rainfed and irrigated crops, source Secretariat of Water Resources of the Ministry of Environment SRH/MMA available on `https://www.codevasf.gov.br/linhas-de-negocio/irrigacao/a-irrigacao-no-brasil/comparativo`. Accessed on August, 24, 2020.

steel pipe suspended by towers with wheels at the base having water emitters along their length. This type of system irrigates a circular area by rotating this structure around a fixed point, called a pivot point, which serves to anchor the system and extract the water [Maranha 2018].

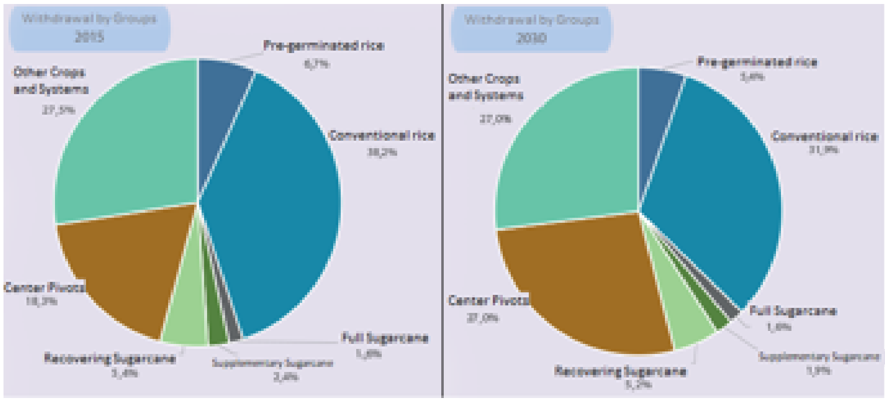


Figure 1. Water withdrawal by crop types and irrigation systems in 2015 and prevision for 2030. Adapted from ANA [2017].

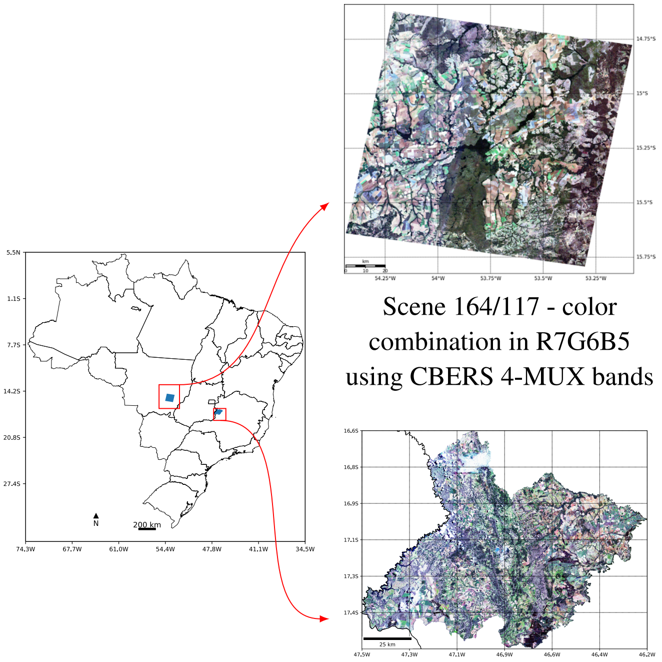
In the literature review for this work, it was evident that in the vast majority of studies the remote sensing data is used only as a support tool associated with statistical data from official surveys or measures in situ [Maranha 2018], mainly in the visual analysis of satellite images for the mapping of irrigation pivot circles. According to Zhang et al. [2018], although the visual analysis of images is relatively simple, the identification and digitizing for a wide range of areas can be very time and labor consuming. Nowadays, with the advent of very high-resolution images and the increasing use of Machine Learning (ML) techniques, the use of automatic and semi-automatic techniques for the identification and classification of fields with irrigation crops has increased [Aksoy et al. 2012]. However, most approaches are limited to small areas of study (partial scene) and specific periods of the year. For example, Santos et al. [2015], present an approach to characterize areas irrigated by center pivots, based on the adjustment of vegetation response thresholds using the Normalized Difference Vegetation Index (NDVI) and Soil Adjusted Vegetation Index (SAVI) for two periods, rainy and dry, in an irrigated area of the municipality of Paranapanema-SP. This type of extremely localized approach does not allow the reproducible and generalization of the application of this methodology to other areas and regions, so much so that the values found by the authors differ from other study proposed for Demarchi et al. [2011] with the same type of approach to other regions and dates.

Based on the aforementioned considerations, this paper proposes a novel approach based on multistage processing to locating and quantifying center pivot irrigation systems based on target detection using Circular Hough Transform (CHT) over images of widely used spectral vegetation indices: NDVI and SAVI. Multistage involves the application of post-processing steps for handling errors that increase the false positive cases during the process of detecting circles by CHT. The main problems are false pivots detection in Riparian woods areas, urban areas and vegetated fields not irrigated by pivots.

2. Material and Methods

2.1. Study Areas

The main goal of this study is to identify a method ready to use in multisource remote sensing images. For this reason, we have chosen the scene of CBERS 4-Multispectral Camera (MUX) at path/row 164/117, this image has an important indigenous reserve of the Xavánte's, Sangradouro-Volta Grande in Mato Grosso state, Brazil, surrounded by a natural forest reserve. The scene also has a large number of agricultural fields employing center pivots. The second area of study is the region of city Paracatu in Minas Gerais state, according ANA and Embrapa [2019], this is one of the three main irrigating municipalities of Brazil in 2017, together with Unaí, also in Minas Gerais state; and Cristalina in Goiás state. These three cities form the highest concentration of pivots in Brazil with 2558 units occupying 191 thousand hectares, for this second area the Landsat 8-Operational Land Imager (OLI) scene at path/row 220/072 was analyzed (Figure 2).



Scene 164/117 - color combination in R7G6B5 using CBERS 4-MUX bands

Detail of Paracatu-MG scene 220/072 - color combination in R4G3B2 using Landsat 8-OLI bands

Figure 2. Study areas using CBERS 4 and Landsat 8.

2.2. Data

The scene of Landsat 8-OLI covering Paracatu region was acquired for the year 2014 using the Google Earth Engine's (GEE) application programming interface, because this platform enables easy access to a product of surface reflectance from radiance measured by sensor, the process involves a detailed radiometric correction of solar energy scattered and reflected from the atmosphere and earth surface processed using algorithms supplied by U.S. Geological Survey. The GEE is a freely accessible, cloud-based platform designed to enable remote sensing studies over long time scales and large spatial extents [Gorelick et al. 2017]. The surface reflectance product from CBERS 4-MUX for the year 2017 was downloaded from the repository provided by Martins et al. [2018], which makes available the product of the images processed using Coupled Moderate Products for Atmospheric Correction (CMPAC) approach. The CMPAC uses atmospheric products from Moderate-Resolution Imaging Spectroradiometer (MODIS) and Visible Infrared Imaging Radiometer Suite (VIIRS) for atmospheric correction of CBERS 4-MUX level-4 images.

One of the post-processing stages involves the use of information over water bodies presence to eliminate false circles of pivots, to decrease the False Alarm Ratio (FAR), for this task we used Water Bodies for Brazil From RapidEye Images [Namikawa et al. 2016]. This mapping of water bodies has a pixel size of 5 m and use an approach based on the thresholding of the Hue component of the conversion of the color system from RGB to Hue-Saturation-Value (HSV), which has better results when compared to single-band thresholding or the use of Normalized Difference Water Index [Namikawa et al. 2016]. The results were organized in tiles with 125 km×125 km in eight bits image having values from 1 to 7 to indicate the confidence of being a water pixel, from 1 as more confident to 7 as the less confident. The tiles can be downloaded using project interface[2], for our work the tiles 451, 452, 482, 483, 508, 509, and 510 needed for covering the Landsat scene, and tiles 301, 302, and 340 for covering CBERS scene.

The detection of center pivots was developed using remote sensing data from the years 2014/2017 and the geospatial vector dataset of center pivots mapped from ANA[3] in collaboration with Brazilian Agricultural Research Corporation (Embrapa) for the same years to validate results. This mapping was performed through visual analysis of Landsat, Sentinel, and other satellite images [ANA and Embrapa 2019].

2.3. Circular Hough Transform (CHT)

Duda and Hart [1972], present an approach to improve line and curve detection on digital images derived from the idea of parameter space or Hough Space (HS) originally defined by the parametric representation used to describe lines in the picture plane using Hough Transform (HT) [Hough 1962]. According to the authors, the general approach of the Hough Method can be extended to detect circular configurations (CHT), using a parametric representation for the family of all circles inside a region determined with maximum distance in the relation to the origin of parametric space. This way each figure point will be transformed into a circular cone in a three-dimensional parameter space represented for a 3D accumulator array, where the number of intersections between cones surfaces (votes) determines the centroids of geometric circles [Duda and Hart 1972]. The CHT

[2] Water Bodies for Brazil From the RapidEye Images repository, provided by Namikawa et al. [2016]. Available online: `http://www.dpi.inpe.br/waterbodies/files/2014_2015_v3/download_wb_2015v3.php`. Accessed on August, 25, 2020.

[3] GeoNetwork is the online repository of geospatial data made by ANA. Please refer to this address `https://metadados.ana.gov.br/geonetwork/srv/pt/main.home?uuid=e2d38e3f-5e62-41ad-87ab-990490841073` to take access of geospatial vector dataset of center pivots. Accessed on January, 22, 2020.

is a simple feature extraction technique widely used in digital image processing for detecting circles in low-quality images, because of its robustness in the presence of noise, occlusion, and varying illumination [Dembele 2015].

We use the CHT method implemented in the OpenCV python library (The 2-1 Hough Transform - 21HT). It uses the gradient information of edges to decompose the circle finding problem into two stages, reducing the requirements of storage. The combining of CHT and vegetation indices, enable detect circular targets with the high response of vegetation indicating the health of the crop due to the use of center pivot irrigation systems.

2.4. Vegetation Indexes

Our approach used a pair of vegetation indexes to characterize vegetation density and enable detect crop fields irrigated of center pivot with high values of vegetation index. According Yin et al. [2012], NDVI can be explored for monitoring agricultural yield, using vegetation properties, such as length of the growing season, the onset date of greenness, and date of maximum photosynthetic activity. This way, Remote Sensing-based measurement is widely used to obtain phenological data in order to emphasize characteristics of terrestrial ecosystems to identify land cover and natural or anthropogenic changes.

Although the NDVI has been found highly correlated with vegetation parameters, some factors, such as atmospheric influences and soil substrate differences, impact the response obtained with him. Atmospheric turbidity generally inhibits reliable measures of vegetation and may delay the detection of the onset of stress in canopies such as green leaf area, biomass, percent green cover, productivity, and photosynthetic activity [Huete 1988]. For this reason, it is very important to combine NDVI with information from other indices, to analyze the behavior of vegetation response more carefully.

The SAVI was developed to enable correction to the response of partial energy reflected from soil surface exposed when the vegetative cover is low:

$$SAVI = \frac{NIR - RED}{NIR + RED + L} \times (1 + L), \tag{1}$$

where NIR is the reflectance value of the near infrared band, RED is reflectance of the red band, and L correction factor adjusts the original equation of NDVI to correct the soil brightness. The L value varies conform the presence of vegetation, very high ($L = 0$) to no vegetation ($L = 1$), but in most cases $L = 0.5$ is ideal to minimize soil brightness variations and eliminate the need for additional calibration for different soils [Huete 1988]. We adopted $L = 0.5$ to generate our SAVI images used to edge detection and extracting stats for detected circles.

2.5. Balanced Random Forest (BRF) For Pivot Identification

Our methodology described in Figure 3, shows the steps necessary to retrieve candidate circles of center pivot (x, y, radius), label the stats extracted from candidates circles, classify using BRF, and save pivots identified to compare with our knowledge database from ANA.

The major work consists of the following steps: (i) Apply the CHT technique over Canny edge detection results from vegetation index images (NDVI/SAVI) to identify

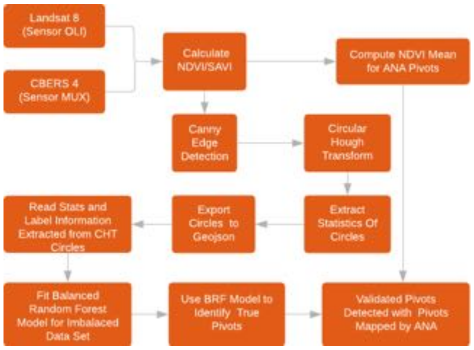

Landsat 8 (Sensor OLI)
CBERS 4 (Sensor MUX)
Calculate NDVI/SAVI
Compute NDVI Mean for ANA Pivots
Canny Edge Detection
Circular Hough Transform
Read Stats and Label Information Extracted from CHT Circles
Export Circles to Geojson
Extract Statistics Of Circles
Fit Balanced Random Forest Model for Imbalaced Data Set
Use BRF Model to Identify True Pivots
Validated Pivots Detected with Pivots Mapped by ANA

Figure 3. The framework for identification of center pivots using CHT and Balanced Random Forest Classification.

possible circles of the center pivots; (ii) Extract statistics from these circles, these samples of information was be labeled in pivot/not pivot through the spatial intersection of these circles with the pivots mapped by ANA; (iii) Fit a BRF model with labeled data to filter false alarm circles of pivots; (iv) Finally, validate filtered circles with ANA information.

Generally, pivot shape and vegetation index values vary significantly between different center pivot systems as a result of the variation of cultivation and irrigation. Our method is sensitive to the shape of the targets that are delimited using the Canny algorithm. Therefore, we decided to evaluate only those circles that had an average NDVI > 0.5. This made it possible to identify areas with the greatest photosynthetic activity, improving the delimitation of crop fields, and allowing better identification of center pivots.

2.5.1. Training Data

For each image processed, a set of circles is generated (Table 1). Each candidate circle is then examined to identify in its delimitation area the mean value and standard deviation of the vegetation indices, pixels of water bodies, edge pixels, and percentage difference of points (Figure 4). This information corresponds to the percentage difference between all points (edge pixels) included in the circle and points that form the convex hull of these points. This ratio is a good indicator of false detection of circles in cases where the analyzed object has many internal pixels.

Table 1. Candidates circles of pivots generated by CHT.

	NDVI	SAVI
Landsat 8	3856	11874
CBERS 4	1741	3083

All information generated during the analysis of circles builds a data set that after being labeled serves as input to train the classification model. As a result after classification, all the samples (circles) which received a positive indication of pivot are saved on geojson file to validate with pivots mapped by ANA.

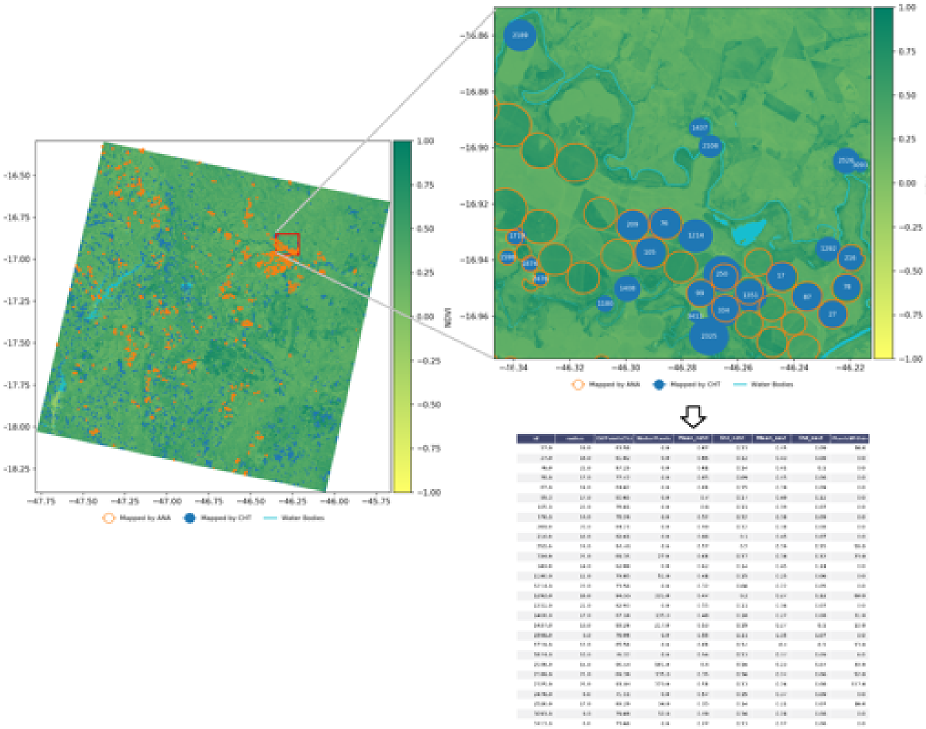


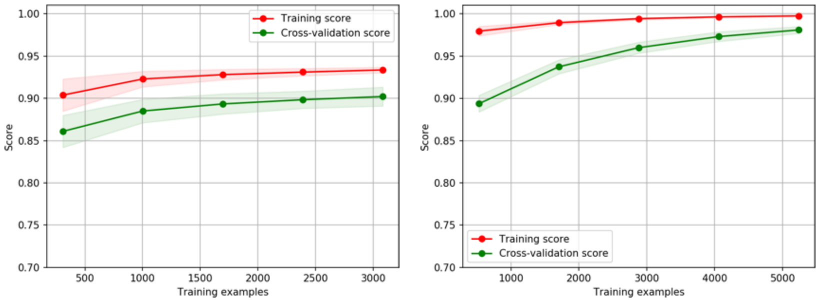
Figure 4. Illustration of the stats information extraction from circles detected by CHT.

2.5.2. Data Analysis

The data analysis and processing were carried out using Python programming language due to the facilities provided by their libraries and packages. For instance, the Pandas and GeoPandas libraries were used to manage stats information of circles identified in images through CHT and label this information through intersects of pivots mapped by ANA [GeoPandas Development Team 2019].

Another important package used was Scikit-learn [Pedregosa et al. 2011], which includes a useful set of methods for the construction and evaluation of ML methods, such as Random Forest. However, as the problem we are deal with is characterized by unbalanced classes, it was necessary to use techniques to care with this imbalance. Because the learning and prediction of ML models are usually affected by the imbalance problem of a data set [Lemaître et al. 2017]. The balancing issue corresponds to the difference of the number of samples in the different classes, for example, the samples generated in the processing of the Landsat SAVI image, only 850 (7.16%) corresponded to pivots, of which 444 had mean NDVI > 0.5.

We employed two techniques to handle the problem of unbalanced classes, Random Over Sampler (ROS) and a classifier including inner balancing samplers (BRF), both were implemented in the imbalanced-learn package [Lemaître et al. 2017]. ROS uses a naive strategy to generate new samples of data set by randomly sampling with replacement of the currently available samples (Upsampling). While BRF is an ensemble method in which each tree of the forest will provide a balanced bootstrap sample [Chen et al. 2004]. These methods combined increased considerably the accuracy of the model (Figure 5) and consequently the result of the classification and filtering of the pivots.



(a) Without Upsampling

(b) With Upsampling

Figure 5. Learning results for BRF varying the number of training samples.

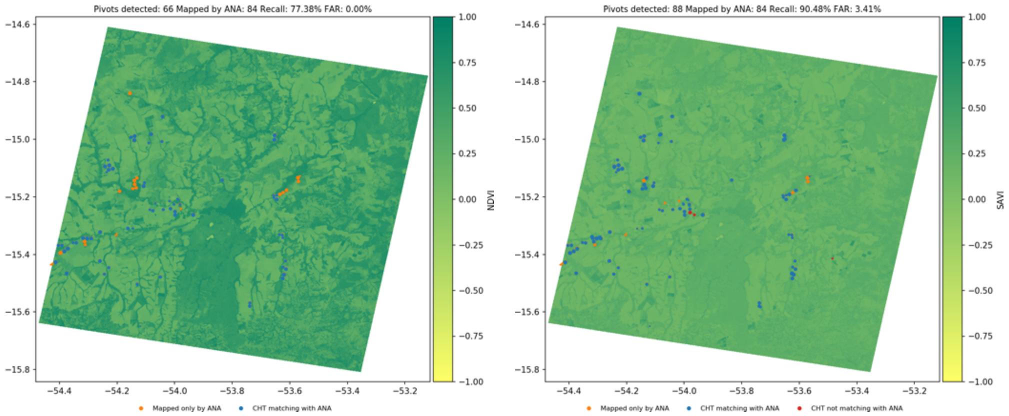
Processing was performed on a machine with 8 cores and 8 GB of memory. Training and predict with the BRF model took a few seconds because the data set is small (maximum 11874 samples). However, the iterative process for extracting statistics for each circle identified by the CHT method reached the time of 132 min. But using the multiprocessing module for Python which allowed us to reduce that time to 1/4 by executing the extraction function in multiple processes occupying all cores at the same time.

3. Results

As previously presented, the BRF model achieved high accuracy for classifying the pivots. In this manner, we have the maximum Recall limited by the CHT method and the FAR minimized by the classification provided by the BRF. Basically, Recall means how much of the total of pivots are in this area the method can be recover, and FAR how much of the total pivots identified by BRF not match with pivots mapped in this area by ANA. It is important to mention that the CHT method can generate the identification of overlapping circles (Overlapping Identifications) because there is a trade-off between the necessity to detect small and large circles of pivots (400 to 1000 m).

Figure 6 present the results achieved for the CBERS scene, the blue-dots represent pivots correctly identified by our approach, red-dots false alarm of pivots and orange-dots pivots not identified. This result, emphasizes that the method is optimal to filters circles of not pivot, but the capacity to recover all pivots depends on a variety of characteristics linked to the vegetative response of the targets that depend on agricultural production cycles.

To quantify and analyze the results on this scene, we compare the number of pivots mapped by ANA in this testing area. The Recall and FAR of identification were calculated with data in Table 2. They show that in this area there are 84 pivots mapped by ANA with (Mean NDVI > 0.5), while our approach was able to identify 66 pivots all matching with pivots of ANA, but exist one overlapped detection, this manner based on the NDVI image we achieved a Recall of 77.38% and FAR of 0%. When the image used is SAVI, we identified 88 pivots, but among them, three no match with ANA and



(a) NDVI

(b) SAVI

Figure 6. Blue-dots represent pivots correctly identified, red-dots false alarm of pivots and orange-dots pivots not identified. Pivots identification based on vegetation images of scene CBERS 4 path/row 164/117 in July 2017.

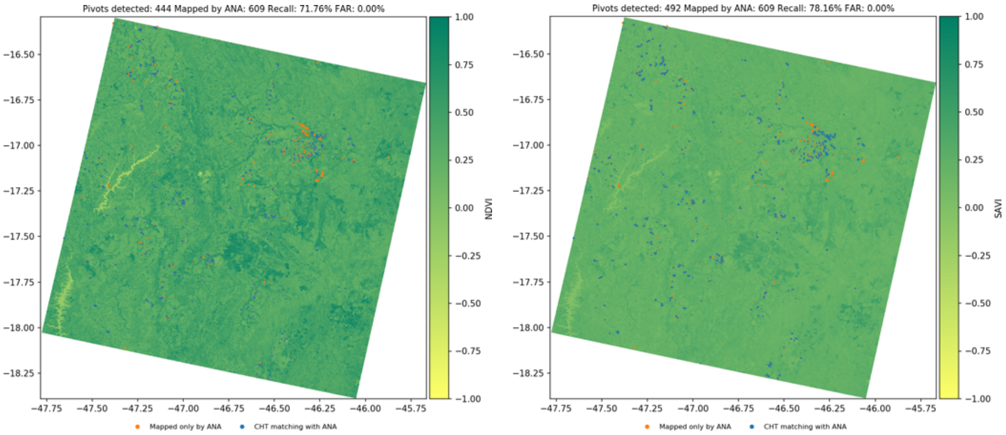
nine overlaps achieving a Recall of 90.48% and FAR of 3.41%. This proves that although the detection with the SAVI image generates a higher processing volume, more circles not corresponding to pivots, it allows a greater recall thus enhancing its advantage over NDVI, due to the brightness adjustment factor for exposed soil.

Table 2. Quantity analysis results for CBERS scene.

	Mapped by ANA	Pivots Identified	Overlapping Identifications	Match ANA	Not Match ANA
Count by NDVI	84	66	1	65	0
Count by SAVI	84	88	9	76	3

For further analysis of the Paracatu-MG region, we needed before process the scene of Landsat 8 path/row 220/072. For this reason, we decided to show the results achieved for both, the entire scene and clipping to the region (Figures 7 and 8). The parameters of the classifier are tuned for each type scene and satellite, we believe that this process should guarantee the best result taking into account the intrinsic differences between the spectral responses recorded by each sensor (satellite) and by each type of image (vegetation indexes) regardless of the chosen date and geographical position.

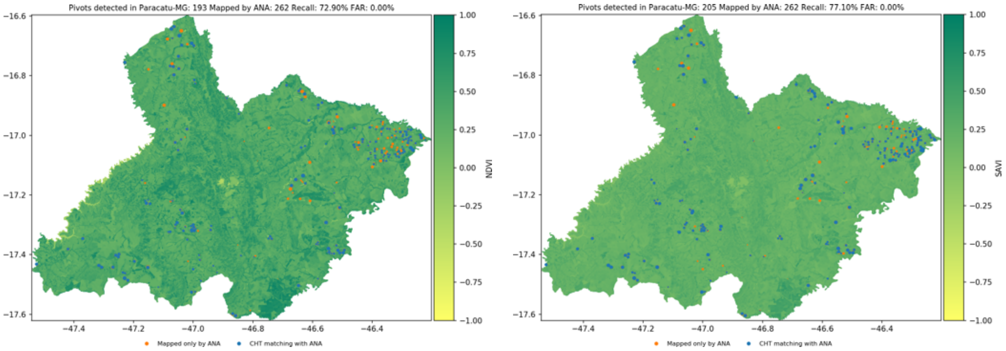
Table 3, compiles both the information for the entire scene and the clipping region. For this area, ANA mapped 609 pivots with (Mean NDVI > 0.5) for the entire scene and 262 for Paracatu-MG region, while our approach identified 444 and 193 pivots respectively all matching with pivots of ANA, but with seven/two overlaps. Therefore, based on NDVI image we achieved a Recall of 71.76% and 72.90% respectively, and FAR of 0% for both. When the image used is SAVI, we identified 492 pivots for the scene and 205 pivots for clipping all matching with pivots of ANA, but there are 16/3 overlaps, resulting in Recall of 78.16% and 77.10% and FAR of 0%. Once again the response with the SAVI image was better. Although for this scene the Recall had lower values than about the other satellite (CBERS), this is possibly linked to the condition of vegetation in this date and agricultural calendar of this region.


Pivots detected: 444 Mapped by ANA: 609 Recall: 71.76% FAR: 0.00%
NDVI
Pivots detected: 492 Mapped by ANA: 609 Recall: 78.16% FAR: 0.00%
SAVI
Mapped only by ANA
CHT matching with ANA

(a) NDVI

(b) SAVI

Figure 7. Pivots identification based on vegetation images of scene Landsat 8 path/row 220/072 in August 2014.


Pivots detected in Paracatu-MG: 193 Mapped by ANA: 262 Recall: 72.90% FAR: 0.00%
NDVI
Pivots detected in Paracatu-MG: 205 Mapped by ANA: 262 Recall: 77.10% FAR: 0.00%
SAVI
Mapped only by ANA
CHT matching with ANA

(a) NDVI

(b) SAVI

Figure 8. Pivots identification (Paracatu-MG region) based on vegetation images of scene Landsat 8 path/row 220/072 in August 2014.

4. Conclusion

The aim of this study was to automatically identify center pivots on multisource remote sensing images using image processing techniques and ML models. The proposed approach identified pivots successfully using the BRF model trained with different types of stats extracted of targets detected by CHT over scenes from Landsat 8 and CBERS 4. The high accuracy achieved shows the robustness of the method and data independence. The results were validated with mapping carried out by ANA, presenting less economic cost and less time than a visual analysis approach. Several factors can negatively influence our results, such as the highest presence of clouds and shadow. Besides that, urban areas and water bodies can result in high levels of false positives, due to a large number of edge pixels generated in these regions, but this problem was handled with success using the Random Over Upsampling technique and Balanced Random Forest classifier to filter not pivot circles in this areas. As future work, we plan to use a collection of a year images to identify the greenest pixels (maximum NDVI), this would allow improving the

Table 3. Quantity analysis results for Landsat scene an Paracatu-MG region.

	Mapped by ANA Scene/Clipping	Pivots Identified Scene/Clipping	Overlapping Identifications Scene/Clipping	Match ANA Scene/Clipping
Count by NDVI	609/262	444/193	7/2	444/193
Count by SAVI	609/262	492/205	16/3	492/205

***Attention** - this table does not have collumn ”Not Match ANA” because for both areas the FAR values are null.

delimitation of pivots by the Canny algorithm. In addition, we also intend to adopt an approach based on time series similar to that used by Rodrigues et al. [2020], who used them to adjust thresholds for identifying pivots in the MATOPIBA region. However, our goal is to use the time series classification obtained from the detected targets in order to characterize the region based on land use and crop cycles.

Acknowledgements

This study was supported in part by grants 2017/24086-2 and 2018/16221-0, São Paulo Research Foundation (FAPESP), the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and the Brazilian National Council for Scientific and Technological Development (CNPq, grant 303360/2019-4). Also, we thank the National Institute for Space Research and the subproject Brazil Data Cube, of the Environmental Monitoring Project for Brazilian Biomes, financed by the Amazon Fund, through the financial collaboration BNDES and FUNCATE nº 17.2.0536.1.

References

Aksoy, S., Yalniz, I. Z., and Tasdemir, K. (2012). Automatic detection and segmentation of orchards using very high resolution imagery. *IEEE Transactions on geoscience and remote sensing*, 50(8):3117–3131.

ANA (2017). Atlas irrigação: Uso da água na agricultura irrigada. Technical report, National Water Agency (ANA). arquivos.ana.gov.br/imprensa/publicacoes (10 May 2019).

ANA and Embrapa (2019). Levantamento da agricultura irrigada por pivôs centrais no Brasil (1985-2017). Technical report, National Water Agency (ANA) and Brazilian Agricultural Research Corporation (Embrapa) Maize/Sorghum. ana.gov.br (10 May 2019).

Britannica Escola Web (2020). Irrigação. In Britannica Escola. Web, escola.britannica.com.br/artigo/irrigação/481588 (18 October 2020).

Chen, C., Liaw, A., Breiman, L., et al. (2004). Using random forest to learn imbalanced data. Technical report, University of California, Berkeley. statistics.berkeley.edu (20 September 2020).

Demarchi, J. C., Piroli, E. L., and Zimback, C. R. L. (2011). Analise temporal do uso do solo e comparação entre os índices de vegetação ndvi e savi no município de santa cruz do rio pardo–sp usando imagens landsat-5. *Raega-O Espaço Geográfico em Análise*, 21.

Dembele, F. (2015). Object detection using Circular Hough Transform. Université Laval. wcours.gel.ulaval.ca/2015/a/GIF7002/default/5notes [Lectures supplémentaires C11:d] (25 January 2019).

Duda, R. O. and Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1).

GeoPandas Development Team (2019). GeoPandas Library, Version 0.5.1. GeoPandas developers Revision 4a4ede8b. geopandas.readthedocs.io/en/v0.5.1 (27 August 2019).

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., and Moore, R. (2017). Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*.

Hough, P. V. (1962). Method and means for recognizing complex patterns. US Patent 3,069,654.

Huete, A. (1988). A soil-adjusted vegetation index (savi). *Remote sensing of environment*, 25:295–309.

Lemaître, G., Nogueira, F., and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

Maranha, L. G. (2018). Mapeamento semiautomático de áreas irrigadas por pivôs centrais por meio de análise espacial orientada a objetos em imagem Landsat 8.

Martins, V. S., Soares, J. V., Novo, E. M., Barbosa, C. C., Pinto, C. T., Arcanjo, J. S., and Kaleita, A. (2018). Continental-scale surface reflectance product from cbers-4 mux data: Assessment of atmospheric correction method using coincident landsat observations. *Remote Sensing of Environment*, 218:55–68.

Namikawa, L. M., Körting, T. S., and Castejon, E. F. (2016). Water body extraction from rapideye images: An automated methodology based on hue component of color transformation from rgb to hsv model. *Revista Brasileira de Cartografia*, 68(6).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rodrigues, M. L., Körting, T. S., de Queiroz, G. R., Sales, C. P., and d. Silva, L. A. R. (2020). Detecting center pivots in Matopiba using hough transform and web time series service. In *2020 IEEE Latin American GRSS ISPRS Remote Sensing Conference (LAGIRS)*, pages 189–194.

Santos, J. É. O., Nicolete, D. A. P., Filgueiras, R., Leda, V. C., and Zimback, C. R. L. (2015). Imagens do Landsat-8 no mapeamento de superfícies em área irrigada. *Irriga, Edição Especial, IRRIGA & INOVAGRI*, pages 30–36.

Yin, H., Udelhoven, T., Fensholt, R., Pflugmacher, D., and Hostert, P. (2012). How Normalized Difference Vegetation Index (NDVI) Trends from Advanced Very High Resolution Radiometer (AVHRR) and Système Probatoire d'Observation de la Terre VEGETATION (SPOT VGT) Time Series Differ in Agricultural Areas: An Inner Mongolian Case Study. *Remote Sensing*, 4(11):3364–3389.

Zhang, C., Yue, P., Di, L., and Wu, Z. (2018). Automatic identification of center pivot irrigation systems from landsat images using convolutional neural networks. *Agriculture*, 8(10):147.

SOLAP Query Processing over IoT Networks in Smart Cities: A Novel Architecture

João Paulo Clarindo dos Santos[1], **João Pedro de Carvalho Castro**[1,2], **Cristina Dutra de Aguiar Ciferri**[1]

[1]Institute of Mathematics and Computer Science – University of São Paulo – Brazil

[2]Computing Center – Federal University of Minas Gerais – Brazil

jpcsantos@usp.br, jpcarvalhocastro@ufmg.br, cdac@icmc.usp.br

Abstract. *Spatial data generated by an Internet of Things (IoT) network is important to assist the decision-making in issues related to smart cities. In these cities, IoT devices generate spatial data constantly. Thus, data get increasingly voluminous very fast. In this paper, we investigate the challenge of managing these data through the use of a spatial data warehouse and spatial on-line analytical processing designed over a parallel and distributed processing framework extended with a spatial analytics system. We propose a novel architecture aimed to assist a smart city manager in decision-making, which integrates a cloud layer where these technologies are located with a fog computing layer for extracting, transforming and loading. Furthermore, we introduce a set of guidelines to aid smart cities managers to implement the proposed architecture. We validate our architecture with a case study that uses real data collected by IoT devices in a smart city.*

1. Introduction

In the last few years, the world population has been growing rapidly. From projections made by the United Nations, the population will reach 8 billion people in 2025 [Fraga and Queirolo 2018]. Hence, providing the necessary infrastructure to accommodate a significant amount of people in cities can be a challenge for public authorities and companies. According to Ramaswami et al. (2016), the meta-principles for developing a sustainable and healthy city are "*improvements in transportation, basic sanitation and energy supply*", "'*sustainability*", and "*technology integration*". Thus, the concept of smart cities emerged. It "*involves the implementation and deployment of information and communication technology (ICT) infrastructures to support social and urban growth through improving the economy, citizens involvement, and government efficiency*" [Yeh 2017].

A network of Internet of Things (IoT) devices can be used to provide information in a smart city. According to Patel and Patel (2016), IoT can be classified as "*interconnected objects that have data regularly collected, analysed, and used to initiate action, providing a wealth of intelligence for planning, management and decision-making*". The different layers of an IoT architecture include: (i) the *smart device/sensor layer*, which is responsible for collecting data from the environment through the employment of connection standards such as Wi-Fi, GSM and Bluetooth; (ii) the *network layer*, which is composed of gateways and gateway networks that support different communication protocols for sending data to the *service layer*; and (iii) the service layer, in which data is processed and prepared to obtain the information required by a desired application [Patel and Patel 2016, Atzori et al. 2017].

The IoT technology is very important in a smart city environment. For instance, it is possible to apply this paradigm in a public transportation system, whose fleet contains sensors that collect data related to the number of passengers, vehicle type (buses, trams, etc.), route taken, and maximum speed, aiming to improve the existing lines. Another IoT application scenario includes air monitoring, with sensors scattered around the city collecting data about pollution in order to identify if air quality improvements are necessary in certain regions [Atzori et al. 2017].

An IoT network contained in a smart city tends to generate spatial data, usually represented by geometries (such as points, lines, and polygons) or combinations of these. For example, smartphones can contain sensors that use location data to connect people with the same hobbies or relationship status living in the same area. The spatial properties of IoT devices can be determined directly by the sensors, using satellite positioning techniques, like GPS and GLONASS [van der Zee and Scholten 2014, Eldrandaly et al. 2019].

Performing analytical queries on data generated by an IoT network in a smart city can assist managers in the decision-making process. For instance, a manager of a public transportation system can be interested in determining *how many passengers were transported last month, considering the type of vehicle, route, and region*. The query results can be displayed on a map according to the region, helping the manager to obtain the necessary knowledge in an intuitive manner. In order to enable the execution of this type of query, IoT data needs to be extracted, transformed, and loaded in a spatial data warehouse (SDW). An SDW is a subject-oriented, integrated, time-variant and non-volatile collection of conventional and spatial. It provides support for the costly spatial on-line analytical processing (SOLAP) queries, which are analytical queries extended with spatial predicates [Han et al. 1998, Rivest et al. 2001].

In smart cities, IoT devices generate spatial data constantly [Bonomi et al. 2014]. Also, because sensors all over the city can collect and transmit masses of data, data scale becomes increasingly big [Chen et al. 2014]. To deal with big data, the management of SDWs can benefit from the use of a cloud computing environment as infrastructure and from the employment of parallel and distributed processing frameworks, such as Hadoop [Shvachko et al. 2010] and Hadoop Spark [Zaharia et al. 2016], to reduce the complexity of the cloud. The processing of the SOLAP queries can also benefit from the use of spatial analytics systems (SASs), which are developed on the top of parallel and distributed processing frameworks to provide extended functionalities to deal with spatial data [Castro et al. 2020].

The challenge is to propose an IoT architecture for smart cities that encompasses all these technologies and also provides efficient support for storing SDWs and processing SOLAP queries. Although there are some proposals of architecture proposed in the literature [Yuan and Zhao 2012, Bonomi et al. 2014, Eldrandaly et al. 2019], they do not focus on SDWs and SOLAP in a parallel and distributed processing environment. In this paper, we overcome these shortcomings.

The contributions of our paper are described as follows.

- The proposal of an architecture aimed to help smart cities managers and residents in their decision-making process through the employment of an SDW that uses a parallel and distributed data processing framework in the cloud and also uses SASs to process SOLAP queries.
- The definitions of guidelines to assist in the proposed architecture implementation.
- The validation of the efficacy and effectiveness of the architecture with a case study that describes an application that handles real data generated from a smart city.

This paper is organized as follows. Section 2 reviews related work, Section 3 presents the proposed architecture, Section 4 introduces the guidelines for implementing the architecture, Section 5 describes the case study, and Section 6 concludes the paper.

2. Related Work

There are studies in the literature that present challenges related to IoT-generated data, considering general [Patel and Patel 2016, Atzori et al. 2017] and smart city scenarios [Arasteh et al. 2016, van der Zee and Scholten 2014, Theodoridis et al. 2013]. In the context of big data, some work has been done to manipulate IoT spatial data. These proposals are described as follows.

Yuan and Zhao (2012) propose an architectural solution for SDWs in the context of IoT environments (SDWIT). This architecture has the following layers: data processing layer, storage layer, and analysis application layer. SDWIT features include accessing and analysing IoT data in real time over a traditional SDW. However, the authors did not consider parallel and distributed data processing frameworks in their architecture, making SOLAP operations difficult for very large SDWs.

Bonomi et al. (2012) introduce a highly virtualized platform called *fog computing*, which "*provides computing, storage and networking services between end devices and traditional cloud computing data centres, typically, but not exclusively located at the edge of network*". Fog computing aims at low latency between the edge and the core of the network, very large number of nodes, and wide-spread geographical distribution. Therefore, the platform is appropriate for operations that have IoT services. The fog computing platform is expanded in [Bonomi et al. 2014] to deal with a massively distributed number of sources at the edge. Regarding applications that require analytics over longer periods or wider scenarios, like an SDW, these proposals only suggest that the corresponding operations should be performed in the cloud. No further investigation is conducted.

Eldrandaly et al. (2019) define the concept of Internet of Spatial Things (IoST), which "*is an integrative paradigm of embedded smart devices concerned with collecting spatial data of objects to serve a significant purpose*". The authors also introduce a framework for an IoST network that uses a fog computing platform for real-time spatial computing. Data are collected and then sent to a fog node for temporary storage and processing. Finally, data are extracted, transformed and loaded into a cloud database. Although the framework provides analytics operations that are defined according to the requirements of the enterprise, these operations do not focus on the processing of queries extended with spatial predicates over SDWs.

In contrast to the described approaches, we propose a novel architecture that employs both parallel and distributed data processing frameworks and SASs, allowing the execution of fast and reliable analyses over IoT data from smart cities. Our architecture excels not only in the integration of these technologies with an SDW in the cloud, but also in the inclusion of a fog layer to handle SOLAP data analytics and SDW Extract, Transform, and Load (ETL) processes.

We also introduce a set of guidelines to aid smart cities managers in the process of implementing our architecture. Further, we validate the architecture with a case study that describes an application that handles real data generated from a smart city. The aim of this case study is to investigate the efficacy and effectiveness of the architecture, but not its efficiency. Thus, carrying out performance evaluations is out of the scope of this paper.

3. The Proposed Architecture

In this section we describe a novel architecture for collecting and analysing, in a fast and reliable manner, data from IoT devices in smart cities. The architecture (Figure 1) achieves these goals through the employment of three different layers: (i) the terminal layer; (ii) the fog layer; and (iii) the cloud layer. We discuss each layer as follows.

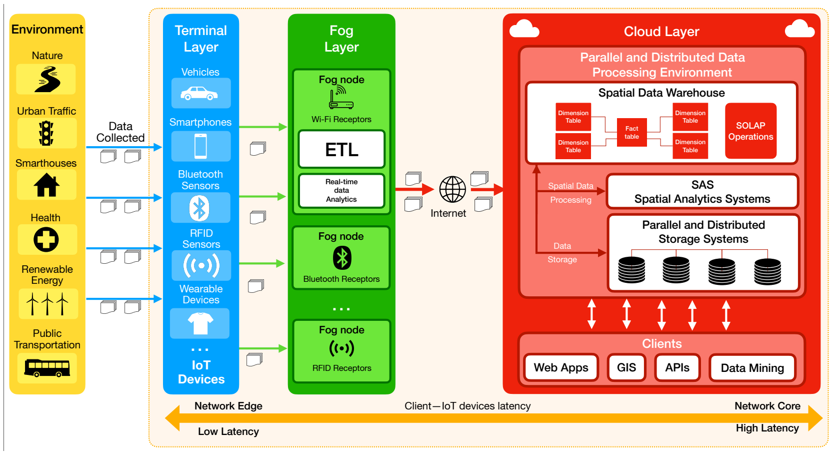


Figure 1. Architecture overview.

Terminal layer. The terminal layer consists of a network of IoT devices, which are interconnected by using technologies such as Radio Frequency Identification (RFID), Global Positioning System (GPS), Wireless Sensor Network (WSN), and network communication standards, such as Ethernet and Bluetooth. These devices are available in many parts of a smart city, such as weather stations, traffic lights, and public transportation. The devices are aimed to collect spatial and conventional data.

Fog layer. Data collected by terminal layer devices are sent to receivers in the fog layer. These receivers, called fog nodes, can be limited with regard to data processing and storage. However, by being located close to the network edge, they are in an optimal position to allow the execution of real-time data analytics and ETL operations. This is due to the low latency in communication between the terminal layer devices and these nodes.

Cloud layer. After the data goes through the ELT/ELT process in the fog layer, it is sent to the cloud layer. In this layer, data are persisted in an SDW stored in a parallel and distributed storage system. This allows SOLAP queries to be processed with the help of a SAS, enhancing their performance considerably. Due to the scalable nature inherent to cloud computing environments, the number of nodes can increase or decrease according to the demand of queries from clients. Examples of clients include web applications, Geographic Information Systems (GIS), and different types of Application Programming Interfaces (APIs).

4. Guidelines for Implementing the Proposed Architecture

In this section, we propose a set of guidelines to aid smart cities managers in the process of implementing the proposed architecture. Because the context behind each smart city may be different, is not mandatory to follow every guideline in its completeness. Managers should choose the appropriated hint provided by the guidelines according to the specific characteristics of the smart city in which the architecture is being employed. Thus, a concise yet general description of each guideline is provided, allowing further specialization based on the requirements imposed by each smart city application.

Guideline 1. Deploying IoT devices on the terminal layer. IoT devices must be deployed in the terminal layer considering the communication protocols supported by each sensor. For instance, vehicle sensors can use GPS, while temperature sensors can use 4G/5G protocols. A smart city manager must also consider the communication compatibility between these devices and the fog nodes. We recommend the framework proposed by [Theodoridis et al. 2013] to assist these managers in the process of integrating these devices in a smart city scenario.

Guideline 2. Distributing fog nodes across the fog layer. After the disposition of the IoT devices in the terminal layer, a smart city manager must define which devices must be used as fog nodes. For instance, some approaches in the literature use Raspberry Pi computers[1], which are small single-boarded computers, as fog nodes, using containerization over these resource-limited devices [Bellavista and Zanni 2017, Xu and Zhang 2019]. Each fog node uses the Docker container technology[2] for creating containers for each application available in fog node (i.e. ETL and real-time data analytics). Because Raspberry Pi computers are low cost and support many communication protocols, they are a viable choice to the heterogeneous nature of an IoT network. Communication between the fog nodes and the cloud layer can be carried out using 4G/5G or Wi-Fi protocols.

Guideline 3. Securing the connection between IoT devices and fog nodes. A smart city manager must be concerned with the dataflow between the IoT devices and the fog nodes, as sensitive information may be transmitted. Malicious attacks in a fog computing environment must also be considered. To deal with these issues, smart cities managers can take decisions using as a basis the work of [Mukherjee et al. 2017]. In this work, the authors determine the impact of security problems on a fog network and also provide solutions to increase the security of these environments.

[1] `https://www.raspberrypi.org/`

[2] `https://www.docker.com/`

Guideline 4. Configuring the ETL process in the fog layer. To enable ETL processing in the fog layer, a smart city manager must select tools that allow programming, scaling and monitoring the tasks of the ETL workflow. There are several tools on the market which support ETL and workflow monitoring. An example is Apache Airflow[3], which is an open-source platform that uses directed acyclic graphs (DAGs) for authoring, scheduling and monitoring workflows. The tasks of the process should be written in the Python programming language, since it is natively supported by Airflow. Airflow provides integration with Hadoop, Spark and several cloud platforms.

Guideline 5. Enabling real-time data analytics in the fog layer. In a fog node, data are loaded constantly, enabling real-time data analytics. To this end, a smart city manager can use multiple data management systems, like NoSQL databases (i.e. Couchbase Server[4] and Apache Cassandra[5]) and event streaming platforms such as Apache Kafka[6], operated in containers inserted in the fog node. These platforms support communication by APIs, enabling real-time spatial data analytics over SDWs.

Guideline 6. Choosing the appropriate SAS to implement the SDW in the cloud layer. The SDW application should process SOLAP queries efficiently. Therefore, a smart city manager must select a SAS that is able to completely fulfill the requirements of the SDW application. Because there several SASs available in the literature with different characteristics and capabilities, the choice of the most appropriate SAS burdens the selection process considerably. Managers should use as a basis of choice the state-of-the-art user-centric comparison of existing SASs described in [Castro et al. 2020].

Guideline 7. Configuring the SDW to process SOLAP queries on the cloud layer. After choosing the appropriate SAS, a smart city manager must configure the SDW environment in the cloud layer to process SOLAP queries. The parallel and distributed processing framework and the distributed file system must be compatible with the chosen SAS. There are several platform-as-a-service (PaaS) on the market that support these frameworks natively, such as Microsoft Azure[7] and Amazon Web Services[8]. The smart city manager must consider the periodicity that data should be extracted from the fog layer, as well as carefully specify data distribution over the SDW. The SOLAP services must support APIs and GIS applications in order to visualize the result of SOLAP queries.

Guideline 8. Ensuring secure SOLAP query processing in the cloud layer. Since cloud computing environments can be virtually accessed from anywhere, smart cities managers should be concerned with security issues related to SDW applications. That is, smart cities managers should define restrict protocols with regard to roles, permissions, and data confidentiality. A solution to ensure confidentiality is the encryption of the data stored in the SDW. Data encryption should be done carefully to not compromise the performance of the SDW application. To this end, the encryption methodology proposed in [Lopes et al. 2014] can be employed, as it allows the efficient processing of analytical queries over encrypted data warehouses.

[3] http://airflow.apache.org/

[4] https://www.couchbase.com

[5] http://cassandra.apache.org

[6] http://kafka.apache.org

[7] http://azure.microsoft.com

[8] http://aws.amazon.com

5. Case Study

In this section, we describe a case study that illustrates the use of the proposed architecture. We define the requirements of a spatial application, whose objective is to process data collected from multiple IoT devices in the context of smart cities. For this case study, we use a dataset provided by [Ali et al. 2015], which contains vehicle traffic data observed between two points. These data were collected from sensors distributed in the municipality of Aarhus, Denmark. The dataset, which is publicly available in the authors' website[9], contains both conventional (i.e., distance in meters between the sensors, type of road, etc.) and spatial data (i.e., the sensors locations, represented by points) referring to the period from February to June 2014. As this dataset only provides data regarding the sensor location (i.e., points), we extended it with new information to enrich the analyses performed in our spatial application. To this end, we use road (i.e., lines) and city (i.e., polygons) data obtained by Geofabrik[10] from OpenStreetMaps, and statistical district data obtained from OpenDataDK[11]. We guarantee the spatial relationship between the data in the sense that a road intersects with sensors, a district contains multiple roads, and a city contains several districts.

The requirements imposed by the SDW application are described as follows. The application should be deployed in the cloud and should communicate with a SAS to process its queries. Furthermore, data handled by the application should be stored in an SDW designed according to the logical schema depicted in Figure 2, which should also be located in the cloud. There are six dimension tables in the SDW: (i) Date and Time, storing the moment in which a measurement occurred; (ii) Report, storing the distance between the two sensors that performed the measurement and their geographic locations; and (iii) Road, District and City, storing the geographic locations associated with the report. The dimension tables are linked through the fact table Measurement, which stores both the measurement time and the vehicle speed. The fact table also stores the vehicle count for each measurement of the IoT sensors.

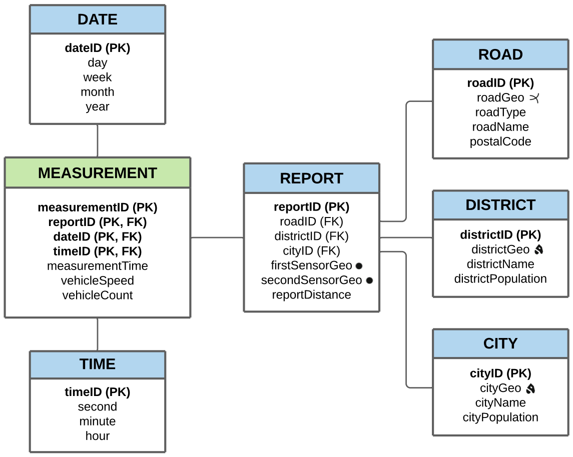


Figure 2. Logical schema of the SDW stored in the cloud.

[9] `http://iot.ee.surrey.ac.uk:8080/`

[10] `https://www.geofabrik.de/`

[11] `https://www.opendata.dk/city-of-aarhus/statistikdistrikter`

Another requirement of the application is that it should support different types of spatial queries based on the definitions of [Gaede and Günther 1998], such as spatial join, containment and k-nearest neighbour queries. The application should also provide good performance results. Finally, it is important to highlight that the developers who are going to implement the application have some previous knowledge of the SQL programming language.

According to the proposed architecture (Section 3) and guidelines (Section 4), the case study application should be implemented as follows: (i) use of the Apache Airflow to perform the ETL process; (ii) storage of the SDW data in the HDFS as the application requires data storage in the cloud; and (iii) selection of GeoSpark [Yu et al. 2019] as the SAS to process the SOLAP queries, as it complies with the application's requirements regarding performance and spatial queries, as well as supports the SQL programming language through the use of GeoSparkSQL [Pandey et al. 2018, Castro et al. 2020].

Data loading into the cloud layer. The dataset used in this case study consists of 449 reports. Data from these reports are stored in comma-separated values (CSV) files. To be loaded into the SDW, the data must go through an ETL process in the fog layer (Guideline 4). Thus, Apache Airflow should be employed to: (i) extract the data from the CSV files; (ii) perform transformations to arrange the data according to the logical schema depicted in Figure 2; and (iii) load the data into HDFS for later use by GeoSpark. To accurately simulate the fog layer, Airflow should be executed from a Docker container.

Converting textual representations of spatial data into spatial objects. In order to employ GeoSparkSQL for processing SOLAP queries over the SDW stored in HDFS, it is necessary to load its tables into structures called DataFrames. These structures, which resemble relational tables, do not transform the textual representations of the spatial data into spatial objects by default. GeoSparkSQL provides a function to convert well-known text (WKT) representations into spatial objects. An example of using this function during the process of loading the Report table is detailed in the following query:

```
SELECT reportID, roadID, districtID, cityID, reportDistance,
       ST_GeomFromWKT(firstSensorGeo) AS firstSensorGeo,
       ST_GeomFromWKT(secondSensorGeo) AS secondSensorGeo
FROM sensor
```

Once the process of loading the data provided by the IoT sensors into the SDW is complete, smart cities managers are able to execute different types of SOLAP queries using GeoSparkSQL. Some query examples that address key points of the application's requirements are defined as follows. We employ QGIS[12] to visualize the query results.

Spatial Join Query. This query returns the districts in which the average vehicle speed reported from the set of sensors that intercept it is greater than 60 km/h (37.28 mph). This analysis is necessary to check if there are districts where the maximum permitted speed is not being respected by the drivers. The query results are depicted in Figure 3, with each selected district being highlighted in red and the average vehicle speed (in km/h) displayed in its centre. The following command expresses this query:

[12] https://qgis.org/

```
SELECT districtGeo, AVG(vehicleSpeed) AS a
FROM measurement, report, district
WHERE ST_Intersects(ST_MakeLine(firstSensorGeo, secondSensorGeo),
                    districtGeo)
AND measurement.reportID = report.reportID
AND report.districtID = district.districtID
GROUP BY districtGeo
HAVING a >= 60
```

The analysis of the query results indicate that the average vehicle speed is higher in the northern districts of the municipality of Aarhus. A smart cities' manager can extract different types of knowledge from this information. An example is the fact that drivers can be less inclined to drive over the speed limit in central areas of the municipality (highlighted in purple in Figure 3), probably due to the increased number of pedestrians in these areas. Another example resides in the assumption that the average speed in the northern districts is higher due to the fact that some of them connect with external highways (displayed as pink lines in Figure 3).

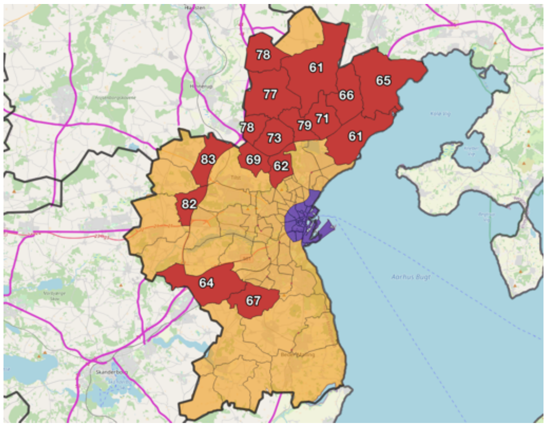


Figure 3. Spatial join query results.

Containment query. This query returns the quantity of vehicles that travelled in Aarhus University/Community Hospital district grouped by day. An interesting knowledge that can be obtained from this type of analysis is to identify the days in which the district had the largest number of vehicles and to investigate whether a holiday or an event happened, as displayed in Figure 4. The following command expresses this query:

```
SELECT day, SUM(vehicleCount)
FROM measurement, report, district, road
WHERE ST_Contains(districtGeo, roadGeo)
AND ST_Intersects(roadGeo,
                  ST_MakeLine(firstSensorGeo, secondSensorGeo))
AND measurement.reportID = report.reportID
AND report.districtID = district.districtID
AND report.roadID = road.roadID
AND district.name = 'Universitetet/Kommunehospitalet'
GROUP BY day
ORDER BY day
```

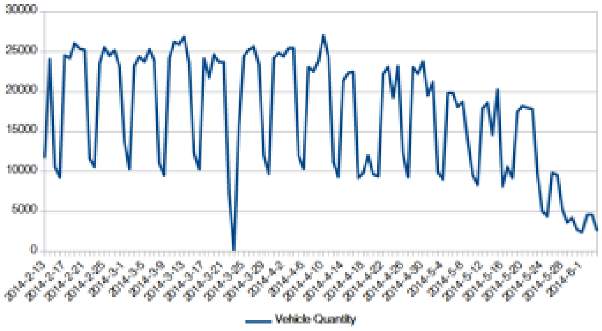


Figure 4. Containment query results.

By interpreting the query results, a smart cities manager can obtain different types of knowledge. For instance, the measurement of zero vehicles in 2014-3-25 could indicate that this was a day in which the sensors in the designated district were entirely disabled. Another interesting knowledge that can be obtained is that the traffic in this district seems more intense in weekdays when compared to weekends.

K-nearest neighbours query. This query returns the average vehicle speed identified by the 10 nearest reports from the Aarhus Cathedral, which is represented by a point (10.210556, 56.156944). This type of analysis is necessary to verify if drivers are respecting the speed limit in the surrounding area of a highly accessed point of interest, as shown in Figure 5. The following command expresses this query:

```
SELECT AVG(vehicleSpeed),
       ST_MakeLine(firstSensorGeo, secondSensorGeo) AS reportGeo
FROM measurement, report
WHERE measurement.reportID = report.reportID
GROUP BY reportGeo
ORDER BY ST_Distance(reportGeo,
      ST_GeomFromWKT('POINT(10.210556 56.156944)'))
LIMIT 10
```

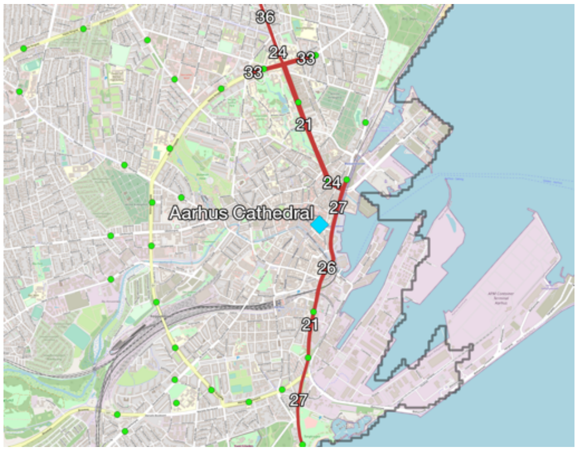


Figure 5. K-nearest neighbours query results.

The results displayed in Figure 5 can enable smart cities managers to perform a wide variety of analyses. In particular, one can identify that the highest average speeds around Aarhus Cathedral can often be observed in the main streets of its district, which are highlighted in red. Smart cities managers can also observe that these speeds do not go over 33 km/h. This can indicate that drivers do not tend to speed up in this region, a fact that might indicate the occurrence of heavy traffic.

6. Conclusions and Future Work

In this paper, we propose a novel architecture for enabling the execution of fast and reliable analyses over IoT data from smart cities. The architecture employs parallel and distributed data processing frameworks and spatial analytics systems in a cloud computing environment. Besides this cloud layer, our architecture also includes a fog layer, responsible for handling both real time data analytics and ETL processes; and a terminal layer, where the IoT devices are located. Further, we introduce a set of guidelines in order to aid smart cities managers in the process of implementing our architecture. Finally, we validate the proposed architecture by employing it to implement an SDW application that analyses data collected from real IoT devices in a smart city.

Future work includes describing additional case studies with sensors that collect measurements from different contexts, such as temperature and pollution levels. Another future work consists in the proposal of algorithms to optimize SOLAP query processing using as a basis the components of the proposed architecture.

Acknowlegments

This work was supported by Brazilian National Council for Scientific and Technological Development (CNPq) and by the São Paulo Research Foundation (FAPESP). C.D.A. Ciferri has been supported by the grant #2018/22277-8, FAPESP.

References

Ali, M. I., Gao, F., and Mileo, A. (2015). CityBench: A configurable benchmark to evaluate RSP engines using smart city datasets. In *LNCS*, volume 9367, pages 374–389.

Arasteh, H., Hosseinnezhad, V., Loia, V., Tommasetti, A., Troisi, O., Shafie-khah, M., and Siano, P. (2016). Iot-based smart cities: A survey. In *2016 IEEE 16th EEEIC*, pages 1–6. IEEE.

Atzori, L., Iera, A., and Morabito, G. (2017). Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56:122–140.

Bellavista, P. and Zanni, A. (2017). Feasibility of fog computing deployment based on docker containerization over RaspberryPi. In *ACM ICPS*, pages 1–10. ACM.

Bonomi, F., Milito, R., Natarajan, P., and Zhu, J. (2014). Fog computing: A platform for internet of things and analytics. *Studies in Computational Intelligence*, 546:169–186.

Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the MCC '12*, page 13.

Castro, J. P. C., Carniel, A. C., and Ciferri, C. D. A. (2020). Analyzing spatial analytics systems based on Hadoop and Spark: A user perspective. *Software: Practice and Experience*.

Chen, M., Mao, S., and Liu, Y. (2014). Big data: A survey. *Mobile Netw Appl.*

Eldrandaly, K. A., Abdel-Basset, M., and Shawky, L. A. (2019). Internet of Spatial Things: A New Reference Model With Insight Analysis. *IEEE Access*, 7:19653–19669.

Fraga, E. and Queirolo, G. (2018). Crescimento populacional fará mundo mudar de cara até 2100. `https://folha.com/ne67804j`. [Online; access sep. 20].

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

Han, J., Stefanovic, N., and Koperski, K. (1998). Selective materialization: An efficient method for spatial data cube construction. In *LNCS*, volume 1394, pages 144–158.

Lopes, C. C., Times, V. C., Matwin, S., Ciferri, R. R., and Ciferri, C. D. A. (2014). Processing olap queries over an encrypted data warehouse stored in the cloud. In *16th DaWaK*, pages 195–207. Springer.

Mukherjee, M., Matam, R., Shu, L., Maglaras, L., Ferrag, M. A., Choudhury, N., and Kumar, V. (2017). Security and Privacy in Fog Computing: Challenges. *IEEE Access*, 5:19293–19304.

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

Patel, K. K. and Patel, S. M. (2016). Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. *IJSR*, 6122.

Ramaswami, A., Russell, A. G., Culligan, P. J., Sharma, K. R., and Kumar, E. (2016). Meta-principles for developing smart, sustainable, and healthy cities. *Science (New York, N.Y.)*, 352(6288):940–3.

Rivest, S., Bédard, Y., and Marchand, P. (2001). Toward better support for spatial decision making: defining the characteristics of Spatial On-Line Analytical Processing (SOLAP). *Geomatica*, 55(4):539–555.

Shvachko, K., Kuang, H., Radia, S., and Chansler, R. (2010). The Hadoop Distributed File System. In *2010 IEEE 26th MSST*, pages 1–10.

Theodoridis, E., Mylonas, G., and Chatzigiannakis, I. (2013). Developing an IoT Smart City framework. In *4th IISA*, pages 180–185.

van der Zee, E. and Scholten, H. (2014). Spatial dimensions of big data: Application of geographical concepts and spatial technology to the internet of things. *SCI*, 546:137–168.

Xu, Q. and Zhang, J. (2019). PiFogBed: A Fog Computing Testbed Based on Raspberry Pi. In *2019 IEEE IPCCC*. Institute of Electrical and Electronics Engineers Inc.

Yeh, H. (2017). The effects of successful ICT-based smart city services: From citizens' perspectives. *Government Information Quarterly*, 34(3):556–565.

Yu, J., Zhang, Z., and Sarwat, M. (2019). Spatial data management in apache spark: the geospark perspective and beyond. *GeoInformatica*, 23(1):37–78.

Yuan, L. and Zhao, J. (2012). Construction of the system framework of Spatial Data Warehouse in Internet of Things environments. In *5th IEEE ICACI*, pages 54–58.

Zaharia, M., Franklin, M. J., Ghodsi, A., Gonzalez, J., Shenker, S., Stoica, I., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., and Venkataraman, S. (2016). Apache Spark. *Communications of the ACM*, 59(11):56–65.

Spatiotemporal disease tracking through open unstructured data and GIS

Luiz H. A. Cardim[1], **Nádia P. Kozievitch**[1]

[1]Departamento de Informática - Universidade Tecnológica Federal do Paraná (UTFPR)

luizcardim@alunos.utfpr.edu.br, nadiap@utfpr.edu.br

Abstract. *Automated disease tracking has become an increasingly important tool today. This article describes the prototype of a disease tracking system for the Brazilian territory. This study aims to extract relevant information in the health segment from unstructured data, extracted from news portals. The system should generate data that allows analysis at different levels of granularity, from small municipalities to the national level. The results of the study demonstrated the viability of the system and allowed the authors to identify some patterns in the processed data.*

1. Introduction

The dynamics of the current world, where millions (or even billions) of people move every day between different neighborhoods, cities, countries, or even continents, created the need to think about equally dynamic ways of monitoring some types of information, such as the spread of communicable diseases. COVID-19 showed us that a virus can, in a matter of months, and starting from some local cases, quickly turn into a global pandemic [World Health Organization 2020]. The impact of this pandemic has even changed the way data is shared [RDA COVID-19 Working Groups 2020], trying to make data sharing simpler. We add to this highly dynamic scenario of human mobility the growing urban cluster that has occurred in recent years, creating cities that are increasingly larger and with greater population density [United Nations 2019]. In this scenario, the delay in identifying the outbreak of a new disease can generate disastrous conditions, putting human lives at risk.

The internet offers us a very rich source of information about the occurrence of diseases in certain regions, like open data from cities[1], open data from public-private partnerships[2], open data from researchers[3], open unstructured data (like news portals), and others, however, its immense volume and heterogeneity of data makes the task of synthesizing all this information manually very costly or even impractical in the case of large geographic spaces or very long periods. In this way, several types of research have been carried out to automate this collection and synthesis of data into relevant information, making it possible to geographically monitor outbreaks at a global level and for extended periods.

However, despite all efforts in the segment, the Brazilian territory still lacks adequate tools for such finality, since the diseases relevant to Brazil can be different from

[1] As example we cite the Curitiba Open Data Portal (`https://www.curitiba.pr.gov.br/dadosabertos/`) used in this study

[2] `https://repositoriodatasharingfapesp.uspdigital.usp.br/`

[3] `https://dataverse.harvard.edu/`

those of other countries [Kindhauser, Mary Kay and World Health Organization 2003] and, in addition, most disease tracking platforms are based primarily on the English language. We also emphasize that, by limiting the tracking of diseases to the territory of only one country, it is also possible to achieve a greater degree of data granularity, also covering small and medium-sized municipalities.

This study presents an alternative to fill this gap, with a prototype for a disease tracking system. The system performs the collection and processing of data in an automated way through news portals, linking the processed information with spatial data from Brazilian municipalities. The rest of this paper is organized as follows: Section 2 presents the related work. Section 3 describes the project architecture. Section 4 presents the results. And finally, section 5 contains the conclusions of the study.

2. Related Work

Among the works in this research segment, one of the pioneers is the alert system through e-mails from ProMed-Mail[4] [Madoff 2004]. This system, which continues to be widely used today, has become the source of data for several other disease tracking platforms developed later. In its flow, before the news received on the portal are published, they are checked by specialists, which makes the platform a reliable and recognized source of data in the disease tracking segment. The World Health Organization (WHO) also maintains an alert portal for the emergence of new communicable diseases[5] with a similar model to ProMed-Mail, however with a much lower update frequency.

Another reference study in the segment is the work of [Freifeld et al. 2008] that presents the initial architecture of the HealthMap[6] platform, one of the first disease tracking platforms based on unstructured data. HealthMap extracts alert on communicable diseases on a global basis by extracting data from several sources, including news sites and also ProMed reports.

A similar approach is used in the study by [Lan et al. 2012], which presents the STEWARD[7] platform. However, in this system, only ProMED-Mail records are used, organizing them in the dimensions of space and time. According to the authors, using only the ProMed database can limit the dynamics of identifying new outbreaks, but it also reduces noise in the data presented because it is a more reliable source of information. In a subsequent study, [Lan et al. 2014] presented the Newsstand[8] platform. A system that can track different types of subjects, including health-related news in the dimensions of space and time.

Some studies tried to track the location and timing of diseases using even more dynamic methods for data extraction, like Social Media data. Among them, the study of [Jayawardhana and Gorsevski 2019], that tries to track the location and timing of a disease occurrence (flu) using data from Twitter.

Another example is the study of [Sankaranarayanan et al. 2009] which processes Tweets by identifying whether they are news and also which news segment they belong

[4] https://promedmail.org/

[5] https://www.who.int/csr/don/en/

[6] https://www.healthmap.org/

[7] http://steward.umiacs.umd.edu

[8] http://newsstand.umiacs.umd.edu/web/

to. The platform also offers a web interface for consulting processed data.

Another approach, used by [Chunara et al. 2013], was to extract data through crowdsourcing, in a platform called flu near you[9], that provides a form for users to self-report symptoms of respiratory diseases, such as fever, cough, shortness of breath, among others. The platform also allows data visualization through a web view of the maps.

Among the review studies in the segment, [Choi et al. 2016] carried out a systematic review of the main disease tracking systems and studies related to them. The study presents the differences between the main platforms and their strengths and weaknesses. The authors also highlight the importance of these systems and the need for countries with a shortage of them to seek to implement it.

[Mohanty et al. 2019] present a review of the disease tracking applications available for Android and IOS platforms. The study concluded that there is great potential in this segment, especially for solutions that serve health professionals and public health authorities.

We also cite studies in related areas or support of disease tracking, such as the study of [Castro and Jr. 2018], which describes the prototype of a tool to index textual and geographical information in a combined way. For textual indexing, the study used NLP techniques such as removing stop words and ranking through the Inverse Document Frequency (IDF).

Considering the public health data from Curitiba, several studies can be mentioned [de Oliveira et al. 2018, Cavalcante et al. 2018, Lima et al. 2019]. The study of [de Oliveira et al. 2018] presented a characterization of Paraguay's public health data, and using the information about the city of Asuncion a comparison was made with Curitiba's public health data. [Cavalcante et al. 2018] carried out a survey with the citizens of Curitiba to list the most important features in a health app. The study also presented a prototype of the application's screens for the features most required in the survey. In the study of [Lima et al. 2019], open data of Curitiba public health is aggregated with transportation data, analysing the accessibility to Curitiba public health units via public transportation.

3. System Architecture

As described in the study of [Lan et al. 2012], the extraction of correct places from unstructured documents is a challenging task, which evolves processing data through a pipeline, that breaks the data cleaning and formatting into several subsequent steps. The general architecture of the prototype developed in our study is shown in Figure 1. The system has three sources of data:

1. HTML from the news portals extracted through web crawling.
2. The shapefile of all the Brazilian municipalities and other information like the population of each city, extracted from the IBGE[10].
3. The list of infectious and parasitic diseases, obtained from the SUS[11] and manually inputted in the system.

[9] `https://fluneayou.org/`

[10] Instituto Brasileiro de Geografia e Estatística (Brazilian Institute of Geography and Statistics)

[11] Sistema Único de Saúde (The Brazilian Universal Health Care System)

The choice of what diseases to track was based on data published by the Ministry of Health of Brazil [Silva and Ferreira 2006]. All the diseases listed on this document were searched, except rabies. The reason for this exception is that rabies in Portuguese is called raiva and the word raiva also means angry in Portuguese, a very common word that could generate a lot of noise in the extracted data.

The complete list of tracked diseases is: aids, amebíase, ancilostomíase, ascaridíase, botulismo, brucelose, cancro mole, candidíase, coccidioidomicose, cólera, coqueluche, criptococose, criptosporidíase, dengue, difteria, doença de chagas, doença de lyme, diarréia, doença meningocócica, donovanose, enterobíase, escabiose, esquistossomose mansônica, estrongiloidíase, febre amarela, febre maculosa brasileira, febre purpúrica brasileira, febre tifóide, filaríase por wuchereria bancrofti, giardíase, gonorreia, hanseníase, hantavirose, hepatite a, hepatite b, hepatite c, hepatite d, hepatite, e, herpes, histoplasmose, HPV, influenza, leishmaniose, leptospirose, linfogranuloma venéreo, malária, meningite, mononucleose, oncocercose, paracoccidioidomicose, parotidite, peste, poliomelite, psitacose, rubéola, sarampo. shigelose, sífilis, cisticercose, tétano, toxoplasmose, tracoma, tuberculose and varicela.

Among the software used in the project, in the data extractor we use the python[12] language (version 3.8.5) with the scrapy framework[13] (version 2.3.0) to perform the data crawling. For NLP we use the spacy[14] library (version 2.3.2) configured for the Portuguese language. The database used was PostgreSQL[15] (version 12.3) with the Postgis[16] extension (version 2.5).

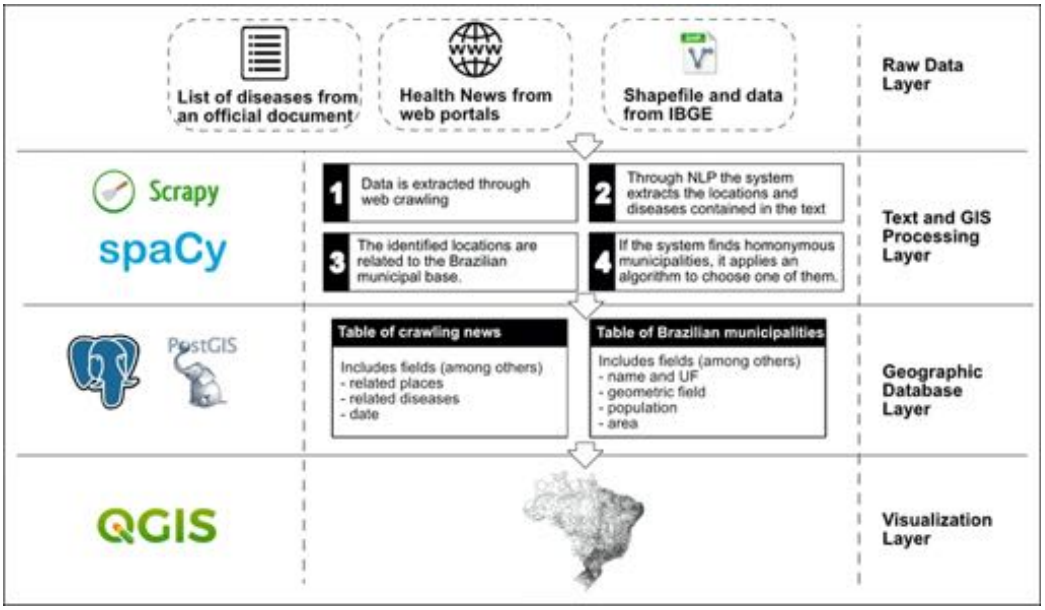

List of diseases from an official document
Health News from web portals
Shapefile and data from IBGE
Raw Data Layer
Scrapy
spaCy
1 Data is extracted through web crawling
2 Through NLP the system extracts the locations and diseases contained in the text
3 The identified locations are related to the Brazilian municipal base.
4 If the system finds homonymous municipalities, it applies an algorithm to choose one of them.
Text and GIS Processing Layer
PostGIS
Table of crawling news
Includes fields (among others)
- related places
- related diseases
- date
Table of Brazilian municipalities
Includes fields (among others)
- name and UF
- geometric field
- population
- area
Geographic Database Layer
QGIS
Visualization Layer

Figure 1: The architecture of the data extraction system.

[12] https://www.python.org/

[13] https://scrapy.org/

[14] https://spacy.io/

[15] https://www.postgresql.org/

[16] https://postgis.net/

3.1. Processing the Unstructured Data

The processing of unstructured data (HTML obtained from news portals) had the following steps:

1. First, the system searches for the term bairro (neighborhood) or bairros (neighborhoods). If it finds, the processing of the current page is interrupted, as the system may confuse neighborhood names with city names.
2. The system extracts the date of publication (or last update) of the news.
3. The system searches in the text content (body of news and title) all occurrences of the diseases being tracked, using regular expressions.
4. Diseases found with different terms are grouped into only one key term. For example, coronavirus, COVID, and COVID-19 are grouped into COVID-19.
5. If the system finds any diseases, then it does an NLP to identify the locations contained in the text.
6. The system removes the names of cities that can generate too much noise in the data, such as the municipality of Saúde (health) in the state of Bahia.
7. The locations found in the previous steps are searched at the base of Brazilian municipalities, previously extracted from IBGE.
8. If there are homonymous municipalities, the system selects only one of them, following two rules: first, it checks whether the publication portal is in the same state as any of the identified municipalities (proximity rule, like in the study of [Lan et al. 2014]). If the first strategy is not met, it then selects the municipality with the largest population.
9. Finally, if the system has identified at least one disease and one location in the news, it adds the record to the database.

4. Results and Discussion

4.1. Analyzing data on a state scale

To our analysis, we use the state of Paraná as a reference. The state is located in the south of Brazil and has the 5th largest population in the country, estimated in 2020 at 11,516,840 inhabitants according to the IBGE. Paraná has 399 municipalities, and its three most populous cities are Curitiba (the capital), Londrina, and Maringá.

We used data from five news portals of the state: The SESA (Health secretary of the government of Paraná) Portal[17], Bandab[18], Bem Paraná[19], AEN[20] (the official news agency of Parana government) and Tribuna do Paraná[21], with the data range from January 1, 2019 to August 20, 2020. The distribution in time of the extracted news is presented in Figure 2 for 3 diseases (Dengue, Yellow Fever, and COVID-19), and also the totals. Note that the Tribuna do Paraná portal does not keep a long historical period of its news, so in the last extraction of our algorithm, it obtained data only from the last 2 months of this portal. Another important point is the correlation between the number of news extracted

[17] http://www.saude.pr.gov.br/

[18] https://www.bandab.com.br/

[19] https://www.bemparana.com.br/

[20] http://www.aen.pr.gov.br/

[21] https://www.tribunapr.com.br/

on the SESA and the AEN portal. The reason is that both portals are managed by the same organization, the Paraná government.

The data processed and used in this study were made available in the Harvard Dataverse repository[22].

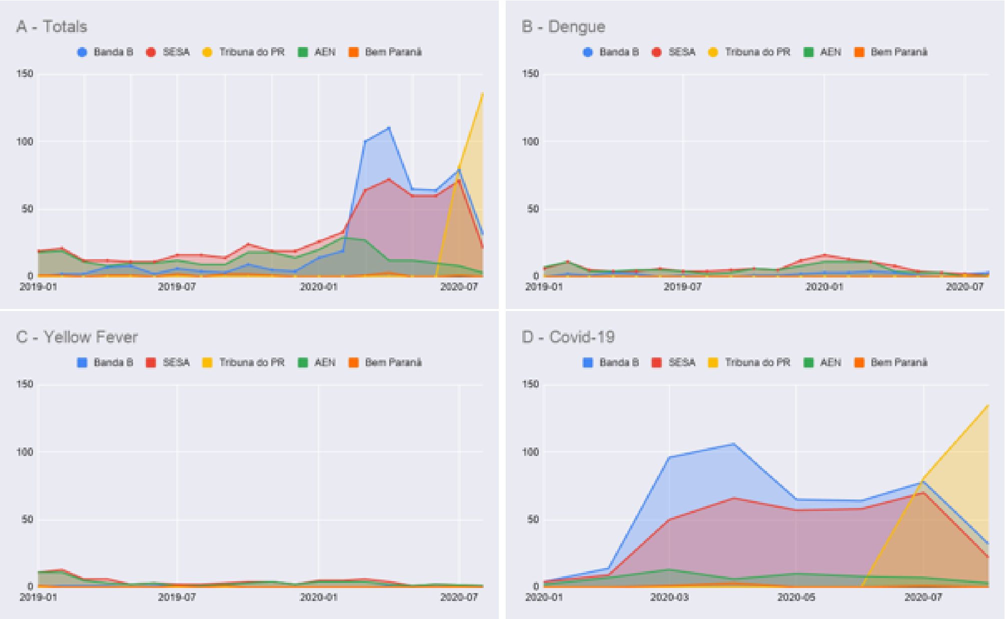


Figure 2: Number of news extracted from each portal by month and disease.

Figure 2 presents an increase in the total number of health-news since about February, although news related to Dengue and Yellow Fever, two highly relevant diseases in the state of Paraná in the last few months, has been declining. The reason is the high frequency of news related to COVID-19, which, as we can see in Table 1, has a greater number of news published than all other diseases combined.

Disease	Related news
COVID-19	1072
Dengue	271
Measles	177
Yellow Fever	161
Influenza	104
Rubella	54
HIV	46
Meningitis	36
Varicella	34
Tuberculosis	33

Table 1: The top 10 diseases by the number of news.

[22] https://doi.org/10.7910/DVN/1YY646

To check the extracted data, we also selected the cities identified in states other than the news source (in this case we extracted the news identified in municipalities outside Paraná), and, from this selection, we separated the news related to capitals. Thus, the news identified was segmented into three groups: news within the state of Paraná, news from capitals outside Paraná, and others. The totals for each of these three groups are shown in 3.

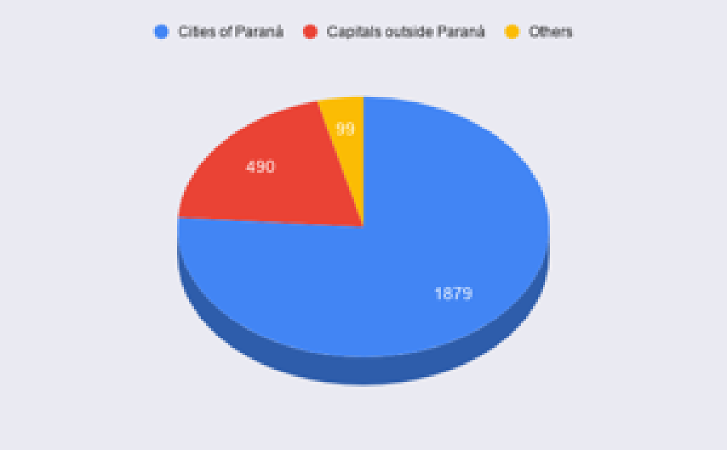


Figure 3: Number of news extracted from 3 groups.

On the group of ”others”, a manual check on the data was carried out. In this group there are cities with great potential to generate false alerts due to having names with famous international places (like the city of Colômbia in the state of São Paulo or the city of Tailândia in the state of Pará) or with very common words in the Portuguese language (like the city of Central (Central) in the state of Bahia or the city of Campanha (Campaign) in the state of Minas Gerais). In these cases, the name of these cities was added as an exception to be ignored by the system pipeline (like described in step 5 of processing).

We then generated choropleth maps (Figure 4) for the two diseases with the highest number of news in the observed period, trying to identify the regions with higher relevance for each one of them. For each map, we present three versions:

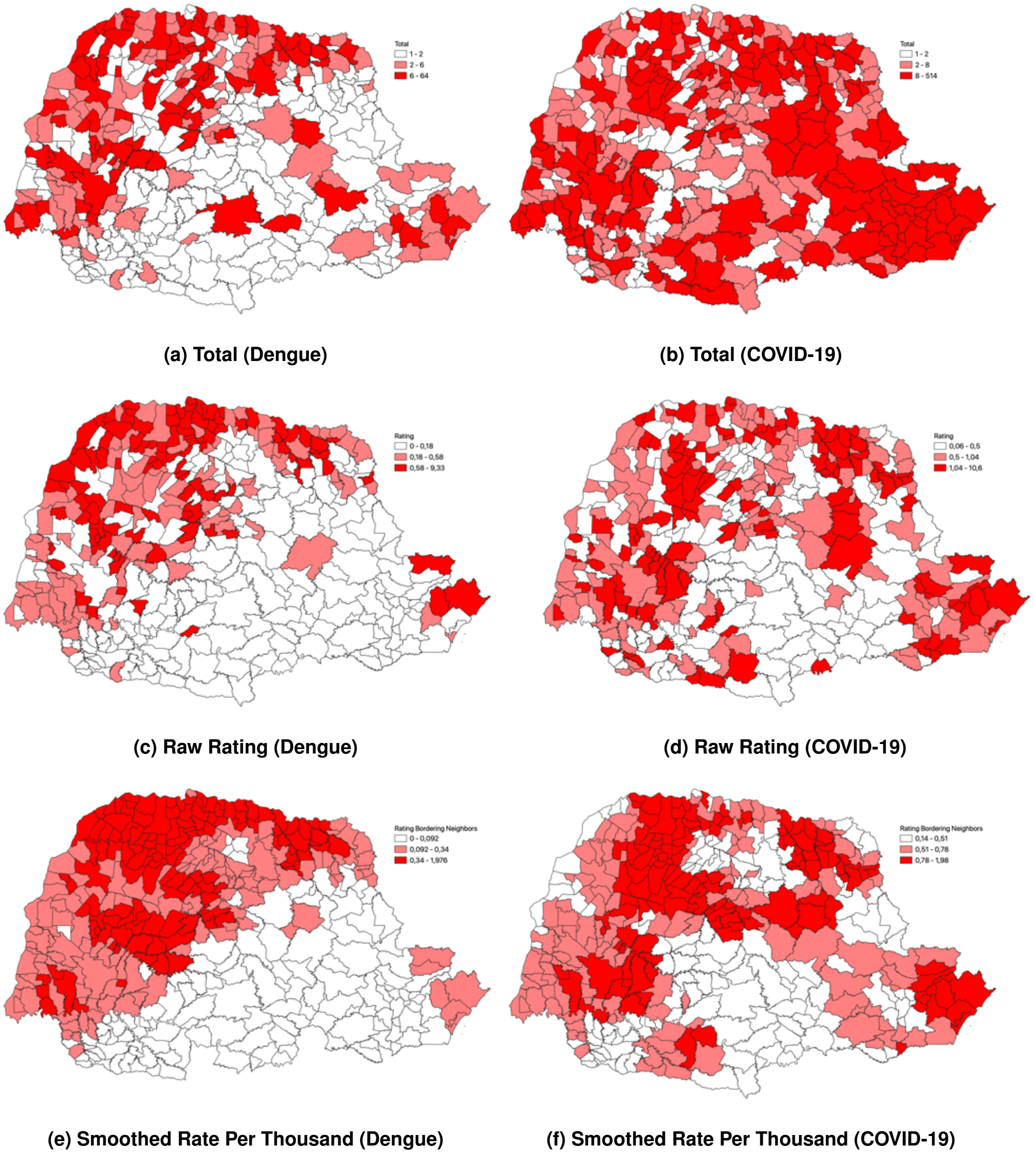
1. A version based on the total number of news of a given disease-related to each city.
2. A version based on a raw ratio per thousand (rrpt), obtained using the equation 1.
3. A version based on a smoothed rate per thousand (srpr), obtained using the equation 2.

$$rrpt = \frac{T}{P} * 1000 \tag{1}$$

$$srpt = \sum_{i=1}^{n} \frac{T(i)}{P(i)} * 1000 \tag{2}$$

Where:

- n - is the number of bordering neighbors (including the city itself).
- T - is the total of news related to some disease for the city.
- P - is the population of the city.



(a) Total (Dengue)

(b) Total (COVID-19)

(c) Raw Rating (Dengue)

(d) Raw Rating (COVID-19)

(e) Smoothed Rate Per Thousand (Dengue)

(f) Smoothed Rate Per Thousand (COVID-19)

Figure 4: The distribution of the identified diseases on the map.

Through the raw rate per thousand (equation 1), we seek to reduce the bias of the system alerts for larger cities and highlight the alerts of small and medium cities, considering population density. According to [Anselin et al. 2006] raw rate serves as estimates for an underlying risk, i.e., the probability for a particular event to occur. The smoothed rate per thousand (equation 2) tries to balance the data and thus allows the identification of patterns by regions. According to [Anselin et al. 2006] smoothed rates tend to empathize broad trends.

Lastly, we chose to classify the map with a quantile scale because, as [Brewer and Pickle 2002] demonstrated in their study, this is one of the most efficient scales in the presentation of choropleth maps, being suitable for several different types of epidemiological data.

Figure 4 shows complementary visualizations of the diseases. For example, to identify the regions most affected by dengue, Figure 4-E showed a better-defined pattern, while for COVID-19 the map in Figure 4-B was closer to reality. The reason for this is because COVID-19 has an equivalent publication frequency in almost all regions of the state, making it difficult to identify patterns by region.

4.2. Analyzing data on a city scale

We also analyzed the data on a municipal scale using the municipality of Curitiba as a reference. Curitiba is the capital and largest city in the state of Paraná, with an estimated population in 2020 of 1,948,626 million inhabitants according to IBGE. An important point is that the city government of Curitiba makes available various types of data through an open data portal[23] and, one of them, is the data of health appointments of its health units. Among the fields contained in the health appointments data is the ICD[24] of the diagnosed disease. Thus, we filter the cases of measles, influenza, and dengue using their respective ICDs described in Table 2:

Disease	**ICD**
Measles	B05
Influenza	J11
Dengue	A90

Table 2: The ICD of searched diseases.

To obtain the number of COVID-19 cases, we used another dataset, also from the open data portal of the municipality of Curitiba. This dataset was more up to date and contained data from March until August 2020.

The initial idea was to compare the number of cases of each disease from the beginning of 2019 to the current date (August 2020) with the number of related news processed by our system. However, we encountered two obstacles: The first was that Curitiba data from health appointments has a gap in January 2019; the second was that the health appointments dataset was limited until mid of February 2020. In this way, we removed January 2019 and included January 2020 in our analysis, thus comprising one year. The only exception was COVID-19, which was analyzed between January and August 2020.

[23] `https://www.curitiba.pr.gov.br/dadosabertos/`

[24] International Classification of Diseases

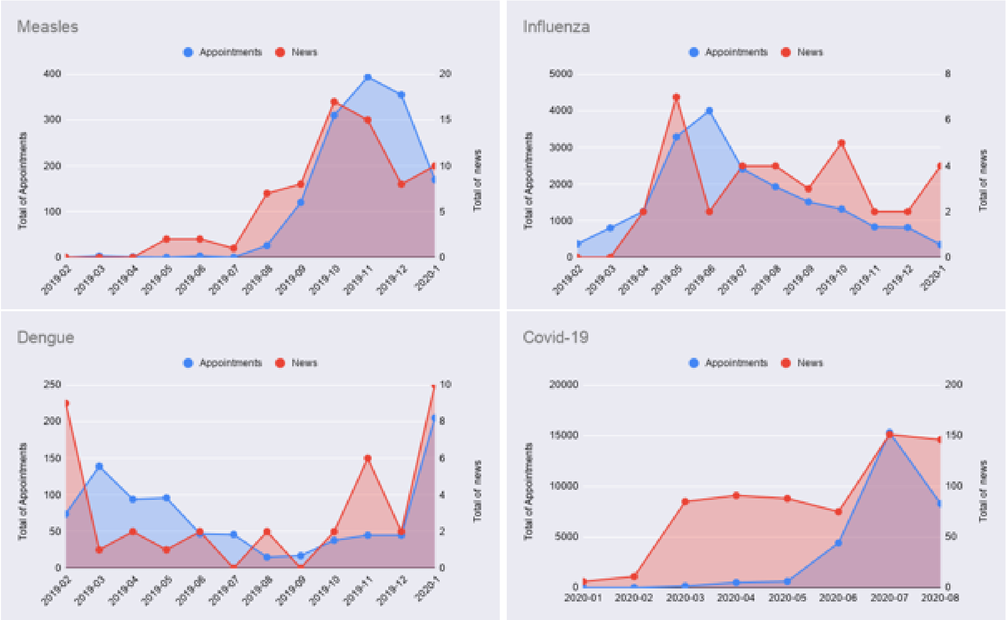


Figure 5: Comparison between the number of diagnoses and the number of news per month in the city of Curitiba.

4.3. Findings

In the temporal analysis of the data, we identified an abnormal increase in the volume of news related to infectious diseases between February and August (Figure 2A), generated mainly by the great attention that COVID-19 has received. This increase generated a distortion in the visualization of other diseases when analyzed on the same scale as the COVID-19 (Figure 2). However, when the data were analyzed in isolation, it was possible to perceive a relationship between its volume of occurrence and the volume of news related to it (Figure 5).

Analyzing the spatiality of the data (Figure 4), we realized that diseases strongly influenced by environmental factors, such as dengue, were highlighted in more well-defined regions. Another observed fact was that the relevance of diseases could be identified for a large number of cities, even those of small and medium-size.

Among the challenges we faced in the development of this study, we cite the difference in the nomenclature of some Brazilian municipalities, between data from the polygons base and data from the population base, both obtained from the IBGE website. In these situations, we did a manual check on the municipality's website to verify the correct spelling. Another challenge was to define a period for data analysis, as some databases had a very short data history (or a gap in the data for a given period), while others comprised a very long uninterrupted period. We try to cover as much data as possible within the periods where most data sources were available.

5. Conclusions

This study presented the architecture and preliminary results of a prototype system for disease tracking system on a national scale. As a preliminary test, for analysis on a state scale, data from Paraná were used with 5 different sources. For the analysis on a municipal scale, open data from the municipality of Curitiba were also used. The graphs and maps generated on the data extracted by the platform showed some patterns and confirmed that the system can extract relevant information even on small municipalities. However, the system needs further testing before it can be made available for public access. In future studies, we intend to expand the base of news portals extracted and processed by the system, covering the entire national territory and develop a mobile interface for users to consult information. Future studies can also apply machine learning to segment and filter the types of news processed by the system. Other processing steps should also be added to the system, such as the differentiation between large cities and states with the same name, such as São Paulo and Rio de Janeiro.

6. Acknowledgments

The authors thank the Brazilian Institute of Geography and Statistics (IBGE), Curitiba Urban Research and Planning Institute (IPPUC), and the city government of Curitiba for sharing part of the data used in this study.

References

Anselin, L., Lozano, N., and Koschinsky, J. (2006). Rate transformations and smoothing. *Spatial Analysis Laboratory Department of Geography*.

Brewer, C. A. and Pickle, L. (2002). Evaluation of methods for classifying epidemiological data on choropleth maps in series. *Annals of the Association of American Geographers*, 92(4):662–681.

Castro, M. Q. and Jr., C. A. D. (2018). Ferramenta para recuperação de informação utilizando indexação espacial e textual. In Vinhas, L. and Campelo, C. E. C., editors, *XIX Brazilian Symposium on Geoinformatics - GeoInfo 2018, Campina Grande, PB, Brazil, December 5-7, 2018*, pages 158–163. MCTIC/INPE.

Cavalcante, J. L. S. B., Neto, M. S., and Kozievitch, N. P. (2018). Utilização e estudo de dados de saúde georreferenciados para desenvolvimento de aplicação móvel. In Vinhas, L. and Campelo, C. E. C., editors, *XIX Brazilian Symposium on Geoinformatics - GeoInfo 2018, Campina Grande, PB, Brazil, December 5-7, 2018*, pages 170–175. MCTIC/INPE.

Choi, J., Shim, E., and Woo, H. (2016). Web-based infectious disease surveillance systems and public health perspectives: A systematic review. *BMC Public Health*, 16.

Chunara, R., Aman, S., Smolinski, M., and Brownstein, J. (2013). Flu Near You: An Online Self-reported Influenza Surveillance System in the USA. *Online Journal of Public Health Informatics*, 5(1).

de Oliveira, M. F. A., Kozievitch, N. P., Bim, S. A., and Legal-Ayala, H. (2018). Caracterização dos dados públicos de saúde do paraguai. In *ERBD 2018*, page 12, Porto Alegre, RS, Brasil. SBC.

Freifeld, C., Mandl, K., and Brownstein, J. (2008). Healthmap: Global infectious disease monitoring through automated classification and visualization of internet media reports. *Journal of the American Medical Informatics Association : JAMIA*, 15:150–7.

Jayawardhana, U. K. and Gorsevski, P. V. (2019). An ontology-based framework for extracting spatio-temporal influenza data using twitter. *International Journal of Digital Earth*, 12(1):2–24.

Kindhauser, Mary Kay and World Health Organization (2003). Communicable diseases 2002 : global defence against the infectious disease threat / edited by mary kay kindhauser. https://apps.who.int/iris/handle/10665/42572.

Lan, R., Adelfio, M. D., and Samet, H. (2014). Spatio-temporal disease tracking using news articles. In *Proceedings of the Third ACM SIGSPATIAL International Workshop on the Use of GIS in Public Health*, HealthGIS '14, page 31–38, New York, NY, USA. Association for Computing Machinery.

Lan, R., Lieberman, M. D., and Samet, H. (2012). The picture of health: Map-based, collaborative spatio-temporal disease tracking. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Use of GIS in Public Health*, HealthGIS '12, page 27–35, New York, NY, USA. Association for Computing Machinery.

Lima, C. D., Peixoto, A. M., Gomes-JR, L. C., Luders, R., and Fonseca, K. V. O. (2019). Avaliação da qualidade do transporte público no acesso a unidades de saúde de curitiba. In *Anais do III Workshop de Computação Urbana (COURB 2019)*, volume 1, Gramado, RS, Brasil. SBC.

Madoff, L. (2004). Promed-mail: An early warning system for emerging diseases. *Clinical infectious diseases : an official publication of the Infectious Diseases Society of America*, 39:227–32.

Mohanty, B., Chughtai, A., and Rabhi, F. (2019). Use of mobile apps for epidemic surveillance and response – availability and gaps. *Global Biosecurity*, 1:37.

RDA COVID-19 Working Groups (2020). *RDA COVID-19 Working Group Recommendations and Guidelines, 1st release*. Research Data Alliance.

Sankaranarayanan, J., Samet, H., Teitler, B. E., Lieberman, M. D., and Sperling, J. (2009). Twitterstand: news in tweets. In Agrawal, D., Aref, W. G., Lu, C., Mokbel, M. F., Scheuermann, P., Shahabi, C., and Wolfson, O., editors, *17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2009, November 4-6, 2009, Seattle, Washington, USA, Proceedings*, pages 42–51. ACM.

Silva, T. P. T. e. and Ferreira, I. d. L. M. (2006). Doenças infecciosas e parasitárias: guia de bolso. *Cadernos de Saúde*, 22:2498 – 2498.

United Nations (2019). World urbanization prospects: The 2018 revision. https://www.un-ilibrary.org/content/publication/b9e995fe-en.

World Health Organization (2020). Timeline of who's response to covid-19. https://www.who.int/news/item/29-06-2020-covidtimeline. accessed June 3, 2020.

Mobipy - A Python Library for Analyzing Mobility Patterns

Pedro Henrique Costa Maia[1], **Claudio E. C. Campelo**[1]

[1]Systems and Computing Department
Federal University of Campina Grande (UFCG)
Campina Grande – PB – Brazil

pedro.maia@ccc.ufcg.edu.br, campelo@dsc.ufcg.edu.br

Abstract. *With the increase in the availability of georeferenced databases, interest in analyzing them in research that requires an understanding of people's mobility patterns, especially in urban centers, has increased. Given this range of applications, several metrics have been proposed in the literature to infer patterns of people's movement, however, they are often not available for use in other studies. In this article, we introduce Mobipy, a Python library that brings together metrics and functions frequently used to calculate people's mobility patterns. It was developed with a focus on usability and compatibility with multiple datasets, facilitating research and data analysis tasks. We hope that Mobipy will provide researchers with new possibilities, enriching their analysis and generating new knowledge.*

1. Introduction

The study of urban mobility is important for various types of applications [Hasan et al. 2013]. Methods of identifying and interpreting people's mobility can provide useful knowledge for urban planning in large cities [Yuan et al. 2012], since, by knowing how people move, more efficient strategies can be used in relation to infrastructure, public safety, urban traffic, among others.

The large amount of geolocation data generated by the massive use of social networks and other applications installed on smartphones enabled researches that analyze different social phenomena, from the spread of a disease [Vazquez-Prokopec et al. 2013], to human behavior in the face of a natural disaster [Song et al. 2014]. The location, trajectory and habits of people that can be inferred from geo-temporal data can also help develop recommendation systems [Kong et al. 2017], define strategies for socio-economic development [Pappalardo et al. 2015], generate alternatives to census data [Jerônimo et al. 2017], among other applications.

Given the scope of these applications, several techniques were created to extract movement patterns from spatio-temporal data. However, applying these metrics to a dataset is not always a simple task. There are cases in which, despite the metric being implemented and documented, the data are not compatible, adding a new job to the researcher to organize the data in order to execute the algorithm. In addition, there are cases in which the metric is not even implemented, with only a pseudocode available or even just a textual description of the algorithm.

The process of applying metrics on geotemporal data must be simple and easy for the researcher/developer, so that he can dedicate more time to carry out his analyzes. In this context, we developed the *Mobipy: A Python Library for Mobility Patterns* [1], that brings together multiple functions for calculating metrics of mobility patterns. These functions were selected based on their relevance and some of them have adapted for greater coverage in other research. Mobipy aims at providing geoinformatics researchers and developers with a useful and easy to use tool.

The rest of this article is structured as follows. Section 2 define some basic concepts used in the following sections. Section 3 shows an overview of the library, with metrics and auxiliary functions. Section 4 includes demos of the system in use with real data. Finally, Section 5 concludes the paper and points to future work.

2. Basic concepts

This section introduces some concepts and technologies that were adopted in the development of the library.

2.1. Clustering

With the increased availability of geospatial data, the demand for ways to conduct exploratory analyzes on these data has also increased. For this, one of the most used processes is data clustering. As discussed by [Han et al. 2001], this process groups a set of objects into classes or clusters so that the grouped objects are highly similar to each other. This procedure makes it possible to identify relationships and characteristics that may exist implicitly in spatial databases.

Many different clustering methods have been proposed in the literature: based on wavelets [Sheikholeslami et al. 2000], data density [Wang and Hamilton 2003], presence of physical obstacles [Zaïane and Lee 2002], among others. In this work, the clustering algorithm DBSCAN (Density Based Spatial Clustering of Applications with Noise) [Ester et al. 1996] will be used. This algorithm was chosen because it does not require the user to specify the number of clusters to be created, and needs only two parameters to work, making its use simpler.

Figure 1 shows an example of how the algorithm works. In this example, three clusters were generated, based on similar spatial characteristics of their containing elements. In addition, it can be seen that there are black dots around the clusters, called noise, representing data located in low density regions. This approach will be used in the calculation of some metrics in the library, which will be described in Section 3.1.

2.2. Dataframe

Performing operations on a large dataset can be a costly task for the machine. Therefore, it is necessary to use a structure that can efficiently filter, iterate and access data. For this purpose, Mobipy uses as its primary data structure the DataFrame of the Pandas library [McKinney 2011]. This structure offers powerful and flexible tools that help manipulate the functions that will be presented in Section 3.

Essentially, DataFrame is a two-dimensional data structure, consisting of rows and columns, referring to a spreadsheet. It can be created from files, web pages or code-generated data. Table 1 shows an example of the structure of a *DataFrame* with the dataset we used to perform the library tests, described in Section 4.

[1] Repository link: github.com/pedrohcm/mobipy

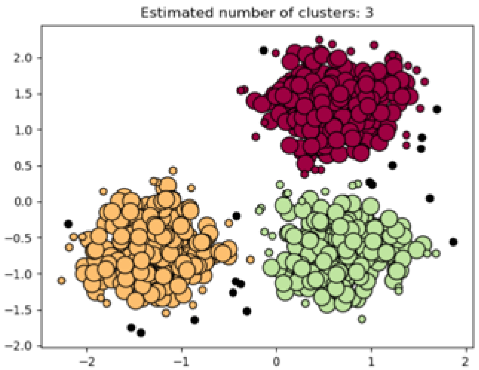


Figure 1. Example of operation of the DBSCAN algorithm, using Scikit-learn.

Table 1. Description of *DataFrame* used in Section 4.

	local time	latitude	longitude	horizontal accuracy	speed accuracy	time	speed	tz	userid
0	1.270654e+09	46.4509	6.86953	87.5226	16.992	1.270661e+09	4.320	-7200.0	6181.0
1	1.270654e+09	46.4509	6.86941	68.2173	11.998	1.270661e+09	6.624	-7200.0	6181.0
2	1.270654e+09	46.4508	6.86946	113.5750	11.998	1.270661e+09	6.624	-7200.0	6181.0
3	1.270654e+09	46.4507	6.86932	56.7157	11.998	1.270661e+09	6.624	-7200.0	6181.0
4	1.270654e+09	46.4506	6.86910	75.9002	9.504	1.270661e+09	3.528	-7200.0	6181.0

From the creation of the dataframe, it becomes possible to perform manipulations on the data in a simple way, in addition to providing useful information for exploratory analyzes to be carried out with this data.

3. Solution Architecture and Design

Mobipy is a code library writen in Python programming language that aims at facilitating the calculation of metrics and patterns of mobility of from geo-temporal data. Common input data are those produced by smartphones and from geotagged posts from social network, although other data in the same format can also be used. It focuses on ease and versatility of use, making it possible to filter data before performing the calculations, in addition to providing *insights* on the analyzed datasets. Currently, there are five metrics in the library, each of which can be used for multiple applications. Due to Mobipy's flexibility, new functions can be added, as well as existing ones expanded to more specific cases.

Initially, a literature review was carried out to identify candidate metrics to be implemented in Mobipy. Among them, some were originally in other languages, needing certain adaptation for Python and its libraries. However, others only had a pseudo-code or just a textual description, which led to the development of an approach that both suited the structure of Mobipy and was generalized in order to be used in different situations. Such generalization is necessary due to the data entry format for Mobipy's functions (*DataFrame*). As explained in the subsection 2.2, the advantage of using *DataFrame* is the high performance when processing a large amount of data, which is essential for calculating metrics.

3.1. Metrics

The next subsections describe the mobility metrics calculation algorithms implemented in Mobipy and its auxiliary functions.

3.1.1. Radius of Gyration

The metric Radious of Gyration (a.k.a. Radious of Inertia) is commonly used in data from *tweets* [Yin et al. 2016], as it can measure how far and how often a user moves [Jerônimo et al. 2017]. The metric measures the accumulated distances from the center of mass of a user's trajectories, indicating their spatial coverage. The resulting value is directly proportional to the travel distance in relation to a center of mass, and can provide knowledge about its movement patterns.

For this computation, Mobipy can receive as an input the center of mass (previously calculated by the user) or, alternatively, it can be calculated by the library. Thus, the function only requires one parameter: points (p), since the center of mass (p_c), is optional. The turning radius (r_g) metric is calculated as shown in Equations 1 and 2.

$$r_g = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - p_c)^2} \tag{1}$$

$$p_c = \frac{1}{n}\sum_{i=1}^{n} p_i \tag{2}$$

3.1.2. Total Distance Displacement

Another metric available at Mobipy is the total travel distance, which is the sum of the distances between the consecutive routes performed by the user. This metric can be useful, for example, to identify users who work as application drivers, taxi drivers, among others.

3.1.3. Activity Centers

From the user data, and applying the DBSCAN algorithm, shown in Section 2.1, it is possible to infer the activity centers. These centers can represent places where the user remains most frequently, being possible to process such places due to the clustering technique applied.

3.1.4. Home Location Detection

The location of the user's home can express social conditions that, in turn, can influence how citizens move within an urban center [Jerônimo et al. 2017]. In this case, Mobipy considers only activities carried out at night, and on weekdays, to minimize the incidence of false positives. Thus, the DBSCAN clustering technique is applied again here to identify the user's place of residence. Although based on a simple heuristic, [Jerônimo et al. 2017] noted that this method proved to be useful in identifying residences and that it can be reused in other research.

3.1.5. Group by Closeness

Grouping two datasets together can provide important insight into how they relate. An example would be to link a user's activity centers with Points of Interest (POI), such as restaurants, squares and schools. This can provide important data for companies about the types of people who visit such places, how much time they spend, other places frequented by these people, among others.

The proximity grouping function takes two datasets, A and B, and lists all items in B that are closest to items in A. The output of the algorithm is a list containing all the elements of A and their respective elements of B. Optionally, if the A dataset contains temporal data, such as *timestamps*, the metric will also return the quartiles of the duration that each element of A was close to the elements of B. These values are used to generate a *box-plot* for a better view of the metric.

However, this processing requires an intensive calculation of the distance between each element $A_i \in A$ and all elements of B, which can be costly depending the number of elements in each dataset. For this, the algorithm performs a "cut" in the dataset B in order to increase performance by not having to calculate the distance for all elements. This cut is performed by creating a *bounding-box* around the A_i element, which filters the elements from B to consider only elements that are within that structure. This parameter is called *searchTolerance*, which defaults to 50m.

In addition, there is also a parameter that indicates the tolerance, in meters, of the distance between two elements of B to be considered close to the same element A. This helps to consider elements of B that can belong to more than one element of A.

3.2. Helper Functions for Calculating Metrics

Mobipy also has auxiliary functions that have as input the same types used in the calculation of the metrics, that is, they can be used with the same dataset, expanding its set of applications. Among the functions, there are those used by the metrics themselves, such as the calculation of distance between two points, center of mass, dataframe items closer to a point, execution of DBSCAN, among others. These functions can be used separately, just importing the Mobipy emph utils module.

3.3. Helper Functions for Datasets

Mobipy includes some auxiliary functions for reading the datasets, described below.

3.3.1. Data Identifier

Given the vast amount of datasets available on the internet, it is easy to come across different types of data, organized in many different ways. Since Mobipy must be used with a dataset, it was thought of as a way to support a greater variety of these. For this, the library has a structure called *DataIdentifier*. This structure contains the following attributes:

- latitude;
- longitude;
- startTime;
- endTime;
- timestamp;
- itemId.

These attributes refer to how the corresponding values are found in the dataset. For example, the latitude field can be like *lat* or *geolat*, so the user can create an instance of *DataIdentifier* with the names of each of the fields, so that the algorithms can access the data normally, without having to change the database structure.

3.3.2. Data Filter

Mobipy can also filter data from datasets before calculations, being useful for the user who wants to calculate metrics in just one time interval, without having to create another dataset before using the library. The user can filter the dataset from:

1. Date range;
2. Weekday (monday to sunday);
3. Time interval of the day (considering the day divided into 24 hourly intervals).

The filter was implemented in such a way that it is not necessary to specify the time zone, being directly compatible with types of temporal data present in the datasets. The input (1) is composed of two *strings* in date format, while (2) and (3) are configured from a *string* containing binary values (0 or 1), where 1 means the item is of interest and 0 indicates the item must be ignored. There are seven values to be set for (2) and 24 values for (3). Table 2 shows an example of how each filter works according to an input.

Table 2. Example of data filter operation with *Selector*.

Example	Filter
“2019-04-23” “2019-04-29”	April 23 to 29, 2019
“1001000”	Wednesday and Sunday
“110001110000011110000000”	0-2h, 5-8h, 13-17h

Such parameters allow the calculation of metrics, for example, using data from a certain month, but considering only weekends from 19h to 23h. Another example would be to apply the metrics only to data collected at night, to compare the results with data collected during the day, among others. In this way, filters can provide new perspectives on user mobility, increasing the possibilities for the researcher.

4. Examples of Usage

This section presents examples of system execution scenarios for calculating the metrics described in Section 3.1. For these case studies, we use a *dataset* with real data and produce artifacts to visualize the outputs of the functions.

4.1. Dataset

The data to be used in the tests come from a survey carried out by Nokia, called Mobile Data Challenge (MDC) [Laurila et al. 2012, Kiukkonen et al. 2010]. In this survey, carried out between 2009 and 2011, in the city of Lake Geneva, Switzerland, sensors embedded in smartphones were used in order to record continuous geolocation data of about 200 people.

4.2. Scenario I

In the first scenario, the system was run to evaluate the Radius of Gyration metric (Section 3.1.1), Total Distance Displacement (Section 3.1.2) and Activity Centers (Section 3.1.3). For that, data from a user present in the dataset whose trajectory has been short was selected, in order to provide a better visualization of the results. Figure 2 shows, in green, the geographic data of this user.

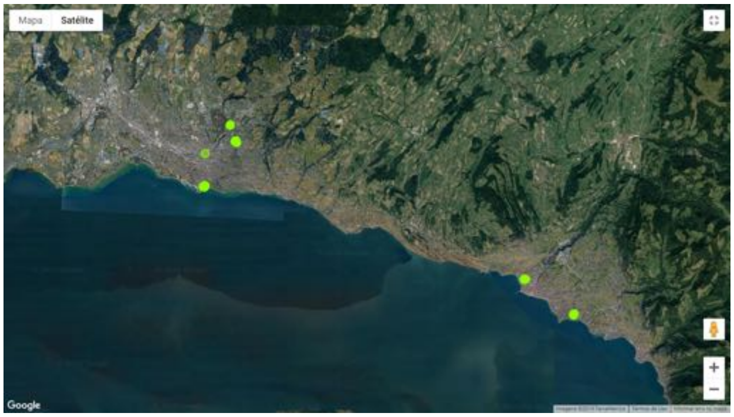


Figure 2. User 1 data on the map.

When executing the Radius of Gyration function, the midpoint was not specified. Therefore, the algorithm did the calculation of this location before following its execution. The midpoint found was *(46.501, 6.694)*, identified in Figure 3 in the yellow circle in the center of the map. This point was then used as a basis to identify the greatest distance. After execution, the function returned the Radius of Gyration, which is approximately 7665 *m*.

From the geographic data, the execution of the total distance displacement calculated that user 1 had a total trajectory of 38.6 *km* during the time interval from 07 to 20 April 2010. The calculation of the Activities identified 2 elements, each involving data at the ends of the map, as seen in Figure 3.

4.3. Scenario II

In the second experimental scenario, we performed the Home Location Detection algorithm. For this, another user was selected with much more data, since, in this way, more clusters are generated, so that the algorithm identifies the one that corresponds to the home location.

As described in Section 3.1.4, the algorithm selects the data sent at night and on the days of the week. Out of the *49.557* Dataframe lines, *17.958* were considered. Then,

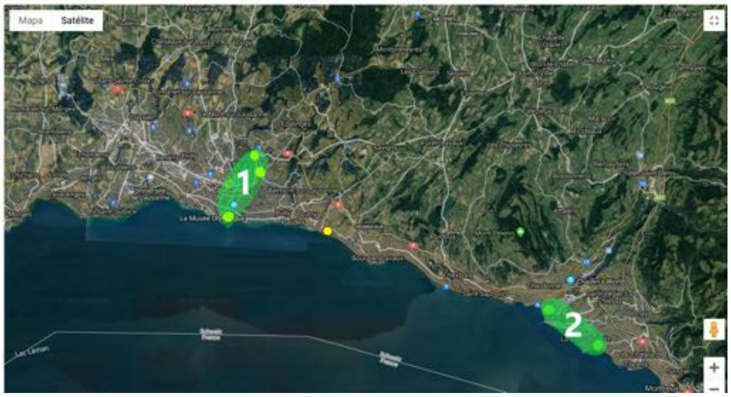


Figure 3. Midpoint and the two user 1 activity centers resulting from the *Radius of Gyration* and *Activity Center Detection* functions, respectively.

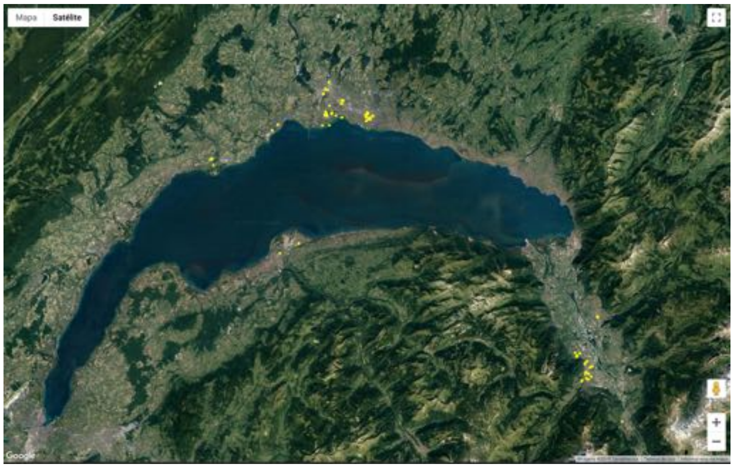


Figure 4. Geographic data of user 2.

from those remaining ones, 7 clusters were created, as shown in the map of Figure 5, where each cluster is shown in a different color. From them, it appears that the densest regions (i.e., the place where the user spends the most time at night on weekdays) are their home. In Figure 5 the identified home location is shown, together with the other clusters created.

4.4. Scenario III

For the third and final scenario, we used two datasets. The first consists of stop regions, which are aggregations of a user's data. The second consists of POIs obtained from Google Maps [2]. It is noteworthy that this is only a scenario of application of the proximity grouping metric (Section 3.1.5), which, in this case, can provide information about the profile of the places that the user most frequently visits. The datasets were passed as a function parameter, while the function output (including the POIs closest to each stop

[2] POIs were obtained programmatically, from the Google Maps API, and then used directly to run the tool, without storing them after execution, according to guidelines specified in the API license

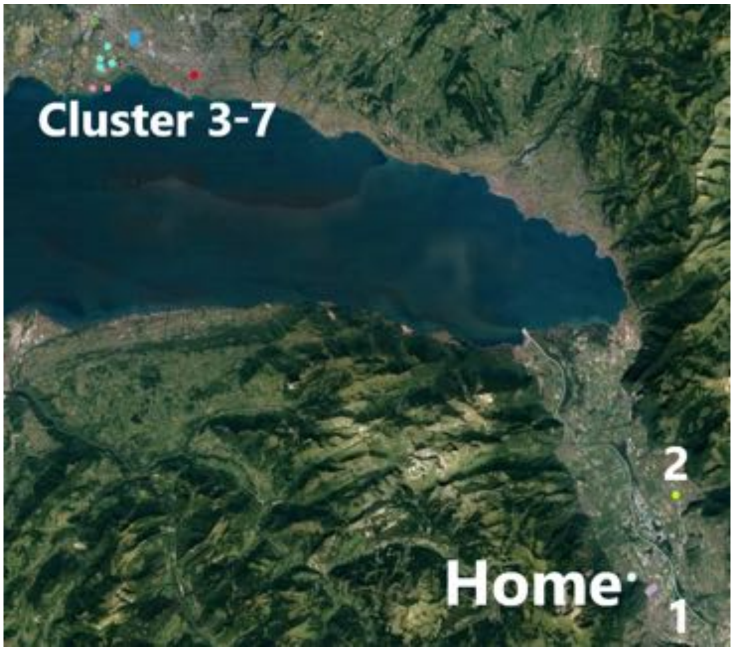


Figure 5. Clusters produced by the *Home Location Detection* algorithm.

region) was drawn on the map of Figure 6.

Based on Figure 6, it is possible to observe the stop regions *(stop region)* (in pink) with their closest POIs, limited to 10 (in dark blue). The nearest POI *(closest because)* is identified on the map by the red square around it.

5. Final Remarks

For the implementation of the library, the Python language was chosen due to the wide range of geospatial data processing tools available [Garrard 2016, Graser and Olaya 2015, Dobesova 2011] as well as to the ease of coding, readability and import in other projects in different systems. There were also decisions about the structure of Mobipy in relation to which and how each of the metrics would be translated into a functional algorithm, as well as that it would be efficient and general to the point of being compatible with a variety of datasets. Furthermore, decisions were made in favor of the performance of the algorithms, since datasets with thousands of records are generally used, and operations such as those performed in the calculation of the metrics tend to be costly in terms of time and processing.

Regarding improvements that can be made at Mobipy in future work, the following can be listed:

(i) implementation of additional metrics or evolutions of existing ones;
(ii) development of more advanced dataset filters;
(iii) integration with database APIs;
(iv) native tool for viewing plots results.

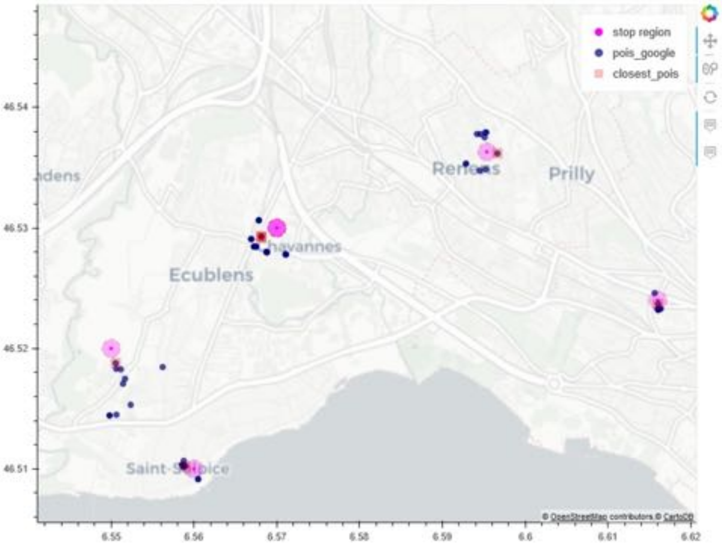


Figure 6. Grouping of stop regions and points of interest.

References

Dobesova, Z. (2011). Programming language python for data processing. In *2011 International Conference on Electrical and Control Engineering*, pages 4866–4869. IEEE.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

Garrard, C. (2016). *Geoprocessing with Python*. Manning Publications Co.

Graser, A. and Olaya, V. (2015). Processing: A python framework for the seamless integration of geoprocessing tools in qgis. *ISPRS International Journal of Geo-Information*, 4(4):2219–2245.

Han, J., Kamber, M., and Tung, A. K. (2001). Spatial clustering methods in data mining. *Geographic data mining and knowledge discovery*, pages 188–217.

Hasan, S., Zhan, X., and Ukkusuri, S. V. (2013). Understanding urban human activity and mobility patterns using large-scale location-based data from online social media. In *Proceedings of the 2nd ACM SIGKDD international workshop on urban computing*, page 6. ACM.

Jerônimo, C. L. M., Campelo, C. E. C., and de Souza Baptista, C. (2017). Using open data to analyze urban mobility from social networks. *JIDM*, 8(1):83.

Kiukkonen, N., Blom, J., Dousse, O., Gatica-Perez, D., and Laurila, J. (2010). Towards rich mobile phone datasets: Lausanne data collection campaign. *Proc. ICPS, Berlin*, 68.

Kong, X., Xia, F., Wang, J., Rahim, A., and Das, S. K. (2017). Time-location-relationship combined service recommendation based on taxi trajectory data. *IEEE Transactions on Industrial Informatics*, 13(3):1202–1212.

Laurila, J. K., Gatica-Perez, D., Aad, I., Bornet, O., Do, T.-M.-T., Dousse, O., Eberle, J., Miettinen, M., et al. (2012). The mobile data challenge: Big data for mobile computing research. Technical report.

McKinney, W. (2011). pandas: a foundational python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, 14.

Pappalardo, L., Pedreschi, D., Smoreda, Z., and Giannotti, F. (2015). Using big data to study the link between human mobility and socio-economic development. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 871–878. IEEE.

Sheikholeslami, G., Chatterjee, S., and Zhang, A. (2000). Wavecluster: a wavelet-based clustering approach for spatial data in very large databases. *The VLDB Journal—The International Journal on Very Large Data Bases*, 8(3-4):289–304.

Song, X., Zhang, Q., Sekimoto, Y., and Shibasaki, R. (2014). Prediction of human emergency behavior and their mobility following large-scale disaster. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 5–14. ACM.

Vazquez-Prokopec, G. M., Bisanzio, D., Stoddard, S. T., Paz-Soldan, V., Morrison, A. C., Elder, J. P., Ramirez-Paredes, J., Halsey, E. S., Kochel, T. J., Scott, T. W., et al. (2013). Using gps technology to quantify human mobility, dynamic contacts and infectious disease dynamics in a resource-poor urban environment. *PloS one*, 8(4):e58802.

Wang, X. and Hamilton, H. J. (2003). Dbrs: a density-based spatial clustering method with random sampling. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 563–575. Springer.

Yin, J., Gao, Y., Du, Z., and Wang, S. (2016). Exploring multi-scale spatiotemporal twitter user mobility patterns with a visual-analytics approach. *ISPRS International Journal of Geo-Information*, 5(10):187.

Yuan, J., Zheng, Y., and Xie, X. (2012). Discovering regions of different functions in a city using human mobility and pois. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 186–194. ACM.

Zaïane, O. R. and Lee, C.-H. (2002). Clustering spatial data when facing physical constraints. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 737–740. IEEE.

Spatial-temporal Analysis of active fire classified by INPE's Fire Risk Model in Brazil using Python language

Gabriel Máximo da Silva[1], Bruno Vargas Adorno[1], Gilberto Ribeiro Queiroz[1], Thales Sehn Körting[1], Fabiano Morelli[1], Silvana Amaral[1], Yosio Edemir Shimabukuro[1]

[1] Divisão de Observação da Terra e Geoinformática - Instituto Nacional de Pesquisas Espaciais (INPE)
12.227-010 – São José dos Campos – SP – Brazil

{gabriel.maximo,bruno.adorno,gilberto.queiroz,thales.korting,
fabiano.morelli,silvana.amaral,yosio.shimabukuro}@inpe.br

Abstract. *The use of programming languages such as Python to analyse geospatial data has simplified the analysis of remote sensing data. In this study, we evaluated the spatial-temporal distribution of active fires detected by MODIS in Aqua Satellite by classes of fire risk, as defined by INPE's Fire Risk model. The study area was the federative units, geographical regions, and biomes in Brazil. The temporal assessment comprised different time seasons and the years from 2015 to 2019. Active fires were higher during winter and spring seasons when also more Critical and High risks were noticed. The same risk was prevalent in Northeastern and Southeastern Brazil, as well as in the Caatinga and Cerrado biomes.*

1. Introduction

Fire in natural environments is the result of a combination of factors, such as vegetation type and climate combined with human actions or natural causes [Fearnside 2006, Phillips et al. 2009]. Therefore, it requires a complex territory-level network to be properly managed. For such endeavors, large databases are necessary to gather information aiming the support of fire monitoring as a way of preventing serious forest fires, especially in fire-prone biomes, such as the Brazilian Cerrado [Schmidt et al. 2016, Tedim et al. 2016].

Fire use can be convenient for anthropological purposes but its abusive use can worsen health problems, due to smoke-carrying combustion products [Souza et al. 2012]. Also, it features a positive feedback relationship with climate change, which can cause harm to fire-prone ecosystems. Lately, these ecosystems have had more contact with fire, especially in the Amazon [Aragão et al. 2018]. Monitoring fire risk is, therefore, necessary for land management over time and within the same year, according to the region under analysis. Considering a continental country like Brazil, with such heterogeneity of vegetation cover, time seasons, land use, and occupation, a complex regional monitoring system is needed [Nogueira et al. 2017].

As consequence, another typical challenge on this issue is dealing with large volumes of data. In Brazil, the National Institute for Space Research (INPE) gathers a database with more than 10 satellites, capable of detecting the occurrence of active fire, generally indicating fire occurrences. Besides, Weather Forecasting and Climate Studies Center – CPTEC/INPE, has implemented fire risk observation and forecasting models, including as variables: vegetation cover, accumulation of days without rain, air temperature, altitude and latitude, and the active fire observation itself [Setzer et al. 2019].

The use of computational languages, such as Python allows the creation and documentation of geospatial data processing architectures. By doing so, the scripts can either be improved in different versions afterward or be easily adapted to other regions of interest [Teodoro and Duarte 2013]. In this sense, such routines related to fire monitoring can be adjusted to allow more input data or even improve time series to perform different analyses and their new results are easily compared to previous ones [Gomes et al. 2017].

Based on the preceding, this work aimed to evaluate active fire occurrence relative to different risk classes, according to the Fire Risk Model developed by INPE's Wildfire Monitoring Program. For this purpose, the analysis was performed across all Brazilian regions, states, and biomes, also considering different seasonality, using programming languages to provide adaptive and replicable processing routines.

2. Material and Methods

The following sections give details on the study area, data, and processing routines chosen for this study. In summary, active fire classified by fire risk were spatially distributed, considering seasonal assessments from 2015 to 2019. Python libraries inserted in the programming routine, which was developed and documented in a Github platform using Jupyter Notebook application, supported all the processing and plotting of main results.

2.1. Study Area

Three Brazilian territory delimitations were considered as spatial analysis unit: Federative Units, or States, Geographical Regions, and Biomes (Figure 1). Either considering the regions or states, the analysis intends to show important results for administrative purposes [Nogueira et al. 2017]. Moreover, important insights can be realized regarding the active fires on the biome delimitation, according to its specific vegetation and other environmental characteristics.

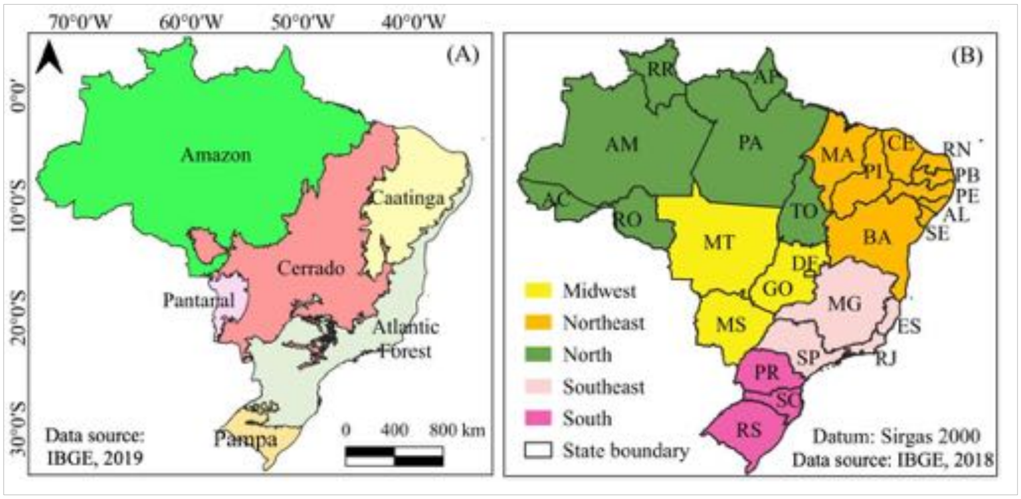


Figure 1. Study area and spatial analysis units: A) Brazilian Biomes B) Geographical Regions and Federative Units (states).

Besides, active fires were also evaluated for different seasons (summer, fall, winter, and spring) from 2015 to 2019. Fundamentally, time assessment may stress the possibilities of different patterns in the monitored events during a defined period.

Identifying these spatial and temporal fire risk patterns are relevant considering the changes in climate the world is facing lately or even the planning of public policies directed to land and environmental administration [Fonseca et al. 2019].

2.2. Database

The database for this study were mainly active fire points (shapefiles) from January 1st to December 31st between 2015 and 2019, available at INPE's Wildfire Monitoring website. Only data from the reference satellite (Aqua afternoon) were selected for this study. The active fire data already bring attributes on date/time, state, biome, and fire risk when registered. Only geographical region delimitation was complemented with data from the Brazilian Institute of Statistics and Geography (IBGE). Season information derived from date/time attribute in terms of days of the year. Summer was defined from December 21st to March 20th; Fall, from March 21st to June 20th; Winter, from June 21st to September 20th, and Spring, from September 21st to December 20th.

2.3. Data processing and analysis

For data processing and analysis, we used the following Python's libraries: Pandas, Geopandas, Numpy, and Matplotlib. The entire sequence of commands was organized into two Jupyter Notebooks and a module, named as riscofogo.py[1]. Some active fires, whose fire risk values were null, were excluded from the analysis. A random 5% sample was selected from active fire data in each a series of routines were applied to consolidate all other necessary information (time seasons and regions) in the same Geodataframe (i.e Geopandas' data organized in table structure including the object geometry as an attribute).

Numpy, Pandas, Geopandas were libraries applied for organizing the data and operating on them. Thus, logic operators to derive time season attributes from date/time and spatial join function were used to include geographical region classes in active fire Geodataframe. Another fundamental process was to define classes for fire risk attributes using logical operators, according to Setzer et al. (2019) classification (Table 1).

Table1. Risk classes according to Fire Risk Values registered in active fire database

Risk classes	Fire Risk Values (RF)
Minimum	RF < 0,15
Low	0,15 < RF ≤ 0,40
Medium	0,40 < RF ≤ 0,70
High	0,70 < RF ≤ 0,95
Critical	RF > 0,95

Adapted from Setzer et al. (2019)

Pandas' Pivot Table function was applied to summarize the results per season, geographical regions, biomes, and federative units, which were then represented in stack horizontal bars, showing the relative occurrence of active fire in different risk classes. Figures and maps showing active fire distribution in the study area were plotted using Matplotlib. Other details can be found in the Jupyter Notebooks developed for this work.

[1] https://github.com/ser-347/risco-de-fogo

3. Results

We observed more occurrences of active fire in the winter and spring seasons. Notably, there is a pattern that persists for the entire time studied. The Critical risk class showed more active fires in the Midwest and Southeast during winter and the Northeast during spring (Figure 2). There are also considerable active fires in the Minimum risk class, mainly in the Amazon and Pampa biomes. Likewise, in the summer, states such as Pará (PA), Mato Grosso (MT), Rondônia (RO), and Tocantins (TO), which are in the Arc of Deforestation, presented a substantial number of active fires with Minimum risk.

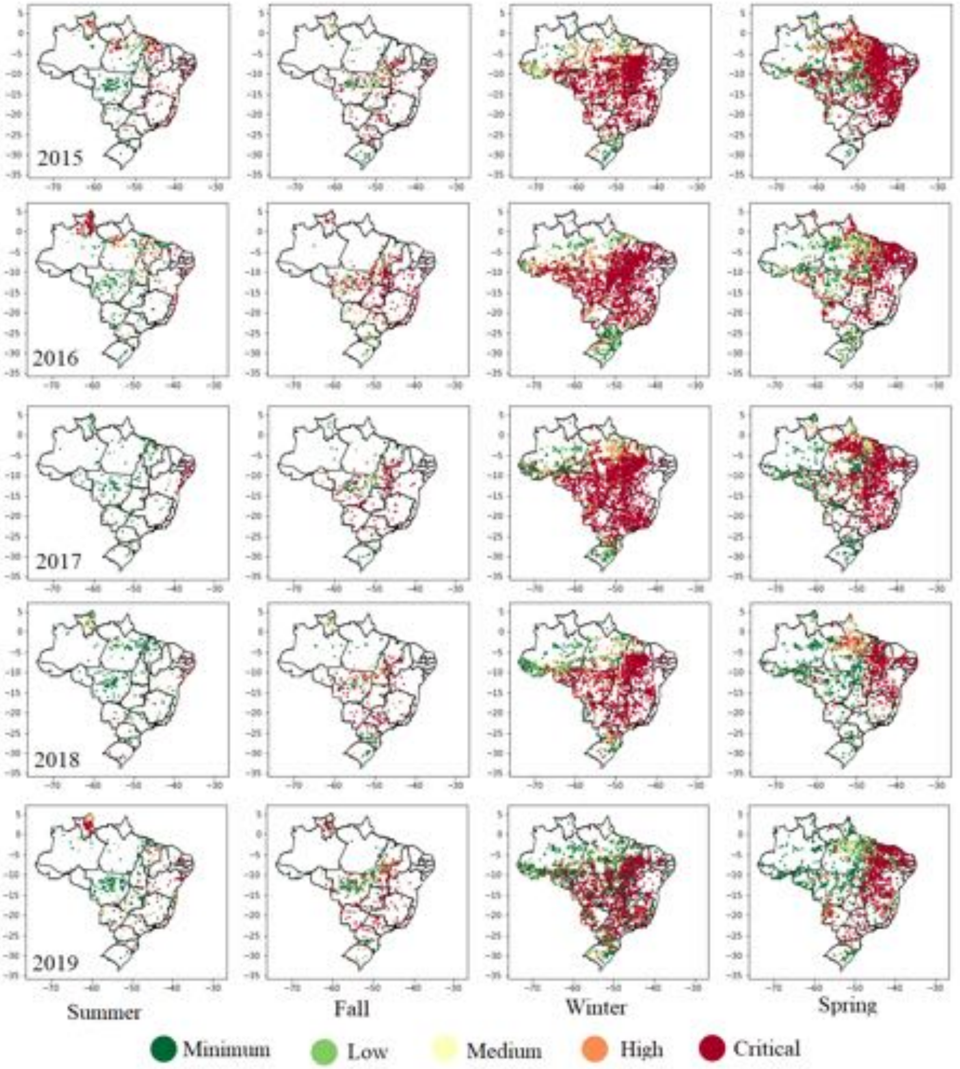


Figure 2. Spatial distribution of active fire in Brazil per fire risk classes from 2015 to 2019 and in different seasons.

Considering the absolute number of active fires, the Amazon biome exhibited more active fires in the Critical than in the Minimum class between 2015 and 2017, but this pattern reversed from 2018 onwards. In 2019, there was a peak of 1,900 active fires in the Minimum risk class for the Amazon, reaching about twice the amount of Critical risk active fires for the same year in the same biome. In contrast, although active fires were classified as Minimum risk increased, Critical risk class was dominant in the Cerrado (Figure 3).

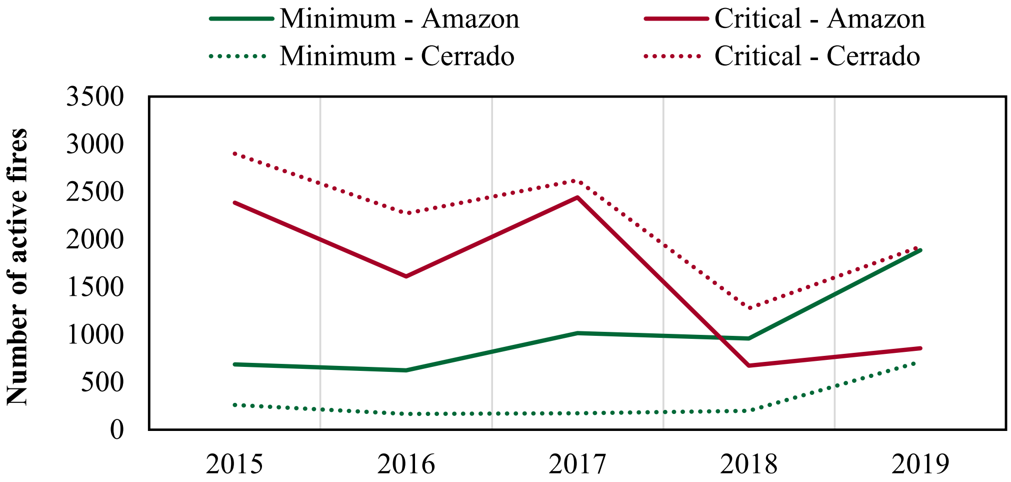


Figure 3. Number of active fires classified as Minimum and Critical fire risk for the Amazônia and Cerrado biomes between 2015 and 2019.

The mean relative occurrence of active fire by seasons, regions, biomes, and states are shown in Figure 4. The model was more accurate during the winter and spring seasons, followed by fall and summer when more active fires were detected at Minimum risk locations (Figure 4.A). Regarding Brazilian administrative regions (Figure 4.B), Northeast and Southeast presented about 80% of active fire classified as Critical risk, whereas North and South less than 40%. Midwest showed almost 60% of active fire in the Critical risk region.

Given the Brazilian biomes (Figure 4.C), the patterns are somewhat related to those observed among geographical regions. The Caatinga biome (located mainly in northeastern Brazil) presented a greater relative occurrence of active fire in Critical risk areas (about 90%), followed by the Cerrado (about 80%). The opposite took place in the Pampa biome, which is found in southern Brazil, followed by the Amazon, showing less than 40% of active fires classified as Critical risk.

Finally, with concerns to the Brazilian States (Figure 4.D), they are somehow linked to the results presented in the geographic regions (Figure 4.B). For instance, states like the Rio Grande do Norte (RN), Sergipe (SE), and Piauí (PI), which are located in the Northeast, and São Paulo (SP), Rio de Janeiro (RJ), and Minas Gerais (MG), in the Southeast, were among those with the greater relative occurrence of active fire classified as a Critical risk. Conversely, Acre (AC), Amazonas (AM) from the North or the Rio Grande do Sul (RS) and Santa Catarina (SC) from the South showed the greater occurrence of active fire classified as Minimum risk.

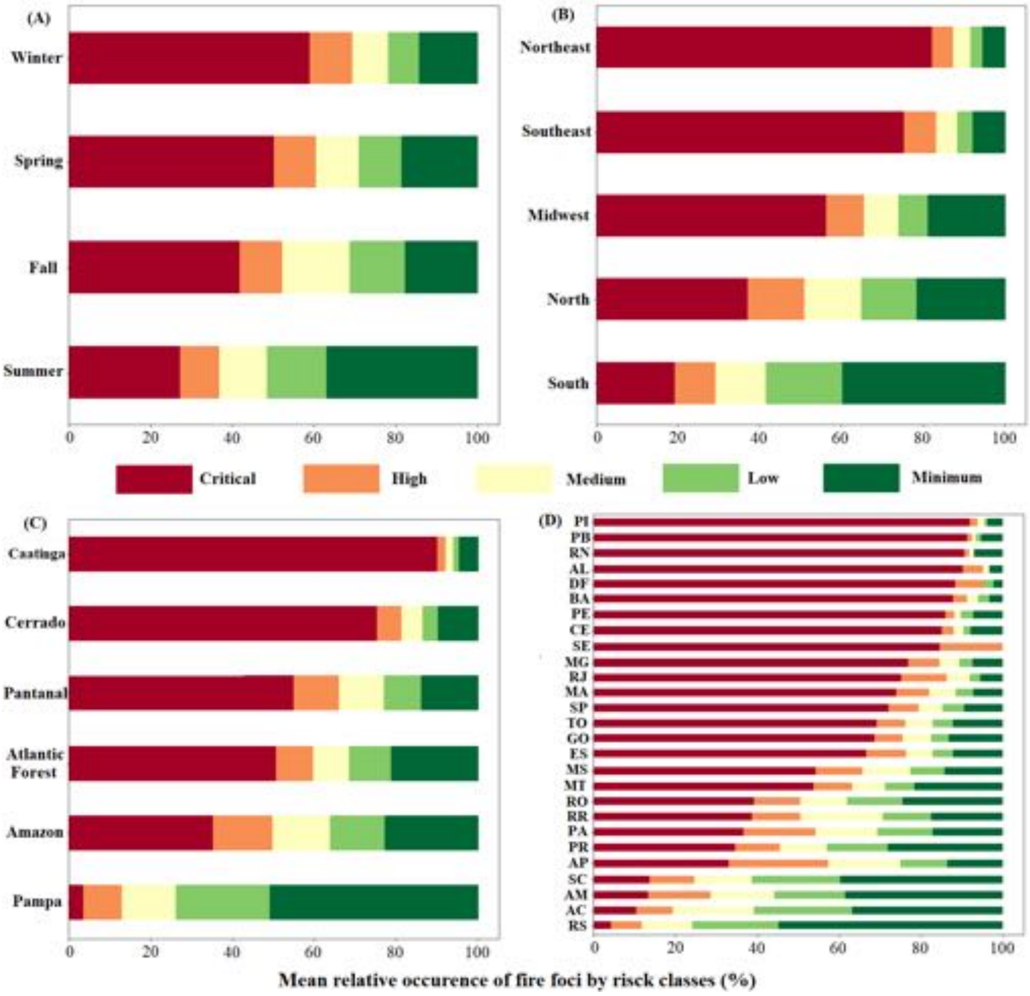


Figure 4. Mean relative occurrence of active fire by fire risk classes in different seasons (A), regions (B), biomes (C), and states (D). The bars were arranged in descending Critical risk order.

4. Discussion

Python libraries used in this study enabled active fire analysis in different states, biomes, regions, and seasons in Brazil from 2015 to 2020. By documenting and making the routines available in an open source platform, however, one could easily reproduce them in other analysis units (e.g. municipal level) and/or time span.

A possible limitation to consider before replicating this analysis is the spatial resolution of the model. Its input variables are typically in kilometric-scale, which may restrict local studies depending on the size of the study area. However, taking advantage of current open source culture and the popularization of geoinformatics, new modeling of fire risk with local variables (generated from finer spatial resolution) is encouraged. Once developed, its architecture could be shared, so other researchers could either test it, by applying variables from their research area or contribute to its processing efficiency [Teodoro and Duarte 2013].

Our results indicated to be coherent to other studies [Nogueira et al. 2017]. More than 80% relative occurrence of fire in Critical or High-risk areas from Cerrado and Caatinga, for instance, maybe explained by many days of drought and available biomass as fire fuel in those biomes, which are important parameters for the fire risk model [Setzer et al. 2019]. Fire is commonly used as a technique for deforestation and pasture management in the Midwest [Schmidt and Eloy 2020] and in the North and Northeast regions, to where agricultural frontiers are expanding [Aragão 2018, Silva 2020].

On the other hand, in the Amazon or Pampa active fires were majorly classified as Minimum to Medium risk. According to the model's parameters, Lower risks may be associated to regions with dense forest vegetation and frequent rainfall as commonly seen in the Amazon, or Lower mean temperatures, as typically noticed in the Pampa [Setzer et al. 2019]. Perhaps, considering the current shift in political scenario [Fonseca et al. 2019, Amigo 2020] and agricultural expansion in the Amazon biome, other parameters should be considered for future modeling, such as areas threatened by deforestation, as they have been increasingly associated with fire events [Silvério et al. 2019].

Between 2015 and 2019, INPE's database recorded more than 19,000 active fires spread in the Amazon. Even though 2017 had the largest absolute number (5,336 active fires), 2019 had the highest percentage of active fires from deforested areas (~34%) [Silvério et al. 2019]. This pattern has changed over the past years and mainly after the current Brazilian government's antienvironmental agenda [Escobar 2019]. Therefore, despite important Brazilian monitoring programs such as the Real-Time Deforestation Detection System (DETER) and the Program for Monitoring Deforestation of the Amazon by Satellite (PRODES), and the Wildfire Monitoring program itself, more stakeholders, including politicians, are needed to mitigate this problem.

Finally, even though uncertainties may be expected in fire risk modeling, the ongoing active fire monitoring is necessary to support the environmental agenda in Brazil. Since 2004, Brazilian policies have played an important role to reduce deforestation and carbon emissions and thus helping improve air quality and human health [Reddington et al. 2015, Wiedinmyer 2015, Oliveira 2020]. However, as stated before, this scenario has changed over the past four years, when decision-makers have been favoring agricultural expansion over natural areas, besides dismantling environmental laws [Fonseca et al. 2019, Take action to stop the Amazon burning 2019, Amigo 2020]. Consequently, respiratory problems may be aggravated by the constant burning of vegetation at large distances due to particulate transport through the atmosphere [Aragão et al. 2020].

5. Conclusions

In this study we demonstrated how using programming language, like Python, may be useful for visualizing and analyzing geospatial data in time and space. Our focus was on active fire and risk distribution exploratory analysis throughout Brazilian states, regions, and biomes from 2015 to 2019 and in different seasons. Thus, it could be noted how fire risk and possible occurrence behave differently across the study area and time considered in the assessment.

Programming language contributed to the spatial-temporal analysis of active fire by risk classes and allowed documentation of every processing routine to adapt or update it in the future, whenever needed. Python's libraries like Pandas, Geopandas, Numpy, and Matplotlib were useful for processing and analyzing data, as well as presenting the results.

In addition, using GitHub as an open source development platform to host codes like the ones developed on these analyses may allow developers to share and adapt different versions of processing routines.

Active fires classified as High and Critical fire risk is more frequent during winter and spring. Regarding space, the same was verified in the Northeast and Southeast regions, as well as in the Caatinga and Cerrado biomes. In the Amazon we identified an interesting pattern of a greater relative register of active fires classified as Critical fire risk until 2017, which reversed from 2018 onwards, indicating a greater likelihood of fire occurrence in less fire-prone regions in the last years.

Lastly, although this study used different geographical limits in the analysis, fire negative effects are usually boundaryless. This implies that, in addition to all the monitoring work being carried out through satellite products, more actions should be taken by the government and other agencies to curb fire and activities associated with it, which might be challenging by itself, considering the climate change scenario.

References

Amigo, I. 2020. “The Amazon’s Fragile Future.” *Nature* 578(February): 505–7.

Aragão, L. E. O. C.; et al. 21st Century drought-related fires counteract the decline of Amazon deforestation carbon emissions. *Nature Communications*, v. 9, n. 1, pp. 1–12, 2018.

Aragão, L. E. O. C.; et al. O desafio do Brasil para conter o desmatamento e as queimadas na Amazônia durante a pandemia por COVID-19 em 2020: implicações ambientais, sociais e sua governança. São José dos Campos, 2020. 34p. SEI/INPE: 01340.004481/2020-96/5543324.

Escobar, H. Brazil’s deforestation is exploding—and 2020 will be worse. *Science* (80-.). 10 (2019), doi:10.1126/science.aba3238.

Fearnside, P. M. Desmatamento na Amazônia: dinâmica, impactos e controle. *Acta Amazonica*, v. 36, n. 3, p. 395–400, 2006.

Fonseca, M. G. et al. Effects of climate and land-use change scenarios on fire probability during the 21st century in the Brazilian Amazon. *Global Change Biology*, v. 25, n. 9, p. 2931–2946, 15 set. 2019.

Gomes, V.; et al. Um ambiente para análise exploratória de grandes volumes de dados geoespaciais: Explorando risco de fogo e focos de queimadas. Proceedings of the Brazilian Symposium on GeoInformatics, pp. 301–309, 2017.

IBGE – Instituto Brasileiro de Geografia e Estatística. Malha Municipal Digital da Divisão Político - Administrativa Brasileira. Versão 2018. Rio de Janeiro: IBGE, 2018. Retrieved from: https://www.ibge.gov.br/geociencias/organizacao-do-territorio/malhas-territoriais/15774-malhas.html?=&t=sobre.

IBGE – Instituto Brasileiro de Geografia e Estatística. Mapa de biomas e sistema costeiro-marinho do Brasil. Rio de Janeiro: IBGE, 2019. 1 mapa. Escala 1:250 000. Projeção policônica. Retrieved from: https://www.ibge.gov.br/geociencias/informacoes-ambientais/estudos-ambientais/15842-biomas.html?edicao=25799&t=acesso-aoproduto.

Nogueira, J. M. P.; et al. Spatial pattern of the seasonal drought/burned area relationship across Brazilian biomes: Sensitivity to drought metrics and global remote-sensing fire products. *Climate*, v. 5, n. 42, pp. 1-21, 2017.

Oliveira, G. de.; et al. Smoke pollution's impacts in Amazonia. *Science*. 2020;369(6504):634-635. doi:10.1126/science.abd5942.

Phillips, O. L.; et al. Drought Sensitivity of the Amazon Rainforest. *Science*, v. 323, n. 5919, p. 1344–1347, 6 Mar. 2009.

Reddington, C. L.; et al. Air quality and human health improvements from reductions in deforestation-related fire in Brazil. *Nature Geoscience*, v. 8, n. 10, p. 768–771, 16 out. 2015.

Schmidt, I. B.; et al. Experiências internacionais de manejo integrado do fogo em áreas protegidas – recomendações para implementação de manejo integrado de fogo no Cerrado. *Biodiversidade Brasileira*, v. 6, n. 2, pp. 41–54, 2016.

Schmidt, I. B.; Eloy, L. Fire regime in the Brazilian Savanna: Recent changes, policy and management. *Flora: Morphology, Distribution, Functional Ecology of Plants*, v. 268, n. October 2019, p. 151613, 2020.

Setzer, A. W.; Sismanoglu, R.A.; Santos, J.G.M. Método do cálculo do risco de fogo do programa do INPE - Versão 11, junho/2019. São José dos Campos: INPE, 2019. 27 p.

Silva, P. S.; et al. Drivers of Burned Area Patterns in Cerrado: The Case of Matopiba Region. n. Idl, p. 542–547, 2020.

Silvério, D. V.; et al. Amazônia em chamas. Brasília, DF. IPAM, 2019.

Souza, L. S. de; et al. Air quality photochemical study over Amazonia Area, Brazil. *International Journal of Environment and Pollution*, v. 48, n. 1–4, pp. 194–202, 2012.

“Take Action to Stop the Amazon Burning.” 2019. *Nature* 573(7773): 163–163. http://www.nature.com/articles/d41586-019-02615-3.

Tedim, F. et al.; A wildfire risk management concept based on a social-ecological approach in the European Union: Fire Smart Territory. *International Journal of Disaster Risk Reduction*, v. 18, pp. 138–153, 2016.

Teodoro, A. C.; Duarte, L. Forest fire risk maps: A GIS open source application - a case study in Norwest of Portugal. *International Journal of Geographical Information Science*, v. 27, n. 4, pp. 699–720, 2013.

Wiedinmyer, C. Breathing easier in the Amazon. *Nature Geosci* 8, 751–752 (2015). https://doi.org/10.1038/ngeo2550.

Evaluating the usage of exact queries on 3D spatial databases

Matheus A. de Oliveira[1], Marcelo de M. Menezes[1], Salles V. G. de Magalhães[1],Bruno F. Coelho[1]

[1]Departamento de Informática, Universidade Federal de Viçosa (UFV)
Campus da UFV, Viçosa, MG, Brazil

{matheus.a.aguilar, marcelo.menezes, salles, bruno.f.coelho}@ufv.br

Abstract. *The availability of big geospatial databases has increased the necessity of having efficient algorithms for processing them. Furthermore, as datasets grow, the chance of having failures due to rounding-errors increases, which makes exact (but typically slower) algorithms more important. This paper presents an evaluation of PostGIS exact and approximate backends. In our experiments, the exact backend was up to* 27 *times slower than the approximate one. We also observed that the straightforward usage of some spatial queries may lead to a poor performance, what requires more care when they are programmed. These results suggest applications requiring exact computation could benefit from the development of faster exact backends, which is the long-term goal of the research project this paper is part of.*

1. Introduction

The ability of storing and processing 3D data in Geographic Information Systems (GISs) has become very important. This type of modeling is especially necessary in areas such as urban planning, environmental monitoring, telecommunications, rescue operations, landscape planning, geology and mining [Zlatanova et al. 2002].

Despite such importance, this processing still faces a major challenge: performing robust computation while maintaining a good performance. This is fundamental for current data sets, since their size and complexity have been increasing, which makes them more prone to roundoff errors caused by floating point arithmetic. This kind of error is especially problematic, since it can propagate and generate inconsistent results or even cause systems to crash [Goodchild and Gopal 1989] .

This work has been developed in the scope of a project whose long-term goal is to optimize (using *GPUs*) exact geometric algorithms and spatial databases.

This paper describes our first case studies: the calculation of exact 3D intersection between segments and triangulated meshes and exact 3D intersection between triangles, using PostGIS, an extension for spatial data from the PostegreSQL Database Management System (DBMS), through its exact backend SFCGAL. This DBMS was chosen because it is open source, widely used and is one of the systems with the best support for spatial data with both exact and approximate arithmetic [Real et al. 2019]. The goal was to evaluate the support for 3D spatial data and the performance obtained by different queries.

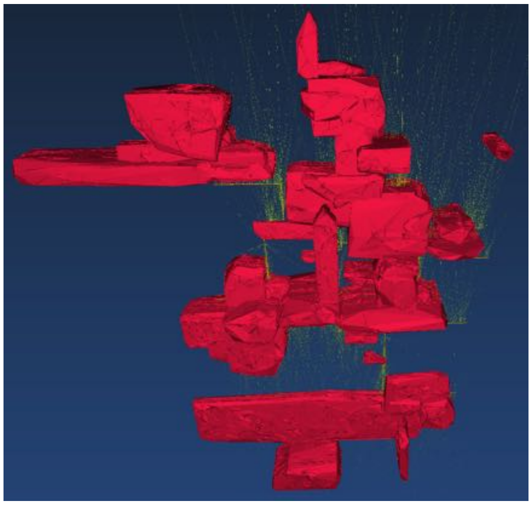


Figure 1. An example of a synthetic mine model. Source: [Real 2020]

2. Case Studies

As a case study, the following problems were considered: given a set of line segments and a set of triangulated meshes, both in 3D, to detect which pairs of segments and triangles do intersect and given two sets of triangles, also in 3D, find the pairs of intersecting triangles. These problems have several applications in computational geometry, GIS, CAD, etc. An example of application is studying the interaction of geological objects in a 3D mining model.

Thus, a 3D mining model provided by [Real et al. 2019] was employed in our experiments (Figure 1 illustrates this dataset, drill holes are in green and minerals are in red). In this domain, the intersection of segments with 3D objects is employed by geologists for studies related to the intersections between survey drill holes (represented by segments) and mineral layers (represented by triangulated meshes). The goal is to verify which layers of minerals were reached by each one of the drill holes in order to estimate the amount of ore that can be extracted.

The intersection of pairs of triangles, on the other hand, may be employed to to intersect a mining model with a shape representing an excavation area. This computation would result in shapes representing the minerals that could be extracted by the excavation.

Geologists typically use spatial queries (such the ones provided by PostGIS) to do these studies. However, the processing time spent by these systems is often prohibitive [Real et al. 2019].

2.1. PostGIS

The tool used to perform the experiments was PostGIS, since it has a large amount of resources for 3D geometries, such as: geometric data types, spatial indexes and intersection functions, unlike other database management systems [Real et al. 2019].

In addition, PostGIS has another important functionality to obtain robustness in spatial data: a backend capable of performing operations with exact arithmetic, the SFCGAL. This backend employs the CGAL library to achieve exactness.

The following query illustrates a problem that occurs when computation is performed using floating point arithmetic (subject to errors). This query checks whether the intersection point between two segments intersects one of the segments (clearly such an intersection occurs). However, as mentioned by [Mercier 2013], this specific intersection is not detected when the (inaccurate) GEOS backend is used (on the other hand, it is correctly computed by SFCGAL). While these errors may be rare, they make software relying on floating-point arithmetic unreliable.

```
SELECT ST_Intersects(ST_Intersection(
  'LINESTRING(0 0,2 1)'::geometry,'LINESTRING(1 0,0 1)'::geometry),
  'LINESTRING(0 0,2 1)'::geometry);
```

SFCGAL uses the *kernel* of CGAL which ensures that both geometric constructions (for example, points constructed as the intersection of two segments) and predicates (e.g., detecting whether two segments do intersect) are performed exactly. However, such an implementation has some challenges, such as the difficulty of parallelization (the *kernel* is not *thread-safe*) and the computational cost is higher than that obtained with floating point arithmetic[The CGAL Project 2019].

In the experiments, both GEOS (inaccurate) and SFCGAL (exact) backends were evaluated in order to assess the impact of exact arithmetic on the performance of the queries.

The data was modeled using tables with the following internal structure: the table of drillholes was composed of objects of type *LineString Z* whereas the mineral layers were divided into a set of indexed triangles, which were stored in a table of objects of type *Polygon Z*.

In addition, two types of spatial indices were evaluated in the table geometries, a 2D GiST index (PostGIS default), that drops the Z coordinate and is applied to a projection of the objects in a plane, and a GiST 3D index, that uses all the coordinates. Thus, it focused on assessing which index was most suitable for this data set, since the 3D index is slightly more computationally expensive, but allows for greater filtering on queries.

Another strategy employed in the experiments was to change the cost value of the intersection functions to $100,000$, in order to force the query planner to perform parallel scans instead of sequential ones (as suggested by [Real et al. 2019]).

3. Results and Discussion

The main idea of the experiments was simulate queries that would be useful in the field of mining. First, we considered the problem of detecting intersections between line segments (drill holes) and 3D objects (minerals) represented by triangles. In the segment/object intersection experiments we employed the dataset provided by [Real et al. 2019], which contains $7,846$ segments, 71 objects and the objects are composed of a total of $3,215,052$ triangles. In the triangle/triangle intersection experiments we employed two datasets (generated with the syntetic mine maker available at [Real 2020]) representing mines and containing, respectively, $125,258$

and $138,964$ triangles. All the experiments were performed on a machine with a Ryzen 5 1600 AMD CPU with 6 cores at 3.2 GHz, $16GB$ of RAM, Kingston A400 SSD, Ubuntu Linux 20.04, PostgreSQL 12.4 and PostGIS 2.5.5. We evaluated both the exact (SFCGAL 1.3.8) and approximate (GEOS 3.8.0) PostGIS backends.

This could be evaluated by selecting pairs of segments and objects that do intersect. However, PostGIS spatial index would index the objects, which leads to a poor performance because, after culling the pairs of segments/objects that may intersect, PostGIS would process each pair, potentially testing the segment for intersection against all the triangles in each object. To improve this performance, we created a table of triangles (each row contains a triangle, its id and *objectId* the id of the 3D object it belongs to) with a spatial index on the geometry. Since the spatial index is now on the triangle level, it can perform a better culling. Considering this table, the intersections can be found with the query *SELECT DISTINCT s.id, t.objectId FROM Triangles t, Segments s WHERE ST_3DIntersects(s.geom, t.geom)* (this approach will be referred as *DISTINCT* in this section).

A drawback with the previous strategy is that, given a segment s, the query planner tests s for intersection with many triangles from the same $3D$ object and, only after all intersections are detected, the unique intersections are filtered. In order to try to obtain a better performance, one could try to employ an approach using an exists clause in order to try to avoid this. Thus, we also evaluated the following approach: *SELECT s.id, o.id FROM Segments s, Objects o WHERE o.geom && s.geom AND EXISTS(SELECT 1 FROM Triangles t WHERE t.objectId = g.id AND s.geom ST_3DIntersects(t.geom, s.geom))* (this approach will be referred as *EXISTS*). This strategy employs two levels of indexing: the *select* clauses selects pairs of segments and objects that may intersect (using an indexed bounding-box check on the object level). Then, for each pair of segment s and object o that may intersect, the exists clause checks if there exists a triangle t from o intersecting s (this step employs the indexing at the triangle level).

The previous two queries present a pitfall that may degrade the performance of naive solutions: even 3D predicates (such as *ST_3DIntersects*) employ 2D indices in PostGIS by default (even when a 3D index is available). Internally, *ST_3DIntersects* is implemented using a 2D bouding-box intersection test using the && operator (which employs a 2D index) followed by a call to the _ST_3DIntersects function that tests the pair for intersection after the bounding-box detects a potential intersection. Thus, the culling is performed by evaluating the projection of the geometric data onto the xy plane.

In order to actually use a 3D index, one should add to the query a 3D bounding-box intersection test (using the &&& operator). We evaluated queries using both the 2D and 3D indices in order to show how the performance of a naive solution could be affected.

Table 1 presents the times (in seconds) obtained by these two approaches using the two kinds of indices and the two options of backend.

As it can be seen, the 3D index significantly reduces the running times in comparison with the 2D index. The performance improvement is higher when the

	2D index		3D index	
Query	GEOS	SFCGAL	GEOS	SFCGAL
seg/tri (DISTINCT)	71.7	2352.8	26.7	220.4
seg/tri (EXISTS)	238.1	26354.8	167.6	318.0
Triangle/triangle	25.8	1625.0	3.1	83.4

Table 1. Times (in seconds) for detecting segment/triangle intersections (first and second rows) using the two different queries (DISTINCT and EXISTS) and for testing pairs of triangles for intersection (third row).

Index	Without the Index	With the index
2D	25,225,297,992	89,523,915
3D	25,225,297,992	7,571,816

Table 2. Number of pairs of segments and triangles tested for intersection considering the 2D and 3D indices

exact backend is employed (for example, considering the *DISTINCT* queries, using the 3D index leads to a running time 11 times smaller when SFCGAL is employed, while the difference when GEOS is employed is 3 times). This can be explained because of the higher cost of evaluating geometric predicates using SFCGAL associated to the better culling of the 3D index, which performs a more significant reduction (in comparison with the 2D index) in the number of intersection predicates that actually need to be evaluated after the culling.

Considering the 3D index and the fastest query (the *DISTINCT* one), the exact backend was 8 times slower than the inexact one. While in some applications this difference is acceptable, in big datasets and applications requiring fast answers this may not be suitable.

Table 2 presents the number of intersection tests performed when the 2D and 3D indices are employed. As it can be seen, the $3D$ index reduces in 12 times the number of pairs being evaluated (this reduction explains the performance improvement obtained with the 3D index) in comparison with the 2D index (the default one employed by PostGIS).

Considering the triangle/triangle intersection tests (third row of Table 1), we employed a straightforward query for the tests: *SELECT COUNT(*) FROM triangles1 AS t1, triangles2 AS t2 WHERE t1.geom OP t2.geom AND _st_3dintersects(t1.geom, t2.geom);*, where OP is && for the 2D index and &&& for the 3D index.

Since testing a pair of triangles for intersection employs more arithmetic operations than intersecting segments with triangles, the performance difference between the exact and approximate backends was more noticeable than in the segment/triangle tests. Considering the $3D$ index, SFCGAL was 27 times slower than GEOS.

4. Conclusions and future work

This paper presented a performance analysis of PostGIS over 3D spatial data on a mining dataset. Our experiments have showed that some naive queries performed on PostGIS could present a high performance penalty. For example, applying a trivial intersection test on 3D data uses, by default, 2D bounding-box tests and indices, which was up to 83 times slower than the query employing the 3D index.

We also evaluated PostGIS exact and approximate backends. Experiments in two kinds of analysis showed a performance difference of 8 and 27 times between the two backends (when a 3D index was employed). This suggests employing faster techniques for exact computation could benefit applications demanding both exactness and performance. This could be particularly important for big datasets containing millions of features.

Researchers have recently been employing GPUs for accelerating queries employing the approximate backend [Real et al. 2019]. Similarly, in a previous paper we have proposed the use of GPUs for accelerating the exact evaluation of geometric predicates [Menezes et al. 2019], which led to a performance improvement of up to 40 times over the sequential implementation. As future work, we intend to combine the two ideas, i.e., accelerate the exact PostGIS backend with GPUs. Thus, GIS applications requiring exactness could benefit from this performance improvements while also benefiting from the modularity and simplicity of a DBMS.

References

Goodchild, M. F. and Gopal, S. (1989). *The accuracy of spatial databases.* CRC Press.

Menezes, M. M., Magalhães, S. V. G., Franklin, W. R., de Oliveira, M. A., and Chichorro, R. E. O. B. (2019). Accelerating the exact evaluation of geometric predicates with GPUs. In *28th International Meshing Roundtable*, Buffalo, NY, USA.

Mercier, H. (2013). 3d and exact geometries for PostGIS. `https://wiki.postgresql.org/images/3/36/Postgis_3d_pgday2013_hm.pdf`. (Retrieved on 09/02/2020).

Real, L. C. V. (2020). Synthetic mine maker. `https://github.com/lucasvr/synthetic-mine-maker`. (Retrieved on 09/24/2020).

Real, L. C. V., Silva, B., Meliksetian, D. S., and Sacchi, K. (2019). Large-scale 3d geospatial processing made possible. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 199–208.

The CGAL Project (2019). *CGAL User and Reference Manual.* CGAL Editorial Board, 4.14.1 edition.

Zlatanova, S., Rahman, A., and Pilouk, M. (2002). 3d gis: current status and perspectives. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(4):66–71.

Integração dos ambientes Brazil Data Cube e Open Data Cube

Felipe Menino Carlos[1]**, Vitor C. F. Gomes**[2]**, Gilberto Ribeiro de Queiroz**[1]**,**
Karine Reis Ferreira[1]**, Rafael Santos** [1]

[1]Instituto Nacional de Pesquisas Espaciais (INPE)
12227-010 – São José dos Campos – SP – Brasil

[2] Divisão de C4ISR – IEAv/DCTA
12.228-001 – São José dos Campos – SP – Brasil

{felipe.carlos,gilberto.queiroz,karine.ferreira}@inpe.br
rafael.santos@inpe.br, vitorvcfg@fab.mil.br

Abstract. *The availability of large quantities of Earth observation data by satellite images has enabled different technologies and research. Given the challenges that this volume of data presents, the organization of these in data cube formats becomes fundamental for any large scale study. Providing tools that use these data cubes simpler allows their widespread use in various contexts. This article aims to present the ongoing work on the extension of the data cube analysis and visualization tools produced by the Brazil Data Cube project by integrating these tools with the Open Data Cube framework.*

Resumo. *A disponibilização de grandes quantidades de dados de observação da Terra por imagens de satélite tem possibilitado o desenvolvimento de diferentes tecnologias e pesquisas. Frente aos desafios que esse volume de dados apresenta, a organização desses em formatos de cubos de dados passa a ser fundamental. Disponibilizar ferramentas que tornem o uso desses cubos de dados mais simples, permite sua ampla utilização em diversos contextos. O presente artigo tem por objetivo apresentar o trabalho em andamento da extensão das ferramentas de analise e visualização dos cubos de dados produzidos pelo projeto Brazil Data Cube através da integração dessas com o framework Open Data Cube.*

1. Introdução

Nos últimos anos, a ciência e a indústria geoespacial têm desenvolvido inúmeras aplicações inovadoras graças à maior disponibilidade de dados de observação da Terra (EO, do inglês *Earth Observation*). Tanto os avanços tecnológicos dos equipamentos de coleta de dados quanto de armazenamento, aliadas à adoção de políticas de dados abertos pelas agências espaciais, têm propiciado a criação dessas aplicações [Soille et al. 2018]. No entanto, lidar com esses conjuntos massivos de dados ainda representa um grande desafio para a extração de todo o potencial e valor destes [Appel and Pebesma 2019]. Frequentemente, esses dados excedem as capacidades de memória, armazenamento e processamento de computadores tradicionalmente utilizados para esta finalidade [Câmara et al. 2014].

Para lidar com estes desafios, a comunidade científica tem utilizado o conceito de *Earth Observation Data Cube* (EODC), que através de conjuntos especializados de tecnologias, busca solucionar os problemas com grandes volumes de dados [Giuliani et al. 2020]. Uma das principais plataformas que tem se destacado nesse cenário é o Open Data Cube (ODC) [Gomes et al. 2020], que possui várias ferramentas e serviços para a análise e gerenciamento de grandes volumes de dados e que vem sendo utilizada por diversas iniciativas e instituições ao redor do mundo [Dhu et al. 2019].

No contexto brasileiro, o projeto Brazil Data Cube (BDC) é uma iniciativa criada em 2019, pelo Instituto Nacional de Pesquisas Espaciais (INPE), que tem por objetivo produzir cubos de dados multidimensionais para todo o território brasileiro, possibilitando a geração e análise de informações sobre o uso e cobertura do solo através de diferentes métodos, como a análise de séries temporais e uso de algoritmos de Aprendizado de Máquina. Para realizar suas atividades, o projeto BDC atualmente utiliza tecnologias de *Big Data* e ambientes de computação em nuvem para processar os cubos de dados [Ferreira et al. 2020].

Esse artigo apresenta o trabalho em andamento de integração entre os produtos de *software* do projeto BDC e a plataforma Open Data Cube. O objetivo dessa integração é possibilitar o acesso, processamento e análise das coleções de imagens e dos cubo de dados gerados pelo projeto BDC na plataforma ODC. Essa integração amplia os serviços e ferramentas que podem ser utilizadas para acessar, visualizar e analisar os dados produzidos pelo projeto BDC.

2. Open Data Cube (ODC)

O ODC é um *framework* analítico composto por uma série de estruturas de dados e ferramentas que facilitam o gerenciamento e análise de dados de EO [Gomes et al. 2020]. Ele permite a catalogação de conjuntos massivos de dados *raster*, possibilitando também o trabalho com dados que possuem alta dimensionalidade temporal. O ecossistema do ODC é composto pelos seguintes componentes [Open Data Cube 2019]:

- **Ferramentas de linha de comando**: Ferramentas para o gerenciamento dos dados registrados no ODC;
- **ODC Explorer**: Interface *web* que possibilita aos usuários explorar e buscar os dados que estão registrados no ODC;
- **ODC Stats**: Aplicação que facilita a análise estatística de grandes conjuntos de dados *raster*;
- **Web User Interface**: Interface para a visualização interativa dos resultados de execução dos algoritmos de análise;
- **OGC Web Services**: Serviços que possibilitam o uso interoperável dos dados registrados no ODC; e
- **Interface de programação (API)**: API em linguagem Python que possibilita a busca, acesso, análise e visualização dos dados, podendo ser utilizado junto ao ambiente interativo `Jupyter Notebook`.

Além desses, o ecossistema do ODC possui o componente `ODC Core`, uma camada entre os conjuntos de dados e os componentes citados anteriormente, responsável por fornecer uma estrutura analítica e de catalogação, capaz de lidar com um conjunto massivo de imagens de Sensoriamento Remoto. Para seu funcionamento, o ODC Core utiliza o conceito de indexação, que é o processo responsável por catalogar os dados que estarão disponíveis para uso. Nesse processo, são registrados os metadados das imagens e os locais de armazenamento, que pode ser um sistema de arquivos distribuídos ou um serviço de armazenamento na nuvem. Uma vez registrados, os dados podem ser consumidos pelos demais componentes do ecossistema ODC.

3. Integração BDC-ODC

Com o objetivo de permitir que os produtos de dados gerados pelo BDC sejam disponibilizados por meio das ferramentas e serviços do ODC, faz-se necessária a integração dos dois ambientes. Para o processo de integração, três fases foram definidas. A primeira delas trata da indexação dos dados do BDC dentro do ODC Core. A segunda fase realiza a seleção, integração e configuração dos serviços e ferramentas do ODC, de modo que essas permitam o uso dos dados do BDC considerando suas características específicas. Por fim, na terceira fase, faz-se a criação de um projeto de infraestrutura computacional que permita aos usuários consumir facilmente as ferramentas e facilidades resultantes dessa integração. Os passos realizados em cada uma dessas etapas são especificados nas próximas subseções.

3.1. Indexação

O primeiro passo necessário para a integração entre os ambientes é a indexação dos produtos de dados disponibilizados pelo BDC no catálogo do ODC. Para isso, inicialmente foi feita a configuração de um *container* Docker com o ODC Core. Em seguida, foi desenvolvida uma ferramenta de indexação, nomeada de `stac2odc` [1], responsável por fazer a busca e recuperação dos dados disponíveis no catálogo BDC-STAC [Zaglia et al. 2019] e registrá-los no catálogo de imagens do ODC Core. A Figura 1 ilustra esse processo.

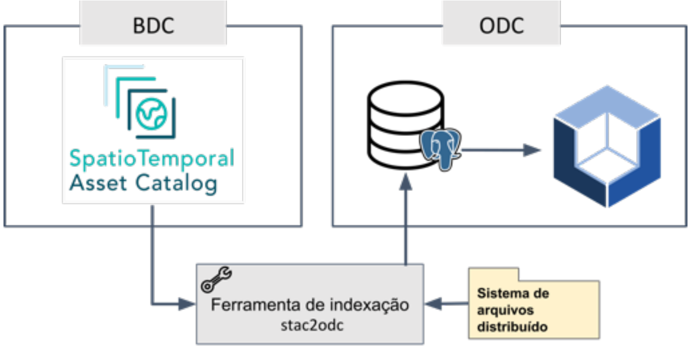


Figura 1. Fluxo de indexação dos dados BDC-ODC

Para evitar a movimentação e replicação de dados, o ODC Core tem acesso direto ao sistema de arquivos distribuídos utilizado no projeto BDC, o qual é utilizado pela ferramenta `stac2odc` durante a indexação. Para usos em que não esteja disponível acesso direto à infraestrutura, a ferramenta `stac2odc` disponibiliza a opção de *download* dos dados durante a indexação.

[1] Disponível em: `https://github.com/brazil-data-cube/bdc-odc`

3.2. Ferramentas e serviços integrados

Com o processo de indexação realizado, por padrão, os dados do BDC podem ser gerenciados através das ferramentas de linha de comando ou serem acessados pela API em Python, ambas disponíveis no ecossistema ODC.

Para permitir a visualização e recuperação facilitada aos dados indexados no ODC, optou-se, nesse trabalho, pela implantação das ferramentas `ODC-Explorer` e `OGC Web Services`. Diferente da primeira fase, o processo de implantação dessas ferramentas exigiu configurações específicas para o funcionamento com os dados do BDC. Nessa etapa, foi necessária a modificação do código fonte dessas ferramentas, pois elas não estavam preparadas para lidar com dados gerados em grades espaciais de referência customizadas, que é o caso dos dados produzidos no BDC. Essas alterações estão disponíveis nos repositórios de código `datacube-explorer` e `datacube-ows` do BDC[2].

3.3. Infraestrutura

Para facilitar o uso das ferramentas analíticas disponibilizadas com integração do ODC e BDC, fez-se a definição de um projeto de infraestrutura computacional, ilustrada pela Figura 2. A ferramenta JupyterHub disponibiliza ambientes interativos, para que cada usuário, através de um navegador *web*, possa acessar e utilizar o ambiente, sem a necessidade de realizar nenhum tipo de configuração de *software* ou movimentação de dados.

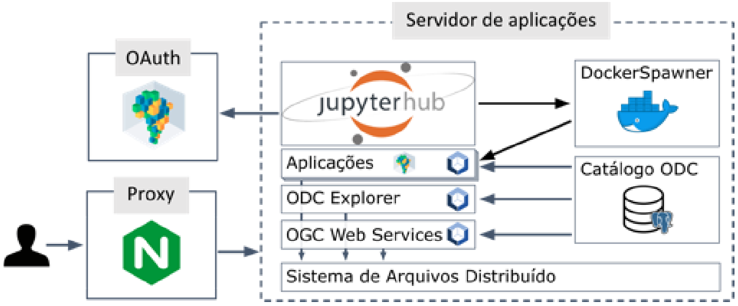


Figura 2. Infraestrutura computacional BDC-ODC

Na arquitetura implementada, o usuário pode realizar acessos ao ambientes através da autenticação no serviço *OAuth* criado no BDC. Uma vez autenticado, o usuário tem à sua disposição um ambiente de Jupyter Notebooks, criado com o uso de *containers* Docker. Por fim, estando dentro deste ambiente, o usuário pode utilizar as ferramentas de análise do ODC e BDC para realizar o processamento e análise dos dados.

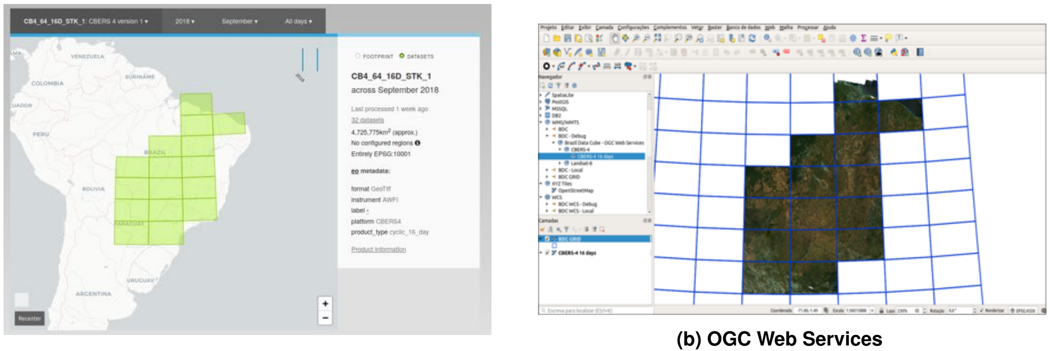
4. Resultados

Nesta seção, são apresentados exemplos de uso dos serviços e ferramentas que foram implementados nesse trabalho de integração.

A customização e configuração dos *OGC Web Services* tornou possível o consumo dos produtos de dados do BDC através dos serviços WCS, WMS e WMTS. Além disso,

[2] Disponível em: `https://github.com/brazil-data-cube`

todos os dados que estão registrados no catálogo ODC podem ser facilmente encontrados com as buscas visuais oferecidas pelo ODC-Explorer. As Figuras 3a e 3b mostram os cubos de dados gerados a partir de imagens do satélite CBERS-4, para todo o bioma Cerrado, sendo apresentado na ferramenta ODC Explorer e consumido via OGC WMS no *software* QGIS.



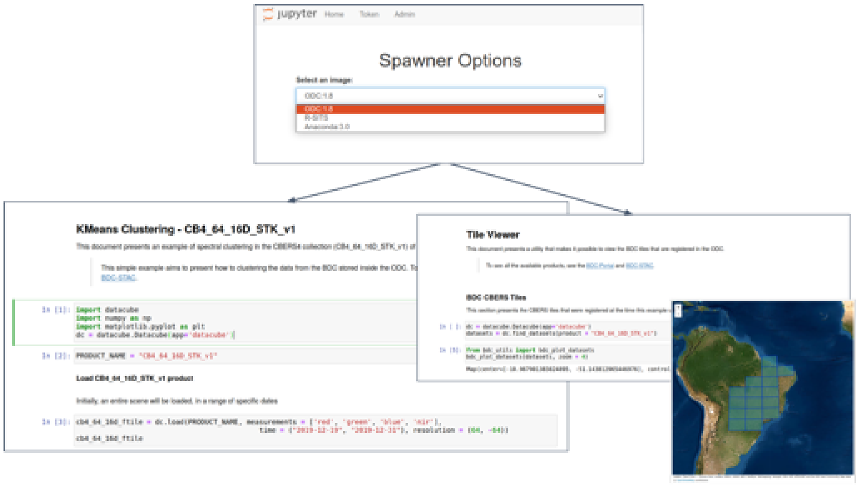
CB4_64_16D_STK_1
across September 2018
Last processed 1 week ago
32 datasets
4,725,775km² (approx.)
No configured regions
Entirely EPSG:10001
eo metadata:
format GeoTiff
instrument AWFI
label -
platform CBERS4
product_type cyclic_16_day
Product information

(a) ODC-Explorer

(b) OGC Web Services

Figura 3. Interface do ODC-Explorer e OGC Web Services

A Figura 4 apresenta a janela de seleção de ambientes oferecida pelo JupyterHub aos usuários, bem como exemplos[3] de Jupyter Notebooks que podem ser carregados para análise dos dados após a seleção do ambiente com as ferramentas ODC.



Spawner Options

Select an image:

ODC:1.8

ODC:1.8
R-SITS
Anaconda:3.0

KMeans Clustering - CB4_64_16D_STK_v1

This document presents an example of spectral clustering in the CBERS4 collection (CB4_64_16D_STK_v1) of

> This simple example aims to present how to clustering the data from the BDC stored inside the ODC. To
> BDC-STAC.

```
In [1]: import datacube
        import numpy as np
        import matplotlib.pyplot as plt
        dc = datacube.Datacube(app='datacube')

In [2]: PRODUCT_NAME = 'CB4_64_16D_STK_v1'
```

Load CB4_64_16D_STK_v1 product

Initially, an entire scene will be loaded, in a range of specific dates

```
In [3]: cb4_64_16d_ftile = dc.load(PRODUCT_NAME, measurements = ['red', 'green', 'blue', 'nir'],
                                     time = ("2019-12-19", "2019-12-31"), resolution = (64, -64))
        cb4_64_16d_ftile
```

Tile Viewer

This document presents a utility that makes it possible to view the BDC tiles that are registered in the ODC.

> To see all the available products, see the BDC Portal and BDC-STAC.

BDC CBERS Tiles

This section presents the CBERS tiles that were registered at the time this example

```
In [ ]: dc = datacube.Datacube(app='datacube')
        datasets = dc.find_datasets(product = 'CB4_64_16D_STK_v1')

In [3]: from bdc_utils import bdc_plot_datasets
        bdc_plot_datasets(datasets, zoom = 4)

        Map(center=[-10.96790138382495, -51.143812005480576], control
```

Figura 4. Interface de usuário do JupyterHub

Cabe notar que o projeto de infraestrutura computacional apresentado, permite o uso de outros ambientes além do ODC, como pode ser visto na Figura 4.

5. Considerações finais

Este artigo apresentou o trabalho em andamento da integração entre os ambientes Brazil Data Cube (BDC) e do framework Open Data Cube (ODC). A integração inicial, feita

[3] Exemplos disponíveis em: `https://github.com/brazil-data-cube/bdc-odc`

com algumas das ferramentas do ecossistema do ODC, mostraram que a realização deste processo fornece aos usuários novas formas de consumo dos produtos de dados do BDC, representando mais opções e flexibilidade aos usuários que vão consumir esses produtos.

O trabalho apresentou também o projeto de uma infraestrutura computacional que pode ser utilizada pelos usuários para que as ferramentas de análise, disponíveis após o processo de integração, sejam facilmente consumidas junto aos produtos de dados do BDC, sem a necessidade de nenhuma configuração ou movimentação de dados para os equipamentos dos usuários.

Como trabalho futuro, será feita a integração dos demais componentes do ODC no ecossistema do BDC, além da adição de facilidades na interface da plataforma, como opções para compartilhamento de resultados e publicação de dados gerados.

Agradecimento

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e ao subprojeto Brazil Data Cube do Projeto Monitoramento Ambiental dos Biomas Brasileiros, financiado com recursos do Fundo Amazônia, por meio da colaboração financeira BNDES e FUNCATE nº 17.2.0536.1

Referências

Appel, M. and Pebesma, E. (2019). On-demand processing of data cubes from satellite image collections with the gdalcubes library. *Data*, 4(3):92.

Câmara, G., Egenhofer, M. J., Ferreira, K., Andrade, P., Queiroz, G., Sanchez, A., Jones, J., and Vinhas, L. (2014). Fields as a Generic Data Type for Big Spatial Data. *Geographic Information Science*, page in press.

Dhu, T., Giuliani, G., Juárez, J., Kavvada, A., Killough, B., Merodio, P., Minchin, S., and Ramage, S. (2019). National Open Data Cubes and Their Contribution to Country-Level Development Policies and Practices. *Data*, 4(4):144.

Ferreira, K. R., Queiroz, G. R., Camara, G., Souza, R. C. M., Vinhas, L., Marujo, R. F. B., Simoes, R. E. O., Noronha, C. A. F., Costa, R. W., Arcanjo, J. S., Gomes, V. C. F., and Zaglia, M. C. (2020). Using remote sensing images and cloud services on aws to improve land use and cover monitoring. In *2020 IEEE Latin American GRSS ISPRS Remote Sensing Conference (LAGIRS)*, pages 558–562.

Giuliani, G., Chatenoux, B., Piller, T., Moser, F., and Lacroix, P. (2020). Data Cube on Demand (DCoD): Generating an earth observation Data Cube anywhere in the world. *International Journal of Applied Earth Observation and Geoinformation*, 87:102035.

Gomes, V. C. F., Queiroz, G. R., and Ferreira, K. R. (2020). An Overview of Platforms for Big Earth Observation Data Management and Analysis. *Remote Sensing*, 12(8):1253.

Open Data Cube (2019). ODC: Architecture and Ecosystem - A High-Level Overview. Technical report, Open Data Cube.

Soille, P., Burger, A., De Marchi, D., Kempeneers, P., Rodriguez, D., Syrris, V., and Vasilev, V. (2018). A versatile data-intensive computing platform for information retrieval from big geospatial data. *Future Generation Computer Systems*, 81:30–40.

Zaglia, M. C., Vinhas, L., Queiroz, G. R., and Simoes, R. E. O. (2019). Catalogação de metadados do cubo de dados do Brasil com o SpatioTemporal asset catalog. *Simpósio Brasileiro De Geoinformática (GEOINFO)*, pages 280–285.

Building Coverage Ratio estimate from LiDAR remote sensing data: an experiment in São Paulo (Brazil)

Luis Felipe Bortolatto da Cunha[1], Carolina Moutinho Duque de Pinho[1], Flavia da Fonseca Feitosa[1]

[1]Laboratório de Estudos e Projetos Urbanos e Regionais (LEPUR)
Universidade Federal do ABC (UFABC)
Alameda da Universidade, s/n° – Anchieta – São Bernardo do Campo – SP – Brasil

{luis.cunha,carolina.pinho,flavia.feitosa}@ufabc.edu.br

Abstract. *Urban planning assumes, at its core, the use of geospatial information to drive decision-making. Nevertheless, data acquisition and development at the intra-urban scale can be resource-expensive, especially for developing countries. This paper presents an ongoing research that explores the potential of LiDAR remote sensing data to monitoring the urban environment with the use of Zoning Parameters. More specifically, it presents an automated methodology to estimate the Normalized Digital Surface Model and Building Coverage Ratio, alongside its application for the Historical City Center of São Paulo. The experiment suggests promising results, with the main advantages of use of open source software, replicability, and fast processing time.*

1. Introduction

The regulation of Land Use and Land Cover of the urban environment through the definition of Zoning Parameters aims at contributing on the pursuit of sustainable urban development and equity. Zoning Parameters may include the control of the Building Coverage Ratio, Floor Area Ratio, permeability ratio, building height, number of floors, setbacks, amongst others [São Paulo 2014].

However, computing Zoning Parameters is a challenging assignment, that often includes field surveying and visual interpretation of very-high resolution remote sensing imagery, which can be cost-expensive and time-demanding. This ongoing research aims to develop an automated methodology for estimating Zoning Parameters at the land lot planning unit, with the use of LiDAR remote sensing technology data. This paper focuses on one specific Zoning Parameter – the Building Coverage Ratio, which refers to the ratio of the building area divided by the land lot area.

LiDAR stands for Light Detection And Ranging, a technology that measures distances (or ranges) based on the time between the transmission and reception of laser signals. LiDAR is an active remote sensing method that can be used on airborne, spaceborne and ground-based platforms. The past decades have seen a rapid increase in applications of LiDAR remote sensing technology in various fields because of its capacity of obtaining data with a high level of detail and tridimensional information. One fundamental attribute of LiDAR data, that is usually stored in a laser point cloud data format, is the classification of the points, which tells whether the laser point is returned from the ground, vegetation, building, water etc. It is considered the second most important information, next to the 3D coordinates, as they allow the conduction of useful analysis [Dong & Chen 2018].

LiDAR's level of detail presents itself both as an advantage and a challenge since processing point cloud data may require the use of proprietary software and high-end hardware. This paper presents a methodology which addresses these challenges by exploring open-source software alternatives that runs at medium hardware requirements [Roussel et al. 2020].

The experiment, carried out in São Paulo's Historical City Center, consisted in the estimate of the Normalized Digital Surface Model (NDSM) and Building Coverage Ratio (BCR) from 2017 LiDAR data. Furthermore, the NDSM and BCR are extracted from high detailed geospatial information of 2004, which allows for the computation of a BCR change index, that captures this complex intra-urban territorial dynamic related to the Land Use and Land Cover of the urban environment.

The results are in compliance with urban planning and management goals and further research should focus on the estimate of other Zoning Parameters, alongside its applications, analysis and assessment.

2. Materials and methods

2.1. Study area

The study area consists of República and Sé districts, which corresponds to São Paulo's Historical City Center. They are part of the City Center Urban Operation Area, foreseen in the master plan as a priority area for urban requalification, a consequence of the demand for business and housing development in areas with established infrastructure and high accessibility [São Paulo 2014]. Figure 1 illustrates the study area location and data.

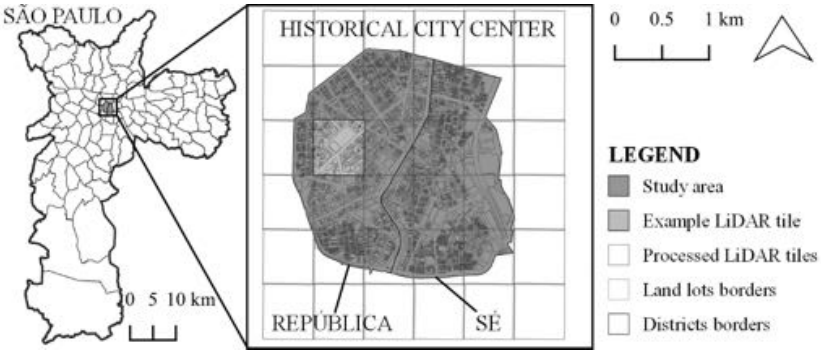


Figure 1. Study area location and data

2.2. Materials

The LiDAR point cloud data was produced in 2017 by the Green-SP Consortium, with the supervision and validation of the São Paulo City Hall. It was obtained with the sensors attached to a helicopter and the points were processed and classified into 5 categories: Soil, Buildings, Vegetation, Road Works, and Other Features. The processed LiDAR data consists of 36 tiles comprising 196.84 million tridimensional points (with longitude, latitude, and altitude attributes), which is used to generate the 2017 NDSM. The building segments, generated manually from the photogrammetric rendering of buildings rooftops in 2004, is used to compute the 2004 NDSM. The 2020 land lot segments, from São Paulo's Finance Secretary, is used as the planning unit for the computation of the 2004 BCR, 2017 BCR and BCR change index. All data is openly available and was acquired from GeoSampa Portal[1].

2.3. Methodology

Figure 2 illustrates the methodology, which consisted in the 2017 NDSM estimate from LiDAR remote sensing data, the 2004 NDSM extraction from the building segments data and the BCR computation from the 2017 NDSM, 2004 NDSM and land lot segments data. All computation was done with R Statistical Software and a reproducible example is available at GitHub[2], with the complete list of packages used.

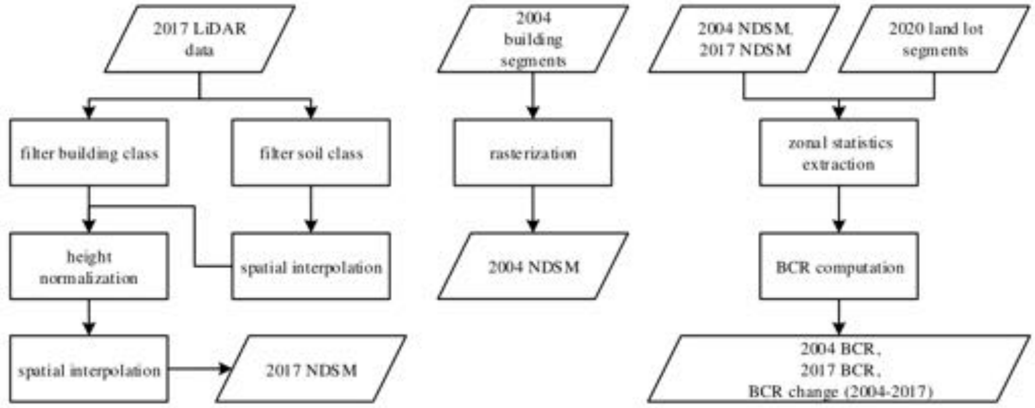


Figure 2. Building Coverage Ratio estimate methodology

The left side of the flowchart refers to the generation of the 2017 NDSM from LiDAR remote sensing data. Since the point cloud data already included a classification, the first step refers to the application of a filter to obtain the classes of interest: buildings and soil. If the classification was not available, it should be carried out before this procedure.

From the soil class, a Digital Terrain Model (DTM) is obtained by the spatial interpolation of points. The algorithm applied is the Triangular Irregular Network (TIN), which derives a Delaunay triangulation and estimates the terrain values at unsampled locations. It is the simplest DTM algorithm, because it involves no parameters, but it was chosen because it has the fastest processing time between the algorithms available. Its drawback is that the interpolation is weak at the edges, a condition that was bypassed with the definition of a 30m buffer from other tiles for every computational step.

Afterward, a height normalization was applied to each point to obtain its height from the soil. This method is superior in terms of computational accuracy if compared to a raster-based height normalization because it is done in a continuous terrain instead of a discretized terrain.

[1] http://geosampa.prefeitura.sp.gov.br/

[2] https://github.com/luisfelipebr/geoinfo2020

The NDSM is then derived from a TIN algorithm that is applied to obtain a continuous surface, i.e. to mask the cells with missing points, with the definition of an additional argument that specify the maximum edge length of a triangle as 4m, resulting in an interpolation that includes an estimate for cells with missing points, but not outside the building class.

To obtain the 2004 NDSM, a rasterize algorithm was applied to the 2004 building segments to convert it to the same data type and spatial resolution of the 2017 NDSM, making them comparable.

From the 2004 and 2017 NDSM, the building area was extracted to the land lot segments, which allowed for the computation of the 2017 BCR and 2004 BCR – dividing the land lot building area by the total area – and the BCR change index – subtracting the 2004 BCR from the 2017 BCR.

3. Results and discussion

The execution of the methodology (Figure 2) in the study area took approximately 45 minutes in a notebook with 256GB SSD, 8GB RAM and an Intel Core i5-8250U CPU.

From Figure 3 it is possible to visually compare, in the example LiDAR tile, the 2004 NDSM and 2017 NDSM. The visual comparison indicates that the generated 2017 NDSM has achieved promising results, although a statistical evaluation was not conducted as they represent different periods of time.

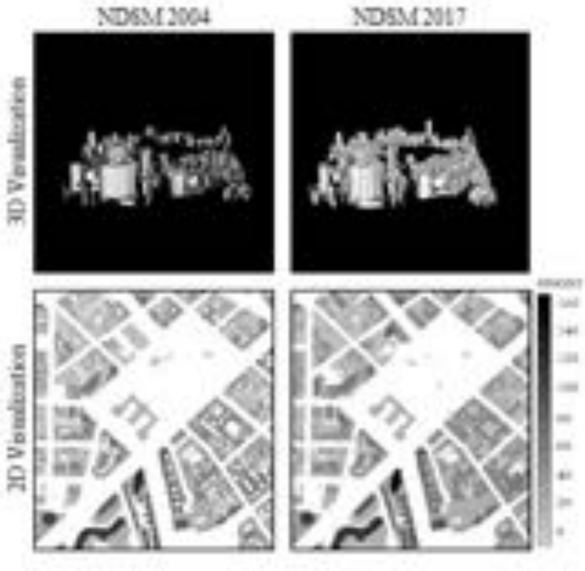


Figure 3. NDSM comparison (2004 and 2017)

Figure 4 evidence the 2004 BCR, the 2017 BCR and the BCR change index, while Table 1 presents its summary statistics. From 2004 to 2017 both the land lots quantity and area of the 75 to 100% BCR class have increased by 7%. Also, even though they represent a low quantity (7%), the 0 to 25% BCR class comprises more than 20% of the total land lots area, which may refer to public spaces or potential building area.

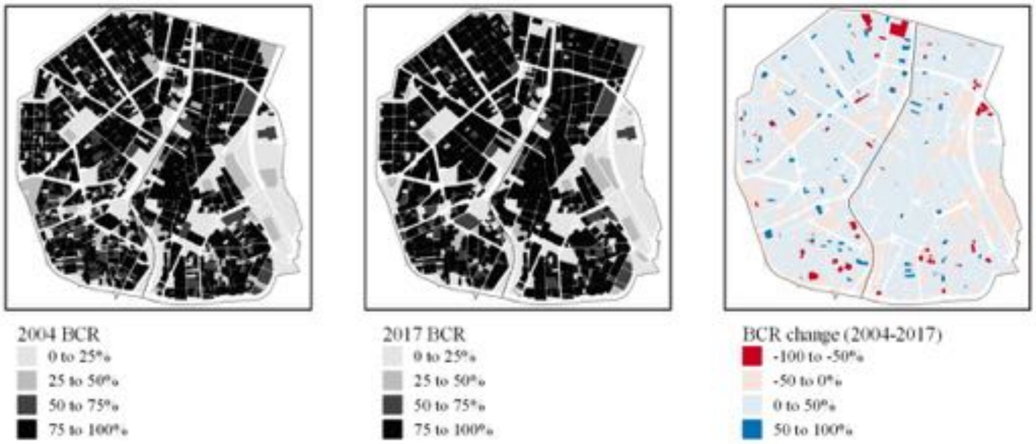


Figure 4. 2004 BCR (left), 2017 BCR (center), and BCR change index (right)

From 2004 to 2017, there have been substantial change on 4% of the land lots quantity and total area, half of them referring to buildings demolition (-100 to -50% class) and the other half to buildings construction (50% to 100% class). From the BCR change map it is possible to spatially identify these dynamics (dark red and dark blue, respectively), although it is not possible to conclude their causes and if there is a spatial pattern related to them.

Table 1. Quantity and area of land lots per class

2004 BCR	Count	Area (km²)	2017 BCR	Count	Area (km²)	BCR change (2004-2017)	Count	Area (km²)
0 to 25%	382 (7%)	2.45 (21%)	**0 to 25%**	417 (7%)	2.57 (22%)	**-100 to -50%**	122 (2%)	0.24 (2%)
25 to 50%	171 (3%)	1.00 (8%)	**25 to 50%**	106 (2%)	0.67 (6%)	**-50 to 0%**	700 (12%)	2.45 (21%)
50 to 75%	595 (10%)	1.65 (14%)	**50 to 75%**	205 (4%)	1.08 (9%)	**0 to 50%**	4757 (84%)	9.02 (76%)
75 to 100%	4544 (80%)	6.82 (57%)	**75 to 100%**	4964 (87%)	7.60 (64%)	**50 to 100%**	113 (2%)	0.21 (2%)
Total	5692 (100%)	11.92 (100%)	**Total**	5692 (100%)	11.92 (100%)	**Total**	5692 (100%)	11.92 (100%)

The estimated Zoning Parameters, in combination with other relevant features of the land lots (function, ownership etc.), can be used to the identification of non-built and underutilized properties and foster the application of the social function of property [Denaldi et al. 2017]. They may also be applied to the estimate of dwelling units density [Lwin & Murayama 2010] and population density [Frizzi et al. 2019], the characterization of deprived settlements [Kuffer & Barros 2011, Feitosa & CDHU 2018, Ribeiro et al. 2019], amongst other urban planning and management paradigms. But its use in these contexts should include additional ancillary data in a more robust framework.

4. Final remarks

This paper presented an automated methodology for the BCR estimate from LiDAR remote sensing technology data and its application to São Paulo's Historical City Center. It included the computation of the NDSM as an intermediary product, which have achieved promising results by visual comparison, while the 2017 BCR, 2004 BCR and BCR change index evidence an important aspect of the intra-urban environment. This approach has the main advantages of relying on open-source software for computation and features a fast processing time, being replicable to other study areas.

Further research should include the estimate of other Zoning Parameters, alongside its analysis and assessment, while also considering urban planning questions, such as the identification of non-built and underutilized properties and the characterization of deprived settlements.

5. References

DENALDI, R.; BRAJATO, D.; SOUZA, C. V. C.; FROTA, H. B. A aplicação do Parcelamento, Edificação ou Utilização Compulsórios (PEUC). urbe, Rev. Bras. Gest. Urbana, 9 (2), 172–186, 2017.

DONG, P.; CHEN, Q. LiDAR remote sensing and applications. Boca Raton: Taylor & Francis, 2018.

FEITOSA, F.; Companhia de Desenvolvimento Habitacional e Urbano do Estado de São Paulo (CDHU). Desenvolvimento e Aplicação de Metodologia para Identificação, Caracterização e Dimensionamento de Assentamentos Precários. Relatório de Pesquisa. São Bernardo do Campo: UFABC, 2018.

FRIZZI, G.; SILVA, G. M.; GONÇALVES, G.; PETRAROLLI, J. G.; FEITOSA, F. F.; PINHO, C. M. D. Estimativa de domicílios em favelas a partir de imagens de alta resolução: resultados para o município de Santos/SP. In: Simpósio Brasileiro de Sensoriamento Remoto, 19. (SBSR), 2019, Santos. Anais... São José dos Campos: INPE, 2019.

KUFFER, M.; BARROS, J. Urban morphology of unplanned settlements: the use of spatial metrics in VHR remotely sensed images. Procedia Environmental Sciences, 7, 152-157, 2011.

LWIN, K. K.; MURAYAMA, Y. Estimation of building population from LiDAR derived digital volume model. In: MURAYAMA, Y.; THAPA, R. B. (eds.). Spatial analysis and modeling in geographical transformation process. Dordrecht: Springer, 2011.

RIBEIRO, S. C. L.; JARZABEK-RYCHARD, M.; CINTRA, J. P.; MAAS, H.-G. Describing the vertical structure of informal settlements on the basis of LiDAR data – a case study for favelas (slums) in Sao Paulo City. In: ISPRS Annals Photogrammetry, Remote Sensing & Spatial Information Sciences, IV-2/W5, 2019.

ROUSSEL, J.-R. et al. lidR: An R package for analysis of Airborne Laser Scanning (ALS) data. Remote Sensing of Environment, 251, 112061, 2020.

SÃO PAULO (Município). Lei 16.050, de 31 de julho de 2014. Institui o Plano Diretor Estratégico do Município de São Paulo. Secretaria do Governo Municipal, São Paulo, 31 jul. 2014.

Identificação de pivôs centrais usando composições de bandas e um método rápido de *Deep Learning*

Denis M. de A. Eiras[1], Mikhaela A. J. S. Pletsch[1], Marcos L. Rodrigues[1], Karine R. Ferreira[1], Thales Sehn Körting[1]

[1]Instituto Nacional de Pesquisas Espaciais – INPE
Av. dos Astronautas, 1758 – 12227-010 – São José dos Campos – SP – Brasil

{denis.eiras, mikhaela.pletsch, marcos.rodrigues, karine.ferreira, thales.korting}@inpe.br

Abstract. *This paper presents a technique to identify central pivot (CP) using patches of images containing only one CP, composed by varied bands of the Landsat 8 OLI sensor and spectral indices, through a fast Convolutional Neural Network (CNN). Different combinations of bands and indexes were tested in this work, as infrared band and NDVI. The obtained results indicate best accuracy (95,85%) when using bands not commonly used in CNNs, surpassing some works. CNN also demonstrated advantages in terms of speed, by classifying a patch of image containing CP in only 0.28 milliseconds, revealing great potential for CP identification in remote sensing images, available in official catalogs.*

Resumo. *Este trabalho apresenta uma técnica para identificar pivô central (PC) usando partes de imagens que contém um PC, compostas por bandas variadas do sensor OLI do Landsat 8 e índices espectrais em uma Rede Neural Convolucional (CNN) rápida. Diferentes combinações de bandas e índices foram testadas, como banda infravermelha e NDVI. Os resultados obtidos indicaram a melhor acurácia (95,85%) utilizando bandas não comumente utilizadas em CNNs, superando alguns trabalhos. A CNN também demonstra vantagens em termos de velocidade ao classificar uma imagem em 0,28 milissegundos, revelando grande potencial para identificação de PC em imagens de sensoriamento remoto, disponíveis em catálogos oficiais.*

1. Introdução

Pivô central (PC) é o sistema de irrigação mecanizada que mais cresce no país, uma vez que garante as necessidades hídricas de diferentes culturas. No Brasil, PCs são responsáveis por irrigar cerca de 20% da área irrigada total. A identificação de PCs no Brasil é de grande importância, tanto para gerir o balanço hídrico, bem como para estudar os impactos ambientais decorrentes desse tipo de prática [Fontelle, 2017]. Apesar disso, ainda não existem metodologias automatizadas para identificação de PCs em imagens de sensoriamento remoto.

Rodrigues *et al.* (2020) estudaram a identificação de PCs no Cerrado utilizando técnicas de Canny, transformada circular de Hough e séries temporais sobre índices de vegetação extraídos de produtos do MODIS, com acurácias de até 90%. Ferreira *et al.* (2011) utilizaram um método de segmentação que buscou contabilizar a área irrigada por pivôs, resultando em um índice kappa de 0,94.

Visando resultados ainda mais acurados, alguns trabalhos utilizaram as Redes

Neurais Convolucionais (ou *Convolutional Neural Networks – CNN*), utilizadas no reconhecimento de objetos em imagens, devido à sua adaptabilidade à formas e tons. A MobileNet é uma das arquiteturas de CNN que apresenta resultados similares ou superiores a outras arquiteturas mais pesadas em um tempo consideravelmente menor [Howard *et al.*, 2017]. O trabalho de Zhang *et al*. (2018) foi um dos pioneiros no uso de CNN para identificação de PCs, utilizando imagens *Red-Green-Blue* (RGB) do satélite Landsat 5. Cada PC está contido em 25 imagens, com uma pequena diferença de posição, para a identificação do centro de cada PC, resultando em uma precisão de 95,85% e um *recall* de 93,33% em uma CNN Le-Net adaptada. Albuquerque *et al*. (2020) usaram imagens do Landsat 8 para avaliar diferentes ambientes no Brasil central e mudanças sazonais. Foram testados três modelos de CNN e técnicas de reconstrução de imagem através de intervalos de sobreposição entre quadros, totalizando 10.997.161 PCs. Os melhores resultados apontaram a CCN U-Net, com F1-score e Kappa de 0,9638 e acurácia de 98,80%. Saraiva *et al.* (2020) também utilizaram a CNN U-Net, buscando velocidade de treinamento. Foram utilizadas 4 bandas do satélite PlanetScope, totalizando 42.000 imagens com resolução espacial aproximada de 3 m, levando à acurácia de 99%. Seu método levou aproximadamente 22 h para o treinamento e 10 minutos para testar cada grade de sua área de estudo, utilizando uma máquina com processador de 4 núcleos, 50 GB de RAM e uma GPU NVIDIA Tesla K80.

No seu uso mais comum, imagens de fotos são utilizadas para a classificação em CNNs. Tratando-se da identificação de PCs com CNNs na literatura atual, encontram-se trabalhos que utilizaram somente as bandas do espectro visível (RGB). Outras bandas não tradicionais, como o infravermelho, e índices espectrais, como o *Normalized Difference Vegetation Index* (NDVI) [Vermote *et al.*, 2016], que são muito utilizados no mapeamento de alvos de vegetação, poderiam substituir bandas do espectro visível para compor as imagens utilizadas pela CNN. Sendo assim, esse trabalho tem como objetivo aplicar uma técnica baseada em CNN para testar a acurácia da identificação de PCs em recortes de imagens, compostas por bandas do Landsat 8 e índices espectrais. Os resultados demonstram quais foram as melhores e piores acurácias obtidas com cada composição de bandas e os tempos de treinamento e validação.

2. Materiais e métodos

2.1. Área de estudo

A área de estudo foi selecionada com base na concentração de PCs, principalmente no Cerrado Brasileiro, exibida à esquerda e ampliada à direita da Figura 1.

2.2. Materiais

A Agência Nacional de Águas (ANA) realizou o mapeamento dos PCs ativos e inativos entre 1985 e 2017 de todo o território brasileiro [Fontenelle *et al,* 2019]. Foram mapeados 23.181 polígonos de PCs (em azul na Figura 1) em um arquivo do tipo *shapefile,* utilizados para extrair partes de imagens de PCs. Foram utilizadas as bandas 1 a 8 do Landsat 8, cujas cenas estão localizadas com contorno em preto (Figura 1):

A) Órbita 220 e ponto 71, de 19/09/2017, 1290 PCs: treinamento da CNN;
B) Órbita 220 e ponto 72, de 19/09/2017, 1290 PCs: treinamento da CNN;
C) Órbita 221, ponto 71, de 10/09/2017, 1461 PCs: validação das composições;
D) Órbita 221 e ponto 76, de 24/07/2017, 1441 PCs: validação adicional, em região com maior diversidade de formas de polígonos;

E) Órbita 220 e ponto 69, 731 PCs: validação adicional, em outros períodos (03/11/2016; 06/01/2017; 14/05/2017; 18/08/2017).

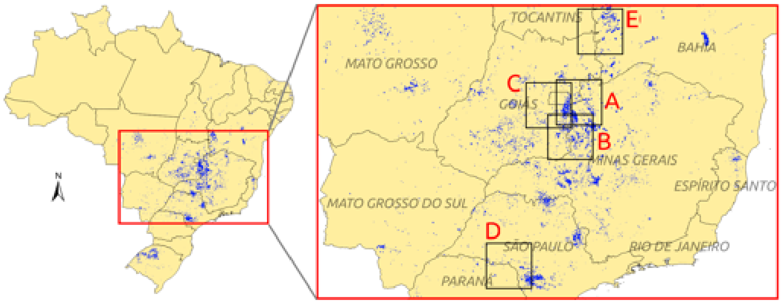


Figura 1. PCs (polígonos em azul), e localização das cenas utilizadas, no Cerrado, para treinamento (A e B) e validação (C, D e E) da CNN.

As cenas (Figura 1) foram adquiridas no catálogo de imagens do INPE (2020), selecionadas com base na grande quantidade de PCs em um mesmo bioma do Brasil (Cerrado) e baixa cobertura de nuvens, em sua maioria, com exceção de uma imagem de 06/01/2017, localizada no oeste da Bahia (E), com 8% de cobertura.

A MobileNet foi a arquitetura de CNN utilizada neste trabalho, que possui parâmetros configuráveis, como o Multiplicador de Largura, que reduz uniformemente a largura em cada camada (0% a 100%), e o parâmetro Multiplicador de Resolução (128, 160, 192 e 224), que reduz a representação interna de cada camada pelo mesmo multiplicador [Howard *et al.*, 2017]. Embora a arquitetura básica MobileNet seja pequena e de baixa latência, pode ser necessário diminuir seu tamanho e aumentar a velocidade do modelo, através da configuração destes parâmetros.

Os métodos propostos a seguir foram implementados em uma aplicação*,* construída com a linguagem Python v.3.7, a biblioteca TensorFlow 1.13.11 e outras bibliotecas, utilizando em um *laptop* com processador Intel i7 de sétima geração e 16 GB de memória RAM, sem a utilização de GPU.

2.3. Métodos

As seguintes etapas automatizadas da aplicação foram executadas (Figura 2):

1. geração de imagens como composição de bandas, para treinamento e validação;
2. extração de imagens com PCs e não pivôs centrais (NPC), através do mapeamento dos polígonos*;*
3. treinamento e testes da CNN, utilizando diferentes configurações MobileNet;
4. validação da melhor configuração da CNN, nas cenas de treinamento;
5. validação da melhor configuração da CNN, nas cenas de validação.

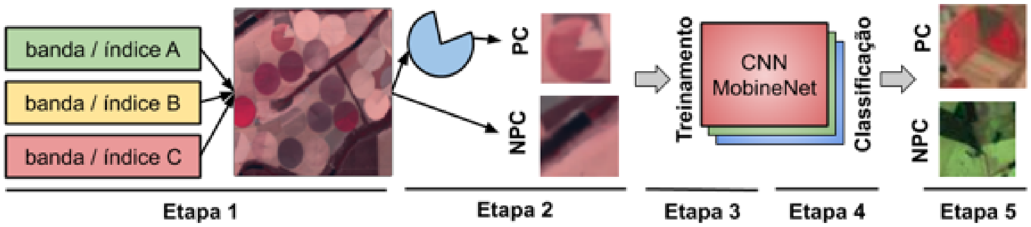


Figura 2. Etapas do método proposto.

A etapa 1 gerou uma imagem para treinamento e outra para validação, como composição de 3 bandas ou índices espectrais, com valores normalizados e reescalados entre 0 e 255, para a redução de dimensionalidade. A etapa 2 efetuou recortes nas imagens, envolvendo os polígonos de PCs contidos no arquivo *shapefile.* O tamanho da imagem foi acrescido de uma margem de 10% do tamanho do PC, para que estes sempre estejam enquadrados na imagem. Um processo semelhante extraiu imagens contendo NPC de tamanhos aleatórios, dentro da faixa de tamanho de PC, sendo a mesma quantidade de imagens de PCs. A etapa 3 realizou diversos treinamentos como combinações dos parâmetros: arquivos com as bandas do Landsat 8, multiplicador de resolução da rede MobileNet e parâmetros para pré-processar as imagens, tais como espelhamento e brilho aleatório, a fim de aumentar a acurácia da validação. Na etapa 4, para cada treinamento gerado na etapa 3, a classificação foi realizada em 20% das imagens de treinamento. A melhor configuração de treinamento identificada com base na classificação foi armazenada para a etapa 5, a qual realizou testes com imagens de validação, excluindo-se daí os recortes de PCs utilizados no treinamento. Essa última etapa permite computar às acurácias e tempo de execução para cada composição de banda testada.

3. Resultados

O melhor resultado foi alcançado utilizando uma CNN com as seguintes configurações: Multiplicador de resolução 224, Multiplicador de Largura 1, sem uso de pré-processamento nas imagens. Essa rede foi treinada com 5160 imagens das cenas em A e em B (Figura 1), em um tempo médio de 7 minutos e 36 segundos, suficientes para atingir a máxima acurácia na validação. A validação foi executada em 2922 recortes da cena C (Figura 1) em aproximados 0,42 segundos (0,28 milissegundos por recorte, em média), onde se considerou somente o tempo do teste de cada imagem na memória.

Dado o conjunto de 35 combinações possíveis das bandas utilizadas, temos na Tabela 1 sumarizados nas 5 primeiras linhas os melhores resultados e nas 5 últimas os piores, em termos de acurácia estimada através da validação. A precisão e o *recall* foram calculados sobre a classe PC. A composição RGB (4, 3, 2) ficou apenas na 27ª posição entre as melhores acurácias (92,84%). Para comparação, a Tabela 2 apresenta os melhores resultados obtidos com a combinação de índices espectrais e bandas do sensor OLI. De maneira a demonstrar a generalização alcançada, foram utilizadas imagens de outras regiões e períodos, usando as bandas 4, 6 e 7. A Tabela 3 apresenta os resultados obtidos nas posições D e E (Figura 1).

Tabela 1. Melhores e piores acurácias na validação das composições

Composição	Acurácia (%)	Precisão (%)	Recall (%)	F1-Score	Kappa
Bandas 4, 6, 7	95,85	96,91	94,72	0,9580	0,9585
Bandas 3, 5, 6	95,45	97,36	93,42	0,9535	0,9544
Bandas 5, 6, 7	94,96	94,81	95,13	0,9497	0,9496
Bandas 2, 4, 5	94,90	94,50	95,35	0,9492	0,9490
Bandas 2, 6, 7	94,76	96,05	93,36	0,9469	0,9476
Bandas 4, 5, 6	91,99	98,12	85,62	0,9144	0,9198
Bandas 1, 5, 6	91,51	98,48	84,32	0,9085	0,9150
Bandas 1, 3, 4	90,97	98,54	83,16	0,9020	0,9096
Bandas 1, 2, 3	90,52	95,68	84,87	0,8995	0,9051
Bandas 1, 2, 5	81,86	99,47	64,07	0,7794	0,8185

Tabela 2. Validação de composições com índices espectrais e bandas

Índices+bandas	Acurácia (%)	Precisão (%)	Recall (%)	F1-Score	Kappa
NDVI, 6, 3	94,10	92,93	95,47	0,9418	0,9410
SAVI, 6, 7	93,15	92,00	94,52	0,9324	0,9315
NDVI	92,02	89,05	95,82	0,9231	0,9202
SAVI	91,20	87,95	95,48	0,9156	0,9119
EVI	88,26	84,80	92,32	0,8816	0,8825

Tabela 3. Validações em outras cenas e períodos

Cena, data	Acurácia (%)	Precisão (%)	Recall (%)	F1-Score	Kappa
E, 18/08/2017	95,69	95,13	96,30	0,9571	0,9568
E, 14/05/2017	94,66	99,24	90,01	0,9404	0,9465
E, 06/01/2017	89,67	100,00	79,34	0,8848	0,8965
E, 03/11/2016	88,61	99,47	77,67	0,8723	0,8861
D, 24/07/2017	75,53	97,79	52,25	0,6811	0,7552

4. Discussão

A CNN deste e outros trabalhos superaram resultados obtidos por técnicas baseadas em detectores de bordas, círculos e segmentação, utilizados nos trabalhos de Rodrigues *et al.* (2020) e Ferreira *et al.* (2011), devido à sua adaptabilidade à diversidade de formas. No entanto, as CNN requerem uma grande quantidade de amostras para treinamento, que devem ser extraídas de regiões e períodos próximos às amostras de validação.

A utilização de índices não apresentou melhores resultados que a utilização exclusiva de bandas, pois a CNN encontrou parâmetros ótimos que superaram índices. De outro lado, a utilização de diferentes bandas melhorou a acurácia da CNN com relação às bandas RGB. As bandas 6 e 7 (infravermelho de ondas curtas) e 5 (infravermelho próximo) apareceram mais vezes nas 10 melhores composições (Figura 3), superando acurácias do trabalho de Zhang *et al.* (2018), que utilizou bandas RGB.

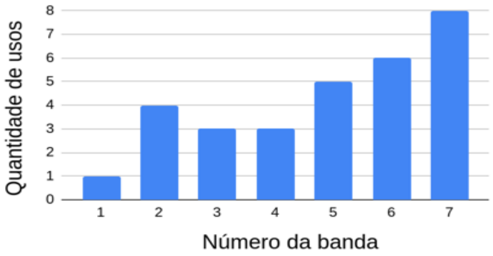


Figura 3. Quantidade de usos das bandas nas 10 melhores composições.

Albuquerque *et al.* (2020) usaram uma técnica de reutilização de exemplos, e Saraiva *et al.* (2020) imagens de melhor resolução, superando a acurácia desse trabalho em, respectivamente, 3,03% e 3,15%, mas utilizando CNN de arquiteturas maiores e dedicadas. Saraiva *et al.* executaram o treinamento em 22 h, muito superior ao tempo da técnica proposta neste trabalho, que levou 7 minutos para o treinamento, demonstrando um ótimo balanceamento entre qualidade e velocidade de identificação de PCs.

5. Conclusões

As melhores composições foram atingidas ao utilizar as bandas 6 e 7, sendo que a melhor composição (95,85%) utilizou as bandas 4, 6 e 7. A validação de uma cena mais distante (Figura 1-D) piorou a acurácia em aproximados 20%, indicando que há características distintas de vegetação, forma e tamanho de PC. A validação de cenas próximas em diferentes períodos (Figura 1-E) apresentou uma variação de acurácia de até 7,8%. Estas validações sugerem que é preciso realizar treinamentos em localidades e períodos mais próximos da cena a ser validada para se atingir melhores resultados.

Em trabalhos futuros será desenvolvida um CNN direcionada para identificar e quantificar PCs, aproveitando o potencial demonstrado pelo uso das composições de bandas não RGB, além de utilizar mais de 3 bandas como entrada, possibilitando a rápida identificação de PCs em catálogos oficiais de sensoriamento remoto.

Agradecimentos

Os autores agradecem aos Prof. Dr. Gilberto R. de Queiroz, Dra. Lúbia Vinhas e Dra. Karine R. Ferreira e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processos 140377/2018-2 e 303360/2019-4. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Referências

Albuquerque, A. O. de, de Carvalho Júnior, O. A., Carvalho, O. L. F. D. …. & Fontes Guimarães, R. (2020). Deep semantic segmentation of center pivot irrigation systems from remotely sensed data. Remote Sensing, 12(13), 2159.

Ferreira, E., Dantas, A. A. A., de Toledo, J. H. (2011). Classificação de áreas irrigadas por pivôs centrais utilizando como base a segmentação. Irriga, 16(2), 145-152.

Fontelle, T., Ferreira, D., Guimarães, D., & Landau, E. (2019). Levantamento da agricultura irrigada por pivôs centrais no Brasil. Livro científico (ALICE).

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., …. & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications.

INPE – Instituto Nacional de Pesquisas Espaciais (2020). www.dgi.inpe.br/catalogo

Rodrigues, M. L., Körting, T. S., de Queiroz, G. R., …. & da Silva, L. A. R. (2020). Detecting Center Pivots In Matopiba Using Hough Transform And Web Time Series Service. In 2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS), 189-194.

Saraiva, M., Protas, É., Salgado, M., & Souza Jr, C. (2020). Automatic Mapping of Center Pivot Irrigation Systems from Satellite Images Using Deep Learning. Remote Sensing, 12(3), 558.

Vermote, E., Justice, C., Claverie, M., & Franch, B. (2016). Preliminary analysis of the performance of the Landsat 8/OLI land surface reflectance product. Remote Sensing of Environment, 185, 46-56.

Zhang, C., Yue, P., Di, L., & Wu, Z. (2018). Automatic identification of center pivot irrigation systems from landsat images using convolutional neural networks. Agriculture, 8(10), 147.

Espectrorradiometria da folha de *Terminalia catappa* sp. em diferentes estádios de desenvolvimento

Isadora H. Ruiz[1], Philipe S. Simões[1], Gabriel M. da Silva[1], Andeise C. Dutra[1], Yosio E. Shimabukuro[1], Leila M. G. Fonseca[1], Lênio S. Galvão[1]

[1] Divisão de Observação da Terra e Geoinformática – Instituto Nacional de Pesquisas Espaciais (INPE)
Caixa Postal 12.227 – 010 – São José dos Campos, SP – Brasil

{isadora.ruiz, philipe.simoes, gabriel.maximo, andeise.dutra, yosio.shimabukuro, leila.fonseca, lenio.galvao}@inpe.br

Abstract. *This study describes the spectral behavior of isolated leaves of T. catappa sp. in different growth stages and observation angles. For this purpose, the Bidirectional Reflectance Factor was calculated for a photosynthetically active leaf, a senescent leaf and a non-photosynthetically active leaf using radiance measurements at 0° and 14°. The continuum removal technique was applied to analyze the absorption bands. Physiology, structure, and biochemical vegetation indexes were also calculated. Results showed that the Plant Senescence Reflectance Index (PSRI) was the most sensitive VI to biophysical changes in the studied leaves, showing also changes with viewing geometry.*

Resumo. *Este estudo descreve o comportamento espectral de folhas isoladas de T. catappa sp. em diferentes estádios de desenvolvimento e sob diferentes ângulos de observação. Obteve-se o Fator de Reflectância Bidirecional da folha fotossinteticamente ativa, senescente e não sinteticamente ativa, a partir de medições de radiância realizadas em laboratório, sob os ângulos de observação de 0° e 14°. Nos parâmetros das bandas de absorção, aplicou-se a técnica de remoção do contínuo e calculou-se índices de vegetação relacionados à fisiologia, estrutura e bioquímica da vegetação. Os resultados mostraram que o índice mais sensível às mudanças biofísicas das folhas estudadas foi o Plant Senescence Reflectance Index (PSRI).*

1. Introdução

A utilização do sensoriamento remoto no estudo da vegetação evoluiu com a chegada de diferentes tecnologias e metodologias de aquisição de dados. Com isso, a compreensão dos processos de interação da radiação eletromagnética (REM) com a vegetação tem permitido maior entendimento das respostas da vegetação a doenças, ciclos fenológicos e/ou distúrbios fisiológicos. Em geral, as características da resposta da interação da radiação eletromagnética com a vegetação são devidas aos pigmentos fotossintetizantes como as clorofilas, xantofilas e carotenos e pela água presente nos vegetais [Ponzoni, Shimabukuro e Kuplich 2012].

Antes de mais nada, há a necessidade de entender as respostas obtidas no espectro eletromagnético de acordo com o alvo de estudo. Precisamente, folhas isoladas e dosséis de uma mesma espécie podem apresentar respostas diferentes quanto à absorção da radiação de acordo com as características bioquímicas e biofísicas de cada alvo [Asner 1998, Fourty et al. 1996]. Entre os fatores que influenciam na interação da REM com o vegetal, destacam-se os morfológicos, como a organização espacial dos elementos envolvidos na captação da REM e os fatores fisiológicos relacionados ao estágio de vida e as condições do vegetal [Bernardes 1987].

Em plantas sadias, os sensores detectam a absorção da radiação em comprimentos de onda do azul e do vermelho na faixa espectral do visível que são utilizadas nos processos fotossintéticos [NASA 2010]. Também, há respostas importantes de absorção de energia no infravermelho próximo relacionadas à quantidade de água líquida presente nas folhas. Com isso, a absortância e reflectância da REM em diferentes regiões do espectro eletromagnético podem variar na observação de um mesmo indivíduo quando o objeto de estudo é o dossel de uma árvore ou apenas uma folha isolada.

Neste contexto, destaca-se a importância da compreensão da interação da radiação eletromagnética com as folhas isoladas dos vegetais nos diversos estádios vegetativos em diferentes regiões do espectro eletromagnético. Portanto, esta pesquisa buscou analisar folhas isoladas de *T. catappa* sp., popularmente conhecida como Amendoeira da Praia, por meio de espectrorradiometria de laboratório. Os objetivos foram analisar a curva espectral da espécie, identificar as feições de absorção por meio da remoção do contínuo e aplicar diferentes índices de vegetação para os três estádios vegetativos. Ao final, buscou-se comparar a reflectância multiespectral simulada para o sensor MSI (*MultiSpectral Instrument*) a bordo da plataforma Sentinel-2A sob diferentes ângulos de observação.

2. Materiais e métodos

Neste estudo foram utilizadas medidas radiométricas da parte superior das folhas isoladas de *T. catappa* sp. em três estádios vegetativos: a) folha sadia (fotossinteticamente ativa), b) folha amarelada (fase de senescência) e c) folha seca (fotossinteticamente não ativa). O experimento foi realizado no Laboratório de Radiometria (LARAD) do Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos - SP, em março de 2017 [Dutra et al. 2019], com auxílio de um espectroradiômetro *FieldSpec* modelo Standard-Res (ASD, Boulder, CO, USA) de amplitude espectral de 350 nm a 2.500 nm e resolução espectral de 3 nm para a faixa do visível e infravermelho próximo e 10 nm para a faixa do infravermelho de ondas curtas. Como fonte de radiação foi utilizado uma lâmpada halógena de 250W e para calibração do equipamento utilizou-se uma placa lambertiana ideal (Spectralon de ~100% de reflectância). Com IFOV (*Instantaneous Field of View*) de 25º do sensor sobre o alvo e fonte de iluminação com fluxo colimado, calculou-se o Fator de Reflectância Bidirecional (FRB) (Equação 1)

$$FRB_{\lambda}=L_{\lambda,a} / L_{\lambda,p} \quad (1)$$

onde $L_{\lambda,a}$ refere-se a radiância espectral do alvo e $L_{\lambda,p}$ a radiância espectral da superfície lambertiana ideal, sob as mesmas condições de iluminação e observação [Jensen 2009, Novo 2010].

O primeiro experimento foi direcionado à análise do espectro de reflectância das folhas isoladas em estágios de vida distintos (Folha Sadia – FS; Folha Amarelada – FA; Folha Seca – FC). A técnica de remoção do contínuo foi aplicada para isolar e qualificar parâmetros de bandas de absorção específicas dos espectros de reflectância [Clark, Rough 1984], o que possibilitou a extração de parâmetros como profundidade, largura à meia altura e assimetria. Além disso, foram calculados os índices de vegetação relacionados com os parâmetros biofísicos baseados em diferentes comprimentos de onda do espectro eletromagnético, de acordo com a literatura (Tabela 1).

Tabela 1. Formulações dos Índices de Vegetação, onde ρ representa a reflectância da banda no comprimento de onda original das formulações.

Índice de Vegetação	Fórmula	Referência
PRI	(ρ531-ρ570) / (ρ531+ρ570)	Gamon et al. (1997)
RENDVI	(ρ752-ρ701) / (ρ752+ρ701)	Gitelson et al. (1996)
NDVI	(ρ864-ρ660) / (ρ864+ρ660)	Rouse et al. (1974)
NDWI	(ρ860-ρ1.240) / (ρ854+ρ1.240)	Gao (1996)
PSRI	(ρ680-ρ500) / ρ750	Merzlyak et al. (1999)

Fonte: Adaptado de Galvão et al. (2009); Sano et al. (2019)

Em sequência, considerando um segundo experimento, foram extraídas as medidas de radiância para a placa de referência e folha sadia com a geometria de observação ao nadir (0º) e a 14°. Por fim, a partir de fatores de correção da resposta espectral de bandas espectrais fornecidos pela *European Space Agency*, foi realizada a simulação da reflectância multiespectral esperada para uma imagem do sensor MSI a bordo da plataforma Sentinel-2A.

3. Resultados e discussão

A curva do fator de reflectância bidirecional (FRB) dos estádios vegetativos da folha revelaram comportamento típico das mudanças em pigmentação nos estádios, conforme resultados encontrados por Dutra et al. (2019). A folha sadia apresentou as feições de absorção relacionadas aos pigmentos fotossintetizantes nas bandas do azul e vermelho, com pico de reflectância no verde [Ponzoni, Shimabukuro e Kuplich 2012, Sano et al. 2019]. As folhas em estádios sucessionais perderam substancialmente a absorção na região do visível pela perda de clorofila e foram caracterizadas pelo aumento da reflectância nesta região.

No infravermelho próximo, a reflectância manteve-se alta e constante, com presença de absorção nas fases sadia e amarelada em 980 nm e 1.200 nm, associada a presença de água foliar [Sano et al. 2019]. No infravermelho de ondas curtas, o FRB manteve-se igual para FS e FA, e diferente para FC, relacionada principalmente ao conteúdo de água líquida na folha [Ponzoni, Shimabukuro e Kuplich 2012, Sano et al. 2019]. Em virtude da degradação de pigmentos fotossintetizantes, estruturas celulares e redução do conteúdo de água na folha seca, feições de absorção associadas a presença de lignina e celulose foram reveladas na folha seca [Kokaly et al. 1998].

De forma complementar, a análise sobre a remoção do contínuo para discriminação dos estádios vegetativos da folha de *T. catappa* sp. permitiu observar, na região entre 550 nm e 750 nm, mudança na profundidade da banda, associada ao conteúdo de pigmentos fotossintetizantes. O comprimento de onda de maior absorção foi de 673 nm para FS e 678 para FA, com parâmetros de profundidade e largura da banda distintos. Na faixa de 900 nm a 1.200 nm não foi observada banda de absorção de água foliar para a folha seca, enquanto para FS e FA demonstraram profundidade e meia altura similar e comprimento de onda de maior absorção centrado em 977 nm para ambos os estádios.

No infravermelho de ondas curtas, todos os estádios vegetativos apresentaram feição de absorção, principalmente bandas de absorção pela água líquida na folha. A fase seca apresentou a maior profundidade de banda nesta região, no comprimento de onda de 1.718 nm. Os demais experimentos caracterizaram profundidade menor, centradas em 1.776 nm (FA) e 1.780 nm (FS). Como apresentado na análise do espectro de reflectância, as bandas de absorção da celulose (2.050 nm - 2.220 nm) e lignina (2.240 nm - 2.372 nm) ocorrem em

FC, com profundidade e largura da banda maior para a celulose e comprimento de onda de maior absorção, sendo em 2.146 nm para a celulose e 2.303 nm para a lignina.

Adicionalmente, os Índices de Vegetação foram calculados para sintetizar parâmetros biofísicos das folhas. Com enfoque na fisiologia, o *Photochemical Reflectance Index* (PRI) demonstrou a eficiência da absorção da radiação eletromagnética [Gamon et al. 1997] nos diferentes estádios. Conforme expresso na Tabela 2, com o avanço dos estádios sucessionais da folha, houve redução do PRI, por assim indicar a reduzida capacidade da folha em captar REM e transformá-la em energia fotossintetizada. Esse processo foi corroborado pelas reduzidas feições de absorção, quanto profundidade e largura das bandas apresentadas pela folha seca.

Tabela 2. Índice de Vegetação nos respectivos estágios vegetativos

Índice de Vegetação	Folha Sadia	Folha Amarelada	Folha Seca
PRI	0,12	0,09	-0,23
RENDVI	0,59	0,04	0,11
NDVI	0,95	0,67	0,78
NDWI	0,04	0,04	-0,08
PSRI	0,003	0,45	0,54

Ainda sobre parâmetros fisiológicos da folha, o *Red Edge Normalized Difference Vegetation Index* (RENDVI) apresentou maior valor para a folha sadia e o menor para folha amarelada, sinalizando a distância entre as bandas da região final do vermelho e começo do IV-próximo. Isso demonstra a caracterização da feição borda vermelha (*red edge*) nos espectros de vegetação [Gitelson et al. 1996]. Com respeito à estrutura da folha e pigmentação associada, o *Normalized Difference Vegetation Index* (NDVI) apresentou variações consistentes em sua resposta [Rouse et al. 1974]. Por sua vez, o *Normalized Difference Water Index* (NDWI) detectou as mudanças no conteúdo de água líquida em plantas [Gao 1996], conforme o padrão vegetativo encontrado na literatura. O comportamento crescente do *Plant Senescence Reflectance Index* (PSRI) [Merzlyak et al. 1999] caracterizou, de forma eficiente, os estádios vegetativos, sendo os valores mais altos relacionados a senescência da folha.

A simulação da reflectância multiespectral do sensor MSI a bordo da plataforma Sentinel-2A nos ângulos de observação de 0º e 14º produziu variações espectrais não associadas à resposta biofísica das plantas. Contudo, as características principais da curva foram preservadas. Feições de absorção não puderam ser detectadas. Entretanto, o comportamento das bandas espectrais simuladas mostrou a menor reflectância no visível (490 nm e 665 nm) e um pico discreto de reflectância em 560 nm. As bandas do infravermelho próximo apresentaram-se com alta reflectância nas bandas de 740 nm, 783 nm, 842 nm e 865 nm. Na faixa do infravermelho médio ocorreu a maior perda de informação sobre a reflectância simulada, pois apenas duas bandas compreendem essa faixa, em 1.610 nm e 2.190 nm.

A diferença no valor da reflectância foi na ordem de 4% nas bandas do visível, 6% e 7% no SWIR 1 e 2 (*Short Wave Infrared*), respectivamente, onde a maior diferença foi verificada nas bandas RE3, RE4 (*Red Edge*) e NIR (*Near Infrared*) com 8% (Figura 1). Essa diferença pode ser associada ao comportamento anisotrópico da vegetação, ou seja, a reflexão da REM ocorre de forma desigual em diferentes direções [Jensen 2009, Cardozo et al 2011, Ponzoni, Shimabukuro e Kuplich 2012]. Com isso, a quantidade de REM retroespalhada para o sensor diferiu em até 8,97% quando observada à 14º, em relação ao nadir. Esse aumento também pode estar associado a área de observação do sensor que passou de 0,00205 m² (0º) para 0,00211 m² (14º), possibilitando, assim, que a REM refletida por maior área da folha chegasse ao sensor.

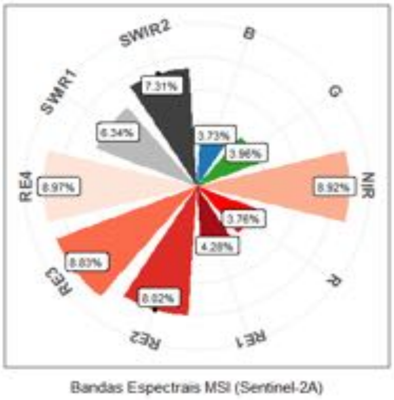


Figura 1 - Diferença da reflectância multiespectral simulada entre os ângulos de observação do sensor de 90° e 14° (R = Red; G = Green; B = Blue; NIR = Near Infrared; RE = Red Edge; SWIR = Short Wave Infrared)

4. Conclusão

O cálculo dos índices de vegetação se mostrou uma estratégia mais efetiva para diferenciar os estádios vegetativos das folhas de *T. catappa* sp, principalmente em termos de aspectos estruturais, fisiológicos e bioquímicos. O PSRI apresentou melhor resposta às mudanças biofísicas das folhas de *T. catappa* sp do que os demais índices de vegetação analisados. O experimento da reflectância multiespectral, com simulação das bandas do sensor MSI a bordo da plataforma Sentinel-2A, mostrou que as bandas espectrais do *Red Edge* (3 e 4) e NIR apresentam diferenças acima de 8% e as bandas espectrais do visível caracterizam as menores diferenças observadas.

Referências

Asner, Gregory P. (1998) "Biophysical and Biochemical Sources of Variability in Canopy Reflectance." *Remote Sensing of Environment*, v.64, n.3, p.234–253.

Bernardes, M. S. (1987) Fotossíntese no dossel de plantas cultivadas. In: CASTRO, P.R. Ecologia da produção agrícola. Piracicaba, SP: Associação Brasileira para Pesquisa da Potassa e do Fósfato, 249 p.

Cardozo, F. S., Oliveira, G., Ferreira, M. P., Moraes, E. C. (2011) "Função de distribuição de reflectância bidirecional (FDRB) de uma superfície vegetada sob diferentes geometrias de visada e condições de alagamento". Anais do XV Simpósio Brasileiro de Sensoriamento Remoto, Curitiba, PR, Brasil, INPE, p.8516-8523.

Clark, R. N. e Roush, T. L. (1984) "Reflectance spectroscopy: quantitative analysis techniques for remote sensing applications", In *Journal of Geophysical Research*, v.89, n.B7, p.6329-6340.

Dutra, A. C., Prudente, V. H. R., Vieira, C. D., França e Silva, N. R., Junior, C. H. L. S., Moraes, E. C., Shimabukuro, Y. E. e Sanches, I. D. (2019) "Fator de reflectância de diferentes folhas de vegetação de Amendoeira da praia (*T. catappa* sp.)", Anais do XIX Simpósio Brasileiro de Sensoriamento Remoto, Santos, SP, Brasil, INPE, p.3084-3087.

Fourty, Th., Baret, F., Jacquemoud, S., Schmuck, G., and Verdebout, J. (1996), Leaf optical properties with explicit description of its biochemical composition: direct and inverse problems. *Remote Sens. Environ*. 56:104–117.

Galvão, L. S., Formaggio, A. R. and Breunig, F. M. (2009) "Relações entre índices de vegetação e produtividade de soja com dados de visada fora do nadir do sensor Hyperion/EO-1", Anais XIV Simpósio Brasileiro de Sensoriamento Remoto, Natal, RN, Brasil, INPE, p.1095-1102.

Gamon, J. A., Serrano, L. and Surfus, J. S. (1997) "The photochemical reflectance index: an optical indicator of photosynthetic radiation use efficiency across species, functional types, and nutrient levels", *Ecologia*, v. 112, n, 4, p.492-501.

Gao, B. C. (1996) "NDWI - a normalized difference water index for remote sensing of vegetation liquid water from space", *Remote Sensing of Environment*, v. 58, p.257-266.

Gitelson, J. A., Merzlyak, M. N. e Lichtenthaler, H. K. (1996) "Detection of red edge position and chlorophyll content by reflectance measurements near 700 nm", *Journal of Plant Physiology*, v. 148, n. 3-4, p.501-508.

Jensen, J. R. (2009) "Sensoriamento Remoto do Ambiente: uma perspectiva em Recursos Terrestres", Tradução de J. C. N. Epiphanio. São José dos Campos, SP: Parênteses, pp.598. Tradução de: *Remote Sensing of the Environment: an Earth Resource Perspective* (Prentice Hall Series in Geographic Information Science).

Kokaly, R., Clarck, R.N., Livo, K.E. (1998) "Mapping the biology and mineralogy of Yellowstone National Park using imaging spectroscopy". Summaries of the 4th Annual JPL Airborne Geoscience Workshop, JPL Publication 97-21, v.1, p.235-244.

Merzlyak, M. N., Gitelson, A. A., Chickunova, O. B. e Rakitin, V. Y. (1999) "Non-destructive optical detection of pigment changes during leaf senescence and fruit ripening", *Physiologia Plantarum*, v. 106, n. 1, p.135-141.

NASA - National Aeronautics and Space Administration, Science Mission Directorate. (2010). Reflected Near-Infrared Waves. Disponível em: < http://science.nasa.gov/ems/08_nearinfraredwaves> Acesso em: 01 de abril de 2020.

Novo, E. M. L. de M. (2010) "Sensoriamento Remoto: Princípios e Aplicações", 4 ed. Blucher: São Paulo, pp. 387.

Ponzoni, F. J., Shimabukuro, Y. E., Kuplich T.M. (2012). "Sensoriamento Remoto da vegetação". 2.ed. São Paulo: Oficina de Textos.

Rouse, J. W., Haas, R. H., Schell, J. A. e Deering, D. W. (1974) "Monitoring vegetation systems in the Great Plains with ERTC" In: ERTS-1 Symposium, 3. Proceeding … ASA Goddard, NASA SP-351, p.309-317.

Sano, E. E., Ponzoni, F. J., Meneses, P. R., Baptista, G. M. M., Toniol, A. C.; Galvão, L. S., Rocha, W. J. S. F. (2019) "Reflectância da vegetação". In: Meneses, P. R., Almeida, R., Baptista, G. M. M. Reflectância dos materiais terrestres: análise e interpretação. São Paulo: Oficina de Textos, p.189-219.

Aplicação do Modelo Aditivo Generalizado espacial para a modelagem da susceptibilidade a ocorrência de deslizamentos

Tatiana Dias Tardelli Uehara[1], Eduardo Celso Gerbi Camargo[1], Camile Sothe[2], Thales Sehn Körting[1]

[1]Divisão de Observação da Terra e Geoinformática (DIOTG)
Instituto Nacional de Pesquisas Espaciais (INPE)
Caixa Postal 515 – 12227-010 – São José dos Campos – SP – Brasil

[2]School of Earth, Environment and Society – McMaster University
Hamilton – ON – Canadá

{tatiana.uehara, eduardo.camargo, thales.korting}@inpe.br,

sothec@mcmaster.ca

Abstract. *The susceptibility mapping is a fundamental task to prevent social and environmental impacts resulting from landslides, which makes it possible to identify areas with greater or lesser probability of occurring the event. In this context, this work uses a spatial approach based on Generalized Additive Models (GAM) associated to driving factors, to estimate and analyze the spatial distribution of landslide susceptibility in the Rio Rolante Hydrographic Basin (RS/Brazil). The results reveal that the model was able to satisfactorily distinguish areas with greater or lesser susceptibility of landslide occurence. All points of landslides occurrence have estimated probabilities greater than 0.999. The non-occurrence class showed greater variability in estimated values.*

Resumo. *O mapeamento de susceptibilidade é uma tarefa fundamental para prevenir impactos sociais e ambientais decorrentes de deslizamentos de terra, que possibilita identificar as áreas com maior ou menor probabilidade deste tipo de evento ocorrer. Este trabalho emprega uma abordagem espacial baseada em Modelos Aditivos Generalizados (GAM) associado a fatores potencializadores, para estimar e analisar a distribuição espacial da susceptibilidade a deslizamentos de terra na Bacia Hidrográfica do Rio Rolante (RS/Brasil). Os resultados revelam que o modelo foi capaz de distinguir satisfatoriamente as áreas com maior ou menor susceptibilidade. Todos os pontos de ocorrência de deslizamentos apresentaram probabilidades estimadas superiores a 0,999. O padrão de não ocorrência apresentou maior variabilidade de valores estimados.*

1. Introdução

Deslizamentos são um dos tipos de movimentos de massa, caracterizados por processos geomorfológicos naturais que ocorrem em diversas partes do planeta. Tais processos consistem em movimentos descendentes de material provindo da encosta que podem ser provocados por terremotos, degelo de neve ou chuvas intensas, podendo também ser causados ou intensificados por atividades antrópicas [Guzzetti et al. 2012]. Tal fenômeno causa grandes perdas econômicas e sociais, principalmente quando ocorre em áreas densamente povoadas. O objetivo desta pesquisa foi analisar os fatores potencializadores à ocorrência de deslizamentos de terra empregando um Modelo Aditivo Generalizado (*Generalized Additive Model - GAM*) espacial e elaborar um mapa de suscetibilidade indicando os locais com maior e menor probabilidade de ocorrência.

2. Material e métodos

2.1. Área de Estudo

A área de estudos (Figura 1) inclui toda extensão da bacia hidrográfica do Rio Rolante (RS/Brasil). Sua área de drenagem abrange 828 km^2 e os valores de elevação variam entre 20m e 1040m. A área é predominantemente ocupada por floresta nativa, silvicultura, agricultura, pastagem e ocupação antrópica rural. No que tange aos aspectos geomorfológicos e climáticos, a área encontra-se na unidade Serra Geral com clima subtropical úmido. Trabalhos anteriores detectaram por volta de 300 cicatrizes de deslizamentos no local. Todas as cicatrizes foram geradas após um evento extremo de precipitação ocorrido no dia 5 de janeiro de 2017, em que foram registrados entre 90 e 272mm de chuva em quatro horas [SEMA 2017].

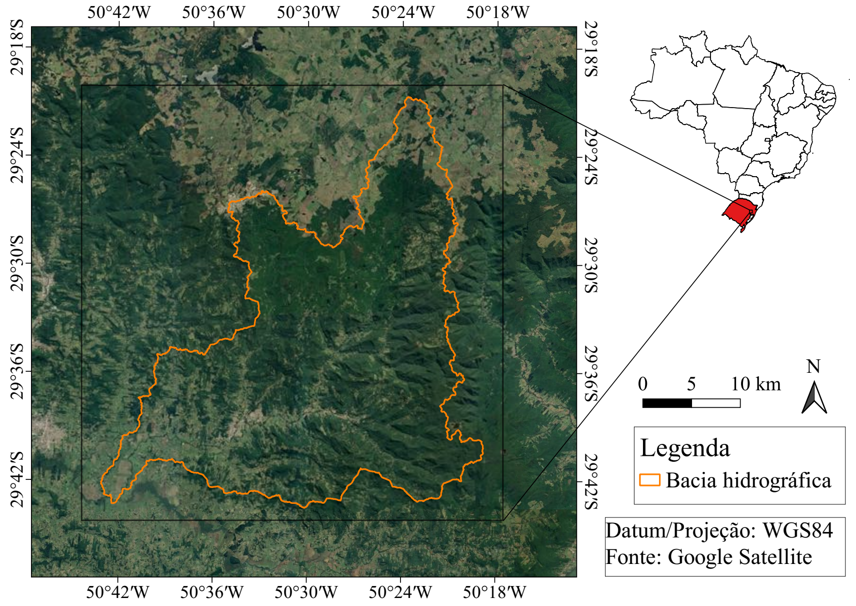


Figura 1. Localização da área de estudo.

2.2. Dados

A partir do inventário de cicatrizes de deslizamento estabelecido na região [Quevedo et al. 2020], foi gerado o centróide de cada cicatriz, o que resultou em 335 *pontos de ocorrência* de deslizamento. A mesma quantidade de *pontos de não ocorrência* foi gerada aleatóriamente, respeitando a distância delimitada por um buffer de 5m ao redor das cicatrizes, para que não houvesse sobreposição entre os *pontos de ocorrência* e *não ocorrência*. Da totalidade do conjunto amostral, 20% dos *pontos de ocorrência* e 20% dos *pontos de não ocorrência* foram retirados e guardados para a validação do modelo.

Para cada ponto amostral há um conjunto de variáveis potencializadoras associadas ao evento investigado. Um valor médio para cada uma das variáveis foi estabelecido dentro de um buffer de 5m em torno de cada ponto amostral. A seleção das variáveis potencializadoras consistiram na extração de atributos a partir do Modelo Digital de Elevação (MDE), mensuração da distância entre as cicatrizes e canal de drenagem mais próximo e a atribuição de pesos às classes de solo, conforme os trabalhos realizados por [Sothe et al. 2017, Saro et al. 2016]. O MDE utilizado provém de dados disponibilizados pelo projeto *Alaska Satellite Facility (ASF)*, no qual, a partir do MDE do *Shuttle Radar Topography Mission (SRTM)* obtem-se um produto de 12,5m de resolução espacial. Por meio da ferramenta SAGA disponível no software QGIS, foram geradas as seguintes variáveis geomorfométricas: declividade, curvatura horizontal e curvatura vertical.

Os tipos de solos foram obtidos do mapa de solos do Brasil, produzido pelo Instituto Brasileiro de Geografia e Estatística (IBGE) em escala 1:5.000.000. Foram identificados quatro tipos de solos e a cada um deles atribuído um peso (0,0 a 0,9) com base no seu grau de erodibilidade [da Silva and Alvares 2007]. Os tipos de solo com seus respectivos graus de erodibilidade e peso são: Cambissolo Húmico e Neossolo Litólico (Muito Alto - 0,8); Argissolo Vermelho-Amarelo (Alto - 0,7) e Nitossolo Vermelho (Médio - 0,5). Os valores da variável distância dos rios foram extraídos a partir da mensuração da distância euclidiana entre os pontos amostrais e o vetor de hidrografia mais próximo.

2.3. Modelo

A estrutura de modelagem empregada é baseada na ideia de processos pontuais espaciais, pela qual se pode estimar uma superfície de suscetibilidade que varia continuamente na região de interesse, conforme proposto por [Kelsall and Diggle 1998]. Neste trabalho, o processo pontual subjacente é caracterizado a partir de um conjunto de pontos espacialmente distribuídos sobre a área de estudo, cuja variável resposta é do tipo binária, isto é, *pontos suscetíveis* com valor 1, e 0 em caso contrário. Isso caracteriza a uma abordagem de regressão binária com a inclusão de fatores potencializadores, um tipo de GAM [Hastie and Tibshirani 1990] que pode ser entendido como um modelo linear generalizado estendido por uma componente aditiva espacial que, por suposição, varia suavemente na região de estudo. Sob estas considerações [Kelsall and Diggle 1998] propõem um GAM, com uma função de ligação *logit*, conforme a Equação 1:

$$logit[p(s)] = log\,\frac{p(s)}{1-p(s)} = \boldsymbol{\beta x} + g(s) \tag{1}$$

em que: $\boldsymbol{x}$ é o vetor de covariáveis, $\boldsymbol{\beta}$ são seus efeitos e g é uma componente espacial (uma função suave), porém desconhecida, das coordenadas espaciais s. Se a suscetibilidade é considerada constante na região de estudo, então $g(s) = 0$ e o modelo reduz-se a um modelo de regressão logística usual [Hosmer et al. 1989]. O procedimento de estimação de $\boldsymbol{\beta}$ e $g(s)$ baseia-se em métodos iterativos usuais de modelos aditivos generalizados, conforme descritos em [Hastie and Tibshirani 1990]. Neste procedimento, $g(s)$ é estimado usando regressão Kernel ponderada [Wand and Ripley 2006].

O modelo proposto (Equação 1) estima uma superfície de suscetibilidade, porém é importante avaliar se esta superfície varia significativamente na região de estudo, ou seja, se existem evidências estatísticas suficientes para rejeitar a hipótese nula de suscetibilidade constante (H0: $g(s) = 0$). Além disso, é de interesse a construção de contornos de tolerância que auxiliam na identificação de áreas onde a suscetibilidade é significativamente superior ou inferior à média global. Para tal, o teste global da suscetibilidade e a identificação de áreas de baixa e alta suscetibilidade foram realizados empregando o método de simulação de Monte Carlo proposto por [Kelsall and Diggle 1998]. A aplicação do modelo foi realizada no software R (versão 2.10), utilizando o pacote "SPGAM".

3. Resultados e Discussão

Inicialmente, a seleção de variáveis foi realizada pelo método *backwards* [Derksen and Keselman 1992]. A variável solo, por apresentar um *p-valor* não significativo, foi removida do modelo. Uma possível explicação pode ser a escala dos dados pedológicos que generalizou os tipos de solo em quatro grandes categorias que não detalham as variações entre áreas mais ou menos susceptíveis. A Equação 2 revela o modelo final. As variáveis potencializadoras incluídas, bem como suas respectivas significâncias estatísticas (*p-valor*), foram as seguintes: $\beta 0$ (1,08E-12), elevação (4,07e-04), declividade (1,96e-30), curvatura vertical (2,59e-04) e curvatura horizontal(1,47e-03).

$$logit[p(s)] = \beta_0 + \beta_{Elev.} + \beta_{Decliv.} + \beta_{Curv.vertical} + \beta_{Curv.horizontal} + g(s) \quad (2)$$

A partir do modelo expresso na Equação 2 estimou-se a suscetibilidade associada à ocorrencia de deslizamento de terra, conforme ilustra a Figura 2. Em seguida, foram realizadas 500 simulações via método de Monte Carlo, cujos resultados observados foram: i) houve uma variação espacial global significativa da suscetibilidade (*p-valor* = 0,001992), portanto a hipótese nula de suscetibilidade constante na região ($g(s) = 0$) foi descartada; ii) a identificação de áreas delimitadas por linhas de contorno de 2,5% e 97,5%, de aproximadamente 95% de tolerância, apontam os locais onde a suscetibilidade é significativamente inferior ou superior à média global, respectivamente. A partir da Figura 2, identifica-se que as áreas com maiores probabilidades de ocorrência de deslizamento (apresentadas em tons vermelhos) localizam-se nas regiões com alta declividade, onde se concentraram as cicatrizes geradas em 2017.

Um estudo semelhante pode ser visto em Sothe et al. (2017), que aplicou um modelo GAM para a análise de susceptibilidade na Bacia do Rio Luís Alves (SC/Brasil). Seus resultados revelaram que 100% das *amostras de ocorrência* foram estimadas com probabilidades acima de 0,9 e 75% das *amostras de não ocorrência*, abaixo de 0,5.

A adequação do modelo empregado foi realizada por meio da comparação: i) entre o conjunto de *pontos de ocorrência* reservado para validação e o conjunto de *pontos de ocorrência* extraído da superfície estimada, ambos sobre a mesma localização geográfica; ii) de forma similar, entre o conjunto de *pontos de não ocorrência* reservado para validação e o conjunto de *pontos de não ocorrência* estimado. O que se espera é que a distribuição estatística de tais conjuntos de pontos (*de ocorrência* e *não ocorrência*) sejam distintas, com média estimada para o conjunto de *pontos de ocorrência* próxima de um (1) e para o conjunto de *pontos de não ocorrência* em torno de zero (0). Isto pode ser avaliado graficamente através da construção de um boxplot, conforme mostra a Figura 3.

Na Figura 3 observa-se que as distribuições das probabilidades estimadas para os *pontos de ocorrência* e *não ocorrência* são bastantes distintas. Pode-se afirmar que

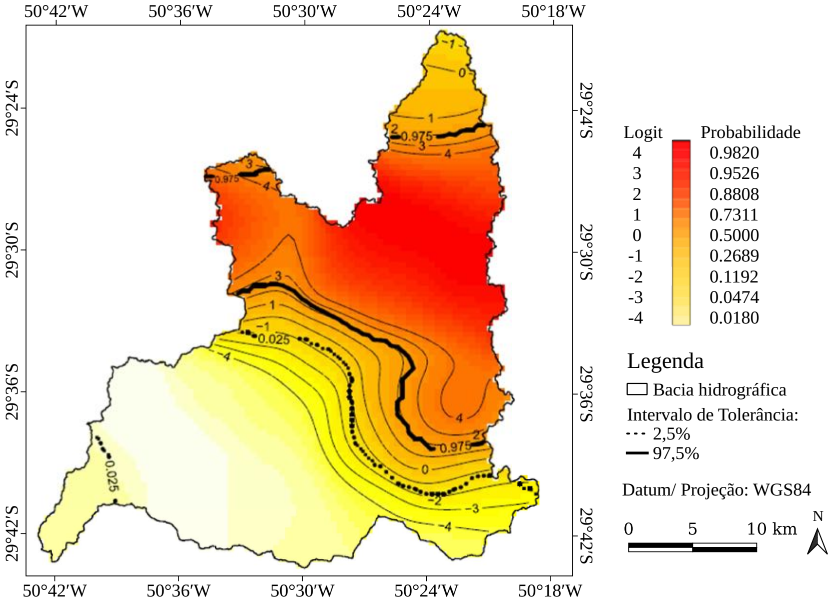


Figura 2. Mapa de susceptibilidade estimada.

100% dos *pontos de ocorrência* apresentam probabilidades estimadas superiores a 0,999. Por outro lado, nas áreas não suscetíveis a deslizamentos de terra, os *pontos de não ocorrência* apresentaram uma maior variação de valores de probabilidade estimado, sendo que cerca de 50% das amostras desta classe apresentaram valores inferiores a 0,472. Esta maior incerteza associada às estimativas dos *pontos de não ocorrência* deve ser objeto de investigação de futuros trabalhos, como por exemplo, a inclusão de outros fatores potencializadores no modelo.

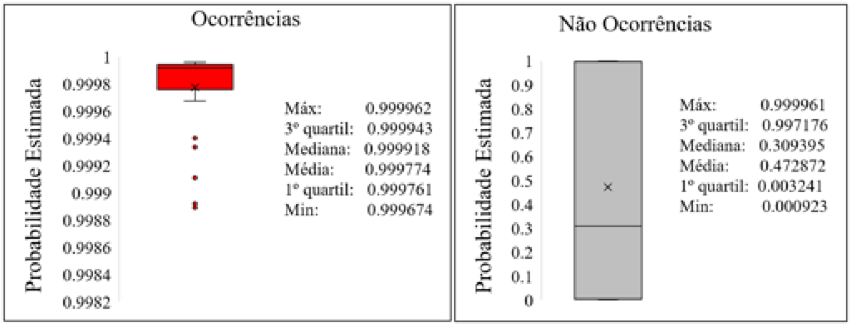


Figura 3. Boxplot da probabilidade estimada para os *pontos de ocorrência* e *não ocorrência*, associados a deslizamentos de terra.

4. Conclusão

Este trabalho apresentou uma abordagem para modelar e mapear a suscetibildade a deslizamentos de terra a partir de informações pontuais. Para tal, empregou-se um método baseado em modelos aditivos generalizados espaciais, com a inclusão de fatores potencializadores correlacionados com o evento investigado. Neste estudo, as variáveis Elevação, Declividade, Curvatura Horizontal e Curvatura Vertical foram fatores potencializadores significativos no modelo empregado. Quanto à avaliação do modelo, ficou evidente que as estimativas de suscetibilidade foram mais precisas nas áreas em que ocorreram deslizamentos de terra, todos os pontos de ocorrência apresentaram probabilidades estimadas superiores a 0,999. Já para os locais menos vulneráveis, as estimativas foram menos precisas, sendo que a metade dos pontos de não ocorrência apresentaram valores inferiores a 0,472. De um modo geral, trata-se de um resultado preliminar, que pode auxiliar no planejamento de ações mais específicas e dirigidas a essas áreas. Para trabalhos futuros recomenda-se investigar a influência da variação dos dados de entrada em diferentes escalas, bem como a exploração e inclusão de outros fatores potencializadores.

Referências

da Silva, A. M. and Alvares, C. A. (2007). Levantamento de informações e estruturação de um banco dados sobre a erodibilidade de classes de solos no estado de são paulo. *Geociências (São Paulo)*, 24(1):33–41.

Derksen, S. and Keselman, H. J. (1992). Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology*, 45(2):265–282.

Guzzetti, F., Mondini, A. C., Cardinali, M., Fiorucci, F., Santangelo, M., and Chang, K.-T. (2012). Landslide inventory maps: New tools for an old problem. *Earth-Science Reviews*, 112(1-2):42–66.

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized additive models*, volume 43. CRC press.

Hosmer, D. W., Lemeshow, S., and Sturdivant, R. X. (1989). The multiple logistic regression model. *Applied logistic regression*, 1:25–37.

Kelsall, J. E. and Diggle, P. J. (1998). Spatial variation in risk of disease: a nonparametric binary regression approach. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 47(4):559–573.

Quevedo, R. P., Guasselli, L. A., De Oliveira, G. G., and Ruiz, L. F. C. (2020). Modelagem de áreas suscetíveis a movimentos de massa: avaliação comparativa de técnicas de amostragem, aprendizado de máquina e modelos digitais de elevação. *Geociências (São Paulo)*, 38(3):781–795.

Saro, L., Woo, J. S., Kwan-Young, O., and Moung-Jin, L. (2016). The spatial prediction of landslide susceptibility applying artificial neural network and logistic regression models: A case study of inje, korea. *Open Geosciences*, 8(1):117–132.

SEMA (2017). Diagnóstico preliminar: Descritivo dos eventos ocorridos no dia 5 de janeiro de 2017 entre as regiões dos municípios de são francisco de paula e rolante/rs. *Secretaria do Ambiente e Desenvolvimento Sustentável, Porto Alegre*, page 26.

Sothe, C., Camargo, E. C. G., Gerente, J., Rennó, C. D., and Monteiro, A. M. V. (2017). Uso de modelo aditivo generalizado para análise espacial da suscetibilidade a movimentos de massa. *Revista do Departamento de Geografia*, 34:68–81.

Wand, M. and Ripley, B. (2006). Kernsmooth: Functions for kernel smoothing for wand & jones (1995). *R package version*, 2:22–19.

Segmentação Semântica de Tipos de Uso de Solo na Amazônia Utilizando Aprendizado Profundo

Joel P. de Oliveira[1], Marly G. F. Costa[2] e Cícero F. F. Costa Filho[2]

[1]Centro Gestor e Operacional do Sistema de Proteção da Amazônia (CENSIPAM)
Manaus – AM – Brasil

[2]Universidade Federal do Amazonas (UFAM)
Manaus – AM – Brasil

joelparente@gmail.com,marlygfcosta@gmail.com,cffccfilho@gmail.com

Abstract. *This work proposes a deep learning methodology to perform the semantic segmentation of LANDSAT-8 images of the following types of uses and land cover: forest, pasture and agriculture. The field of study is the Amazon region. The reference data were extracted from the TerraClass project in 2014. A CNN architecture was evaluated along with three optimization methods: SGDM, ADAM and RMSProp and the dropout and L_2 regularization methods, as methods for generalization improvement. The best results were obtained using the RMSProp optimization method. The accuracy values obtained for the evaluated images were above 93%.*

Resumo. *Este trabalho propõe uma metodologia de aprendizado profundo para realizar a segmentação semântica de imagens LANDSAT-8 dos seguintes usos e cobertura de solos: floresta, pasto e agricultura. O campo de estudo é a região Amazônica. Os dados de referência foram extraídos do projeto TerraClass do ano de 2014. Foi avaliada uma arquitetura CNN juntamente com três métodos de otimização: SGDM, ADAM e RMSProp e os métodos dropout e regularização L_2, como métodos para a melhoria de generalização. Os melhores resultados foram obtidos utilizando o método de otimização RMSProp. Os valores de acurácia obtidos para as imagens avaliadas ficaram acima de 93%.*

1. Introdução

Sensoriamento remoto é a utilização de diversas tecnologias com o objetivo de estudar os fenômenos que ocorrem na superfície da Terra. Essas tecnologias compreendem sensores, equipamentos instalados a bordo de aeronaves, espaçonaves e outras plataformas. Os dados gerados a partir de sistemas de sensoriamento remoto são de grande utilidade para várias aplicações, dentre as quais podemos citar: planejamento urbano, agrícolas, geológicas, monitoramento de desmatamento [Novo 2008].

Com respeito ao monitoramento do desmatamento na Amazônia através de sensoreamento remoto, o Instituto Nacional de Pesquisas Espaciais (INPE) é uma referência mundial. Dentre os vários projetos desenvolvidos pelo INPE para esse monitoramento, destacam-se o Programa de Monitoramento do Desflorestamento na Amazônia Legal (PRODES) e o TerraClass. O PRODES fornece dados por meio de mapas anuais de desmatamento na região Amazônica. O projeto TerraClass utiliza os dados gerados pelo

PRODES para realizar uma classificação de uso e cobertura de solo nas seguintes classes: floresta, pasto, agricultura, áreas urbanas, mineração e outros. Esse tipo de informação ajuda os órgãos do governo a desenvolverem políticas públicas de prevenção para conter o avanço do desmatamento [Noma et al. 2013]. Embora o PRODES e o TerraClass sejam grandes projetos e forneçam dados bastante confiáveis, eles ainda contam com uma significativa parcela do trabalho realizada com intervenção humana.

Para tornar mais eficiente o trabalho que envolva a análise de imagens de sensoriamento remoto, trabalhos têm sido propostos na área de aprendizado de máquina. [Bem et al. 2020] utilizaram aprendizado profundo para mapear o desmatamento entre 2017 e 2018 e entre 2018 e 2019. Foram utilizadas três cenas do LANDSAT-8 de regiões do estado do Pará e do Amazonas. Os dados de referência utilizados foram os dados PRODES dos anos de 2017 a 2019. Os melhores resultados obtidos com a CNN ResUnet foram acurácia e F1-Score de 99,93% e 94,65%, respectivamente. [Adarme et al. 2020] utilizaram aprendizado profundo para detecção automática de desmatamento por meio de duas imagens LANDSAT-8. As áreas de estudo foram duas regiões com diferentes padrões de desmatamento: os biomas Amazônia e Cerrado no Brasil. Os dados de referências utilizados foram os dados PRODES dos anos de 2017 e 2018. Os autores obtiveram como melhores resultados uma acurácia e um F1-Score de 95% e 63%, respectivamente, na Amazônia, e de 97% e 78%, respectivamente, no Cerrado.

Segundo [Aggarwal 2018], a partir da primeira década desse século, a rede neural renasceu sob o novo rótulo chamado de aprendizado profundo. Aprendizado profundo aborda o uso de modelos computacionais com arquiteturas hierárquicas compostas por múltiplas camadas de processamento a fim de ”aprender”determinadas representações de dados nos mais diferentes formatos: áudio, imagens e texto [Lecun et al. 2015]. Quatro modelos principais compõem o aprendizado profundo: Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNN), Máquinas de Boltzmann, *Autoencoders* e *Sparse Coding*. Segundo [Aggarwal 2018], a grande quantidade de dados disponíveis nos últimos anos, juntamente com o aumento do poder computacional, permitiram a utilização de arquiteturas mais profundas em relação ao que era possível anteriormente.

Os trabalhos de [Bem et al. 2020] e [Adarme et al. 2020] utilizaram aprendizado profundo para detectar áreas desmatadas em regiões da Amazônia brasileira. Os resultados alcançados por esses autores foram bastante satisfatórios. No entanto, os autores avaliaram uma região específica da Amazônia. Outra crítica que fazemos aos trabalhos previamente publicados é que os mesmos não se preocupam em disponibilizar o conjunto de dados utilizado para *benchmark*.

Um grande desafio para se treinar CNNs com imagens de sensoreamento remoto é que, normalmente, as classes são desbalanceadas. Em outras palavras, em uma região capturada pela imagem, existe um grande desbalanceamento em termos de área dos diversos tipos de solos. Esse problema pode levar, no treinamento da CNN, aos métodos de otimização terem um melhor desempenho nas classes mais frequentes. Para contornar esse problema, nesse trabalho utiliza-se a técnica de imagem-mosaico. Essa técnica consiste de extrair, a partir da imagem de satélite, pequenas retalhos retangulares uniformes, ou seja, com apenas um tipo de solo. A partir desses retalhos monta-se uma imagem maior, denominada de imagem-mosaico.

Portanto, considerando a ausência de um modelo de rede em aprendizado profundo adequado para classificação dos diversos tipos de solo para a região amazônica, e a dificuldade de treinamento de uma CNN com bom desempenho em todas as classes, esse trabalho pretende, através da técnica de imagem mosaico, utilizar uma CNN para segmentação e classificação de diversas regiões da Amazônia nos seguintes tipos de uso e cobertura de solo: agricultura, pasto e floresta.

2. Material e métodos

2.1. Conjunto de dados utilizados

Neste trabalho foram utilizadas imagens LANDSAT-8/OLI. As imagens estão disponíveis gratuitamente no site: http://earthexplorer.usgs.gov/. Foram utilizadas as seguintes bandas: Vermelho (B4); Infravermelho próximo (B5); Infravermelho de ondas curtas 1 (B6). Segundo [Yu et al. 2019], B4, B5 e B6 é a melhor combinação de três bandas para aplicações de sensoriamento remoto cujo o objetivo é realizar a classificação do solo. As imagens utilizadas abrangem os estados do Amazonas, Mato Grosso, Pará e Rondônia. Para a geração do padrão-ouro do classificador, foram utilizados os resultados do projeto TerraClass, do ano de 2014. Os dados foram adquiridos gratuitamente no site do INPE [Inpe 2019]. Foram extraídas informações sobre áreas de floresta, pasto e agricultura, correspondendo a um problema de reconhecimento de três classes.

Utilizando os dados de referências do projeto TerraClass, foram gerados retalhos de imagens de tamanho 40x40 *pixels* de cada uma das classes de solo. Foram gerados 4.000, 225.000 e 6.000 retalhos de agricultura, floresta e pasto, respectivamente. Utilizando os conjuntos de retalhos de imagens, foram geradas imagens que definiremos como imagens-mosaico. Cada uma dessas imagens-mosaico tem dimensão 400x400 *pixels* e são geradas selecionando-se aleatoriamente retalhos de agricultura, floresta ou pasto. Para cada imagem-mosaico gerada, gera-se também uma imagem-mosaico correspondente ao padrão-ouro da mesma. Para a construção dessas imagens que constituíram o padrão-ouro, os *pixels* que correspondem a região de floresta, pasto e agricultura foram marcados com o valor 255, 100 e 1, respectivamente. O conjunto de dados gerado possui um total de 3.600 imagens-mosaico, sendo que 3000 foram destinadas para o conjunto de treinamento e 600 para o conjunto de validação. Na Figura 1 mostra-se um exemplo de imagem-mosaico e o padrão-ouro correspondente. Pode ser observado que, dos 100 retalhos, 34 são de agricultura, 35 de floresta e 31 de pasto.

2.2. Métodos

Neste trabalho, foi avaliada uma arquitetura de CNN, três métodos de otimização e três métodos para a melhoria da generalização. Os métodos de otimização avaliados foram o Gradiente Descendente Estocástico com Momento (SGDM), Propagação da Raiz Média Quadrática (RMSProp) e Estimativa de Dinâmica Adaptativa (ADAM). Para cada um desses métodos foram empregados as seguintes técnicas para a melhoria de generalização: nenhuma técnica, camada de *dropout*, regularização L2 e camada de *dropout* com regularização L2. Os dados de entrada utilizados foram as imagens-mosaico. Foram realizadas 12 simulações (1 arquitetura x 3 métodos de otimização x 4 métodos para melhoria de generalização). Em seguida, o modelo com o melhor desempenho no conjunto de validação foi selecionado para classificar algumas imagens da Amazônia.

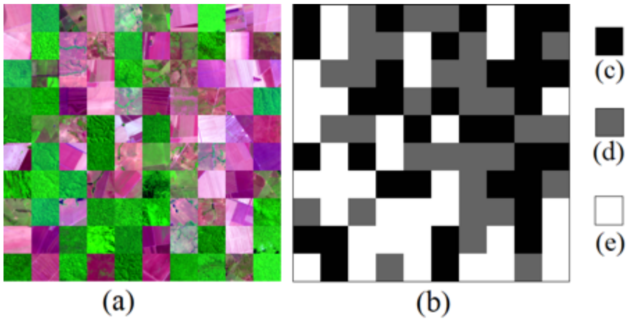


Figura 1. Exemplo de imagem-mosaico. Em (a) mostra-se uma imagem mosaico composta por retalhos de agriculta, floresta e pasto, em (b) a imagem padrão-ouro, e em (c), (d) e (e) temos o padrão-ouro para cada retalho de imagem, correspondente a área de agricultura, pasto e floresta, respectivamente.

2.2.1. Arquitetura CNN, Métricas e Parâmetros de Treinamento

A arquitetura CNN utilizada neste trabalho foi baseada nas arquiteturas propostas no trabalho de [Miyagawa et al. 2018]. Neste trabalho, os autores realizaram a segmentação do lúmen em imagens de tomografia por coerência ótica intravascular (IVOCT). Os melhores resultados para acurácia, valor de Dice e de Jaccard ficaram acima de 99%, 98% e 97%, respectivamente. No trabalho ora apresentado, a CNN utilizada possui duas etapas de subamostragem (*maxpooling*) e duas de sobreamostragem. Antes de cada subamostragem, existem três sequências de camadas convolucionais 3 x 3, camada *batch normalization* e ReLU.

Neste trabalho foram calculadas a Acurácia Global (ACCG), Acurácia Média (ACCM), coeficiente de similaridade de Jaccard (J), coeficiente de similaridade Jaccard ponderado (JP) e Score F1 (F1) [MathWorks 2017]. Uma estação de trabalho com *Windows* 10, Matlab 2019a e com NVIDIA Quadro GV100 32GB e 5120 núcleos CUDA foi utilizado nos experimentos. Em relação aos parâmetros de treinamento da CNN, foi utilizada a taxa de aprendizado inicial = 0,001, fator de queda de taxa de aprendizado = 0,5, número de épocas = 200, tamanho do lote = 2, parâmetro da camada *dropout* = 0,3, fator de regularização L_2= 0,001. Esses valores foram ajustados de maneira experimental.

3. Resultados

A Tabela 1 apresenta o desempenho obtido para a CNN quando combinada com os método de otimização e com os métodos para melhoria da generalização. A Tabela 2 apresenta a matriz de confusão para o modelo com o melhor desempenho no conjunto de validação, que neste caso foi aquele no qual foi empregado o método RMSProp como método de otimização e sem utilizar nenhuma técnica para melhoria da generalização. Esse modelo foi utilizado para segmentar/classificar algumas imagens da região Amazônica. Na Figura 2 mostramos três imagens LANDSAT-8 de regiões da Amazônia com seus respectivos padrão-ouro, e a imagem classificada pelo modelo CNN. As imagens 1 e 2 correspondem regiões da cena 001/66, enquanto que a imagem 3, a uma área da cena 224/68. Os valores de acurácia obtidos para as imagens 1, 2 e 3 foram 98,91%, 96,72% e 93,10% respectivamente.

	Experimento	ACCG (%)	ACCM (%)	J (%)	JP (%)	F1 (%)
1	SGDM	95,984	95,995	92,406	92,389	86,854
2	SGDM/Dropout	94,900	94,919	90,487	90,463	87,111
3	SGDM/L2	96,370	96,381	93,091	93,075	87,981
4	SGDM/Dropout/L2	95,947	95,960	92,328	92,311	89,076
5	ADAM	97,158	97,171	94,537	94,524	89,433
6	ADAM/Dropout	97,265	97,272	94,740	94,727	89,827
7	ADAM/L2	97,095	97,102	94,420	94,408	89,292
8	ADAM/Dropout/L2	96,824	96,838	93,906	93,892	90,229
9	RMSProp	97,467	97,477	95,109	95,097	89,517
10	RMSProp/Dropout	96,807	96,810	93,877	93,865	88,308
11	RMSProp/L2	97,260	97,268	94,729	94,717	90,253
12	RMSProp/Dropout/L2	96,533	96,546	93,387	93,371	90,273

Tabela 1. Desempenho da CNN.

		Classes Preditas		
		Agricultura	**Pasto**	**Floresta**
Classes Reais	**Agricultura**	0,9799	1,96E-02	4,11E-04
	Pasto	4,87E-02	0,9466	4,67E-03
	Floresta	2,56E-04	1,99E-03	0,9978

Tabela 2. Matriz de Confusão para o modelo CNN/RMSProp.

4. Discussão

A partir da Tabela 1, com respeito aos métodos otimização, conclui-se que os resultados para a acurácia obtidos utilizando-se os métodos ADAM e RMSProp ficaram muito próximos, em média 97,08% e 97,017%, respectivamente. Por outro lado, os resultados obtidos com o método SGDM, foram inferiores, com média de 95,80%. A partir da tabela de confusão, verificou-se um maior erro de classificação em regiões em que o padrão-ouro apontava áreas como sendo de pasto mas que foram classificadas como agricultura e vice-versa. O modelo com melhor desempenho foi o modelo CNN com o método RMSProp, sem utilizar nenhuma técnica para melhoria da generalização. Esse modelo foi empregado para avaliar a classificação de algumas regiões na Amazônia apresentadas na Figura 2. A Acurácia obtida para essas três regiões variaram significativamente, entre 98,91% e 93,10%.

5. Conclusões

Este trabalho propôs uma metodologia para segmentar o uso do solo para a região amazônica para as classes de pasto, agricultura e floresta. A metodologia consistiu em avaliar uma arquitetura de CNN e treiná-la utilizando um banco de dados de imagens-mosaico. Foram utilizadas imagens óticas LANDSAT-8 da região amazônica. Diante dos resultados apresentados, pode-se concluir que a metodologia proposta neste trabalho mostrou-se promissora para realizar a tarefa de segmentação/classificação de imagens de sensoriamento para regiões da Amazônia. Com o objetivo de melhorar a acurácia da segmentação das imagens, pretende-se, em trabalhos futuros, avaliar arquiteturas de CNNs mais profundas, utilizar transferência de conhecimento e utilizar mais bandas da imagem LANDSAT-8. Por fim, pretende-se também avaliar a metodologia proposta com um maior número de regiões da Amazônia.

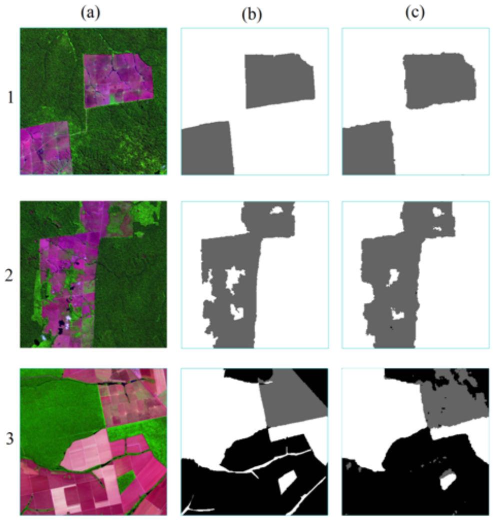


Figura 2. Imagens LANDSAT-8 de regiões da Amazônia. Na coluna (a) temos as imagens originais. Em (b) temos o padrão-ouro. Em (c) temos as imagens classificadas pelo modelo CNN/RMSProp. A acurácia obtida para as imagens 1,2 e 3 foi de 98,91%, 96,72% e 93,10% respectivamente.

Referências

Adarme, M. O., Feitosa, R. Q., Happ, P. N., A., A. C., and Gomes, A. R. (2020). Evaluation of deep learning techniques for deforestation detection in the brazilian amazon and cerrado biomes from remote sensing imagery. *Remote Sensing*, 12:910.

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Springer, New York.

Bem, P. P., Carvalho Junior, O. A., Guimarães, R. F., and Gomes, R. A. T. (2020). Change detection of deforestation in the brazilian amazon using landsat data and convolutional neural networks. *Remote Sensing*, 12:901.

Inpe (2019). Terraclass. *http://www.inpe.br/ cra/projetos_pesquisas/dados_terraclass.php*.

Lecun, Y., Bengio, Y., and Hilton, G. (2015). Deep learning. *Nature*, 521:436–444.

MathWorks (2017). Evaluate semantic segmentation data set against ground truth. *https://in.mathworks.com/help/vision/ref/evaluatesemanticsegmentation.html*.

Miyagawa, M., Costa, M. G. F., Gutierrez, M. A., Costa, J. P. G. F., and Costa Filho, C. F. F. (2018). Lumen segmentation in optical coherence tomography images using convolutional neural network. *40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1:1–4.

Noma, A., Körting, T. S., and Fonseca, L. M. G. (2013). Uma comparação entre classificadores usando regiões e perfis evi para agricultura. *Anais XVI Simpósio Brasileiro de Sensoriamento Remoto*, 1:2250–2257.

Novo, E. M. L. M. (2008). *Sensoriamento Remoto: Princípios e Aplicações*. Edgard Blücher Ltda, São Paulo.

Yu, Z., L, D., Yang, R., and Tang, J. (2019). Selection of landsat 8 oli band combinations for land use and land cover classification. *8th International Conference on Agro-Geoinformatics*, 1:1–5.

Explorando Aspectos Espaciais da Agricultura Familiar Brasileira

Jaudete Daltio, Mário Balan, Marcelo Fernando Fonseca

[1]Empresa Brasileira de Pesquisa Agropecuária (Embrapa)
Campinas – SP – Brazil

{jaudete.daltio,marcelo.fonseca}@embrapa.br,
mario.balan@colaborador.embrapa.br

Abstract. *The term "family farming" has different meanings in academia, government, or social movements. It is characterized by a productive process led by the farmer, in which the family labor surpasses hired labor. The Secretariat for Family Farming and Cooperativism (SAF) is responsible for developing government public policies aimed at this population. The goal of this paper is to present the partial results of cooperation between SAF and Embrapa to explore the geographic aspects of SAF data about Brazilian family farming. We expect that a spatial visualization may contribute to the understanding of territorial coverage of SAF's programs and may be applied as a planning instrument.*

Resumo. *O termo "agricultura familiar" possui diferentes definições na academia, no governo ou movimentos sociais. Ele é caracterizado como um processo produtivo conduzido pelo agricultor, no qual a mão de obra familiar supera a contratada. A Secretaria de Agricultura Familiar e Cooperativismo (SAF) é responsável pelo desenvolvimento de políticas públicas governamentais direcionadas para esta população. O objetivo deste artigo é apresentar os resultados parciais de uma cooperação entre a SAF e a Embrapa para explorar aspectos geográficos dos dados da agricultura familiar brasileira. Espera-se que a visualização espacial possa contribuir na compreensão da cobertura territorial dos programas da SAF e possa ser usada como instrumento de planejamento.*

1. Introdução

A agricultura familiar tem grande relevância na produção de alimentos no Brasil [Silva 2015, Pasqualotto et al. 2019]. Segundo o Censo Agropecuário 2017, a agricultura familiar corresponde a 76,8% dos 5,073 milhões de estabelecimentos rurais do país e foi responsável por 23% da receita agrícola daquele ano-safra [IBGE 2019]. A produção familiar é especialmente significativa na segurança alimentar (mandioca, feijão), horticultura, fruticultura (banana, abacaxi), rebanhos de pequeno porte e pecuária de leite.

No âmbito governamental, a Secretaria de Agricultura Familiar e Cooperativismo (SAF), subordinada ao Ministério da Agricultura, Pecuária e Abastecimento (MAPA), é o orgão responsável pela implantação de políticas públicas para a agricultura familiar. A elegibilidade a essas políticas é mediada pela Declaração de Acesso ao Pronaf [1] (DAP), um instrumento legal que comprova o enquadramento na categoria de pequeno agricultor.

[1] Programa Nacional de Fortalecimento da Agricultura Familiar

A DAP comprova a renda anual e as atividades exercidas e é indispensável para o acesso a linhas de crédito específicas e programas governamentais. A emissão da declaração é gratuita e pode ser requerida nas entidades de assistência técnica públicas ou privadas.

Em 2019, foi estabelecido um Acordo de Cooperação Técnica entre a Embrapa e a SAF com o objetivo de explorar o potencial cartográfico da DAP. O intuito é agregar os microdados em termos espaciais e adotar geotecnologias para apoiar estratégias de inclusão social e produtiva para geração de renda de agricultores familiares atendidos pelo programa. Iniciativas similares têm mostrado o potencial de ferramentas deste tipo na gestão pública [da Silva et al. 2018, da Luz 2019].

O objetivo deste artigo é descrever os resultados iniciais desta cooperação. Partindo de uma extração parcial dos dados de DAPs ativas, foi possível elaborar um novo modelo de dados, agregado espacialmente por município, com um conjunto de atributos descritivos. A partir dos novos dados, implementou-se um conjunto de painéis interativos para a exploração integrada, que permite seleções espaciais e cruzamento de variáveis.

Espera-se que os resultados deste trabalho possam contribuir para a gestão do programa pela SAF, avançando na compreensão do alcance territorial das ações de cooperativismo e agricultura familiar. Acredita-se que será possível identificar especificidades regionais, gargalos ou embasar políticas públicas locais.

2. Dados e Desafios do Projeto

2.1. Dados da Declaração de Aptidão ao Pronaf (DAP)

Para o desenvolvimento das atividades, a SAF disponibilizou o acesso via FTP a um relatório chamado “Layout Único”. Trata-se de um conjunto de 5 arquivos textuais, extraídos diariamente, contendo dados relativos às DAPs ativas [2]. Os arquivos são posicionais (sem delimitadores) e podem ter subdivisões internas, combinando diferentes tipos de registros de dados (identificados por um conjunto de caracteres no início de cada linha).

A SAF disponibilizou um manual contendo diretivas de interpretação do conteúdo de cada arquivo do relatório – quais tipos de registros esperados para cada arquivo, quais campos cada tipo de registro possui e seu tamanho – e tabelas de apoio para a interpretação do conteúdo dos campos com domínio determinado (conjunto de valores válidos). O relatório possui a seguinte composição:
arquivo1.txt: apresenta dados do agricultor, da propriedade e do seu registro no programa. Contém 4 subdivisões: dados do titular, da propriedade e do registro (DAP Principal) e de filhos ou cônjuges com registro próprio (DAP Jovem e DAP Agregada);
arquivo2.txt: dados de pessoas jurídicas com registro de DAP;
arquivo3.txt: dados de caracterização das DAPs, com 3 subdivisões: organizações sociais, atividades principais e uso da terra;
arquivo4.txt: membros das pessoas jurídicas com registro de DAP, com 2 subdivisões: CPFs dos sócios e CNPJs de entidades associadas;
arquivo5.txt: produtos de origem na agricultura familiar associada às DAPs.

O volume de dados em questão é de aproximadamente 5GB, que correspondem a 5.824 DAPs Jovem, 493 DAPs Agregadas, 1.558 DAPs Jurídicas e 2.800.075 registros de

[2] A validade das DAPs atualmente é de dois anos, porém este período tem variado desde a existência do programa.

DAPs ativas, caracterizadas em várias atividades principais, organizadas de forma coletiva ou não e com renda associada a cerca de 300 produtos agropecuários (relatório extraído em 19/10/2020). O acesso a este relatório foi alvo de termo de confidencialidade em consonância com Lei Geral de Proteção de Dados (LGPD) e era um requisito essencial do projeto que os resultados gerados não permitissem a identificação pessoal.

2.2. Aspectos Espaciais

Considerando-se os atributos disponíveis, foram elencadas as possibilidades de espacialização dos dados. O ideal era utilizar a localização da propriedade do agricultor. O relatório possui dois atributos com este fim: (1) **Município do imóvel principal**: atributo do titular da DAP e (2) **Localização do imóvel principal**: atributo da DAP principal. Ambas alternativas se mostraram inviáveis. A opção (1) possui valores nulos para um terço das DAPs – trata-se de uma adaptação recente no formulário de cadastro. A opção (2) apresenta conteúdo textual aberto que não é passível de um procedimento automático de espacialização – nome do sítio (ex. *Sítio Cangandu*), identificação do povoado (ex. *Povoado Taboca*) ou indicações de como chegar na propriedade (ex. *Rod Pitanguipapagaio entrar a direita no km 15 e andar mais 10 km até a propriedade*).

Diante da impossibilidade de adotar a localização da propriedade, a opção encontrada para espacialização foi utilizar o **Município de residência**, um atributo do titular da DAP (agricultor), que corresponde ao padrão de geocódigos adotado pelo IBGE.

2.3. Desafios Enfrentados

O desenvolvimento deste projeto atravessou vários desafios técnicos, iniciando com a interpretação dos dados disponibilizados – a identificação das entidades e seus relacionamentos. A Figura 1 apresenta uma visão geral deste modelo de dados. A entidade central é a **DAP Principal**, que define o registro de um agricultor e sua propriedade no programa de agricultura familiar, e que possui um identificador único denominado *controle externo* usado para relacioná-lo com as demais entidades de caracterização: **Atividade Principal**, **Uso da Terra** e **Produto** (entidades com tabelas de apoio específicas). Como pode ser visto na figura, um mesmo registro de DAP pode ter nenhum ou múltiplos relacionamentos com estas entidades de caracterização (cardinalidade 0:*, em ambas extremidades).

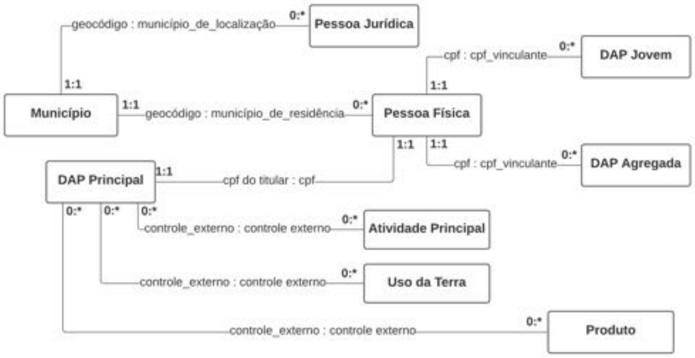


Figura 1. Modelo de Dados Aderente ao "Layout Único"

O relacionamento entre o agricultor (**Pessoa Física**) e a **DAP Principal** é materializado pelo seu *CPF*. Também o *CPF* é usado para associar **DAP Jovem** e **DAP Agregada** que, por definição, precisam estar vinculadas a titulares de **DAP Principal**. Não há caracterização associada a estas DAPs, assim como às DAPs emitidas para pessoas jurídicas. O atributo município está presente em **Pessoa Física** e **Pessoa Jurídica**.

A partir deste modelo de dados, outros dois desafios foram endereçados. O primeiro deles foi a validação da aderência dos dados a este modelo que, embora teoricamente correto, apresentava pontos de divergência quando confrontado aos dados. Vários registros foram descartados, de quase todas as entidades, por não terem relação com nenhuma DAP ativa. Registros cujos atributos tinham valores fora das tabelas de apoio foram corrigidos ou descartados. Embora o *CPF* fosse o único meio de identificação de um agricultor, havia vários *CPFs* com mais de um registro e com valores diferentes nas demais variáveis (múltiplos registros em **Titular Pessoa Física**). Todos os casos foram discutidos e endereçados para que os dados do layout único pudessem dar origem a um conjunto íntegro de registros que serviria de base para as demais etapas do projeto.

O segundo desafio foi a seleção dos atributos (e valores) mais relevantes para fins de classificação, filtragem e agregação na elaboração das visões territoriais. Foram selecionadas 15 em um universo de 100 variáveis para objeto do trabalho. Destas, a maioria demandou classificações ou reclassificações nos registros originais. Encontrar as classes pertinentes para cada atributo foi um processo interativo, que deveria acomodar as necessidades da SAF e também ser aderente à distribuição estatística dos dados.

3. Implementação e Resultados Iniciais

A partir do modelo de dados criado (Figura 1) e da seleção dos atributos a serem trabalhados, os dados foram (re)classificados, agregados e espacializados. Os atributos selecionados, além do município, foram: sexo, idade, escolaridade e quantidade de membros da família (dados do agricultor), enquadramento no programa, total de DAPs por tipo (Principal, Jovem, Agregada e Jurídica), categoria e tamanho da propriedade, atividades principais, uso da terra e produtos associados.

Apenas os atributos sexo e categoria da propriedade não demandaram pré-processamentos antes da agregação e espacialização. Enquadramento, escolaridade, atividades principais, uso da terra e produtos foram reclassificados, tanto para sumarizar a quantidade de categorias, quanto para agregar dados similares e remover dados não relevantes. A classe de idade foi calculada a partir da data de nascimento do agricultor (jovens até 30 anos e idosos a partir de 60 anos). A classe de tamanho da família foi calculada a partir do número de familiares residentes no estabelecimento informada em cada DAP, dado relevante já que grande parte da mão de obra é familiar. O tamanho da propriedade foi classificado de acordo com a quantidade de módulos fiscais [3] que representa. Por definição, as pequenas propriedades alvo de DAPs deveriam ter até 4 módulos, exceto casos de uso coletivo da terra.

A partir destas reclassificações, os dados mostrados na Figura 1 foram agregados por município, dando origem a outras relações. As novas relações foram segmentadas de

[3] Índices básicos de 2013 do Sistema Nacional de Cadastro Rural, disponível em .www.incra.gov.br/pt/modulo-fiscal.html

acordo com cardinalidade original dos atributos. O intuito foi viabilizar filtros cruzados entre os dados – ou seja, dados de cardinalidade ***:*** deram origem a relações próprias, para fins de caracterização do município; e dados de cardinalidade **1:1** foram agregados conjuntamente considerando cada combinação possível de valores (sexo, idade, escolaridade, tamanho da família, enquadramento do agricultor, tamanho e categoria da propriedade, um total de 9.720 combinações de valores diferentes).

A apresentação dos dados foi viabilizada pela elaboração de um conjunto de painéis interativos desenvolvidos na ferramenta *Tableau* – um software para visualização de dados que permite a elaboração de componentes interativos. Os atributos disponíveis são categorizados como dimensões ou medições – dimensões são os valores qualitativos, usados para segmentação e medições são os valores numéricos e quantitativos, passíveis de agregação (ex. soma, média). Todas as classificações criadas nos dados originais foram mapeadas como dimensões e a quantidade de DAPs como medições.

O conjunto de componentes desenvolvidos abrange 5 painéis temáticos (**O Agricultor**, **A Propriedade**, **Atividade Principal**, **Uso da Terra** e **Produto**) e um agregador, que provê o **Panorama Geral** dos dados. Com exceção deste último, os painéis apresentam algum subconjunto dos atributos trabalhados (de acordo com suas agregações temáticas) e sua representação territorial (em mapa). Os dados **(1:1)** estão representados nos dois primeiros painéis e os dados **(*:*)** nos três painéis subsequentes.

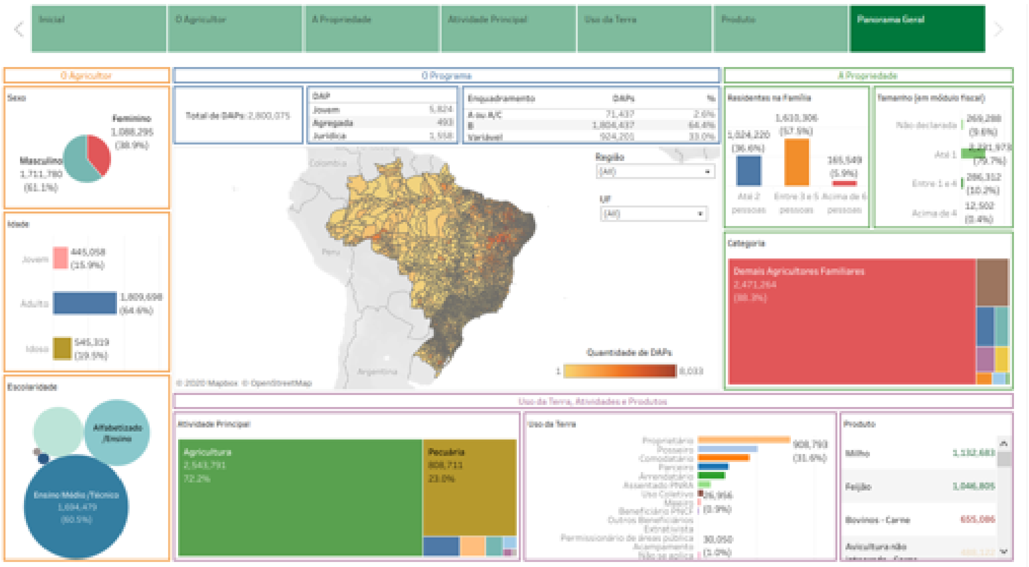
Em cada painel, as classes dos atributos são apresentadas em termos visuais (gráficos) e numéricos, com seus valores absolutos e percentuais. Qualquer seleção espacial no mapa automaticamente gera um re-cálculo de todos os atributos mostrados e a atualização dos dados apresentados. De forma análoga, qualquer seleção de classes de atributos irá atualizar todos os demais dados apresentados e filtrar o mapa para apresentar apenas os municípios onde ocorre a seleção de interesse.

A Figura 2 mostra painel **Panorama Geral** e o menu de navegação para acesso aos demais painéis. O mapa, elemento central deste painel, apresenta a quantidade de DAPs e pode ser filtrado por região e/ou unidade da federação. É possível visualizar no mapa, por exemplo, apenas o quantitativo de DAPs de sexo feminino + jovens + com ensino superior + com até duas pessoas na família + em propriedades de até 1 módulo fiscal + aquicultores. A partir da restrição de munícipios causada por esta seleção, os demais dados do painel (atividades principais, uso da terra e produtos) são atualizados para mostrar a dispersão do total de DAPs nas suas categorias correspondentes. O caminho inverso também é possível – a partir da seleção de um produto, por exemplo, é possível obter o panorama dos municípios que possuem DAPs associadas a este produto em relação a todas as demais variáveis. A versão atual dos painéis está disponível na nuvem pública da plataforma [4] e apresenta dados de todo o território nacional.

4. Contribuições e Próximos Passos

Após um vasto trabalho de interpretação e organização dos dados da agricultura familiar da SAF, foi possível construir uma base de dados espacial que agrega diferentes conceitos relacionados ao programa DAP e, a partir dela, elaborar um conjunto de painéis interativos

[4] Acesso em `https://public.tableau.com/profile/gite3613#!/vizhome/agricultura_familiar/historia`



DAP	
Jovem	5,824
Agregada	493
Jurídica	1,558

Enquadramento	DAPs	%
A ou A/C	71,437	2.6%
B	1,804,437	64.4%
Variável	924,201	33.0%

Produto	
Milho	1,132,683
Feijão	1,046,805
Bovinos - Carne	655,086

Figura 2. Painel elaborado para exploração dos dados espaciais das DAP

para exploração e análise dos dados. Espera-se que esta visão espacial possa contribuir para orientar políticas de cooperativismo da Secretaria nos estados e municípios e que possa apoiar ações de planejamento por parte dos órgãos gestores.

Os resultados parciais estão atualmente em fase de validação pela SAF e, justamente para facilitar este processo, os valores numéricos e percentuais foram mantidos de forma explícita em todas as planilhas dos painéis. Os próximos passos abrangem a verificação da adequabilidade das apresentações visuais propostas (tipos de gráficos e adequações de cores e rótulos) e a consolidação de um processo automático para atualizar a criação da nova estrutura de dados proposta a partir do layout único, gerado diariamente.

Agradecimentos. Os autores agradecem a Luísa Martins Fernandes, da Secretaria de Agricultura Familiar e Cooperativismo (SAF/MAPA), por seu auxílio na interpretação dos dados, solução de problemas e validação dos resultados.

Referências

da Luz, J. S. (2019). Análise do Programa Minha Casa Minha Vida: Um Recorte Espacial do Eixo Goiânia – Anápolis – Brasília. In *Anais do XVI SIMPURB*, pages 4132–4149.

da Silva, J. P., Guarneri, H., Arenas, F. C., de Paula, E. V., and Camboim, S. P. (2018). Uso de um Dashboard Geoespacial como ferramenta de suporte para o diagnóstico socioeconômico e ambiental da Reserva Biológica Bom Jesus – Litoral do Paraná. In *Proceedings XIX GEOINFO*, pages 152–157, Campina Grande, PB.

IBGE (2019). *Censo Agropecuário 2017 - Resultados Definitivos*. IBGE.

Pasqualotto, N., Kaufmann, M. P., and Wizniewsky, J. G. (2019). *Agricultura Familiar e Desenvolvimento Sustentável*. Universidade Federal de Santa Maria.

Silva, S. P. (2015). *A Agricultura Familiar e Suas Múltiplas Interações Com o Território: Uma Análise de Suas Características Multifuncionais e Pluriativas*. Ipea.

Visualização de Dados de Origem-Destino - Foco em Unidades de Saúde e Educação

Fernando X. De Souza[1], Paulo R. Bauer[1], Keiko Fonseca[1], Tatiana Gadda[1], Rita Berardi[1], Nádia P. Kozievitch[1]

[1]Universidade Tecnológica Federal do Paraná (UTFPR)
Avenida Sete de Setembro, 3165
Departamento Acadêmico de Informática – DAINF – Curitiba – Brasil

{fernando.1994,paulobauer}@alunos.utfpr.edu.br,
{keiko,tatianagadda,ritaberardi,nadiap}@utfpr.edu.br

Abstract. *Understanding the dynamics of displacement of people in the public transport system in order to improve this service is a constant challenge in large urban centers. This article presents a visualization tool for origin-destination data (using heatmap, clustering and flux map), with focus on health and education units (citizens attractions and traffic generators). A preliminary evaluation using data from Curitiba showed positive results from the users point of view.*

Resumo. *Compreender a dinâmica de deslocamento das pessoas no sistema de transporte público visando melhorar este serviço é um desafio constante em grandes centros urbanos. Este artigo apresenta uma ferramenta de visualização para dados de origem-destino (usando mapa de calor, clusterização e mapa de fluxo), com foco em unidades de saúde e educação (atratoras de cidadãos e geradoras de tráfego). Uma avaliação preliminar da aplicação usando dados de Curitiba mostrou resultados positivos do ponto de vista de usuários.*

1. Introdução

O crescimento das cidades traz desafios relacionados à mobilidade urbana, e técnicas que envolvem processos automatizados, onde é possível produzir, consumir e distribuir grandes quantidades de informação em tempo real [Lemos 2013] vêm sendo utilizadas para propor novas soluções. Nesse cenário, a administração pública de cidades como Paris[1], Nova Iorque[2] e Moscou[3], disponibilizam seus dados seguindo a tendência de dados abertos. Na mesma tendência, a cidade de Curitiba disponibiliza seus dados através de várias fontes (Instituto de Pesquisa e Planejamento Urbano (IPPUC)[4], Urbanização de Curitiba (URBS)[5] e dados abertos da Prefeitura[6]). Em Curitiba, diariamente, uma média de 1229 veículos transportam aproximadamente 1.365.615 passageiros (sendo 60% deles usando cartões inteligentes) por 251 rotas diferentes e 24 terminais de ônibus, resultando, em média, 14.166 viagens de ônibus em 300.373 km por dia, em 6500 pontos de ônibus.

[1] opendata.paris.fr - Acessado em Mar.2020.

[2] opendata.cityofnewyork.us - Acessado em Mar. 2020.

[3] data.gov.ru - Acessado em Mar. 2020.

[4] ippuc.org.br - Acessado em Mar.2020.

[5] urbs.curitiba.pr.gov.br/ - Acessado em Mar.2020.

[6] www.curitiba.pr.gov.br/dadosabertos/ - Acessado em Mar.2020.

Do ponto de vista de dados e geoprocessamento, o sistema de transporte público possui componentes geográficos e temporais, além de dados sobre sua dinâmica, como por exemplo, viagens. Uma viagem de ônibus (na qual um usuário vai de uma origem a um destino e volta a uma origem) pode ser caracterizada por datas e coordenadas de partidas e chegadas (dados de destino de origem) associados a uma linha de ônibus (rota predeterminada). Uma viagem também pode ter informações como nome e código de linha, código do veículo, ID do cartão inteligente, sexo e data de nascimento do passageiro e latitude e longitude do ônibus (por exemplo, a cada cinco minutos).

Relatórios de origem-destino (OD) como [IPPUC 2017] já indicaram a importância de pontos de interesse, como entidades de saúde e educação (que concentram uma grande quantidade de população), para entender a dinâmica destes dados. Entretanto, não há nenhuma ferramenta desenvolvida para os dados de Curitiba que aborde as variáveis citadas (espaciais/temporais) permitindo compreender a dinâmica do transporte na cidade. A partir das demandas identificadas em sua pesquisa, [Parcianello 2020] desenvolveu uma interface para auxiliar na análise do deslocamento dos passageiros. Este artigo tem como objetivo expandir as opções de análises do trabalho desenvolvido por [Parcianello 2020], por meio de uma extensão de sua ferramenta, incluindo dados da localização de Pontos de Interesse (POI) como escolas e hospitais (equipamentos públicos e privados que concentram população) junto à adaptações da interface para consulta e apresentação desses dados.

2. Trabalhos Relacionados

O transporte público é uma das áreas mais críticas sob a perspectiva da cidade. Os desafios de mobilidade já ganharam atenção da comunidade científica no Brasil [7]. Nesta direção, diferentes serviços telemáticos de transporte já foram propostos em [Diniz Jr. 2017] (como localização de rotas sem ônibus, média de lotação em ônibus, alerta para diferentes rotas e alertas de velocidade), usando os mesmos dados propostos neste artigo. Já dados do Rio de Janeiro foram usados em [Cruz et al. 2018] para a identificação e a classificação de anomalias através de um processo que envolve integração de *open data* e mineração de dados via uso do algoritmo *Apriori*. A pesquisa de [Spadon et al. 2018] também compara dados abertos da rede viária, da demografia, da extensão territorial e de outros indicadores urbanos de 645 cidades do estado de São Paulo. Este último estudo também aplica conceitos de redes complexas e algoritmos de clusterização para classificar tais cidades por semelhança sob diferentes perspectivas.

Por outro lado, as imagens no geoprocessamento são importantes mas no transporte o movimento é crucial (exemplos em [Andrienko and Andrienko 2013]). Como já mencionado em [Ferreira et al. 2013]: grande parte do trabalho é baseada em dados de trajetória. Já em OD, os dados multivariados têm apenas o início e o posições finais juntamente com atributos associados ao movimento e ao tempo gasto. No mesmo artigo, a visualização de dados de táxi é estudada e uma solução é proposta para não especialistas, com diferentes abordagens de visualização. Ao se tratar de apresentação de dados, vários projetos podem ser mencionados, como DataViz[8], VisualComplexity[9] ou

[7] http://www.sbc.org.br/documentos-da-sbc/send/141-grandes-desafios/802-grandesdesafiosdacomputao/no-brasil- Acessado em Jun. 2020.

[8] datavizproject.com - Acessado em Jun. 2020.

[9] www.visualcomplexity.com - Acessado em Jun. 2020.

CityGeographics[10]. Por outro lado, dados abertos já estão sendo disponibilizados com soluções interativas, como Manhattan Population Explorer[11] ou Transit Accident Dashboard from IPPUC[12]. Algoritmos de clusterização ou gráficos)como marcadores, mapa térmico) também podem ser usados na visualização, como em [Vila 2016], onde é proposta uma solução web-mobile que permite a visualização espaço-temporal de dados georreferenciados.

3. O Protótipo

Os dados de OD utilizados no protótipo são os dados mencionados em [Parcianello 2020] (e detalhes mais específicos podem ser verificados em [Diniz Jr. 2017]), e os dados de saúde (hospitais, laboratórios, CAPS, unidade de saúde) são do portal de dados abertos de Curitiba. Para o armazenamento dos dados, foi utilizado PostgreSQL[13] e Postgis[14]. Os pontos de interesse escolhidos foram equipamentos urbanos da cidade: transporte (pontos de ônibus e terminais), educação (escolas de educação básica, de jovens e adultos, especial, infantil, profissional técnica, superior, ensino médio).

A ferramenta em [Parcianello 2020] é dividida em duas partes: busca e visualização (usando clusterização e mapa de calor) e o painel com gráficos (total de embarques e desembarques for faixa etária, total de embarques por sexo, embarques por sexo ao longo do período, total de embarques por idade, e embarques por faixa etária ao longo do período). Este trabalho adicionou na visualização o mapa de fluxo, técnica de visualização que ilustra as viagens realizadas através de setas no mapa ligando os pontos de origem e destino das viagens, além da adição de dados de localização de instituições de saúde e educação como Pontos de Interesse.

A Figura 1 mostra um diagrama de como as tecnologias foram combinadas para o desenvolvimento do protótipo. Na arquitetura original desenvolvida por [Parcianello 2020] foram utilizadas as bibliotecas Bootstrap, Leaflet, Jquery e Open Street Map para o lado do cliente, e neste trabalho foi adicionado a biblioteca Leaflet.Migration.Js para utilização de mapas de fluxo. Para o lado do servidor foram utilizadas tecnologias Apache e PHP, além da base de dados.

Na prática, a interface foi modificada nos seguintes itens: a Figura 2-A mostra a nova opção para selecionar a categoria do tipo de Ponto de Interesse a ser pesquisado, na Figura 2-B é possível selecionar uma subcategoria para afunilar os resultados, na Figura 2-C foi adicionada uma nova opção para utilizar a técnica de mapa de fluxo na exibição dos resultados da pesquisa, e por fim na Figura 2-D encontra-se os botões para realizar a pesquisa, configurar e retirar os resultados da última pesquisa do mapa.

Como exemplo, considere as viagens realizadas no dia 01 de Outubro de 2017, entre as 0:00 até às 23:59, tendo como ponto de destino as instituições ”UTFPR - Sede Central”e ”UFPR - Setor de Ciências Sociais Aplicadas”, ilustrada na Figura 3. As setas verdes ilustram quais as que tiveram como destino a UTFPR, e as setas azuis quais foram

[10] `citygeographics.org` - Acessado em Jun. 2020.

[11] `manpopex.us` - Acessado em Jun. 2020.

[12] `ippuc.org.br/mapasinterativos/AcidentesDeTransito/dashboard.html` -Acessado em Jun. 2020.

[13] `https://www.postgresql.org/` - Acessado em Jun 2020.

[14] `https://postgis.net/` - Acessado em Ago 2020.

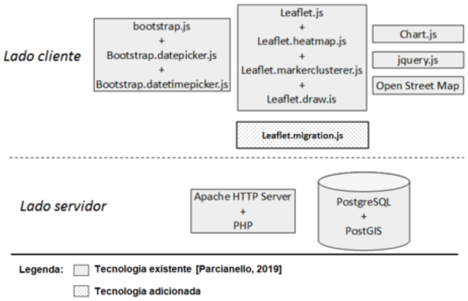


Figura 1. Tecnologias utilizadas no protótipo.

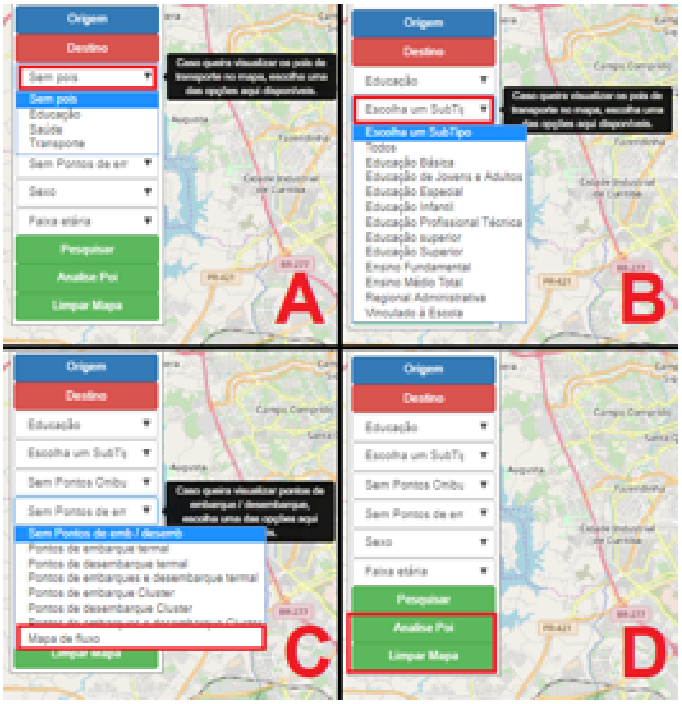
destinadas à UFPR.

A Figura 4-esquerda apresenta as instituições de ensino superior que obtiveram em suas imediações (400 metros) a maior concentração de usuários como destino, entre os dias 23 a 29 de Outubro de 2017. A Figura 4-direita indica a quantidade de usuários que tiveram como destino as imediações de cada Instituição. Note que a instituição Faculdades Integradas Camões possui em suas imediações a maior quantidade de usuários.

Com o intuito de avaliar de maneira preliminar a aplicação, foi desenvolvido um teste contendo 3 tarefas, aplicado a um grupo de cinco pessoas no período entre os dias 11/11/2019 a 20/11/2019. O perfil dos entrevistados é de estudantes da instituição UTFPR, graduandos e mestrandos de computação. Os testes foram realizados separadamente, sob observação direta, com os entrevistados tendo acesso à aplicação, às perguntas e a um mapa da cidade de Curitiba. As tarefas escolhidas foram: (1) Descobrir a origem das pessoas que chegaram de ônibus nas redondezas da UTFPR-Centro, no dia 04 de Outubro de 2017; (2) Descobrir para que bairro mais foram e a porcentagem de pessoas do sexo feminino que utilizaram ônibus na região do Terminal Guadalupe na semana de 8 à 14 de Outubro de 2017 e (3) Descobrir quantas pessoas utilizaram ônibus no Centro com destino à todas as instituições de saúde Especializadas em Saúde Mental, do dia 01 ao dia 31 Outubro de 2017. Como resultados, todos os entrevistados conseguiram resolver as 3 tarefas, apresentando respostas compatíveis com o que era esperado de cada tarefa. Um vídeo da aplicação pode ser encontrado em [15].

Lições aprendidas. Vários fatores devem ser considerados para projetar e impactar os protótipos OD: 1) os filtros e sua classificação (quais são relevantes para o usuário, quão fáceis são seu uso e implementação, e quais resultados são mais bem compreendidos); 2) o usuário final e quais restrições e conhecimentos de tecnologia estão presentes (percepções de como a experiência do usuário com as tecnologias aplicadas impacta na compreensão dos resultados); 3) a visualização geral dos dados e quais estatísticas básicas impactam o usuário (em que visão geral o usuário final está interessado); 4) variações de movimento e região (intra-região, regional, variações menores (como K-Means); 5) a análise da arquitetura, dados, estrutura de banco de dados e consultas a fim de melhorar a otimização (técnicas como otimização de consulta, índices , partições de mesa); 6) teste e integração de diferentes bibliotecas (às vezes não disponíveis gratuitamente, ou difíceis de integrar, ou indisponíveis para diferentes sistemas operacionais).

[15] `https://youtu.be/aj7-TMSBXO4` - Acessado em Jun.2020.



Caso queira visualizar os pois de transporte no mapa, escolha uma das opções aqui disponíveis.

Caso queira visualizar os pois de transporte no mapa, escolha uma das opções aqui disponíveis.

Caso queira visualizar pontos de embarque / desembarque, escolha uma das opções aqui ...s.

Figura 2. Novidades apresentadas na interface.

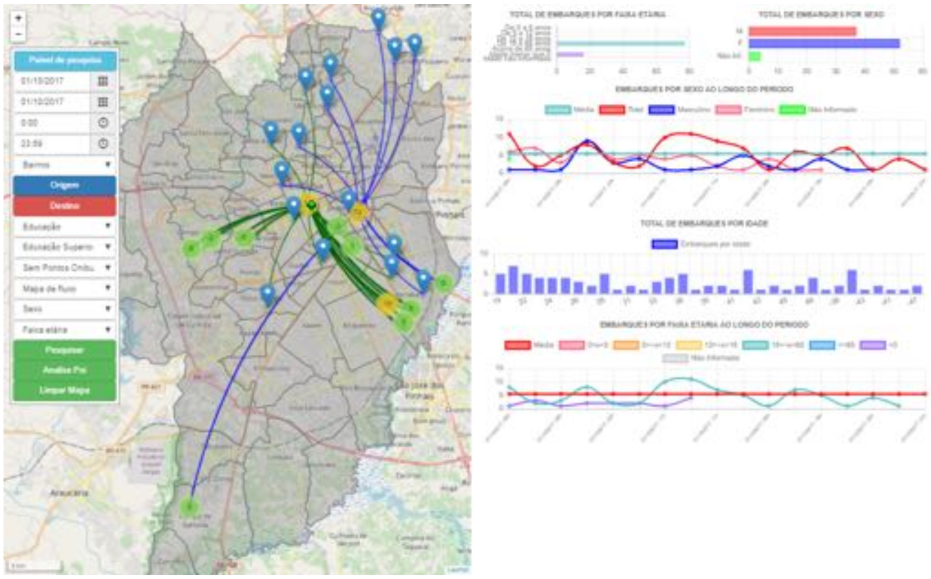


Figura 3. Resultado da consulta de origem-destino.

4. Conclusão

Este artigo apresentou uma aplicação de visualização de dados de Origem-Destino, com um painel de busca e visualização (mapa de calor, clusterização e mapa de fluxo) e um painel de análise (categorização dos dados em gênero, horário e idade). O objetivo é auxiliar no estudo dos padrões de deslocamento dos usuários do transporte coletivo, entre

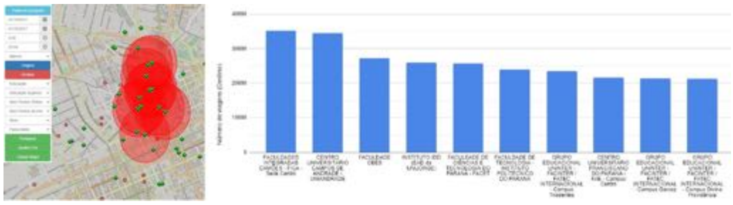


Figura 4. Instituições de Ensino Superior que obtiveram em suas imediações a maior quantidade de usuários como destino (esquerda), e a quantidade respectiva de cada uma (direita).

equipamentos urbanos de transporte, saúde e de educação. Uma avaliação preliminar obteve uma aceitação positiva, onde todos os usuários amostrados conseguiram concluir as tarefas propostas. Como trabalhos futuros, podemos listar a inclusão de outros filtros de pequisa, as rotas e horários de maior fluxo de pessoas, aplicar técnicas de clusterização ao mapa de fluxo e inserir uma interface que permita realizar consultas diretamente em SQL.

Referências

Andrienko, N. and Andrienko, G. (2013). Visual analytics of movement: An overview of methods, tools and procedures. *Information Visualization*, 12(1):3–24.

Cruz, A., Ferreira, J., Carvalho, D., Mendes, E., Pacitti, E., Coutinho, R., Porto, F., and Ogasawara, E. (2018). Detecção de anomalias frequentes no transporte rodoviário urbano. In *SBBD: Brazilian Symposium on Databases*, pages 271–276. SBC.

Diniz Jr., P. C. (2017). Serviços telemáticos em uma rede de transporte público baseados em veículos conectados e dados abertos. Master's thesis, Universidade Tecnológica Federal do Paraná.

Ferreira, N., Poco, J., Vo, H. T., Freire, J., and Silva, C. T. (2013). Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2149–2158.

IPPUC (2017). Consolidação de dados de oferta, demanda, sistema viário e zoneamento - relatório 5 - pesquisa de origem-destino domiciliar. disponível em `http://www.ippuc.org.br/visualizar.php?doc=http://admsite2013.ippuc.org.br/arquivos/documentos/D536/D536_002_BR.pdf`. Technical report, IPPUC, Curitiba.

Lemos, A. (2013). Cidades inteligentes. *GV-executivo*, 12(2):46–49.

Parcianello, Y. (2020). Análise de origem-destino do uso do sistema de transporte coletivo de curitiba sob o ponto de vista de regions of interest. Master's thesis, Universidade Tecnológica Federal do Paraná.

Spadon, G., Scabora, L. C., Nesso-Jr, M. R., Traina Junior, C., and Rodrigues Junior, J. F. (2018). Caracterização topológica de redes viárias por meio da análise de vetores de características e técnicas de agrupamento. In *SBBD'18*, pages 157–168. SBC.

Vila, J. J. F. R. (2016). Clusterização e visualização espaço-temporal de dados georreferenciados adaptando o algoritmo marker clusterer: um caso de uso em Curitiba. Master's thesis, Universidade Tecnológica Federal do Paraná, Campus Curitiba.

Designação de Veículos Autônomos em Abordagens Mono-objetivo e Multiobjetivo

Catrine dos Santos Oliveira[1], Marconi de Arruda Perreira[1]

[1]Departamento de Tecnologia e Eng. Civil, Computação e Humanidades
Universidade Federal de São João Del Rei - Campus Alto Paraopeba MG 443, KM 7
Ouro Branco – MG – Brasil

catrine.sntsoliveira@gmail.com, marconi@ufsj.edu.br

Abstract. *This work proposes alternatives to optimize the designation of autonomous vehicles according to the distance traveled by the automobile to the requester and the time that this client waits for this vehicle to arrive to answer the call. This paper is the continuation of another work, whose exclusive focus was to minimize the distance that vehicles must travel to serve the customer. The tool proposed here presents improvements both in execution time and in a more complete analysis: the optimization model used has become multiobjective, minimizing both the service time and the distance to be covered. The results show 90% of gain in execution time and 15% in service time when compared to the previous version.*

Resumo. *Este trabalho propõe alternativas para otimização da designação de veículos autônomos, em termos da distância percorrida pelo veículo até o solicitante e o tempo que este cliente espera para que o veículo chegue para atender ao chamado. O estudo em questão é a continuidade de outro trabalho, o qual focava exclusivamente na minimização da distância que os veículos devem percorrer para atender o cliente. A ferramenta proposta nesse artigo apresenta melhorias tanto no tempo de execução quanto uma análise mais completa: o modelo de otimização usado agora é multiobjetivo, minimizando tanto o tempo de atendimento quanto a distância a ser percorrida. Os resultados apresentam um ganho da ordem de 90% no tempo de execução e em 15% no tempo de atendimento, em relação à versão anterior.*

1. Introdução

O cenário atual de transporte urbano conta com novas formas de mobilidade, plataformas que conectam veículos diretamente aos passageiros, tais como Uber, 99Taxi e Cabify, os quais oferecem uma forma de utilização de veículos sob demanda. Segundo [Fonseca, 2020] a expectativa é que até 2030, carros autônomos de diferentes níveis se tornem um grande mercado mundial movimentando um capital em torno de US$ 60 bilhões. Desse modo, tanto na conjuntura vigente quanto no futuro próximo, o problema de alocação de veículos mostra-se relevante.

O problema de alocação consiste na atribuição de veículos a um conjunto predeterminado de viagens (passageiros), tendo em vista a minimização dos custos operacionais. Logo, uma alocação otimizada visando a redução do tempo de atendimento ao cliente e a redução da distância percorrida pelo veículo implica em menos gastos, consequentemente o serviço pode ser oferecido a preços mais acessíveis. Percebe-se assim que o problema possui uma natureza multiobjetivo.

[Alcântara & Pereira, 2019] apresentaram um estudo para o problema de alocação de veículos por meio da abordagem mono-objetivo, com intuito de minimizar a distância total percorrida pelos veículos. A abordagem proposta foi comparada a dois outros algoritmos: o primeiro reunia as chamadas recebidas durante uma janela de 50 segundos e executava o Algoritmo Húngaro [Kuhn, 1955] para alocação dos veículos para cada chamada; o segundo é baseado numa estratégia gulosa, na qual, para cada chamada realizada, aloca-se o veículo que necessita percorrer a menor distância para atender à requisição. Como o modelo baseado no algoritmo Húngaro resulta numa resposta eficiente, porém demanda muito tempo de processamento e o modelo baseado na estratégia gulosa gera uma resposta pior que a anterior, apesar de gerar o resultado rapidamente, o trabalho de [Alcântara & Pereira, 2019] focou na proposta de um algoritmo que gerasse uma resposta que tentasse ser o mais próxima o possível da obtida pelo modelo Húngaro, porém gerada num menor tempo, tentando ser tão rápida quanto a obtida pela estratégia gulosa.

O presente artigo propõe melhorias no trabalho supracitado. Inicialmente, o foco voltou-se para a elaboração de versões aperfeiçoadas do algoritmo genético para reduzir o tempo de execução sem perder na qualidade da resposta. Posteriormente, o modelo se tornou multiobjetivo, minimizando tanto o tempo de atendimento quanto a distância a ser percorrida.

2. Algoritmo Genético Mono-objetivo

O Algoritmo Genético proposto em [Alcântara & Pereira, 2019] ainda apresentava um tempo de processamento muito maior que o demandado pelo modelo guloso. Assim, o presente estudo propõe estratégias para reduzir o tempo de resposta do Algoritmo Genético, com a finalidade de aproximar-se do tempo de execução do Algoritmo Guloso, sem perdas na minimização da distância. Nessa fase, manteve-se as configurações do cenário de simulações utilizadas no trabalho anterior, de forma a garantir que as comparações entre os trabalhos sejam válidas.

Uma forma de diminuir o tempo de execução de um algoritmo genético é reduzindo a quantidade de gerações criadas. Usar um número fixo de gerações a serem produzidas pode fazer com que o algoritmo gere novas soluções que não estão mais apresentando melhorias em relação às respostas anteriores. Esse era o caso do trabalho anterior, onde o algoritmo sempre gerava um número fixo de gerações igual a 200.

Com o intuito de interromper a produção de gerações quando não se observava mais melhorias significativas nas soluções geradas, ou seja, nos casos em que a variabilidade das novas gerações é baixa. Elaborou-se três novas versões (Gen-I, Gen-II Gen-III), que apresentam diferentes formas de verificação da evolução da população de soluções geradas. Cada versão possui seus próprios critérios de parada, que, quando alcançados, geram a interrupção da produção de novas gerações. Em todas as versões foi utilizado um tamanho de população igual a 1.000 indivíduos. Os demais parâmetros são os mesmos usados em [Alcântara & Pereira, 2019].

As versões Gen-I e Gen-II buscam verificar se a variabilidade das gerações que estão sendo produzidas continua existindo, parando de produzir novas quando considera que a variabilidade está baixa. A Gen-III é uma combinação das anteriores, motivada pelos bons resultados gerados pelas referidas versões.

A Gen-I analisa se ocorre evolução entre as gerações e nos indivíduos de cada geração. Os critérios são: (1) os 10 melhores indivíduos da geração corrente (1% da população) são iguais entre si; e (2) os melhores indivíduos da geração corrente são iguais aos melhores indivíduos das 10 últimas gerações.

A Gen-II analisa se ocorre evolução somente entre as gerações. O critério é: os melhores indivíduos da geração atual são iguais aos melhores indivíduos das 10 últimas gerações. Aqui foram geradas 2 subversões: a subversão (A) considera os melhores 3% indivíduos da população; a subversão (B) considera 1% dos melhores indivíduos da população.

A Gen-III consiste na junção das versões Gen-I e Gen-II, onde nas 99 primeiras gerações são utilizados os critérios da Gen – I; a partir da geração 100 utiliza-se o critério da Gen – IIB.

3. Algoritmo Genético Multiobjetivo

A abordagem multiobjetivo de um problema de otimização possibilita a obtenção de um conjunto soluções que satisfaça as restrições do problema e aperfeiçoe um vetor de funções objetivo [Gaspar-Cunha, Takashi, & Antunes, 2013]. É muito popular nesse contexto o uso de Algoritmos Evolutivos, pois estes são capazes de evoluir simultaneamente todo um conjunto de diferentes indivíduos, onde cada um deles corresponde a uma possível solução para o problema.

O problema de alocação de veículos pode ser tratado como um problema multiobjetivo visando: (1) minimizar a distância média percorrida pelos veículos para realizar o atendimento, (2) minimizar a distância total percorrida pelos veículos, (3) minimizar o tempo total para atendimento e (4) minimizar o tempo médio para atendimento. Logo, o estudo do tempo de atendimento faz-se necessário, pois embora seja natural pensar que o tempo de atendimento é proporcional à distância percorrida, particularidades no trânsito de cada trajeto impedem que o pressuposto descrito seja sempre verdadeiro.

3.1. Modelagem Matemática

Matematicamente optou-se pela interpretação do problema da minimização da distância como a minimização da função:

$$z_1 = \sum_{i=1}^{m} \sum_{j=1}^{n} g_{ij} * d_{ij} \quad (1)$$

Onde ***n*** representa o número de solicitações de veículo, ou seja, o número de linhas em cada uma das matrizes de dados; ***m*** representa a quantidade de veículos que podem realizar viagens, isto é, o número de colunas em cada uma das matrizes de dados. A variável ***g_{ij}*** é binaria e sinaliza se o veículo ***i*** aceitou ou não a solicitação de viagem ***j***. Assim, $\boldsymbol{g_{ij}}$ = 1, se a viagem tiver sido aceita e $\boldsymbol{g_{ij}}$ = 0, se a viagem não tiver sido aceita. A variável $\boldsymbol{d_{ij}}$ é o valor armazenado na posição ***ij*** da matriz **D** com os valores de distância da viagem do veículo ***i*** para a solicitação ***j***.

A minimização do tempo de atendimento consiste na minimização da função:

$$z_2 = \sum_{i=1}^{m} \sum_{j=1}^{n} g_{ij} * t_{ij} \quad (2)$$

Onde a variável t_{ij} é o valor armazenado na posição ***ij*** da matriz **T** com os valores de tempo da viagem do veículo ***i*** para a solicitação ***j***.

O problema multiobjetivo é resultado da minimização da função:

$$z_3 = \sum_{i=1}^{m} \sum_{j=1}^{n} g_{ij} * (d_{ij} + t_{ij}) \qquad (3)$$

4. Simulações e Parametrização

O software utilizado para simulação do tráfego é o *Simulation of Urban MObility* (SUMO[1]), o qual possibilita simular um veículo percorrendo uma via, configuração de uma malha viária, entre outros recursos. Neste trabalho conservou-se o mapa proposto em [Alcântara & Pereira, 2019]. Os parâmetros utilizados são: 850 veículos autônomos para atender às solicitações, 18.000 veículos pessoais: carros e motocicletas, que transitam em 50.000 rotas distintas. Os algoritmos foram executados em um ASUS Z450U Intel ® Core ™ i5-7200U e 8 GB de RAM. Foram criados dois cenários distintos de simulação, os quais serão descritos a seguir.

4.1. Simulação 1

Aqui, a simulação é configurada de forma semelhante à descrita em [Oliveira et al. 2015]. Ela foi utilizada para testar os algoritmos mono-objeitvo. Seus detalhes:

- Os veículos, autônomos e pessoais, são distribuídos aleatoriamente no mapa. Durante todo o tempo de simulação, os veículos pessoais podem entrar e sair da rede, simulando o fluxo de tráfego;
- Quando um veículo autônomo se torna ocupado, um ocupado torna-se livre, mantendo assim a proporção de veículos autônomos livres e ocupados. No decorrer da simulação sempre haverá metade dos veículos desocupados e a outra metade ocupada;
- O veículo autônomo ocupado não pode ser atribuído à outra solicitação.

4.2. Simulação 2

A Simulação 2 foi aplicada ao algoritmo multiobjetivo, sendo configurada assim:

- Os veículos, autônomos e pessoais, são distribuídos aleatoriamente no mapa. Durante todo o tempo de simulação, todos veículos podem entrar e sair da rede, simulando o fluxo de tráfego e interpretando o mapa como o recorte de uma região maior;
- No decorrer na simulação a proporção entre veículos autônomos livres e ocupados pode variar, simulando a existência de horários com maior número de solicitações;
- O veículo autônomo ocupado não pode ser atribuído à outra solicitação.

5. Resultados e Discussões

5.1. Tempo de Execução dos Algoritmos Genéticos Mono-Objetivo

Foram realizadas 30 execuções para cada versão dos algoritmos, incluindo a versão Gen, apresentada em [Alcântara & Pereira, 2019]. Nessa fase do trabalho, utilizou-se a

[1] http://sumo.sourceforge.net/

simulação 1. Usando ANOVA e teste de Tunkey [Carrano et al 2011] tem-se a seguinte ordenação, do melhor para o pior: Gen-I, Gen-IIA, Gen-IIB, Gen-III e Gen. Os valores de distância foram próximos, de modo que Gen-I, Gen-IIA, Gen-IIB estão empatados e a GenIIA só é melhor que a Gen.

A Fig. 1 apresenta os resultados com relação ao tempo de execução dos algoritmos mono-objetivo. As novas versões foram comparadas com a Gen, constatando melhorias no tempo de execução bastante significativas. A versão Gen-IIA destacou-se com 90% de melhoria, seguida pela Gen-IIB com 88%, Gen-III com 81% e Gen-I com 65%.

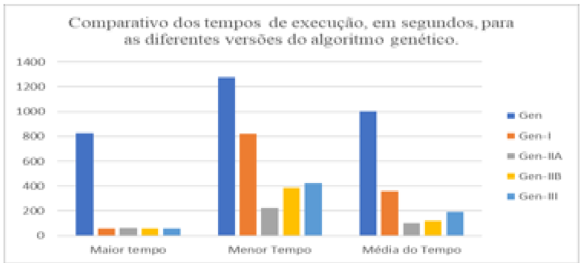


Figura. 1. Comparação dos tempos de execução das versões mono-objetivo.

5.2. Abordagem Multiobjetiva

Na abordagem multiobjetivo utilizou-se a Simulação 2, de modo que a complexidade do problema aumentou. Os valores esperados de distância de atendimento são maiores do que os esperados para a Simulação 1. Com a finalidade de comprovar a eficiência da proposta de minimização multi-objetivo, realizou-se testes de três abordagens distintas: algoritmo de minimização da distância, algoritmo de minimização do tempo de atendimento e algoritmo de minimização multiobjetivo. O algoritmo de minimização da distância usado foi proposto em [Alcântara & Pereira, 2019]; os demais foram desenvolvidos no presente estudo.

Devido ao aumento da complexidade da execução na simulação 2, utilizou-se quantidade máxima de gerações igual a 300 em todos os algoritmos. Realizou-se 30 repetições de cada algoritmo. Nas novas propostas aplicou-se os critérios de convergência estudados na versão Gen II-A.

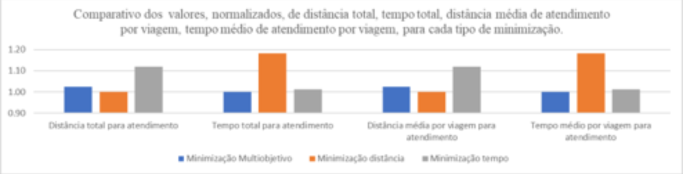


Figura 2. Gráfico comparativo dos valores, normalizados, de distância e tempo de viagem, para cada tipo de minimização.

A eficiência do algoritmo de minimização multiobjetivo pode ser observada na Fig. 2. Comparando ao algoritmo [Alcântara & Pereira, 2019], minimização distância: (1) o algoritmo mono-objetivo de minimização do tempo de atendimento oferece melhoria de 14% no tempo de atendimento; em contrapartida uma piora em 12% na distância para atendimento; (2) o algoritmo multiobjetivo oferece 15% de melhoria no tempo para atendimento, sem melhorias significativas na distância para atendimento. O algoritmo multiobjetivo se destaca, ainda, no tempo de execução, pois a heurística de parada utilizada na Gen-IIA também foi aplicada nele, desde modo não atinge o número máximo de gerações, diferente do algoritmo de minimização da distância, Gen.

6. Conclusão

O desenvolvimento de algoritmos que tenham respostas rápidas e eficientes e ainda assim retratem bem a complexa realidade do trânsito, otimizando a alocação de veículos, pode ser desafiador. Esse artigo apresentou formas alternativas de critérios de convergência em algoritmos genéticos focados na alocação de veículos autônomos. Esse tipo de método pode proporcionar redução no tempo de execução dos algoritmos. Outro ponto de contribuição consiste em um cenário mais abrangente que o apresentado em outros trabalhos: aqui é apresentado um algoritmo multiobjetivo capaz de minimizar a distância total percorrida pelos veículos e o tempo total de atendimento, consequentemente minimizando também a distância média percorrida pelos veículos para realizar o atendimento e o tempo médio para atendimento. Os resultados apresentam a minimização multiobjetivo como uma abordagem mais eficiente do que a abordagem mono-objetivo, dado o problema de alocação de veículos. Finalmente, destaca-se o estudo da evolução das gerações, inicialmente estudado no cenário mono-objetivo e posteriormente aplicada no cenário multiobjetivo, como um bom método para estabelecer critérios de convergência do algoritmo.

Agradecimentos

Os autores agradecem ao CNPq e PROPE/UFSJ pelo suporte financeiro.

Referências

Alcântara, E. B., Pereira, M. A. (2019). A Genetic Algorithm to Autonomous Vehicles Designation. In GEOINFO 2019. pp. 194-199.

Carrano, E. G., Wanner, E. F., Takahashi, R. H. (2011). A multicriteria statistical based comparison methodology for evaluating evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 15(6):848–870.

Fonseca, A. 7 empresas que estão desenvolvendo carros autônomos. (2020) Whow, 20 de março de 2020. Disponível em: < https://www.whow.com.br/novas-tecnologias/7-empresas-desenvolvendo-carros-autonomos/>. Acessado em 02/09/2020.

Goldberg, D. E., Holland, J. H. (1988). Genetic algorithms and machine learning. Machine learning, 3(2):95–99.

Gaspar - Cunha, A., Takashi, R., Antunes, C. H. (2013). Manual de computação evolutiva e metaheurística. Belo Horizonte: Editora UFMG.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2), 83-97.

Oliveira, A. A. M., Souza, M. P., Pereira, M. A., Reis, F. A. L., Almeida, P. E. M., Silva, E. J., Crepalde, D. S. (2015). Optimization of taxi cabs assignment in geographical location-based systems. In GEOINFO 2015, pp. 92-104.

Brazil Data Cube Cloud Coverage (BDC^3) Viewer

Felipe Rafael de Sá Menezes Lucena, Elton Vicente Escobar-Silva, Rennan de Freitas Bezerra Marujo, Matheus Cavassan Zaglia, Lubia Vinhas, Karine Reis Ferreira and Gilberto Ribeiro de Queiroz

Earth Observation and Geoinformatics Division, National Institute for Space Research (INPE), São José dos Campos – SP – 12227-010 – Brazil

{felipe.lucena, elton.silva, rennan.marujo, matheus.zaglia, lubia.vinhas, karine.ferreira, gilberto.queiroz}@inpe.br

Abstract. *Remotely sensed Earth Observation (EO) data have exceeded a large scale and are increasingly available for different communities and thematic applications. In this matter, EO Data Cubes (EODC) are a promising solution to efficiently manage big EO data, enabling its access, processing and analysis. As final products, EODC provides analysis-ready data (ARD) for vegetation and land use and land cover (LULC) studies, for environmental monitoring urban growth studies, among others. One of the most used pre-selection criteria for image retrieval of big EO data is the average cloud cover statistic. This paper describes our initiative in the implementation of an on-demand view-based tool for cloud coverage embedded on the Brazil Data Cube (BDC) project based on the SpatioTemporal Asset Catalog (STAC).*

1. Introduction

In the last decade, remotely sensed Earth Observation (EO) data have exceeded the petabyte-scale milestone [Giuliani et al. 2019] and are increasingly available in diverse free and open access repositories [Giuliani et al. 2017]. This highlighted significant issues regarding storage, organization, management, and EO data analysis due to its volume, velocity, and variety [Giuliani et al. 2019]. However, to exploit the potential benefits of big EO data users are often required to have a high level of expertise and are frequently hampered by engaging in exhausting and tedious processes to discover data, and extract information and knowledge [Plag & Jules-Plag 2019].

Owing to the limitations of traditional acquisition, management, distribution, and analysis approaches of the big amount of satellite EO data available, exploitation of the full information potential of big EO data has not been achieved so far [Giuliani et al. 2019] and is at a lower level than desirable and feasible [Plag & Jules-Plag 2019]. Data size, heterogeneity, and complexity are the most notably limitations [Giuliani et al. 2019]. Therefore, new solutions are dearly needed to fulfill current analysis demands in this fast-changing field.

In this context, EO Data Cubes (EODC) are a promising solution to manage efficiently and effectively big EO data from different data repositories [Baumann et al. 2019; Nativi et al. 2017; Giuliani et al. 2017]. EODC are aimed to be a solution to store, organize, manage, and analyze large amounts of multi-sensor EO data [Poussin et al. 2019], as well as to provide access to EO Analysis-Ready Data (ARD) [Dwyer et al. 2018]. In other words, they are designed to harness big EO data, facilitating the access and use of ARD for immediate analysis in applications and for time-series exploitation

[Poussin et al. 2019]. As final products, EODC provide ARD for vegetation, land use and land cover (LULC), environmental monitoring and urban growth studies, among others [Ferreira et al. 2020; Giuliani et al. 2019; Poussin et al. 2019; Dhu et al. 2019].

One of the most used pre-selection criteria for image retrieval of big EO data is the average cloud cover statistic [Augustin et al. 2019]. This is because, for remote sensing of the earth's surface via optical sensors, some elements such as clouds act as no interest targets, which usually make the surface absent in some portion of the images [Zhong et al. 2017].

In this context, this paper describes our ongoing work to implement a novel viewer tool for cloud coverage embedded on the Brazil Data Cube (BDC) project, the Brazil Data Cube Cloud Coverage (BDC^3) viewer. Descriptions of the BDC project and BDC^3 viewer tool are presented in the following sections.

2. Brazil Data Cube (BDC) project

Since 2019, the Brazilian National Institute for Space Research (INPE) has been working in the BDC project (see http://brazildatacube.org/), which aims (i) to process remote sensing images of medium spatial resolution (10 to 30 meters) of the entire Brazilian territory into ARD datasets and assembling them as multidimensional cubes with at least three dimensions (space, time and spectral properties) [Ferreira et al. 2020; Picoli et al. 2020]; (ii) to propose and develop innovative methods and techniques to store, process and analyze EO data cubes using satellite image time series analysis (TSA), image processing and machine learning methods; (iii) to use the data cubes and methods to improve the generation of land use and cover change (LUCC) information for Brazil [Ferreira et al. 2020].

According to the Committee on Earth Observation Satellites (CEOS), ARD are "satellite data that have been processed to a minimum set of requirements and organized into a form that allows immediate analysis with a minimum of additional user effort and interoperability both through time and with other datasets" [Killough 2016]. In other words, ARD products can be described as data that have been processed to minimize the time and scientific knowledge required of users in such a way that allows analysis with a minimum of additional user effort [Dwyer et al. 2018].

The BDC project is creating ARD image collections and multidimensional data cubes from medium spatial resolution images of the sensors OLI/Landsat-8, MSI/Sentinel2 (A and B), and WFI/CBERS-4 [Picoli et al. 2020; Ferreira et al. 2020]. The latest is the fourth China-Brazil Earth Resources Satellite (CBERS-4) with the Wide-field Image camera (WFI), which produces medium resolution images in both visible and infrared wavelengths of the electromagnetic spectrum [Picoli et al. 2020]. Generally, in a data cube (DC) generation process, data acquisition and preprocessing are the first steps required. In BDC, all available images from the three repositories mentioned above are queried from their providers. To do so, a grid is used to search all available images of a specific grid cell on a given image collection. Then, these images are merged, reprojected, resampled, and gridded to a common spatial reference. Finally, a temporal compositing function is used to build regular intervals (e.g. 16 days or monthly, for instance) and reduce the data dimensionality according to a composition function (such as median or best quality pixel) (Figure 1) [Ferreira et al. 2020].

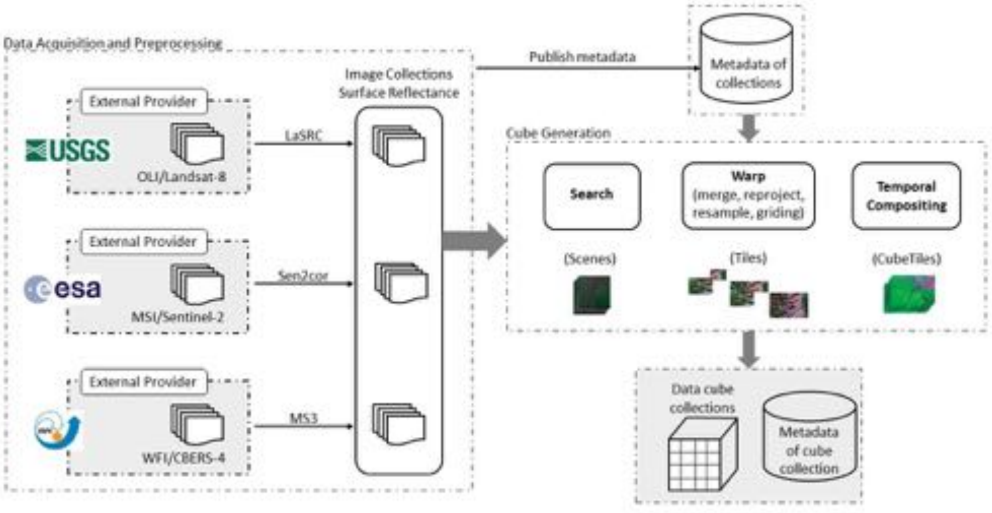


Figure 1: A summary of Brazil Data Cube (BDC) generation process. Adapted from Ferreira et al. (2020).

After the acquisition and preprocessing of OLI/Landsat-8, MSI/Sentinel (2A and 2B) and WFI/CBERS-4, their metadata are stored in an internal database catalog, then processed locally to generate the surface reflectance products through LaSRC [Vermote et al. 2016] and Sen2cor [Louis et al. 2016] atmospheric correction for OLI/Landsat-8 and MSI/Sentinel, respectively. To result in surface reflectance products, visibility masks are generated for each scene and data on the percentage of cloud coverage present in the image is cataloged in the image metadata.

3. Brazil Data Cube Cloud Coverage (BDC^3) Viewer

The BDC³ viewer is a tool to graphically visualize information about cloud cover in EODC, based on the SpatioTemporal Asset Catalog (STAC) specification (see https://github.com/radiantearth/stac-spec). The BDC³ viewer is being implemented in the BDC project using the BDC–STAC service. The BDC-STAC service implements the STAC specification, which defines how metadata of geospatial data are organized, consulted, and made available to users [Zaglia et al. 2019]. Therefore, the service aims cataloging and providing access to BDC metadata. A brief description of the BDC-STAC and BDC³ architecture is shown in Figure 2.

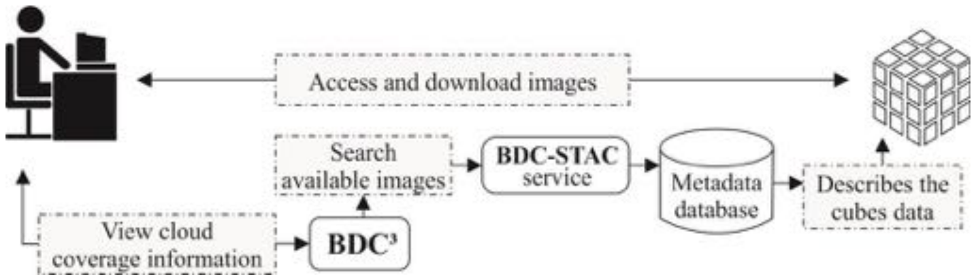


Figure 2: A description of BDC³ view-based tool and BDC-STAC service architecture. Adapted from Zaglia et al. (2019).

The BDC³ viewer proposes an extension of BDC-STAC's query possibilities for acquisition and visualization of cloud coverage information and will be able to answer questions such as *"Which data in my area of interest have less than 10% cloud cover?"* and "*In which year was the highest cloud coverage in December?"*. The tool allows user interaction, and all queries can be made per tile or polygon and are min/max coverage threshold-based. The BDC³ viewer provides four methods to get information about cloud cover from EO data cubes:

(i) seasonal (e.g. monthly) cloud coverage average;

(ii) total annual cloud coverage;

(iii) scene, area or period with maximum or minimum cloud coverage;

(iv) cloud coverage timeseries.

The seasonal average corresponds to the temporal grouping (average) of values for one or more tiles. For example, the monthly average represents the average between the cloud coverage values for each month. The total annual cloud coverage returns all scene coverage values for the queried period in a scatter plot. The selection of maximum and minimum queries the scene, area or period of interest according to the parameters defined by the user. Finally, the cloud coverage time-series query allows the display of coverage values for a given tile within the selected period.

The main idea of the tool is basically to show information about cloud cover in EODC as graphs and tables through an interactive web interface according to the defined parameters. Besides the visual information, the grouped data that generate graphics will be available for download to users. Two examples of possible queries to the metadata database through BDC³ viewer are shown as visual information in Figures 3 and 4. As a result, the users will be able to view information and acquire data on cloud coverage for a spatial extent of interest directly from the web without having to get images files or run complex algorithms.

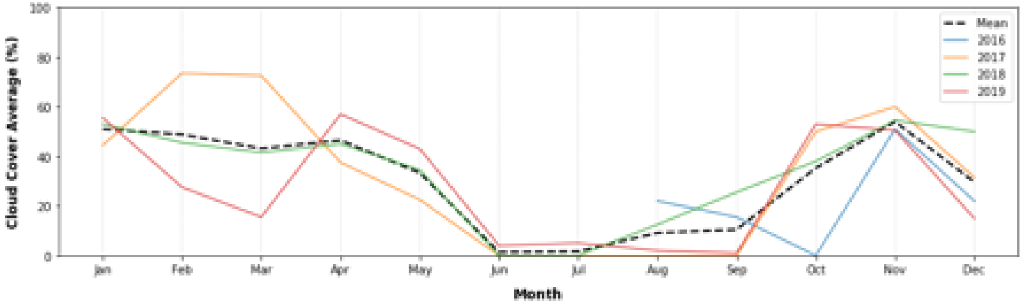


Figure 3: An example of a query-based on the monthly cloud coverage on BDC³. Just for illustration, it was chosen the cloud coverage values of the Landsat Path/Row 222068 from 2016 to 2019.

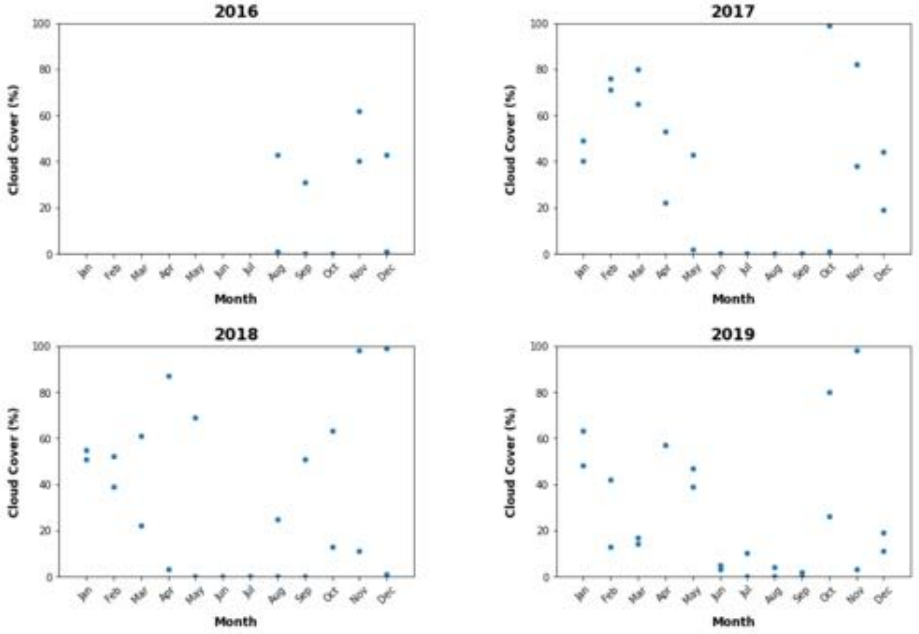


Figure 4: An example of a query-based on the cloud coverage absolute values by month on BDC³. Just for illustration, it was the chosen cloud coverage values of the Landsat Path/Row 222068 from 2016 to 2019.

4. Final considerations

As a work in progress, this paper aims to present the initial stages of the Brazil Data Cube Cloud Coverage (BDC^3) viewer, a novel view-based tool for cloud coverage embedded on the BDC project. BDC^3 viewer has a diversity of applications and its potential uses are the aid in scenes selection for agricultural monitoring, support in sensor selection for study in an area of interest, and the availability evaluation of scenes with clean pixels for a time interval, among others. As the interactive tool is already designed, the next steps are (i) complete the script and (ii) integrate the tool on the BDC as interactive maps.

5. References

AUGUSTIN, Hannah; SUDMANNS, Martin; TIEDE, Dirk; LANG, Stefan; BARALDI, Andrea. (2019). Semantic Earth observation data cubes. Data, 4(3), 102.

BAUMANN, Peter; MISEV, Dimitar; MERTICARIU, Vlad; HUU, Bang P. (2019). Datacubes: Towards space/time analysis-ready data. In Service-oriented mapping (pp. 269-299). Springer, Cham.

DHU, Trevor; GIULIANI, Gregory; JUÁREZ, Jimena; KAVVADA, Argyro; KILLOUGH, Brian; MERODIO, Paloma; MINCHIN, Stuart; RAMAGE, Steven. (2019). National open data cubes and their contribution to country-level development policies and practices. Data, 4(4), 144.

DWYER, John L.; ROY, David P.; SAUER, Brian; JENKERSON, Calli B.; ZHANG, Hankui K.; LYMBURNER, Leo. (2018). Analysis ready data: enabling analysis of the Landsat archive. Remote Sensing, 10(9), 1363.

FERREIRA, Karine. R.; QUEIROZ, Gilberto R.; CAMARA, Gilberto; SOUZA, Ricardo C. M.; VINHAS, Lúbia; MARUJO, Rennan F. B.; SIMOES, Rolf E. O.; NORONHA, Carlos A. F.; COSTA, Raphael W.; ARCANJO, Jeferson S.; GOMES, Vitor C. F; ZAGLIA, Matheus C. (2020). Using Remote Sensing Images and Cloud Services on Aws to Improve Land Use and Cover Monitoring. In 2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS) (pp. 558-562). IEEE.

GIULIANI, Gregory; CHATENOUX, Bruno; BONO, Andrea D.; RODILA, Denisa; RICHARD, Jean-Philippe; ALLENBACH, Karin; DAO, Hy; PEDUZZI, Pascal. (2017). Building an earth observations data cube: lessons learned from the Swiss Data Cube (SDC) on generating analysis ready data (ARD). Big Earth Data, 1(1-2), pp. 100–117.

GIULIANI, Gregory; CAMARA, Gilberto; KILLOUGH, Brian; MINCHIN, Stuart. (2019). Earth observation open science: Enhancing reproducible science using data cubes.

KILLOUGH, Brian. (2016). CEOS land surface imaging analysis ready data (ARD) description document.

NATIVI, Stefano; MAZZETTI, Paolo; CRAGLIA, Max. (2017). A view-based model of data-cube to support big earth data systems interoperability. Big Earth Data, 1(1-2), 75-99.

PICOLI, Michelle C. A.; SIMOES, Rolf; CHAVES, Michel; SANTOS, Lorena A.; SANCHEZ, Alber; SOARES, Anderson; SANCHES, Ieda D.; FERREIRA, Karine R.; QUEIROZ, Gilberto R. (2020). CBERS DATA CUBE: A POWERFUL TECHNOLOGY FOR MAPPING AND MONITORING BRAZILIAN BIOMES. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 5(3).

PLAG, Hans-Peter; JULES-PLAG, Shelley A. (2019). A Transformative Concept: From Data Being Passive Objects to Data Being Active Subjects. Data, 4(4), 135.

POUSSIN, Chartlotte; GUIGOZ, Yaniss; PALAZZI, Elisa; TERZAGO, Silvia; CHATENOUX, Bruni; GIULIANI, Gregory. (2019). Snow Cover Evolution in the Gran Paradiso National Park, Italian Alps, Using the Earth Observation Data Cube. Data, 4(4), 138.

ZAGLIA, Matheus C.; VINHAS, Lubia; QUEIROZ, Gilberto R. de; SIMOES, Rolf. (2019). Catalogação de Metadados do Cubo de Dados do Brasil com o SpatioTemporal Asset Catalog. GEOINFO, 20 Years After!, 280.

ZHONG, Bo; CHEN, Wuhan; WU, Shanlong; HU, Longfei; LUO, Xiaobo; IU, Qinhuo. (2017). A Cloud Detection Method Based on Relationship Between Objects of Cloud and Cloud-Shadow for Chinese Moderate to High Resolution Satellite Imagery. In IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (p. 4898-4908). IEEE.

Dinâmica Espacial Urbana na Amazônia: Modelo de Autômatos Celulares na Simulação da Expansão Urbana no Município de Mocajuba - Pará

Renata Maciel Ribeiro[1]; Leonardo R. Queiroz[1]; Luigi M. Ribeiro[2]; Pedro R. Andrade[1]; Silvana Amaral[1].

Instituto Nacional de Pesquisas Espaciais – INPE[1]
Caixa Postal 515, 12227-010, São José dos Campos, SP – Brasil

Universidade Estadual do Rio de Janeiro – UERJ[2]
R. São Francisco Xavier - 524, 20550-900, Rio de Janeiro, RJ - Brasil

{renata.ribeiro; leonardo.queiroz; pedro.andrade; silvana.amaral}@inpe.br e luigimacielribeiro@gmail.com

Abstract: *Computational models that represent the process of genesis and evolution of cities are simplifications of reality. As the same time, they allow to understand the relationships of these complex systems and enable to make inferences about future dynamics. This work is an exploratory analysis that uses a methodological approach of cellular automata to analyze the dynamics of urban expansion. The results show that the theoretical-methodological approach is effective to model urban expansion, reproducing in a simplified way the patterns of territorial growth of an Amazonian city.*

Resumo: *Modelos computacionais que representam o processo de gênese e evolução das cidades são simplificações da realidade. Ao mesmo tempo, estes modelos permitem entender as relações desses sistemas complexos e possibilitam fazer inferências sobre dinâmicas futuras. Este trabalho é uma análise exploratória que utiliza uma abordagem metodológica de autômatos celulares para análise da dinâmica de expansão urbana. Os resultados mostram que a abordagem teórico-metodológica é eficaz para modelar a expansão urbana, reproduzindo de forma simplificada os padrões de crescimento territorial de uma cidade amazônica.*

1. Introdução

As cidades são organismos complexos, onde diversos agentes e fatores atuam em seu processo de evolução. Os modelos computacionais que representam o processo de gênese e evolução destes espaços são simplificações da realidade. No caso da expansão urbana, a simples decomposição estática de suas partes não permite a compreensão deste fenômeno multidimensional. Com isso, a utilização de modelagem para análise de fenômenos urbanos através das inter-relações de seus diferentes elementos, possibilita entender as relações em sistemas complexos, aportando ferramentas para explorar possibilidades relativas a mudanças futuras [Allen 1997, Batty e Torrens 2005].

Na Amazônia, a dinâmica de evolução das cidades responde à agentes e fatores muito particulares à história de ocupação do território e expressam o efeito da relação entre atores que interagem sobre o mesmo espaço, mas representam diferentes temporalidades. Propondo uma análise exploratória para testar parte da abordagem metodológica descrita por Barredo et al. (2003), no contexto amazônico, este trabalho tem o objetivo de explorar um modelo de autômatos celulares para estudar a dinâmica de expansão urbana da sede do município de Mocajuba, no estado do Pará. Tendo como referência a proposta de gradiente de urbano de Dal'Asta et al. (2016), a evolução urbana do município foi modelada, a partir de dados de mapeamentos de uso e cobertura da terra do Terraclass (ALMEIDA et al, 2016) reamostrados em uma grade celular de 200m², para período de 2004 a 2014, de modo a estabelecer um arcabouço teórico-conceitual para as regras de transição de estados celular.

2. Metodologia

Um modelo de autômatos celulares (AC) é caracterizado por algumas propriedades fundamentais [Weimar 1998]: (i) consistem em uma matriz, ou grade de células; (ii) a evolução do modelo se dá em passos discretos de tempo; (iii) cada célula possui um estado pertencente a um conjunto finito de estados; (iv) evolui de acordo com regras que dependem do estado em que a célula se encontra e do estado de seus vizinhos; e (v) a relação com a vizinhança é local e uniforme. Para além destas características gerais, deve-se ter um modelo teórico-metodológico subjacente que justifique as escolhas para o processo a ser estudado, neste caso, a evolução urbana de uma cidade amazônica.

No escopo metodológico desta pesquisa, adotam-se dois diferentes trabalhos como base teórico-metodológica. Para o norteamento metodológico, propõe-se a aplicação da metodologia de análise da dinâmica urbana a partir de autômatos celulares (AC) testada por Barredo et al. (2003). Para a fundamentação teórico-conceitual, adotou-se a representação espaço temporal do fenômeno urbano na Amazônia de Dal'Asta et al. (2016), que considera o urbano como resultado de um processo *continuum*.

2.1. Fundamentação Teórico-Metodológica

O modelo de Barredo et al. (2003) descreve a dinâmica urbana de Dublin entre 1968 e 1998, considerando cinco vetores para composição celular: (i) *adequação*, definido pela soma linear de fatores físicos, ambientais e institucionais agrupados em um coeficiente entre 0 e 1 que caracterizam a capacidade de transição entre usos da terra; (ii) *acessibilidade*, que é um coeficiente estabelecido em função da distância das redes de transportes públicos; (iii) *status de zoneamento*, definido como o estado inicial da célula pré-definido no modelo; (iv) *influência da vizinhança*, que é a relação da célula central com seus vizinhos; e (v) *perturbações estocásticas*, que é o componente de randomização no modelo, o que faz com que as simulações possuam grande similaridade entre si, mas não sejam exatamente iguais.

O caminho de representação da mudança do paradigma urbano sugerido por Dal'Asta et al. (2016) apoia-se na ideia de sistemas de objetos e sistemas de valores [Lefebvre 1999], intermediados por um sistema de ações [Santos 2006]. Considerando ainda a urbanização como um fenômeno contínuo, que propaga sua lógica de produção, consumo e modo de vida por todo território, em diferentes intensidades. Essa matriz conceitual, se apoia na existência de lógicas produtivas associadas às classes de uso e cobertura que podem identificar diferentes intensidades de urbano.

2.2. Procedimentos Metodológicos

Ao assumir como base teórico-conceitual o gradiente de urbano de Dal'Asta et al. (2016), a classe "não observada" é desconsiderada, não sendo possível calcular a probabilidade de transição desta classe para "urbano". Ainda, visto que, provavelmente devido à presença de nuvens nas imagens, a área não observada contígua à mancha urbana em 2004 é predominantemente classificada como "vegetação secundária" em 2014, a classe foi agrupada à "vegetação secundária". Assim, define-se o sistema de objetos como: (i) Floresta, (ii) Vegetação Secundária, (iii) Pastagem, (iv) Mosaico de ocupações e (v) Urbano. Apesar de não se enquadrar no modelo de transformação urbana, a classe Rio também foi considerada nos estados celulares para composição do mapa final de uso e cobertura da terra.

Para a adaptação das classes de entrada (TerraClass) às classes do gradiente de Dal'Asta et al. (2016), quando as classes observadas correspondem a mais de uma classe no modelo de referência, o valor da classe modificada é obtido pela média das classes que a compõem (Figura 1), entendendo que a tendência de ação humana é menor quanto menor for o coeficiente. Sendo assim, para: (i) Urbano – equivalente às três classes de maior intensidade: Unidade Espacial de Ocupação Humana (UEOH) (8), núcleo urbano (9) e cidade (10), a média igual a 9; (ii) Pastagem, definida pelas classes: pasto sujo (4) e pasto limpo (5), a média igual a 4,5; (iii) Floresta – corresponde ao valor da floresta: 1; (iv) Vegetação Secundária – corresponde ao valor da vegetação secundária: 2; e (v) Mosaico – como área de transição, que pode ser tanto entre floresta e pasto, quanto entre pasto e cidade, abrange diferentes níveis de tecnificação do território. Nesse sentido, definiu-se como mosaico todas as outras classes do gradiente proposto, ou seja, não floresta (3), agricultura de pequena escala (6) e agricultura anual (7), que computaram média igual a 5,3.

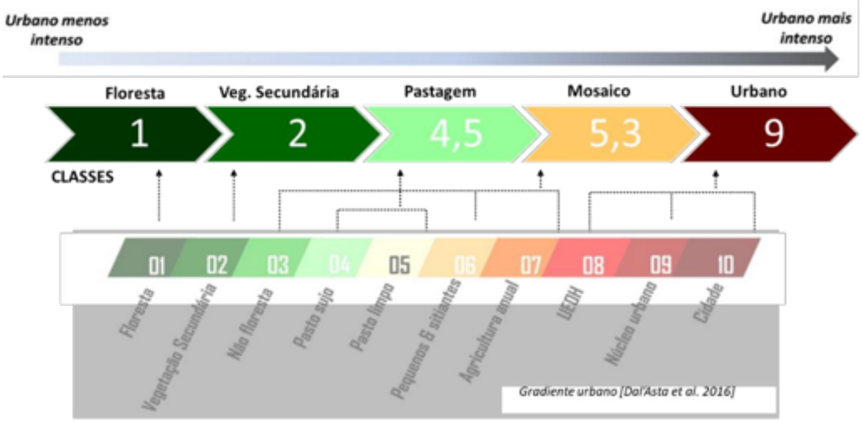


Figura 1 - Sistema de Objetos - Classes adaptadas a partir do gradiente urbano [Dal'Asta et al. 2016].

Partindo dos valores calculados para as classes modificadas, foi possível definir uma especificação matemática para cálculo da tendência de transição entre classes de acordo com a distância entre elas no gradiente (Figura 1). Para se enquadrar em um modelo normalizado de 0 a 1, os coeficientes são expressos pela Equação 1.

$$c = d1/n \qquad \text{(Equação 1)}$$

Onde, *c* é o coeficiente que representa numericamente a tendência de transição entre as classes, *d1* é a classe numericamente menor e *n* o número de classes do modelo. Para calcular a tendência de mudança, usamos uma variante normalizada da lei do inverso do quadrado da distância, utilizada para calcular distâncias em modelos geoestatísticos por interpolação multivariada [Lukaszyk 2004]. Partindo desse conceito, definimos uma fórmula em que *K* é a probabilidade efetiva de mudança entre classes, *c* é o coeficiente de variação de cada classe, *d1* é a classe numericamente menor, denotando menor ação humana, e *d2* é a classe numericamente superior (Equação 2).

$$K = c^{*}\{1/[1+ (d2 - d1)^2]\} \qquad \text{(Equação 2)}$$

Estabelecida a matriz de probabilidade que define a atuação dos vetores de *adequação* e *status de zoneamento*, os outros vetores são descritos diretamente no script (disponível em https://www.lissinpe.com.br/códigos), inserindo-se um efeito randômico para o vetor de *perturbações estocásticas* e a relação de vizinhança. O modelo foi implementado em linguagem Lua no ambiente de programação TerraME. A partir dos valores obtidos pela Equação 2, montou-se a matriz de probabilidade do modelo (Tabela 1), que descreve a probabilidade de transição entre as classes. Destacando que como o interesse do trabalho é estudar a expansão urbana, apenas foram exploradas as transições para a classe Urbano.

A área de estudo localiza-se no município de Mocajuba (Figura 2), região do Baixo Tocantins no estado do Pará, cuja dinâmica de ocupação territorial e atividades econômicas são associadas ao rio Tocantins.

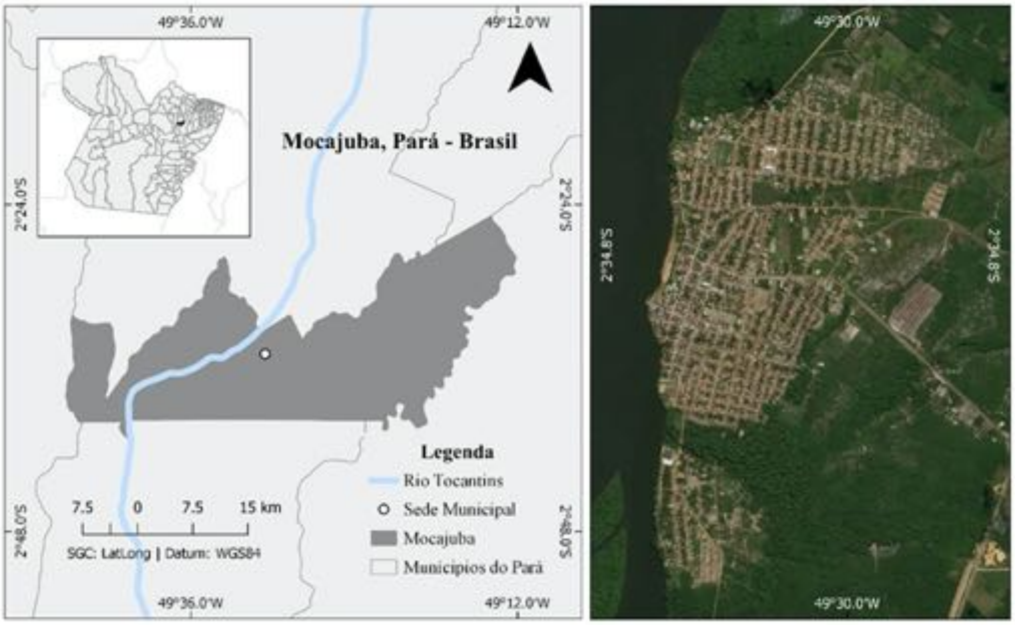


Figura 2 - Localização da Área de Estudo – Mocajuba, Pará.

Como não existem núcleos urbanos afastados da sede de Mocajuba, o crescimento da mancha urbana foi considerado contíguo. Por isso, adotou-se a estratégia de vizinhança de “Moore”. Para a regra de transição de estado, estabeleceu-se que para cada célula que possua um vizinho “urbano”, quando a probabilidade randômica (p) for menor que a probabilidade de transição estabelecida na matriz de probabilidade (I), o estado da célula muda para “urbano”.

O mapeamento de uso e cobertura da terra qualifica o urbano em uma classe única, sendo, portanto, desnecessário um indicador sintético de *adequação*, sendo aqui representado pela capacidade de cada célula de transição para a classe “urbano”. Além disso, Mocajuba, como a maioria das cidades amazônicas, não tem rede de transporte público consolidada e, portanto, o vetor de *acessibilidade* foi desconsiderado no modelo adaptado. Com isso, baseado em Barredo et al. (2003), a especificação do modelo é descrita por:

$${}^{t}piK = f\,({}^{t}SiK,\ {}^{t}ZiK,\ {}^{t}NiK,\ v) \qquad \text{(Equação 3)}$$

Onde, (*p*) é a probabilidade de um lugar (*i*), em uma cidade, ser ocupado por um uso (*K*) em um determinado tempo (*t*). Essa função é definida pelos fatores *adequação* (*S*), *status de zoneamento* (*Z*), *influência da vizinhança* (*N*) e *perturbações estocásticas* (*v*). O modelo foi aplicado para o intervalo temporal de 10 anos (2004 a 2014), estabelecendo o passo de tempo mensal (120 meses).

Para a validação do modelo, foi calculada a acurácia e precisão média de 30 simulações. E para a simulação que obteve melhor acurácia, calculou-se a matriz de confusão, com análise de erros de omissão e inclusão da classificação de “urbano” (classe Urbano) e “não urbano” (demais classes).

3. Resultados e Discussões

Com base na especificação matemática descrita (Equação 2), estabeleceu-se as probabilidades de transição entre classes (Tabela 1). A Figura 3a mostra o resultado da simulação que apresentou melhor acurácia (91%) ao comparada ao dado real do mapeamento TerraClass (Figura 3b).

Tabela 1 - Probabilidade de transição entre classes de uso e cobertura, com destaque para Urbano, classe de interesse para análise da dinâmica de expansão urbana.

De/Para	**Floresta**	**Veg. Sec.**	**Pastagem**	**Mosaico**	**Urbano**
Floresta	-	0,0500	0,0049	0,0035	**0,0012**
Veg. Sec.	0,1000	-	0,0163	0,0106	**0,0031**
Pastagem	-	-	-	0,1339	**0,0149**
Mosaico	-	-	0,4379	-	**0,0369**
Urbano	-	-	-	-	-

A acurácia, dada pela porcentagem de células classificadas corretamente em relação ao total de células, foi de 89% para a média das 30 simulações; e a precisão, métrica que mostra a relação entre as células classificadas como urbanas que de fato são urbanas em relação ao total de células urbanas, foi de 0.93 para a média das 30 simulações. Desta avaliação pode-se afirmar que os valores de acurácia e precisão média das simulações indicam que o modelo é estatisticamente válido.

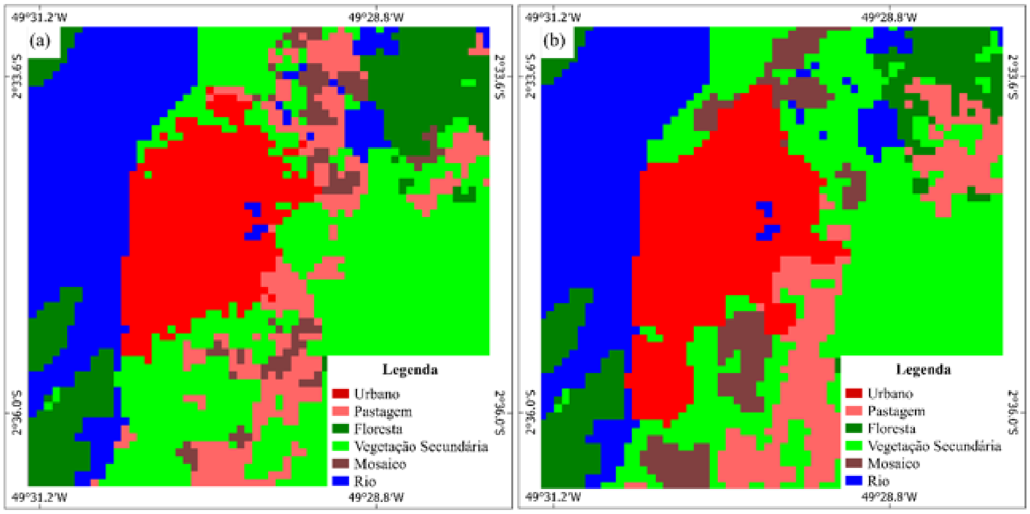


Figura 3 – Mapas de uso e cobertura da terra resultante da simulação (a) e o mapa TerraClass de referência (b) para o ano de 2014.

Além disso, a matriz de erros (Tabela 2) mostrou que não houve significativa confusão entre as classes "urbano" e "não urbano". O índice de exatidão global foi de 91,5%, o que, isoladamente, significa um excelente resultado da simulação. No entanto, para melhor conferir significado às inferências, os erros de omissão e inclusão também foram avaliados. Neste caso, a classe "urbano" apresentou 11,2% de erro de inclusão, significando que 7.473 células foram classificadas como "urbano", mas não eram "urbano". Os erros de omissão foram de 17,1%, significando que 12.232 células que deveriam ser classificadas como "urbano", não foram. A classe "não urbano" obteve resultados melhores do que a classe "urbano" por ser um compilado das outras 4 classes consideradas no modelo. De modo geral, conclui-se que a simulação da expansão urbana obteve bons resultados.

Tabela 2 – Matriz de erros (nº de células) entre o dado de referência e a simulação que obteve, sob o critério de acurácia, o melhor resultado no modelo.

		Simulação			
		NÃO URBANO	**URBANO**	**TOTAL**	**Omissão (%)**
Referência	**NÃO URBANO**	154070	7473	161543	4,6
	URBANO	12232	59514	71746	17,1
	TOTAL	166302	66987	Exatidão global: 91,5%	
	Inclusão (%)	7,3	11,2		

4. Considerações Finais

Apesar do caráter generalista e exploratório do modelo, os resultados observados demonstraram que os protótipos de AC para análise da dinâmica urbana ofereceram resultados satisfatórios. O arcabouço metodológico adotado [Barredo et al. 2003], com as devidas adaptações necessárias, foi capaz de mostrar os padrões de crescimento urbano no município de Mocajuba no Pará. Além disso, foi possível verificar que a proposta de Dal'Asta et. al. (2016) pode ser utilizada como base teórico-conceitual, especialmente no contexto amazônico, para calcular as variações de uso da terra partindo da ideia de gradiente de urbano.

Deste modo, conclui-se que a abordagem metodológica se mostrou uma ferramenta eficiente para reproduzir dinâmicas espaciais complexas, uma vez que permitiu fazer a análise a partir de limites geográficos reais e dados reais em um contexto de urbano diferente do modelo original [Barredo et al. 2003]. Destaca-se, porém, que o modelo necessita ainda de ajustes, pois sendo uma simplificação da realidade, não se adere automaticamente às especificidades locais, que demandariam incorporar as particularidades do processo de expansão urbana desta região Amazônica.

4. Referências Bibliográficas

ALLEN, P. M. Cities and regions as self-organizing systems: models of complexity. Amsterdam, Netherlands: Taylor & Francis, 1997 (**Environmental problems & social dynamics series**).

ALMEIDA, C. A. et al. High spatial resolution land use and land cover mapping of the Brazilian Legal Amazon in 2008 using Landsat-5/TM and MODIS data. **Acta Amazonica**, v. 46, n. 3, p. 291–302, 2016.

BARREDO et al. (2003) Modelling dynamic spatial processes: simulation of urban future scenarios through cellular automata. **Landscape and Urban Planning** Volume 64(3)145-160.

BATTY. M. Cities and complexity: understanding cities with cellular automata, agent-based models and fractals. Cambridge, MA: **The MIT Press**, 2005.

BATTY. M; TORRENS, P. M. Modelling and prediction in a complex world. **Futures**, v. 37, n. 7, p. 745-766, 2005.

DAL'ASTA, A. P.; AMARAL, S.; MONTEIRO, A. M. V. Um modelo para a representação espaço –temporal do fenômeno urbano na Amazônia Contemporânea. **Revista Políticas Públicas & Cidades**, v. 5, n. 2, p. 17 – 37, ago./dez. 2016.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA - IBGE. **Censo demográfico 2010**. 2010. Disponível em < http://censo2010.ibge.gov.br/>. Acesso em: 02 set. 2019.

LEFEBVRE, H., A revolução urbana. Belo Horizonte: UFMG, 1999, p.14-32

ŁUKASZYK, Szymon. A new concept of probability metric and its applications in approximation of scattered data sets. **Computational Mechanics**, v. 33, n. 4, p. 299-304, 2004.

SANTOS, M. **A Natureza do Espaço: Técnica e Tempo**. Razão e Emoção. 4. ed. São Paulo: Editora da Universidade de São Paulo, 2006.

WEIMAR, J. R. **Simulation with Cellular Automata**. Berlin: Logos Verlag Berlin, 1998. 208p.

Projeto ForestEyes: Uma proposta para aliar Ciência Cidadã e Aprendizado de Máquina para monitoramento de desmatamento

Fernanda B. J. R. Dallaqua[1], **Álvaro L. Fazenda**[1], **Fabio A. Faria**[1]

[1]Instituto de Ciência e Tecnologia – Universidade Federal de São Paulo (UNIFESP)
Avenida Cesare Mansueto Giulio Lattes, nº 1201 – Eugênio de Mello – SP – Brazil

{fernanda.dallaqua,alvaro.fazenda,ffaria}@unifesp.br

Abstract. *This paper presents the ForestEyes project methodology, a Citizen Science project in which volunteers analyze and classify segments of remote sensing images. These classifications are used as training set of classifiers, which will then label new remote sensing images to monitor deforestation. The goal is that, with improvement, the project will be able to generate reliable data, being used in areas where there is a deficit of monitoring programs.*

Resumo. *Este artigo apresenta a metodologia do projeto ForestEyes, um projeto de Ciência Cidadã em que voluntários analisam e classificam segmentos de imagens de sensoriamento remoto. Essas classificações são utilizadas no treinamento de algoritmos classificadores, que depois classificarão novas imagens de sensoriamento remoto, a fim de monitorar desmatamento. A ambição é que, com aprimoramento, o projeto consiga gerar dados confiáveis e que possa ser utilizado em áreas onde haja deficit de programas de monitoramento.*

1. Introdução

As florestas tropicais desempenham um importante papel no ecossistema global, uma vez que abrigam mais da metade das espécies do planeta, retêm bilhões de toneladas de carbono, promovem a formação de nuvens e chuvas, e são o lar de inúmeros povos indígenas [Martin 2015].

Infelizmente, milhões de hectares de florestas tropicais são perdidos todo ano, seja por desmatamento ou degradação, fazendo-se necessários programas de monitoramento e detecção de desmatamento, além de políticas públicas para a prevenção e punição. Esses programas, que podem ser governamentais ou de institutos sem fins lucrativos, utilizam de imagens de sensoriamento remoto, processamento de imagens, técnicas de Aprendizado de Máquina e fotointerpretação de especialistas para analisar, identificar e quantificar mudanças na cobertura florestal [Luz et al. 2014].

Um grande desafio para as tecnologias da informação e comunicação (TIC) é a escassez de mão-de-obra especializada e/ou a grande quantidade de dados a serem analisados, tornando o processo custoso [Soares et al. 2010]. Uma possível solução é utilizar voluntários não especializados na coleta, análise e classificação de dados para resolverem problemas técnicos e científicos, o que é conhecido como Ciência Cidadã. É uma área que têm atraído bastante atenção devido a grande quantidade de dados que são gerados, que tendem a ser de boa qualidade e são obtidos a baixo custo [Grey 2009].

A Ciência Cidadã pode ser uma valiosa fonte de dados para a área de Observação da Terra, o que inclui o monitoramento do desmatamento [Fritz et al. 2017]. Para os projetos de Ciência Cidadã *ForestWatchers* [Luz et al. 2014], *EarthWatchers* [Schepaschenko et al. 2019] e *Geo-Wiki* [Fritz et al. 2012], voluntários analisam e classificam imagens de sensoriamento remoto. Essas classificações são utilizadas para a geração de mapas ou alertas de desmatamento. Já o projeto *Forest Watcher* utiliza a coleta de dados de voluntários *in situ* para confirmar alertas de desmatamento emitidos pelo *Global Forest Watch* [Petersen et al. 2017].

As classificações vindas dos voluntários também poderiam ser utilizadas como conjuntos de treinamento de técnicas de Aprendizado de Máquina. Nesse contexto, em abril/2019 foi lançado o projeto *ForestEyes* [Dallaqua et al. 2019], hospedado na conhecida plataforma de Ciência Cidadã, *Zooniverse* [Smith et al. 2013]. Os voluntários classificam segmentos de imagens de sensoriamento remoto e essas contribuições são utilizadas para criar conjuntos de treinamento de algoritmos classificadores, que serão utilizados para monitorar desmatamento em novas imagens de sensoriamento remoto.

Trabalhos preliminares [Dallaqua et al. 2019, Dallaqua et al. 2020] foram feitos para atestar a qualidade das contribuições dos voluntários e a viabilidade de utilizá-las como conjunto de treinamento em Aprendizado de Máquina. No entanto, ainda não existia uma formalização da metodologia do projeto. Este artigo apresenta essa metodologia, que pode ser dividida em 5 módulos.

A organização deste trabalho é a seguinte: a Seção 2 discorre sobre a metodologia do projeto *ForestEyes*; a Seção 3 apresenta um breve resumo dos resultados preliminares já publicados, atestando a viabilidade do projeto; e a Seção 4 conclui este artigo.

2. Metodologia do projeto ForestEyes

A metodologia do projeto ForestEyes pode ser dividida em cinco módulos principais: Módulo de Pré-processamento, Módulo de Ciência Cidadã, Módulo de Organização e Seleção, Módulo de Aprendizado de Máquina e Módulo de Pós-processamento. A Figura 1 apresenta o esquema simplificado dessa metodologia.

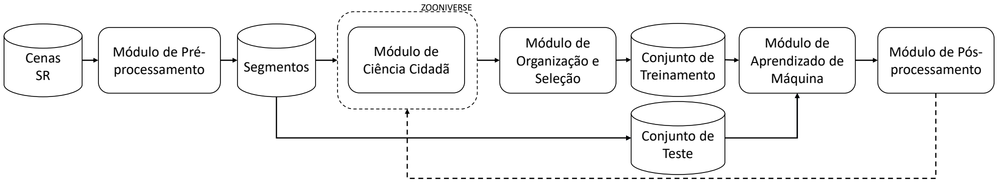

ZOONIVERSE
Cenas SR
Módulo de Pré-processamento
Segmentos
Módulo de Ciência Cidadã
Módulo de Organização e Seleção
Conjunto de Treinamento
Módulo de Aprendizado de Máquina
Módulo de Pós-processamento
Conjunto de Teste

Figura 1. Esquema simplificado da metodologia do projeto *ForestEyes*.

No Módulo de Pré-processamento, exemplificado na Figura 2, as cenas de sensoriamento remoto são adquiridas, processadas e segmentadas. Atualmente, as cenas utilizadas são do satélite Landsat-8, em que 7 de suas 11 bandas são adquiridas na plataforma *EarthExplorer* do *United States Geological Survey* (USGS) (etapa (a)). Essas 7 bandas – costal, azul, verde, vermelho, infravermelho próximo, infravermelho médio I e infravermelho médio II – passam por reamostragem e corte da região de interesse (etapa (b)). Depois, a dimensionalidade é reduzida para 3 (etapa (c)), através de *Principal Component Analysis* (PCA) [Jolliffe 2011], gerando uma imagem (etapa (d)) que é segmentada por um algoritmo de segmentação (etapa (e)), que pode ser *Simple Linear Iterative Clustering* (SLIC) [Achanta et al. 2012], *Image Foresting Transform*-SLIC (IFT-SLIC) [Alexandre et al. 2015], entre outros. Os segmentos (etapa (f)) são enviados a uma base de dados (etapa (g)).

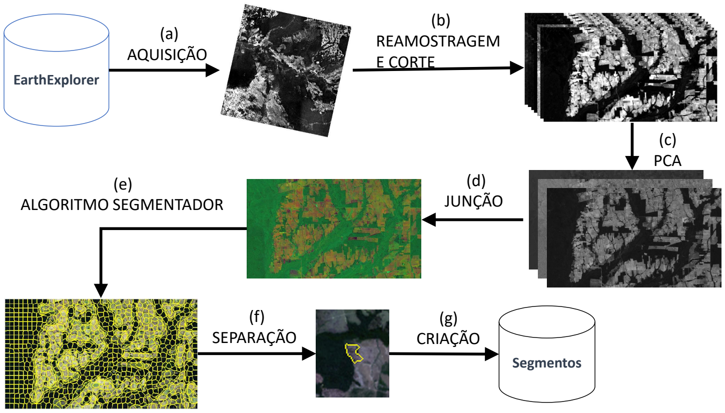


Figura 2. Esquematização do Módulo de Pré-processamento do projeto *ForestEyes*. Fonte: [Dallaqua 2020].

No Módulo de Ciência Cidadã, alguns dos segmentos na base de dados são enviados à plataforma *Zooniverse*, sendo criado um *workflow*, onde é definido o tipo de tarefa que os voluntários terão que realizar, a interface gráfica, as respostas possíveis, além da definição de um tutorial e de textos de ajuda. Os voluntários são instruídos pelo tutorial a classificar um segmento em 'floresta' ou 'não-floresta' se 70% ou mais *pixels* do segmento pertencerem a uma dessas classes; senão, os voluntários devem assinalar 'indefinido'. Atualmente é definido que cada tarefa deverá ser respondida por 15 voluntários distintos. Ao final, é feito o *download* das contribuições dos voluntários em um arquivo CSV.

No Módulo de Organização e Seleção, as contribuições dos voluntários passam por uma filtragem, eliminando contribuições redundantes – mesmo voluntário respondeu a mesma tarefa mais de uma vez – e respostas em excesso. Depois, a classe de cada tarefa é definida pelo voto majoritário das 15 respostas e são feitas análises como nível de dificuldade das tarefas [Arcanjo et al. 2016], convergência do consenso [Arcanjo et al. 2016], acurácia das classificações em relação à uma verdade, cálculo da pontuação dos voluntários [Arcanjo et al. 2016], entre outras. Para compor o conjunto de treinamento do próximo módulo são consideradas apenas as amostras que receberam classe 'floresta' ou 'não-floresta', sendo eliminadas as amostras de classe 'indefinido' e as em que ocorreu empate. Para o futuro também poderiam ser eliminadas amostras de maior dificuldade ou de acordo com alguma outra análise, por exemplo.

No Módulo de Aprendizado de Máquina, o conjunto de treinamento é o gerado pelo módulo anterior. Já o conjunto de teste é composto pelos segmentos que não foram enviados ao Módulo de Ciência Cidadã. Todos esses segmentos têm suas características extraídas por descritores de imagem. Depois, esses vetores de características são normalizados e então uma técnica de Aprendizado de Máquina é treinada e aplicada no conjunto de teste. Diferentes descritores de imagem e técnicas de Aprendizado de Máquina podem ser testadas.

O Módulo de Pós-processamento é planejado para o futuro, onde as classificações realizadas pelo Módulo de Aprendizado de Máquina podem ser validadas por especialistas, gerando alertas de desmatamento às autoridades competentes, por exemplo. Além disso, segmentos interessantes, que trariam informação ao treinamento das técnicas de Aprendizado de Máquina, poderiam ser enviados ao Módulo de Ciência Cidadã.

3. Estudos preliminares

Para atestar a viabilidade do *ForestEyes* foi realizado um estudo de caso para uma pequena área de Rondônia, no ano de 2016, comparando com verdades baseadas no PRODES. O projeto PRODES estima a taxa anual de desmatamento na Amazônia Legal Brasileira. Especialistas fotointerpretam imagens de sensoriamento remoto, coletadas pelos satélites Landsat-8, Sentinel e CBERS-4 e os dados gerados são disponibilizados no portal Terra-Brasilis [Souza et al. 2019].

Como no ano de 2016 as imagens temáticas do PRODES possuíam resolução espacial de $60m$, foi necessário reamostrar as 7 bandas do Landsat-8 de $30m$ para $60m$. Depois, a dimensionalidade foi reduzida através da técnica PCA e a imagem resultante foi segmentada pelo algoritmo SLIC, gerando 1022 segmentos com tamanho médio de 174 *pixels*. Esses segmentos foram apresentados aos voluntários em composições de cor RGB e falsa-cor com as bandas infravermelho médio II, infravermelho próximo e verde do Landsat-8. Foram recebidas 19.807 contribuições, de 227 voluntários [Dallaqua et al. 2019].

Três conjuntos verdade baseados no PRODES foram criados: um baseado em *pixel* (GT-PRODES) e dois baseados em segmentos (GT-U e GT-M). Em GT-PRODES, o mosaico PRODES é binarizado, em que todas as classes diferentes de ‘floresta’ são agregadas como ‘não-floresta’. Já para GT-U e GT-M, classificam-se segmentos de acordo com a proporção de *pixels* ‘floresta’ e ‘não-floresta’ de GT-PRODES. Em GT-M, por exemplo, a classe que possui mais de 50% dos *pixels* será a classe do segmento. Para esses 3 conjuntos verdade foram obtidas acurácias acima de 84%, mostrando como voluntários conseguem criar dados de qualidade [Dallaqua et al. 2019].

Para um experimento de Aprendizado de Máquina, selecionou-se apenas as amostras que receberam classe ‘floresta’ ou ‘não-floresta’, gerando um conjunto de treinamento de 934 amostras, sendo metade de uma classe e metade da outra. Já para o conjunto de teste, adquiriu-se segmentos de outra área de Rondônia, no ano de 2016, que foram rotulados de acordo com a verdade GT-M [Dallaqua et al. 2020].

Para cada segmento, de cada conjunto, 13 descritores de textura de Haralick [Haralick et al. 1973] foram extraídos, gerando vetores de características. Como classificador foi utilizado *Support Vector Machine* (SVM) [Hearst et al. 1998] e realizou-se tanto aprendizado supervisionado quanto aprendizado ativo [Tuia et al. 2011], em que o conjunto de treinamento é iterativamente construído através de uma seleção das amostras que trarão melhor representatividade ao treino. A utilização de aprendizado ativo emulou a situação em que existe o ciclo entre os Módulos de Ciência Cidadã e Pós-processamento (linha pontilhada na Figura 1).

Os resultados mostraram que aprendizado ativo, com uma boa inicialização, constrói um conjunto de treinamento que consegue resultados similares ao aprendizado supervisionado, porém, utilizando bem menos amostras [Dallaqua et al. 2020]. Assim, sua implementação no projeto *ForestEyes* conseguiria baratear ainda mais o processo, uma vez que apenas os segmentos importantes que seriam enviados aos voluntários.

4. Conclusão

Este artigo apresentou a metodologia do projeto *ForestEyes*, que visa aliar Ciência Cidadã com Aprendizado de Máquina para o monitoramento de desmatamento. O projeto é composto de cinco módulos principais, de fácil adaptação, permitindo o teste de diferentes técnicas para processamento de imagens, Aprendizado de Máquina, entre outros.

A viabilidade do projeto foi atestada por estudos preliminares, que validaram os resultados com dados do projeto PRODES. Esses estudos mostraram que os voluntários conseguem boa acurácia na classificação dos segmentos e que essas classificações podem ser utilizadas como conjunto de treinamento de um SVM.

O projeto ainda precisa de aprimoramento, mas pode no futuro ser uma alternativa para áreas onde não existam programas de monitoramento oficiais, ou também ser utilizado como fonte complementar de dados. No entanto, a viabilidade deste projeto não significa que especialistas possam ser substituídos por voluntários, ainda mais para um escopo tão crítico e sensível quanto monitoramento de desmatamento, que exige dados bem acurados para a implementação de políticas públicas.

5. Agradecimentos

Os autores gostariam de agradecer a agência fomentadora CAPES (Código de Financiamento 001), o USGS pelas imagens Landsat-8, o INPE pelos dados do PRODES, a plataforma *Zooniverse* por hospedar o projeto *ForestEyes*, e os voluntários que realizaram as tarefas. Esta pesquisa é parte do INCT da Internet do Futuro para Cidades Inteligentes, financiado por CNPq (proc. 465446/2014-0), CAPES (Código de Financiamento 001) e FAPESP (procs. 14/50937-1 e 15/24485-9).

Referências

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282.

Alexandre, E. B., Chowdhury, A. S., Falcao, A. X., and Miranda, P. A. V. (2015). IFT-SLIC: A general framework for superpixel generation based on simple linear iterative clustering and image foresting transform. In *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 337–344. IEEE.

Arcanjo, J. S., Luz, E. F., Fazenda, A. L., and Ramos, F. M. (2016). Methods for Evaluating Volunteers' Contributions in a Deforestation Detection Citizen Science Project. *Future Gener. Comput. Syst.*, 56(C):550–557.

Dallaqua, F., Fazenda, A., and Faria, F. (2019). ForestEyes Project: Can Citizen Scientists Help Rainforests? In *IEEE 15th International Conference on eScience*, pages 18–27. IEEE.

Dallaqua, F., Fazenda, A., and Faria, F. (2020). Aprendizado Ativo com dados de Ciência Cidadã para o monitoramento de florestas tropicais. In *1ª Escola Regional de Aprendizado de Máquina e Inteligência Artificial de São Paulo (ERAMIA-SP 2020)*.

Dallaqua, F. B. J. R. (2020). *Projeto ForestEyes - Ciência Cidadã e Aprendizado de Máquina na Detecção de Áreas Desmatadas em Florestas Tropicais*. PhD thesis, Universidade Federal de São Paulo. Instituto de Ciência e Tecnologia.

Fritz, S. et al. (2012). Geo-Wiki: An online platform for improving global land cover. *Environmental Modelling & Software*, 31:110–123.

Fritz, S., Fonte, C., and See, L. (2017). The role of citizen science in earth observation. *Remote Sensing*, 9(4).

Grey, F. (2009). Viewpoint: The age of citizen cyberscience. *Cern Courier*, 29.

Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

Jolliffe, I. (2011). *Principal component analysis*. Springer.

Luz, E. F., Correa, F. R. S., González, D. L., Grey, F., and Ramos, F. M. (2014). The ForestWatchers: A Citizen Cyberscience Project for Deforestation Monitoring in the Tropics. *Human Computation*, 1:137–145.

Martin, C. (2015). *On the Edge: The State and Fate of the World's Tropical Rainforests*. Greystone Books Ltd.

Petersen, R., Pintea, L., and Bourgault, L. (2017). Forest Watcher Brings Data Straight to Environmental Defenders. `http://blog.globalforestwatch.org/people/forest-watcher-brings-data-straight-to-environmental-defenders/`. Accessed: 08-07-2020.

Schepaschenko, D. et al. (2019). Recent advances in forest observation with visual interpretation of very high-resolution imagery. *Surveys in Geophysics*, 40(4):839–862.

Smith, A. M., Lynn, S., and Lintott, C. J. (2013). An introduction to the zooniverse. In *First AAAI conference on human computation and crowdsourcing*.

Soares, M. D., Santos, R., Vijaykumar, N., and Dutra, L. (2010). Citizen science-based labeling of imprecisely segmented images: Case study and preliminary results. In *Collaborative Systems-Simposio Brasileiro de Sistemas Colaborativos (SBSC), 2010 Brazilian Symposium of. IEEE*, pages 87 – 94.

Souza, A. et al. (2019). Metodologia Utilizada nos Projetos PRODES e DETER. *São José dos Campos: INPE*.

Tuia, D., Volpi, M., Copa, L., Kanevski, M., and Munoz-Mari, J. (2011). A survey of active learning algorithms for supervised remote sensing image classification. *IEEE Journal of Selected Topics in Signal Processing*, 5(3):606–617.

As dificuldades no rastreio de tempestades com uso de refletividade radar a partir de técnicas de geoprocessamento: Um estudo de caso sobre a região Amazônica

Helvecio B. Leal Neto[1], Adriano P. Almeida[1], Alan J. P. Calheiros[1]

[1]Instituto Nacional de Pesquisas Espaciais (INPE)
CEP 12.227-010 – São José dos Campos – SP – Brasil

{helvecio.neto,adriano.almeida,alan.calheiros}@inpe.br

Abstract. *This work proposes the application of geoprocessing techniques for storm tracking, and demonstrates what are the problems involved in using polygon centroids to track events in the instability lines, common in the Amazon region. The developed algorithm uses the overlapping of consecutive observation times in the weather radar data, to track the storm cell trajectory. Merger events between storm cells occur due to the dynamics of development of the instability lines, which makes tracking difficult. To solve this problem, a correction was applied to the vectors of the systems involved to a certain threshold that best represents the dynamics of storms in this region.*

Resumo. *Este trabalho propõe à aplicação de técnicas de geoprocessamento para o rastreio de tempestades, e demonstra quais os problemas envolvidos na utilização dos centróides de polígonos para rastreio de eventos das linhas de instabilidade, comuns na região Amazônica. O algoritmo desenvolvido utiliza a sobreposição de tempos consecutivos de observações nos dados de radar meteorológico, para rastrear a trajetória de células de tempestades. Eventos de fusões entre células de tempestade ocorrem devido a dinâmica de desenvolvimento das linhas de instabilidade, o que dificulta o rastreio. Para resolver este problema, foi aplicada uma correção nos vetores dos sistemas envolvidos para em certo limiar que melhor representar a dinâmica das tempestades na região.*

1. Introdução

As tempestades são fenômenos meteorológicos que podem gerar grandes transtornos, além de implicar em prejuízos econômicos podem causar perdas de vida devido aos eventos associados, como chuvas excessivas, raios, alagamentos, deslizamento de encostas e etc [Confalonieri 2015]. Uma das dificuldades encontradas para rastreio e previsão de tempestades está associado à dinâmica temporal e espacial destes fenômenos, devido à alta variabilidade de processos físicos, são necessários sistemas capazes de discretizar suas características morfológicas e aplicar técnicas que possibilitam rastrear o deslocamento de suas células convectivas, tais sistemas também podem realizar previsões em curtíssimo prazo (*nowcasting*, em inglês) [Wilson et al. 1998] e são essenciais para tomadas de decisões na prevenção de desastres naturais associados às tempestades.

Na Amazônia, uma grande quantidade de tempestades são formadas em linha e chamadas de linhas de instabilidade (LI), estes eventos produzem volumes significativos de chuva e propagam-se a velocidades médias entre 50 à 60 km/h, formando-se ao longo da costa e se deslocando de noroeste para sudoeste [Garstang et al. 1994]. As estruturas convectivas das LIs durante seu estágio de maturação apresentam uma dinâmica em seu movimento que provoca o desenvolvimento de novos sistemas convectivos, boa parte dessas estruturas são levadas para parte frontal (*gust front*) a direção de deslocamento tempestade, e o restante é espalhado na sua retaguarda [Gamache and Houze Jr 1982].

Algoritmos computacionais são bastante utilizados para identificação e rastreio do ciclo de desenvolvimento das tempestades, [Dixon and Wiener 1993],[Vila et al. 2008]. Técnicas baseadas no centróide das células convectivas são frequentemente aplicadas para definir a trajetória de deslocamento e contribuem para identificação dos processos dinâmicos envolvendo as tempestades (fusões, divisões e continuidades) [del Moral et al. 2018]. Contudo, o posicionamento do centróide em eventos de junções, quando duas células de tempestades se juntam e forma um único sistema, ainda é um desafio a ser resolvido. Devido a mudanças bruscas no deslocamento dos centróides o cálculo das velocidade e direção são comprometidos e na maioria dos casos são não realistas, comprometendo assim à eficácia do monitoramento e previsão dessas tempestades. Como estudo de caso, este apresenta o deslocamento de uma LI sobre a Amazônia e apresentada uma solução para correção da trajetória das células ao utilizar o centróide de células convectivas como característica para rastreio.

2. Área de estudo e dados

Os dados utilizados neste trabalho são observações volumétricas do Radar de Banda S do SIPAM (Sistema de Proteção da Amazônia) localizado na cidade de Manaus (AM) (Figura 1). O projeto GoAmazon (*Green Ocean Amazon*, em inglês) desenvolveu diversas atividades de pesquisas relacionadas à dinâmica da floresta e sua interação com a atmosfera, em uma de suas campanhas foram coletadas informações sobre a física das nuvens e precipitação, onde dados de observações do radar meteorológico foram compilados por [Schumacher and Funk 2018] e utilizados neste trabalho.

Um dos principais produtos obtidos através de varreduras volumétricas em coordenadas polares, é o CAPPI (Indicador de posição do plano de altitude constante, em inglês *Constant Altitude Plan Position Indicator*). Este produto é referente a projeções em um plano horizontal para uma determinada altitude de varredura volumétrica e bastante utilizado para monitoramento de precipitação [Raghavan 2013]. O valor de CAPPI utilizado foi baseado na altura de 2.0 km da superfície.

3. Metodologia

A metodologia aplicada neste trabalho visou extrair características morfológicas das tempestades com intuito de rastrear o deslocamento de uma LI sobre a região de Manaus. Para isso, foi desenvolvido um algoritmo capaz de identificar células de tempestade a partir de multilimiares de refletividade. O processo de rastreio foi dividido em etapas, onde, cada uma destas pretende melhor representar os eventos decorrentes dos processos dinâmicos em uma LI. A análise de estudo de caso é uma prática comum em meteorologia, onde é possível analisar a fundo as características dos eventos que são importante para uma determinada região, dando subsídio ao desenvolvimento de técnicas que possam melhorar a previsão do tempo local. Neste caso foi estudado um evento de LI que ocorreu no intervalo entre as 13:12 min e 14:12 min do dia 18 de Janeiro de 2014.

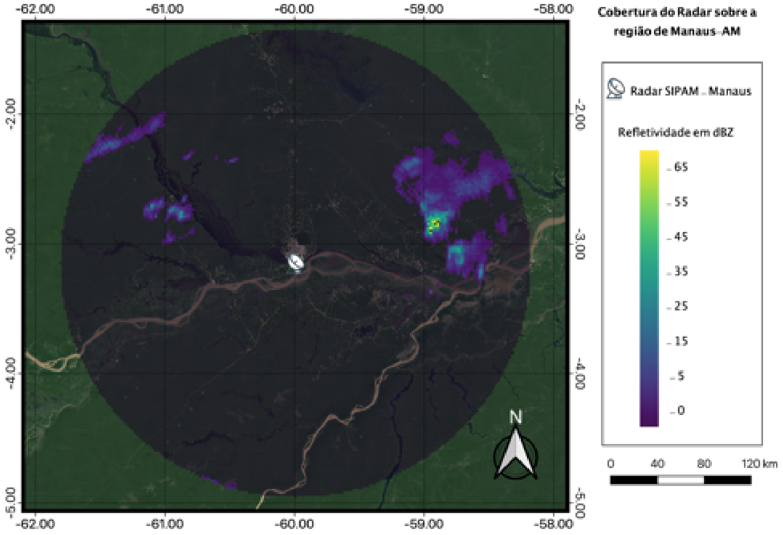


Figura 1. Área de cobertura do Radar de Banda para o CAPPI 2.0km

Na primeira etapa os pontos de grade com valores de refletividade do radar são rotulados com identificadores através do algoritmo DBSCAN (abreviação para *Density-based spatial clustering of applications with noise*, traduzido para português como, agrupamento espacial para aplicações com ruído baseado em densidade). Cada ”*cluster*”é referente à uma célula de tempestade a partir do seu limiar e em um tempo de observação. O processamento das características de contorno do limite exterior de cada clusters é realizado a partir do processo que faz a conversão dos pontos de grade em formato matricial para sua representação vetorial como polígonos (Figura 4). Estes polígonos individuais em tempos consecutivos (*t* e *t-1*) são processados e caso existam sobreposições (”*overlaps*”) o deslocamento é definido a partir dos centróides de cada um.

Devido a grande quantidade de processos dinâmicos que ocorrem nas tempestades, a representação da trajetória dos centróides pode ser comprometida, pois, múltiplas sobreposições entre polígonos de tempos consecutivos podem ocorrer. Isso está relacionado aos eventos de fusões e divisões nas células da tempestade, comuns durante o ciclo de sistemas organizados, como as LIs. Então, o vetor de deslocamento dos centróides precisa sofrer uma correção de modo que rastreio seja mais realístico.

A proposta para correção dos vetores de deslocamento nos centróides em casos de fusões, pode ser feita por meio da autocorreção do vetor deslocamento a partir do cálculo do vetor resultante da média entre as células convectivas que apresentaram junção em *t-1*, que irá substituir os valores do vetor de deslocamento no tempo *t*, esta reamostragem pode ser feita através da seguinte formulação (1).

$$\vec{V}_{corrigido(t)} = \frac{\sum_{i=1}^{n} \vec{V}_{i(t-1)}}{n} \tag{1}$$

Onde, $\vec{V}_{corrigido(t)}$ é o vetor resultante da média entre à soma dos vetores das células observadas antes *(t-1)* da fusão *(t)*.

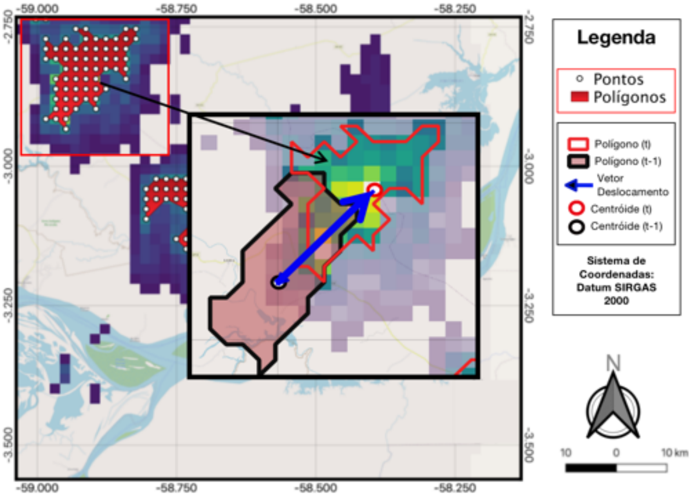


Figura 2. Representação da metodologia de rastreio para uma célula de tempestade em tempos consecutivos polígono vermelho (*t*) e polígono preto (*t-1*).

4. Resultados

O processo de rastreio dos eventos apresentados neste trabalho podem ser visualizados na Figura 3. Nestes eventos, duas zonas de limiar acima de 35 dBZ foram utilizadas para representar os problemas relacionados ao rastreio de células de tempestades, como as LIs. Logo, podem ser observados na figura supracitada o desenvolvimento destas células no decorrer de quatro tempos de varredura. Observa-se no último quadro a junção entre essas duas células convectivas, aqui referenciadas como evento 1 e evento 2. Note que o rastreio da trajetória de cada evento individual (célula convectiva embebida na tempestade) é descrito como linhas a partir do centróide de cada polígono, onde este polígonos são baseados na área correspondente ao limiar de rastreio de 35 dBZ para o nível Z de 2 km de altitude.

A Figura 4a mostra a disposição entre os polígonos do eventos 1 (azul), 2 (verde) e fusão (vermelhos), assim como seus vetores baseados nos centróides. Já a Figura 4b apresenta a relação entre todos os vetores associados a Figura 4a a partir de um mesmo ponto. Aplicando a proposta apresentada na seção de metodologia, utilizou-se a equação (1) para correção do vetor de deslocamento dos centróides entre o Evento 1 (Polígono azul) e o Evento de Fusão (Polígono Vermelho).

Observa-se que logo após a ocorrência da fusão entre os eventos 1 e 2 que o vetor de deslocamento, que é baseado nos centróides do evento 1 (aquele de maior área e que dará continuidade ao ciclo de vida da tempestade) no tempo t-1 e o centróide atual, apresenta uma velocidade muito maior que aquelas apresentadas nos tempos antecedentes e com direção que não condiz a realidade, com isso, o vetor deslocamento deve ser reajustado, uma vez que, se aplicada uma análise subjetiva quadro a quadro, é possível notar que este sistemas têm um deslocamento em direção a sudoeste (225^{o}) com velocidade de aproximadamente 12 km/h, que aqui será referenciado como vetor ”real”.

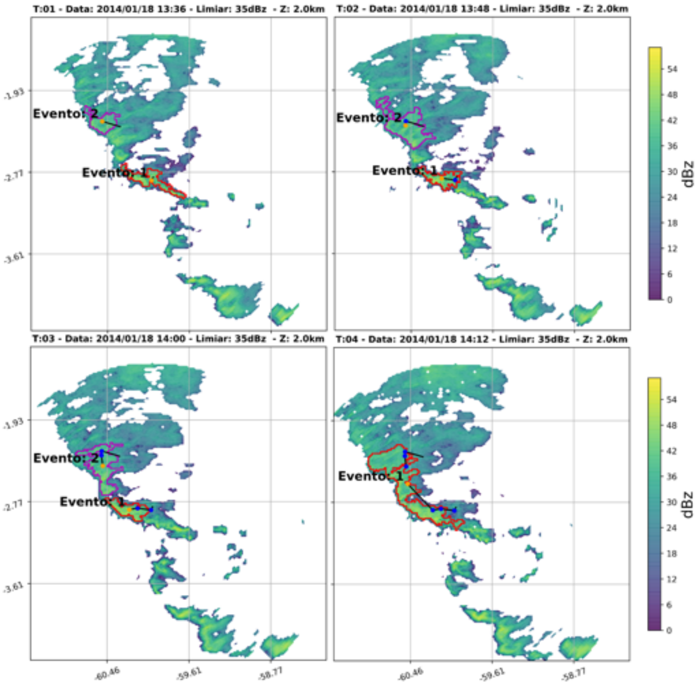

T:01 - Data: 2014/01/18 13:36 - Limiar: 35dBz - Z: 2.0km
T:02 - Data: 2014/01/18 13:48 - Limiar: 35dBz - Z: 2.0km
T:03 - Data: 2014/01/18 14:00 - Limiar: 35dBz - Z: 2.0km
T:04 - Data: 2014/01/18 14:12 - Limiar: 35dBz - Z: 2.0km
Evento: 2
Evento: 1
-1.93
-2.77
-3.61
-60.46
-59.61
-58.77
dBz
54
48
42
36
30
24
18
12
6
0

Figura 3. Rastreio dos Eventos 1 e 2 para Limiar 35 dBz no nível Z de 2 km

O vetor corrigido (Vetor Verde na Figura 4.b) corresponde a média entre os vetores dos Eventos 1 e 2 no tempo T03 para o tempo consecutivo T04. O vetor correspondente ao movimento realístico da LI (Vetor Cinza em ambas as figuras, ”real”) foi adicionado para comparação com os resultados alcançados neste trabalho.

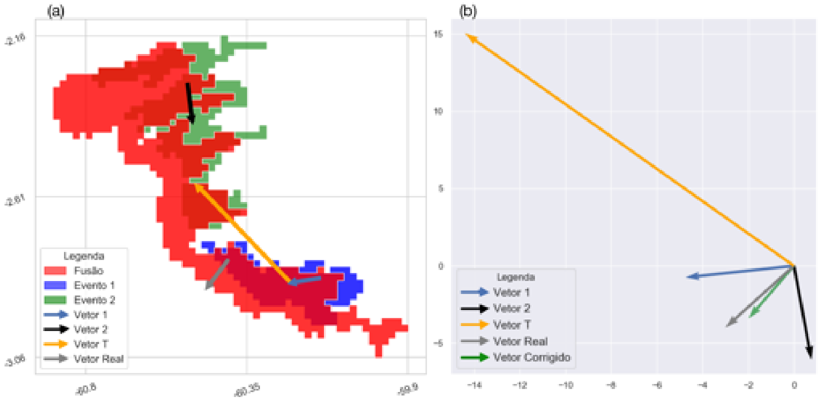


Figura 4. (a) Fusão (polígono vermelho) entre os eventos 1 (azul) e 2 (verde). (b) Relação entre os vetores e aplicação da correção dos vetores.

Nota-se pela Figuras 4b que o vetor corrigido (verde), aquele que substituirá erro ocasionado pela junção (vetor laranja), que é baseado nos vetores azul (Evento 1) e preto (Evento 2) é mais próximo daquele que seria o vetor real (Cinza) do deslocamento do sistemas. Deste modo, fica claro que para este evento em episódios de junção, uma correção simples é suficiente para determinar o deslocamento real de tempestades em linha.

5. Considerações Finais

Como mostrado nos resultados deste trabalho, fica evidente que a aplicações de técnicas simples, como a correção dos vetores de deslocamento, em sistemas mais complexos de rastreio de tempestades, podem ser importantes na melhorias do monitoramento e previsão de sistemas severos, como as linhas de instabilidade. Apesar dos resultados serem aplicados a apenas um sistema, como neste estudo de caso, as diferenças regionais no rastreio deste tipo de tempestade com relação a sua morfologias, são pequenas, e tais resultados podem ser utilizados para outras localidades, e técnicas de rastreios semelhantes que partam do princípio de posicionamento dos centróides para definir o vetor deslocamento. Cabe ressaltar que este estudo de caso faz parte de um trabalho mais profundo sobre rastreio de tempestades que envolve o uso de diferentes limiares e plataformas, cujos os resultados serão mais robusto do ponto de vista estatístico.

Agradecimentos: O presente trabalho foi realizado com apoio das Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 e Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, Processo 438310/2018-7).

Referências

Confalonieri, U. E. (2015). Variabilidade climática, vulnerabilidade social e saúde no brasil. *Terra livre*, 1(20):193–204.

del Moral, A., Rigo, T., and Llasat, M. C. (2018). A radar-based centroid tracking algorithm for severe weather surveillance: Identifying split/merge processes in convective systems. *Atmospheric Research*, 213:110–120.

Dixon, M. and Wiener, G. (1993). Titan: Thunderstorm identification, tracking, analysis, and nowcasting—a radar-based methodology. *Journal of atmospheric and oceanic technology*, 10(6):785–797.

Gamache, J. F. and Houze Jr, R. A. (1982). Mesoscale air motions associated with a tropical squall line. *Monthly Weather Review*, 110(2):118–135.

Garstang, M., Massie Jr, H. L., Halverson, J., Greco, S., and Scala, J. (1994). Amazon coastal squall lines. part i: Structure and kinematics. *Monthly Weather Review*, 122(4):608–622.

Raghavan, S. (2013). *Radar meteorology*, volume 27. Springer Science & Business Media.

Schumacher, C. and Funk, A. (2018). Goamazon2014/5 rain rates from the sipam manaus s-band radar. (2).

Vila, D. A., Machado, L. A. T., Laurent, H., and Velasco, I. (2008). Forecast and tracking the evolution of cloud clusters (fortracc) using satellite infrared imagery: Methodology and validation. *Weather and Forecasting*, 23(2):233–245.

Wilson, J. W., Crook, N. A., Mueller, C. K., Sun, J., and Dixon, M. (1998). Nowcasting thunderstorms: A status report. *Bulletin of the American Meteorological Society*, 79(10):2079–2100.

Dinâmica da intensificação da agricultura temporária na Área de Proteção Ambiental Ilha do Bananal-Cantão

Talita Nogueira Terra[1], Ana Cláudia dos Santos Luciano[1,3], Júlio César Dalla Mora Esquerdo[2], Alexandre Camargo Coutinho[2], João Francisco Gonçalves Antunes[2], João Luís dos Santos[1], Lídia Sanches Bertolo[1]

[1]Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH - Projeto Paisagens Rurais, 70.711-902, Brasília, Distrito Federal, Brasil

[2]Embrapa Informática Agropecuária, 13083-886, Campinas, Brasil

[3]Núcleo Interdisciplinar de Planejamento Energético (NIPE-UNICAMP), 13083-896, Campinas, Brasil

{talita.terra; ana.luciano; joao.santos; lidia.bertolo}@giz.de, {julio.esquerdo, alex.coutinho; joao.antunes}@embrapa.br

Abstract. *This work aimed to map the croplands (harvest 2019-2020) in the Protection Area Ilha do Bananal-Cantão, identifying the number of productive cycles and evaluating the land use changes from 2013 and 2018. For mapping the 2019-2020 harvest, Moderate Resolution Imaging Spectroradiometer (MODIS) images were classified, and the results were scaled up at the Landsat-8/OLI level through visual interpretation. For the years 2013 and 2018, were used data from the TerraClass Cerrado project. We concluded that cropland not only expanded territorially, but also intensified in the region. Most of the expansion took place in previously anthropized areas; however, 1% of the agricultural area in 2020 came from an area deforested as of 2018.*

Resumo. *Este trabalho teve como objetivo mapear as culturas agrícolas temporárias na safra 2019-2020 da APA Ilha do Bananal-Cantão, identificando a quantidade de ciclos produtivos e avaliando a dinâmica de uso e cobertura da terra dos anos de 2013 e 2018. Para o mapeamento da safra 2019-2020 foram classificadas imagens do sensor MODIS e compatibilizadas com a escala de imagens Landsat-8/OLI por meio de interpretação visual. Para os anos de 2013 e 2018 foram utilizados dados do projeto TerraClass Cerrado. Concluiu-se que a agricultura não só se expandiu territorialmente, como também se intensificou na região. A maior parte da expansão ocorreu em áreas previamente antropizadas, sendo que 1% delas em 2020 foi proveniente de áreas desmatadas a partir de 2018.*

1. Introdução

O Tocantins, estado brasileiro localizado na porção oeste do domínio morfoclimático do Cerrado, abrange uma área territorial de 277.466,763km^2. Em se tratando do território brasileiro, é o segundo estado com o maior crescimento na produção de grãos desde 2013 [IBGE, 2020]. No período de 2013 a 2020, o estado do Tocantins apresentou uma taxa de crescimento anual média de 12% em área plantada de grãos [IBGE, 2020], sendo a soja a cultura agrícola predominante, ocupando mais de 70% da área total plantada destinada à produção de grãos do estado [IBGE, 2020]. Neste mesmo período, a Área de Proteção Ambiental (APA) Ilha do Bananal-Cantão, localizada na região oeste do Tocantins, apresentou um crescimento anual médio de 43% em área plantada de grãos [IBGE, 2020].

A expansão e intensificação da atividade agrícola é fruto da implantação de cultivares adaptados às condições edafoclimáticas do Cerrado, da mecanização e automação dos processos de produção de grãos e da intensificação do uso e cobertura da terra, que envolve, em uma mesma área, a prática de mais de um ciclo anual de produção agrícola, usualmente denominados como safra e safrinha.

Conhecer onde as terras agrícolas estão se expandindo, qual a taxa de expansão e quais usos e cobertura da terra estão sendo substituídos é fundamental para a tomada de decisão sobre a região, seja no setor agrícola e ou ambiental. Técnicas de geoprocessamento permitem caracterizar a área analisada nos contextos físico-biótico, infraestrutura logística, delimitação e caracterização de polos agrícolas [Silveira et al., 2015]. A pesquisa sobre a dinâmica de uso e cobertura da terra associada à expansão de áreas agrícolas no Brasil é extensa na literatura [Scaramuzza et al., 2017; Rausch et al., 2019]. No entanto, na maioria das vezes, não são apresentados resultados da quantificação de ciclos agrícolas, dificultando as análises referentes à intensificação da agricultura [Qiu et al., 2014; Picoli et al., 2018].

Nesse contexto, este trabalho teve como objetivo mapear as culturas agrícolas e identificar a quantidade de ciclos produtivos na APA Ilha do Bananal-Cantão, na safra 2019-2020. Além disso, avaliou-se a intensificação da agricultura temporária por meio da análise da dinâmica de uso e cobertura das áreas agrícolas mapeadas em 2020, interpretando as mudanças da série histórica dos anos de 2013 e 2018.

2. Material e Métodos

A APA Ilha do Bananal-Cantão ocupa 6% (1.569.647ha) do estado do Tocantins e abrange nove municípios (Figura 1).

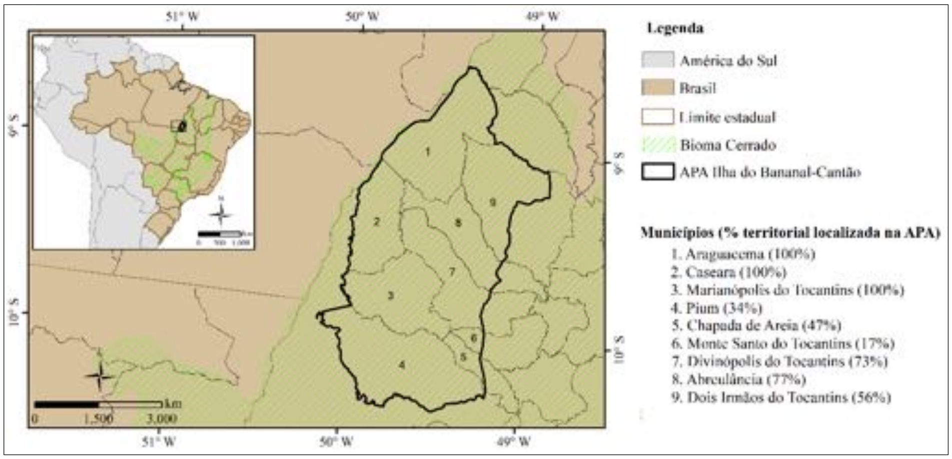

51° W
50° W
49° W
9° S
10° S
1
2
3
4
5
6
7
8
9
0 1.500 3.000 km
Legenda
América do Sul
Brasil
Limite estadual
Bioma Cerrado
APA Ilha do Bananal-Cantão
Municípios (% territorial localizada na APA)
1. Araguacema (100%)
2. Caseara (100%)
3. Marianópolis do Tocantins (100%)
4. Pium (34%)
5. Chapada de Areia (47%)
6. Monte Santo do Tocantins (17%)
7. Divinópolis do Tocantins (73%)
8. Abreulândia (77%)
9. Dois Irmãos do Tocantins (56%)

Figura 1. Delimitação da APA Ilha do Bananal-Cantão no estado do Tocantins e principais municípios que a integram.

A área de estudo está localizada na zona climática Tropical com inverno seco (Aw), segundo a classificação de Köppen-Geiger [Alvarez et al., 2014]. A temperatura média anual varia de 25ºC a 26ºC [Seplan, 2012] e as precipitações médias anuais são acima de 1.900mm, sendo 43% do território acima de 2.200mm, com as chuvas concentradas nos meses de outubro a abril [Alvarez et al., 2014]. O solo predominante da área de estudo é o Plintossolo (78%) seguido do Neossolo (12%) e 96% da área apresenta declividade nas categorias A ou B, ou seja, no máximo de 10% de declive [Seplan, 2012].

O mapeamento da agricultura temporária do ano de 2020 foi feito por meio da análise do comportamento espectro-temporal extraído de imagens do sensor MODIS (produto MOD13Q1), resolução espacial de 250m. O processamento se deu pela entrada de amostras no pacote computacional "sits" - *Satellite Image Time Series Analysis for Remote Sensing Data Cubes*, no programa R, que vem sendo utilizado com sucesso em alguns trabalhos recentes envolvendo o mapeamento do uso e cobertura da terra em larga escala (Picoli et al., 2018; Simões et al., 2020). A partir do sits foi feita a classificação automática da agricultura temporária de 1 ciclo e 2 ciclos. Em seguida, foi feita a passagem de escala para a resolução espacial de 30m, com base em imagens Landsat-8/OLI, por meio de interpretação visual.

Para avaliar a origem das áreas agrícolas do ano de 2020 foi analisado o mapeamento do ano de 2013 do projeto TerraClass Cerrado [Scaramuzza et al., 2017] e o produto preliminar do ano de 2018 (TerraClass Cerrado 2018 em fase final de elaboração sendo produto do projeto Paisagens Rurais do FIP - http://fip.mma.gov.br/projeto-paisagem/). O mapeamento do ano de 2013 apresentou originalmente um conjunto de 11 classes temáticas, as quais foram reorganizadas em sete classes (Tabela 1), segundo critérios específicos de relevância para a análise da dinâmica. A atualização da classe temática agricultura temporária do mapeamento do TerraClass Cerrado 2013 em 1 ciclo ou 2 ciclos se deu pela interpretação do perfil espectro-temporal obtido a partir da ferramenta Web SATVeg (Sistema de Análise Temporal da Vegetação) [Esquerdo et al., 2020].

Tabela 1. Compatibilização da legenda das classes temáticas mapeadas pelo TerraClass Cerrado 2013 e o mapeamento de 2018, com as classes de interesse deste trabalho.

TerraClass 2013	Classes temáticas deste trabalho	TerraClass 2018
Agricultura anual	Agricultura temporária de 1 ciclo	Agricultura temporária de 1 ciclo
	Agricultura temporária de 2 ciclos	Agricultura temporária de 2 ciclos
Área natural	Área natural	Área natural
Pastagem plantada	Pastagem	-
-	Pastagem e vegetação secundária	Pastagem e vegetação secundária
-	Desmatado no ano de 2018	Desmatado no ano de 2018
Agricultura perene	Outros	Agricultura perene
		Agricultura semi-perene
Silvicultura		Silvicultura
Área urbana		Área urbana
Corpo d'água		Corpo d'água
Mineração		Mineração
Mosaico de ocupações		Mosaico de ocupações
Não observado		Não observado
-		Outros
Solo exposto		Solo exposto

Após o processo de compatibilização das legendas e o recorte das áreas de interesse, de acordo com o mapeamento das áreas agrícolas do ano de 2020, os três mapeamentos de uso e cobertura da terra foram espacialmente sobrepostos. Essa abordagem possibilitou calcular a frequência relativa das classes temáticas e apresentar a dinâmica de uso e cobertura da terra por meio do diagrama de Sankey [Sankey, 2014], que favorece a observação de fluxos de transição entre as classes ao longo dos anos.

A partir da compatibilização das classes temáticas, os mapeamentos puderam ser espacialmente comparados para a condução das análises das dinâmicas territoriais ocorridas entre os mapeamentos de 2013, 2018 e 2020. No entanto, uma vez que utilizamos o produto oriundo do TerraClass Cerrado 2013, não foi possível separar a classe temática vegetação primária da classe vegetação secundária, sendo assim considerada uma limitação do trabalho. Além disso, o mapeamento das áreas agrícolas do ano de 2020 foi realizado a partir da classificação de imagens de baixa resolução espacial (MODIS), demandando trabalho manual da passagem de escala (Landsat).

3. Resultados e Discussão

No ano de 2020, a APA Ilha do Bananal-Cantão apresentou 8% (120.013ha) de sua área ocupada pela agricultura temporária. A análise destes 8% indicou que 92% (110.731ha) dessas áreas apresentaram 2 ciclos de produção agrícola (safra/safrinha), o que é possível em regiões com alta pluviometria e quando o período chuvoso é um pouco mais longo [Sano et al., 2019], como ocorre na APA Ilha do Bananal-Cantão.

De acordo com o diagrama de Sankey (Figura 2), as áreas de agricultura temporária encontradas na safra 2019-2020 são oriundas, predominantemente, da classe temática pastagem no ano de 2013. Em 2013, somente 10.048 ha (8%) das áreas já eram ocupadas por agricultura temporária e 20.526 ha (17%) eram áreas de cobertura natural. Em um intervalo de cinco anos (de 2013 a 2018), 19.441 ha (95%) das áreas de cobertura natural foram antropizadas pela incorporação de 8.936 ha de agricultura temporária de 1 ciclo (44%), 6.578 ha de 2 ciclos (32%) e 3.927 ha da associação de pastagem com vegetação secundária (19%).

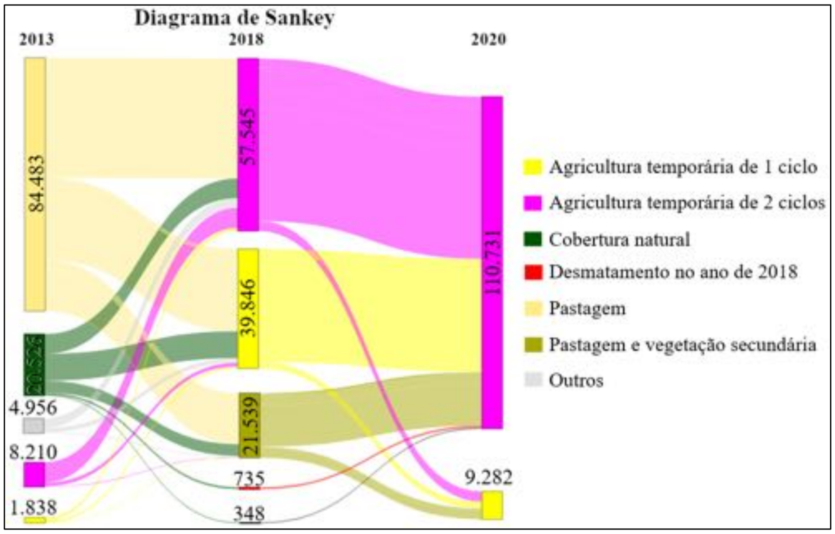


Figura 2. Diagrama de Sankey das áreas agrícolas entre os anos de 2013, 2018 e 2020.

O diagrama de Sankey permite afirmar que a área de estudo vem sofrendo um forte processo de intensificação agrícola, onde 37.648 ha (34%) das áreas cultivadas duas vezes na safra 2019-2020 proveem de áreas agrícolas cultivadas uma vez na safra 2017-2018, e 17.315 ha (16%) são provenientes da classe temática de pastagem associada a vegetação secundária. E não menos importante, têm-se 1.083 ha (1%) de áreas desmatadas convertidas diretamente para uso agrícola com dois ciclos produtivos, resultado que corrobora com o estudo de Rausch et al. (2019).

Os municípios que mais sofreram expansão e intensificação da agricultura temporária foram Araguacema, Caseara e Marianópolis do Tocantins. Estes três municípios representam 5.907 ha (64%) da agricultura temporária de um ciclo na safra 2019-2020 e 80.022 ha (72%) de dois ciclos (Figura 3). Segundo o IBGE (2020), estes municípios são responsáveis por 7% da produção de grãos de todo o Tocantins, tendo como destaque as culturas agrícolas de soja e milho. Ainda de acordo com o IBGE, a cultura do milho tem se expandido também como a cultura da segunda safra, igualmente importante em termos de geração de renda, sendo cultivado após o ciclo produtivo da soja, o que está de acordo com o resultado encontrado por Moreira et al. (2019).

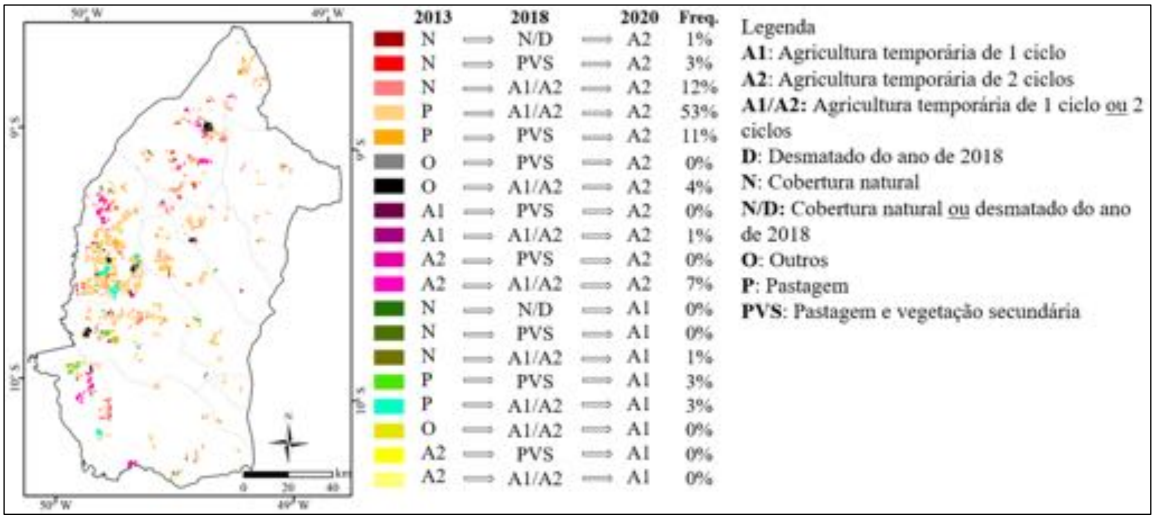


Figura 1. Distribuição espacial da dinâmica de uso e cobertura da terra analisada nos anos de 2013, 2018 e 2020.

4. Conclusões

A APA Ilha do Bananal-Cantão se apresentou como importante fronteira agrícola na última década. A dinâmica do uso e cobertura da terra desde o ano de 2013 indicou não só a expansão da agricultura, como também a sua intensificação na região, sobretudo pelo aumento do cultivo do milho de segunda safra em áreas cultivadas com soja na primeira safra. A maior parte da expansão se deu em áreas previamente mapeadas como pastagem no ano de 2013. No entanto, 1.083 ha da área agrícola de 2020 foi proveniente de áreas desmatadas a partir do ano de 2018, indicando a abertura de áreas para agricultura temporária, favorecidas pela mecanização e intensificação da produção.

5. Referências

Alvares, C.A. et al. (2014). Köppen's climate classification map for Brazil. Meteorologische Zeitschrift 22(6): 711–728.

CONAB - Companhia Nacional de Abastecimento. (2020). Acompanhamento da Safra 2019/2020 - 11º Levantamento.

IBGE - Levantamento Sistemático da Produção Agrícola. (2020). Acessado em 13 de agosto de 2020, disponível em https://sidra.ibge.gov.br/tabela/6588.

Esquerdo, J.C.D.M. et al. (2020). SATVeg: A web-based tool for visualization of MODIS vegetation indices in South America. Computers and Electronics in Agriculture, 175.

Moreira, D.C. et al. (2019). Panorama do cultivo e produtividade da soja na APA Ilha do Bananal/Cantão, Tocantins: safras 2008/2009 a 2015/2016. Journal of Bioenergy and Food Science, 6(4): 119-131.

Picoli, M.C.A. et al. (2018). Big earth observation time series analysis for monitoring Brazilian agriculture. ISPRS Journal of Photogrammetry and Remote Sensing, 145, 328-339.

Qiu, B. et al. (2014). A new methodology to map double-cropping croplands based oncontinuous wavelet transform. International Journal of Applied Earth Observation, 26, 97-104.

Rausch, L.L. et al. (2019). Soy expansion in Brazil's Cerrado. Conservation Letters, 12.

Sankey Diagram Generator by Dénes Csala, based on the Sankey plugin for D3 by Mike Bostock; https://sankey.csaladen.es; 2014.

Sano, E.E. et al. (2019). Cerrado ecoregions: A spatial framework to assess and prioritize Brazilian savanna environmental diversity for conservation. Journal of Environmental Management, 232, 818-828.

Simões, R. et al. (2020). Land use and cover maps for Mato Grosso State in Brazil from 2001 to 2017. Scientific Data, 7(34).

Scaramuzza, C.A.M. et al. (2017). Land-use and land-cover mapping of the brazilian Cerrado based mainly on Landsat-8 satellite images. Revista Brasileira de Cartografia, Edição de Fotogrametia e Sensoriamento Remoto, 69(6): 1041-1051.

Seplan. (2012). Secretaria do Planejamento e da Modernização da Gestão Pública. Base de Dados Geográficos do Tocantins - atualização 2012. Palmas, SEPLAN/DEZ. (Atualização de arquivos em escala 1:1.000.000 da Base de Dados Geográficos do Tocantins). Organizado por Rodrigo Sabino Teixeira Borges e Paulo Augusto Barros de Sousa.

Silveira, G.R.P. et al. (2015). Geoprocessamento aplicado na espacialização da capacidade de uso do solo em uma área de importância agrícola. Revista Energia na Agricultura, 30(4): 363-375.

LapsusVGI: um framework para Sistemas de Gerenciamento de Informação sobre Deslizamento de Terra

Lucas F. Dorigueto, Carlos H. T. Brumatti, Jugurta Lisboa-Filho

Departamento de Informática – Universidade Federal de Viçosa (UFV)
Viçosa, MG, Brasil

{lucas.dorigueto,carlos.h.tavares,jugurta}@ufv.br

Abstract. *Voluntary Geographic Information (VGI) has great potential to contribute to data collection in emergency and humanitarian aid situations. VGI integrated with Disaster Information Management Systems (DIMS) can assist the manager in making decisions. However, failure to adopt standards makes data sharing and DIMS usability difficult, and may be more critical with the integration of VGI collected without the use of standards. This work presents a fully standardized and interoperable framework, which integrates official data and VGI to assist in the implementation of DIMS focused on landslides and dam break.*

Resumo. *Informação Geográfica Voluntária (VGI) tem grande potencial para contribuir com a coleta de dados em situações de emergência e de ajuda humanitária. VGI integrado à Sistemas de Gerenciamento de Informação de Desastres (DIMS) pode auxiliar o gestor na tomada de decisões. Porém, a não adoção de padrões dificulta o compartilhamento de dados e a usabilidade de DIMS, podendo ser mais crítico com a integração de VGI coletada sem o uso de padrões. Este trabalho apresenta um framework totalmente padronizado e interoperável, que integra dados oficiais e VGI para auxiliar na implementação de DIMS focado em deslizamento de terra e rompimento de barragens.*

1. Introdução

Segundo Ellis, Gibbs e Rein (1991), sistemas de software colaborativos possibilitam que um grupo de pessoas alcancem um determinado objetivo, por meio de uma interface que permita aos usuários utilizarem um ambiente de software compartilhado (*groupware*). Neste contexto, sistemas de Informação Geográfica Voluntária (VGI) são sistemas colaborativos nos quais os usuários colaboram com dados associados a uma localização espacial (Goodchild, 2007). Essa categoria de sistema tem sido utilizada para auxiliar na tomada de decisões em situações de emergências ou desastres. Por exemplo, o sistema Ushahidi, de coleta de VGI, foi utilizado após o terremoto ocorrido no Haiti em 2010 (Ushahidi, 2019).

Sistemas de Gerenciamento de Informações de Desastres (DIMS) são sistemas usados para apoiar a integração de ações de diferentes órgãos e instituições quando há ocorrência de catástrofes (Ryoo e Choi, 2006). Assim, VGI tem grande potencial para contribuir com a coleta de dados a serem utilizados em situações de emergência e de ajuda humanitária, podendo ser utilizada de forma complementar por gestores para auxiliar na tomada de decisões nas etapas do gerenciamento de ações de desastres.

De acordo com Ryoo e Choi (2006), interoperabilidade é um dos principais obstáculos dos DIMS devido à falta de padronização nos dados. Com base em um levantamento realizado por Tavares et al. (2018), num período de três anos foram identificados apenas onze sistemas colaborativos voltados para o gerenciamento de ações de emergências, sendo que somente dois sistemas cobriam todas as etapas de uma emergência (pré-evento, durante e pós-evento). Por sua vez, esses dois sistemas não seguem padrões internacionais em sua arquitetura, podendo não garantir a interoperabilidade a qualidade dos dados utilizados/gerados nestas plataformas.

O objetivo deste trabalho é propor um framework e projetar um DIMS com base em padrões internacionais voltados para o gerenciamento de situações de emergência, utilizando VGI como fonte de dados, com o intuito de auxiliar no gerenciamento de ações e análise de desastres para amparar gestores na tomada de decisões, permitindo que gestores e voluntários colaborem durante uma catástrofe, provendo interoperabilidade e qualidade das informações.

O framework proposto, chamado *LapsusVGI*, é voltado para eventos relacionados a deslizamentos de terra e rompimento de barragens, devido a tais eventos serem problemas recorrentes no Brasil. Segundo o IBGE (2018), 15% dos municípios brasileiros já foram atingidos por deslizamentos de terra, além disso, certas áreas brasileiras contam com histórico de rompimento de barragens, por exemplo, o município de Brumadinho.

Na Seção 2 são apresentados alguns trabalhos relacionados. Na Seção 3 são relacionados os padrões internacionais utilizados neste trabalho. Na Seção 4 é apresentado o projeto do framework *LapsusVGI*, por fim, na Seção 5 são apresentadas as conclusões e os próximos passos do projeto.

2. Trabalhos Relacionados

Fathani e Karnawati (2018) descrevem um sistema de aviso precoce de deslizamento de terra de baixo custo, capaz de informar possíveis deslizamentos de terra horas antes da catástrofe ocorrer. Os sensores utilizados no sistema são capazes de informar, por exemplo, o declive do solo e a precipitação de chuva no local. O sistema é construído levando em conta estudos sociais, geotécnicos e geológicos do local.

Ushahidi é um exemplo de sistema VGI colaborativo, o qual tem sido utilizado para auxiliar em diversas situações. Além do terremoto no Haiti, foi utilizado no monitoramento de incidentes, como ofensas eleitorais e violência durante as eleições no Quênia, no ano de 2017, e também no desenvolvimento do SomaliaSpeaks, que permite que usuários expressem como os desastres têm afetado suas vidas (USHAHIDI 2019).

3. Padrões para Interoperabilidade e Gerenciamento de Emergências

O padrão *ISO 22327 - Guidelines for implementation of a comunity-based landslide early warning system* (ISO, 2018) define métodos e procedimentos para serem implementados em sistemas de aviso precoce voltados para comunidades vulneráveis à deslizamento de terra. Entre suas especificações está a utilização de rotas de fuga e elaboração de mapas utilizando símbolos definidos nos padrões ISO 7001 e ISO 7010, além da especificação dos elementos que uma interface deve apresentar aos seus usuários.

O padrão *ISO 22351 - Message strucuture for exchange of Information*, descreve a estrutura de mensagens para serem utilizadas entre organizações envolvidas em situações de emergência. Essa estrutura é chamada de Informações Compartilhadas de Gerenciamento de Emergência (EMSI) (ISO, 2015). Com a utilização de uma EMSI, é possível especificar detalhes do evento, além dos recursos disponíveis para a superação de desastres e tarefas que possam ser feitas para minimizar o impacto de emergências.

Os padrões do Open Geospatial Consortium (OGC) também podem ser utilizados em DIMS. Por exemplo, o padrão *Web Map Service* (WMS) fornece uma interface HTTP para que imagens georreferenciadas possam ser transferidas através de camadas em formatos como JPEG e PNG, para serem exibidas em aplicações Web. O padrão *Web Feature Service* (WFS) permite o compartilhamento de informações espaciais em nível de feição, permitindo o gerenciamento da informação através de operações de descrição e recuperação de feições espaciais (OGC, 2020).

Para auxiliar na customização visual de feições e camadas espaciais, OGC define o padrão *Style Layer Descriptor* (SLD), que é uma extensão do WMS que permite a representação de feições através de símbolos utilizando a linguagem *Symbology Encoding* (SE). Uma de suas funcionalidades é a definição de uma operação para acessar de forma padronizada os símbolos de legenda.

4. Framework *LapsusVGI*

LapsusVGI (*Lapsus*, significa deslizar/escorregar em latim), é um framework em desenvolvimento, com o intuito de amparar gestores na tomada de decisões e auxiliar a comunidade em momentos de emergências. A seguir são descritas as principais especificações do *LapsusVGI*, que considera os requisitos de Ryoo e Choi (2006) e padrões internacionais voltados para interoperabilidade em situações de emergência.

Ryoo e Choi (2006) descrevem requisitos essenciais que todo DIMS deve possuir. Esses requisitos podem ser divididos em quatro grupos: coleta de dados; distribuição e compartilhamento de dados; processamento de dados; e apresentação de dados. Cada requisito possui inúmeras especificações.

Porém, nem todas as especificações dos requisitos foram incorporadas ao framework, dado que o foco são os deslizamentos de terra. Deste modo, somente foram selecionados requisitos que possam ser relacionados diretamente com esse tema e também com a integração com VGI. Segundo Ryoo e Choi (2006) é prioritário incorporar componentes de maior importância em um primeiro momento, e adicionar o restante de forma incremental. Os demais requisitos podem ser acrescentados em trabalhos futuros.

No requisito de coleção de dados, é descrita a habilidade de reconhecer e tratar dados de desastres provindos de diferentes fontes, além da capacidade do sistema conseguir tratar dados em qualquer formato possível, ou adotar padrões dedicados ao armazenamento de informações de desastres. Deste modo, para satisfazer essas condições, foi estabelecido que a alimentação do *LapsusVGI* deve ser feita a partir de duas principais fontes de dados: VGI e dados oficiais.

VGI é incorporada ao framework de duas formas. Primeiro, da plataforma de mapeamento colaborativo OpenStreetMap (OSM) é extraído o mapa base da região do evento. Adicionalmente, a coleta de dados sobre as contribuições dos voluntários é feita por meio de formulários de sistemas Web ou aplicativos móveis, ambos com interface que permitem o registro, sobre o mapa base, da localização espacial da contribuição. Em relação aos dados oficiais são usados, por exemplo, mapas de áreas de risco de deslizamentos de terra, para que gestores e colaboradores possam estar cientes das áreas mais vulneráveis em situações de deslizamento de terra.

Na distribuição de dados, Ryoo e Choi (2006) recomendam a utilização de protocolos de transmissão/recepção de dados conhecidos. Portanto, foi definido o uso do *padrão ISO 22351 - Message strucuture for exchange of Information.* Este padrão, além de ser multilingual, especifica atributos que uma notificação de desastre deve possuir, garantindo deste modo a qualidade da informação contida na aplicação, além de permitir a interoperabilidade entre qualquer organização que segue tal norma.

Para a apresentação dos dados foi selecionado o padrão *ISO 22327 - Guidelines for implementation of a comunity-based landslide early warning system,* que além de definir o modelo de interface gráfica de sistemas de aviso precoce de deslizamento de terra, esse padrão define a utilização de símbolos dos padrões *ISO 7001 – Graphical symbols – Public information symbols* e *ISO 7010 – Graphical Symbols – Safety colours and safety signs – Registered safety signs,* para representarem feições espaciais no mapa.

Segundo Ryoo e Choi (2006), uma das especificações do requisito de apresentação de dados, é a capacidade de um DIMS possuir um conjunto de métodos de recuperação de informação. Deste modo, foram selecionados os padrões WMS, WFS e SLD para auxiliar na recuperação de camadas e feições. Além disso, com tais padrões, é possível aumentar a interoperabilidade do sistema, uma vez que o mapa gerado pode ser replicado em outras aplicações que também seguem esses padrões.

Conforme mostrado na Figura 1, o framework *LapsusVGI*, está dividido em quatro módulos principais: fonte de dados; interface; armazenamento; e saída de dados.

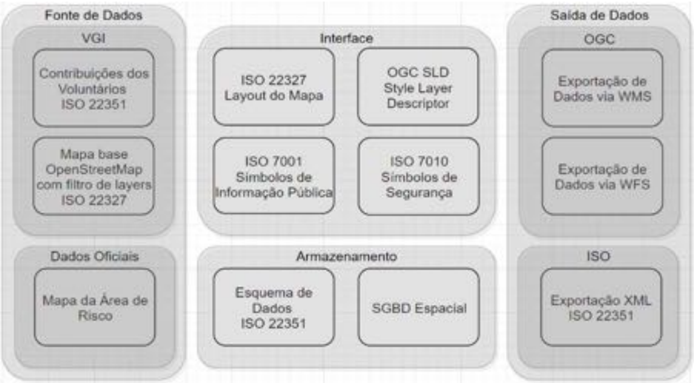

Fonte de Dados
VGI
Contribuições dos Voluntários ISO 22351
Mapa base OpenStreetMap com filtro de layers ISO 22327
Dados Oficiais
Mapa da Área de Risco
Interface
ISO 22327 Layout do Mapa
OGC SLD Style Layer Descriptor
ISO 7001 Símbolos de Informação Pública
ISO 7010 Símbolos de Segurança
Armazenamento
Esquema de Dados ISO 22351
SGBD Espacial
Saída de Dados
OGC
Exportação de Dados via WMS
Exportação de Dados via WFS
ISO
Exportação XML ISO 22351

Figura 1. Módulos e Componentes *LapsusVGI*

Como citado anteriormente, o framework tem duas fontes de dados: VGI e dados oficiais. Então, nesse módulo, as contribuições VGI devem seguir as especificações de atributos que são definidas na norma ISO 22351. Além disso, devido as especificações da norma ISO 22327, durante a construção do mapa base, é ideal manter somente camadas necessárias durante períodos de emergência.

Para o módulo de interface, que é responsável pela experiência visual do usuário, as normas ISO 7001 e ISO 7010 estabelecem cores e símbolos que devem ser utilizados na exibição das feições no mapa, enquanto o padrão SLD da OGC descreve como essa exibição deve ser realizada, para que as feições possam ser exportadas segundo as normas WFS e WMS, enquanto a ISO 22327 descreve o layout do mapa a ser exibido.

Na Figura 2 é mostrado um exemplo de mapa que segue as diretrizes da ISO 22327. Na parte superior da interface é exibido um título que indica a região que é retratada no mapa, na parte central da figura são exibidas as feições julgadas necessárias, no lado direito pode ser vista uma legenda contendo pontos de interesse e símbolos que podem ser reconhecidos facilmente pelo usuário, além de informações de contato que possam ser relevantes durante a emergência, por exemplo, autoridades e hospitais. Na parte inferior é exibido o nome dos chefes de família de cada moradia, juntamente com a referência para suas respectivas residências no mapa.

Devido à necessidade da manipulação de feições espaciais, o módulo de armazenamento é implementado com o uso de um SGBD espacial (MySQL). Para esta etapa foi elaborado um esquema conceitual de dados que engloba o esquema de dados definido pela ISO 22351.

O módulo de saída de dados é responsável pelas exportações e segue padrões da ISO e da OGC. O padrão ISO 22351 especifica o esquema XML de exportação, com os atributos e seus respectivos formatos durante a etapa de exportação. Os padrões WFS e WMS são usados para disponibilizar, por meio de serviços Web, os dados armazenados no framework para outras plataformas, garantindo maior interoperabilidade, uma vez que todos os dados podem ser recuperados via URL.

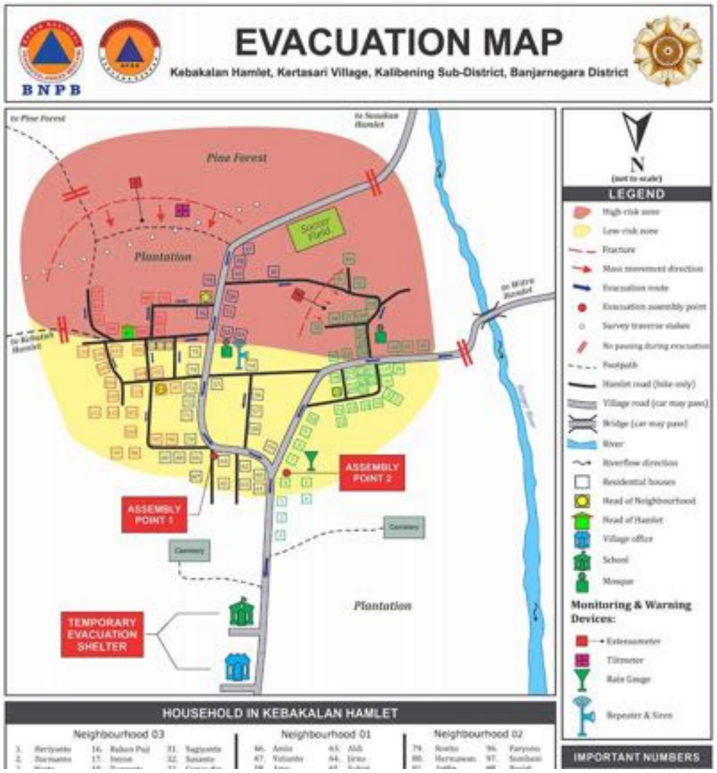


Figura 2. Layout do Mapa de Rotas e Evacuação conforme diretrizes ISO 22327
Fonte: Fathani, Karnawati e Wilopo (2016)

5. Conclusões e Trabalhos Futuros

Este artigo apresentou o projeto do *LapsusVGI*, um framework para implementação de DIMS focado em deslizamento de terra e rompimento de barragens. Esse projeto visa a implementação de um sistema padronizado e interoperável com outros sistemas, com base em padrões ISO e do OGC. Apesar do *LapsusVGI* ter sido projetado com o foco em deslizamentos de terra, o sistema pode ser adaptado para outros tipos de desastres.

Um *DIMS-LapsusTerrae* está sendo implementado no Departamento de Informática (DPI) da Universidade Federal de Viçosa (UFV), e trabalhos futuros incluem os seguintes desafios: ampliar o framework para trabalhar com o restante das especificações dos requisitos mencionados por Ryo e Choi (2006); ampliar as possibilidades de uso de VGI; além de adicionar funcionalidade para atender a gestão de recursos durante o desastre natural, por exemplo, gerenciamento de cadeia de suprimentos e a utilização de sensores físicos para auxiliar na predição de possíveis catástrofes.

Agradecimentos

Projeto parcialmente financiado pela Fapemig e CAPES.

Referências

Ellis, C. A., Gibbs, S. A. e Rein, G. (1991). Groupware: Some issues and experiences. *Communications of ACM*, 34: 39-58. DOI: https://doi.org/10.1145/99977.99987

Fathani, T. F., Karnawati, D. (2018). TXT-tool 2.062-1.1 A Landslide Monitoring and Early Warning System. *In: Sassa K. et al. (eds) Landslide Dynamics: ISDR-ICL Landslide Interactive Teaching Tools*. Pages: 297-308.

Fathani, T. F., Karnawati, D., Wilopo, W. (2016). An integrated methodology to develop a standard for landslide early warning systems. *In: Natural Hazards and Earth System Sciences*. Pages: 2123-2135.

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

Instituto Brasileiro de Geografia e Estatística – IBGE (2019). MUNIC 2017: 45,6 dos municípios do país foram afetados por secas nos últimos 4 anos. Disponível em: https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/21636-munic-2017-48-6-dos-municipios-do-pais-foram-afetados-por-secas-nos-ultimos-4-anos.

ISO - International Organization for Standardization (2007). ISO 7001:2007. Graphical symbols - Public information symbols.

ISO - International Organization for Standardization (2011). ISO 7010:2011. Graphical symbols - Safety colours and safety signs -Registered safety signs.

ISO - International Organization for Standardization (2018). ISO 22327:2018. Security and resilience - Emergency management - Guidelines for implementation of a community-based landslide early warning system.

ISO - International Organization for Standardization (2015). ISO/TR 22351:2015. Societal security - Emergency management - Message structure for exchange of information.

OGC - Open Geospatial Consortium. Web Map Service (2020). Disponível em: https://www.ogc.org/standards/wms.

Ryoo, J., Choi, Y. B. A comparison and classification framework for disaster information management systems. *International Journal of Emergency Management,* 3: 264-279

Tavares, J. F. et al. (2018). Systematic review on the use of groupware technologies in emergency management. *In proc. of the Third IFIP TC 5 DCITDRR Int. Conf. on Information Technology in Disaster Risk Reduction*, pages 22-35.

Ushahidi (2019). Ushahidi, https://www.ushahidi.com, December.

Geographical Complex Networks applied to describe meteorological data

Aurelienne A. S. Jorge[1], **Izabelly C. Costa**[1], **Leonardo B. L. Santos** [2]

[1]Instituto Nacional de Pesquisas Espaciais (INPE)
Cachoeira Paulista, SP – Brazil

[2]Centro Nacional de Monitoramento e Alertas de Desastres Naturais (Cemaden)
São José dos Campos, SP – Brazil

aurelienne.jorge@inpe.br, izabelly.costa@inpe.br, santoslbl@gmail.com

Abstract. *Complex Networks have been widely applied to climate data analysis, identifying relations and patterns in the atmosphere on a long-term scale. However, a few investigations have made use of Complex Networks to study meteorology (dealing with short-term changes in the atmosphere). With this in mind, the purpose of the present work is to make some progress in the spatial analysis of metrics in meteorological networks, specifically in precipitation events. We present some results for a study case comprising the Tamanduateí basin, in which we could analyze the spatial dependence intrinsic in the network structure.*

Resumo. *Redes Complexas têm sido largamente aplicadas na análise de dados climáticos, na tentativa de identificar relações e padrões na atmosfera a longo prazo. Algumas poucas pesquisas fizeram uso das Redes Complexas no estudo da Meteorologia (tratando de mudanças a curto prazo na atmosfera). Com base nisso, o presente trabalho tem o propósito de buscar algum avanço na análise espacial de métricas em redes meteorológicas, mais especificamente, em eventos de precipitação. Alguns resultados são apresentados para um estudo de caso compreendendo a bacia do rio Tamanduateí, nos quais foi possível analisar a dependência espacial intrínsica na estrutura da rede.*

1. Introduction

Based on Graph Theory, the study of Complex Networks represents a relevant contribution to science as a tool to describe the structure of a wide range of complex systems in nature and society, such as climate events [Barabási and Pósfai 2016]. In such a context, Complex networks have been applied to climate data analysis, aiming to identify structural patterns and teleconnections. Those researches use similarity measures such as Pearson correlation, event synchronization, or mutual information to construct the network connections. In terms of data, they are based on long time series of atmospheric variables, ranging from months to several years [Tsonis et al. 2006, Boers et al. 2019].

A few works have been held specifically in the weather domain, dealing with short-term changes in the atmosphere and manipulating spatial and temporal high-resolution data through complex networks. One of those few examples handled precipitation data from weather radar, and they achieved significant results in community detection based on a time series of only ten days, with 1km of spatial resolution [Ceron et al. 2019]. The behavior of topological metrics in meteorological networks is a characteristic that remains unknown.

With this in mind, the purpose of the present work is to make some progress in the spatial analysis of metrics in meteorological networks, specifically in precipitation events.

Due to climate changes, extreme precipitation events are becoming more frequent, with several impacts on society. Finding spatial patterns of precipitation events could represent a significant advance in atmospheric science and several applications, from health geography to resilient urban mobility [Santos et al. 2017].

2. Materials and Methods

2.1. Data

The case study presented here was held in São Paulo Metropolitan Region, specifically comprising the area of Tamanduateí basin, from January of 2015. Located on the Tiete river's left margin, the Tamanduatei basin has its source in the city of Mauá. It also crosses the towns of Diadema, São Caetano do Sul, besides the eastern and central zones of São Paulo [Ramalho 2007].

Due to its spatial and temporal high-resolution data, we used weather radar time series as our base dataset. Considering the mentioned study area, the weather radar located in the city of São Roque is the one that offers the best coverage, with a range of 250 kilometers. Its scans provide data with 1 kilometer of spatial resolution every 10 minutes [DECEA 2010]. The raw data is composed of a volume scan which contains scan values for different angles of elevation. For each of these elevation angles, an azimuth scan is performed, and such a scan is called Plan Position Indicator (PPI). The São Roque radar has 15 elevation angles, starting at 0.5 degrees to approximately 20 degrees [Redemet 2015].

For the first study cases, we used PPI data corresponding to the first level of elevation. These data are available in binary files with a grid of values. Such information is reflectivity value (dBZ), which we can convert into estimated rainfall rate. In summary, the higher the reflectivity value, the more intense is the estimated precipitation. As mentioned before, the selected time series comprises the entire month of January of 2015 with a temporal resolution of 10 minutes, so it is composed of more than 4400 scans in time, each one of them including 783 points in space.

2.2. Network Construction and Analysis

Spatial embedding is a physical property inherent in many phenomena modeled through networks, including the meteorological events addressed in this research. Therefore, we use here geographical graphs, which are graphs whose nodes have a known geographic location, and their edges contain an intrinsic spatial dependency [Santos et al. 2017].

We developed a tool to manage the input data and construct the network taking into account its geographical component. We called it GIS4Graph. It delivers output files with topological metrics calculated. One of these outputs is a shapefile, a file compatible with GIS platforms, and allows graph visualization in geographical space.

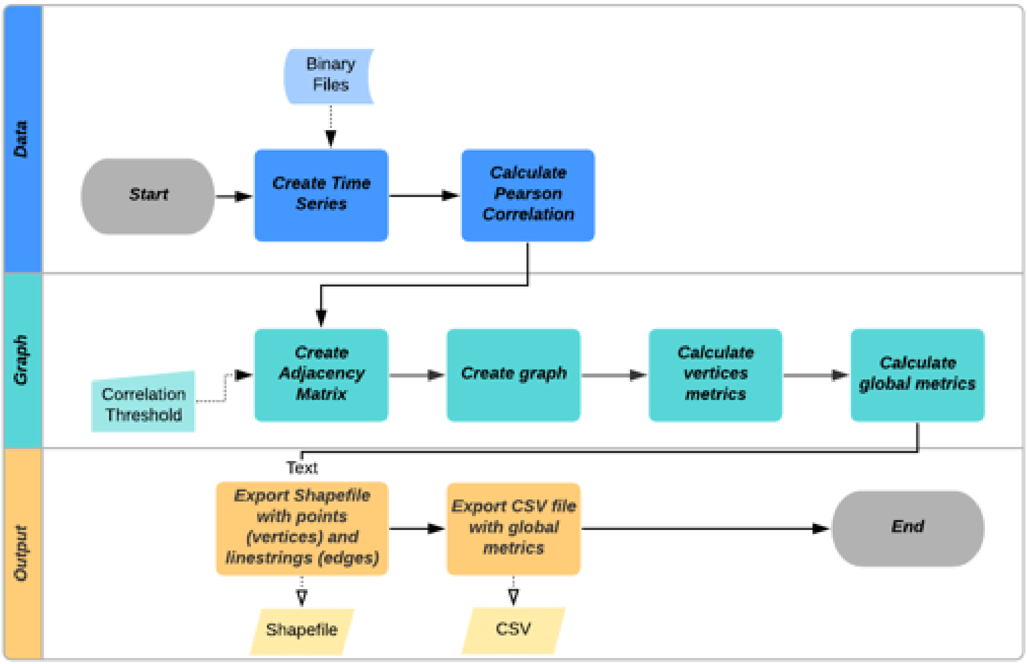

Data
Graph
Output
Binary Files
Start
Create Time Series
Calculate Pearson Correlation
Correlation Threshold
Create Adjacency Matrix
Create graph
Calculate vertices metrics
Calculate global metrics
Text
Export Shapefile with points (vertices) and linestrings (edges)
Export CSV file with global metrics
End
Shapefile
CSV

Figure 1. Graph4GIS flowchart

The application is composed of 3 main modules: Data, Graph, and Output. The developed flowchart is presented in Figure 1. The first module is responsible for dealing with the binary files provided by the weather radar. It reads all data and creates a time series for each grid point. Then, it calculates the Pearson correlation between each pair of them.

The second module deals directly with graph construction and metrics calculation. First of all, it generates a weighted adjacency matrix based on the Pearson correlation values - eliminating the ones under a predefined threshold. Next, it builds the graph with one node for each grid point and the edges with the weights indicated by the adjacency matrix. After that, it calculates the topological metrics, both global and nodes specific. Degree, clustering coefficient, and average shortest path are a few examples of them.

The results exportation is done by the third module, which delivers a shapefile with a set of points and lines, geographically representing the graph's nodes and edges. The metrics of each node appears as an attribute of the point in the shapefile. The application also generates a CSV file with all the global network metrics. The Output module also exports some charts to support auxiliary analysis.

We execute the application inputting different correlation thresholds, and we analyze the network diameter in each case. The final network is the one with the highest diameter metric, aiming to promote the best possible balance between removing the least relevant edges and keeping the most important ones - as applied in previous papers in the literature [Santos et al. 2019, Ceron et al. 2019]. The threshold in which the network achieves the highest diameter value is called a critical threshold.

3. Results

Before analyzing the network built for the mentioned case study, we can observe the spatial dependence inherent in such data on Figures 2 and 3. The first one shows how the (temporal) correlation between the (time series associated with each) pairs of points is related to the geographical (euclidean) distance between them. We grouped correlation values into three categories - minimum, medium, and maximum - respectively colored in red, green, and blue. For the red group, it is possible to see a spatial dependence up to approximately 3 km. Regarding medium and maximum categories, the temporal correlation is considerably high between 1 and 10 km of distance, but we can still observe the influence of spatial dependence until 20 km.

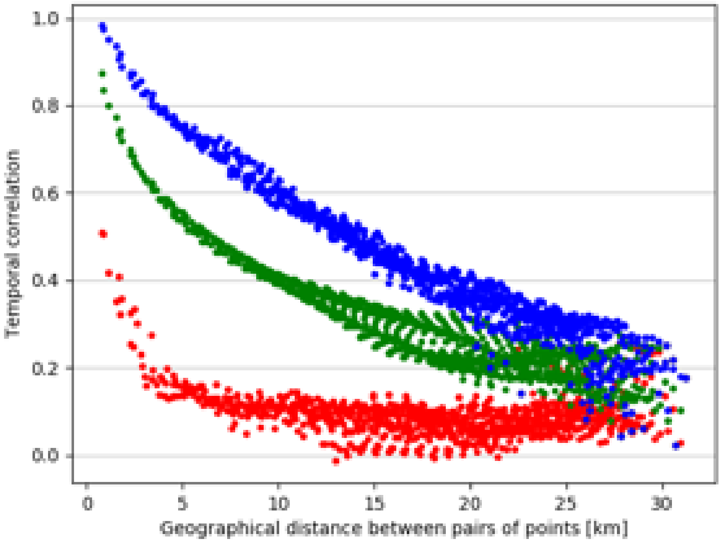


Figure 2. Temporal Correlation versus Geographical distance between each pair of points. Correlation values are grouped into three categories - minimum, medium and maximum values - respectively coloured in red, green and blue.

We can also notice that the minimum correlations for the geographically nearest ten pairs of points are even higher than the maximum correlations for those more distant than 28 km. Such property is an indicator of how well-behaved the relation between temporal correlations and geographical distances in this network structure.

The scatter plot on Figure 3 presents the relation between the euclidean distance and the topological distance between each pair of nodes - the network path with the shortest number of edges between those nodes. We can verify that there is strong linearity in such relation, with a correlation coefficient (R^2) equals to 0.767 and a slope of 1.16. Such a slope value indicates that as the geographical distance increases, the impact is even more significant on the topological distance.

This chart also shows the largest edge in the network (2.5 km), indicated by the maximum geographical distance for the pairs of points within a topological distance of 1 edge. Therefore, there are no pairs of points directly connected in a distance greater than

2.5 km. On the other hand, there are very close nodes, geographically neighbors, but with a high topological distance, up to 12 edges.

The geographical network built up by graph4GIS is introduced in Figure 4. It used a threshold of 0.86, which was the critical threshold for our study case. This output allows us to visualize the structure of network connections spatially.

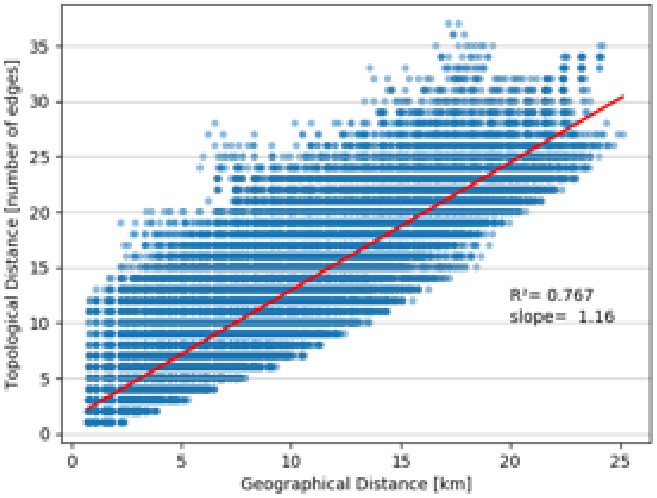


Figure 3. Topological distance versus Geographical (euclidean) distance

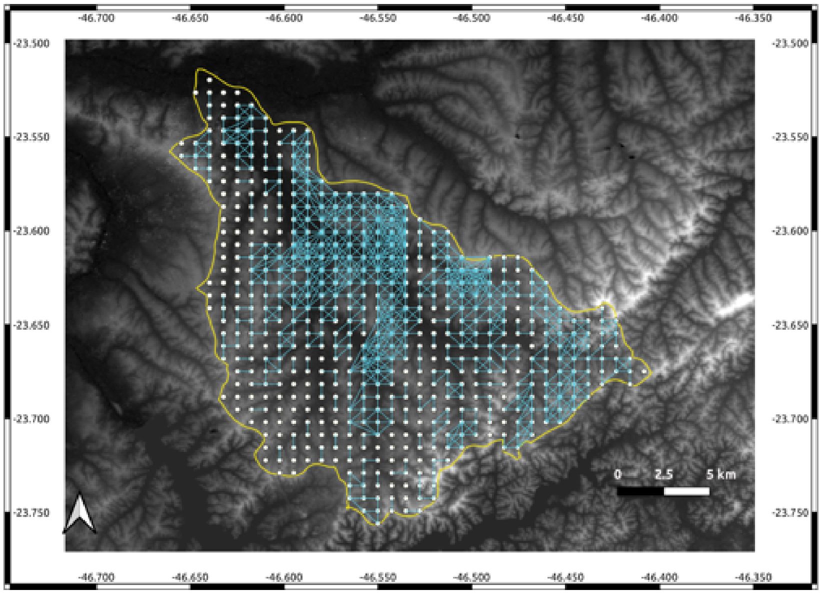


Figure 4. Geographical network for Tamanduateí Basin. The white points represent the nodes, the blue segments are the edges of the network, and the yellow border is the outline of the basin.

4. Final Considerations

This work applied Complex Networks in the study of meteorological networks, aiming to explore topological metrics' behavior in such a context. Based on precipitation time series, this paper introduced some spatial analysis of the system's topological structure.

As a result, we could identify the spatial dependence of temporal correlations, such as the linearity in the relation between the topological and geographical distances between different pairs of points in a hydrological basin. We were also able to verify some peculiarities in the network, such as the maximum geographical length of an edge (2.5 km) and a high maximum topological distance between neighboring nodes (11 edges on the shortest path between nodes closer than 1km to each other).

In future works, we would like to analyze datasets for specific meteorological processes to identify spacial and topological signatures. Besides, we intend to approach larger study areas, including the entire São Paulo Metropolitan Region, and other graph measures, such as degree, clustering coefficient, betweenness, and diameter.

References

[Barabási and Pósfai 2016] Barabási, A.-L. and Pósfai, M. (2016). *Network science*. Cambridge University Press, Cambridge.

[Boers et al. 2019] Boers, N., Goswami, B., Rheinwalt, A., Bookhagen, B., Hoskins, B., and Kurths, J. (2019). Complex networks reveal global pattern of extreme-rainfall teleconnections. *Nature*, 566(7744):373–377.

[Ceron et al. 2019] Ceron, W., Santos, L. B. L., Dolif Neto, G., Quiles, M. G., and Candido, O. A. (2019). Community Detection in Very High-Resolution Meteorological Networks. *IEEE Geoscience and Remote Sensing Letters*, pages 1–4.

[DECEA 2010] DECEA (2010). Ministério da defesa comando da aeronáutica estado-maior da aeronáutica. *Ministério da Defesa Comando da Aeronáutica*, pages 1–24.

[Ramalho 2007] Ramalho, D. (2007). Rio Tamanduateí - nascente à foz: percepções da paisagem e processos participativos. *Paisagem e Ambiente*, (24):99.

[Redemet 2015] Redemet (2015). Redemet. `https://www.redemet.aer.mil.br`.

[Santos et al. 2019] Santos, L. B. L., Carvalho, L. M., Seron, W., Coelho, F. C., Macau, E. E. N., Quiles, M. G., and Monteiro, A. M. V. (2019). How do urban mobility (geo)graph's topological properties fill a map? *Appl Netw Sci*, 4(91).

[Santos et al. 2017] Santos, L. B. L., Carvalho, T., Anderson, L. O., Rudorff, C. M., Marchezini, V., Londe, L. R., and Saito, S. M. (2017). An rs-gis-based comprehensiveimpact assessment of floods—a case study in madeira river, western brazilian amazon. *IEEE Geoscience and Remote Sensing Letters*, 14(9):1614–1617.

[Santos et al. 2017] Santos, L. B. L., Jorge, A. A. S., Rossato, M., Santos, J. D., Candido, O. A., Seron, W., and de Santana, C. N. (2017). (geo)graphs - Complex Networks as a shapefile of nodes and a shapefile of edges for different applications. 321124491(November).

[Tsonis et al. 2006] Tsonis, A. A., Swanson, K. L., and Roebber, P. J. (2006). What do networks have to do with climate? *Bulletin of the American Meteorological Society*, 87(5):585–595.

MobilityHelp: Uma Ferramenta para Análise de Dados no Transporte Público Urbano

José Ivan S. da Cruz Júnior[1], **Claudio E. C. Campelo**[1]

[1]Universidade Federal de Campina Grande (UFCG)
Departamento de Sistemas e Computação
Campina Grande – PB – Brasil

jose.ivan.junior@ccc.ufcg.edu.br, campelo@dsc.ufcg.edu.br

Abstract. *With the increasing growth of urban centres and the demand for public transport, there has been a considerable increase in the demand for tools that help analyse data to better understand the dynamics of the transport system. In this article, we propose MobilityHelp, a software tool, implemented in the R language, which offers a set of useful functions for analysing public transport data, including: spatial analysis of travel origins and destinations; analysis of bus capacity by route; and detection of outliers in the speed of the trips made. The tool has been evaluated through a case study in the city of Curitiba, Brazil. We believe the knowledge extracted by this and other similar tools can contribute to the improvement of the service offered to citizens in different cities.*

Resumo. *Com o crescimento cada vez maior dos centros urbanos e da demanda pelo transporte público, cresceu o interesse por ferramentas que ajudem na análise de dados para melhor compreender da dinâmica do sistema de transporte. Neste artigo, propomos a MobilityHelp, uma ferramenta de software, implementada na linguagem R, que oferece um conjunto de funções úteis para análise de dados de transporte público, incluindo: a análise espacial das origens e destinos das viagens; a análise da lotação dos ônibus por rota; e a detecção de outliers na velocidade das viagens realizadas. A ferramenta foi avaliada através de um estudo de caso na cidade de Curitiba, Basil. Acreditamos que o conhecimento extraído por esta ferramenta e outras similares pode contribuir para melhorar o serviço oferecido a cidadãos em diferentes cidades.*

1. Introdução

Este trabalho tem como objetivo aplicar técnicas de ciência de dados e visual analytics a dados do transporte público, a fim de desenvolver um ferramental útil a desenvolvedores de software para gestão de transporte público, visando facilitar a exploração e análise deste tipo de dado por gestores e tomadores de decisão. Para validação da ferramenta proposta, foi conduzido um estudo de caso com dados de transporte público da cidade de Curitiba (Paraná, Brasil).

Diante da diversidade de análises que podem ser implementadas e propostas, saber quais delas serão, de fato, relevantes para os gestores, pesquisadores e usuários do transporte público em geral é um desafio considerável. Diante disso, três análises são implementadas e propostas: análise espacial das origens e destinos das viagens; análise da lotação dos ônibus das rotas em qualquer hora do dia; e a detecção de outliers em relação à velocidade das viagens realizadas, com foco da detecção de viagens mais lentas.

2. Metodologia

A metodologia adotada para analisar os dados é o KDD (Knowledge-Discovery in Databases)[FAYYAD 2020], que é um processo de extração de informações de base de dados. Esse método consiste na imersão no domínio da aplicação para compreendê-lo de uma forma mais eficiente.

Durante a etapa de processamento e análise dos dados, foi utilizada a plataforma de desenvolvimento RStudio. Nela o desenvolvimento se deu principalmente utilizando a linguagem de programação R, que é voltada para análise estatística e criação de visualizações de dados. Adicionalmente, utilizou-se a linguagem de marcação Markdown, uma linguagem simples de marcação que possibilita a transformação das análises em relatórios.

Os dados utilizados no estudo de caso foram produzidos por Braz [BRAZ 2019], que desenvolveu uma Matriz Origem-Destino a partir dos dados de bilhetagem da cidade de Curitiba-PR [BRAZ 2019]. Os dados das viagens utilizados nas análises correspondem ao período de 01/05/2017 até 17/07/2017. No total, a base de dados contém 4169274 viagens e 246 rotas diferentes registradas.

Para o pré-processamento dos dados, é desenvolvido um script na linguagem de programação estatística R. Através dele, obtém-se as seguintes informações por viagem: *duração*, *quantidade total de viagens*, *distância percorrida*, *velocidade*, *dia da semana* e o *código do ônibus* da viagem realizada.

O procedimento de transformação das variáveis consiste nas seguintes etapas:

- Para obtermos a *duração mediana das viagens*, processamos as informações do horário de embarque e desembarque;
- Processamos a informação de cada viagem individualmente para calcular a *quantidade total de viagens*. Cada viagem era uma linha no dataframe. Logo, agregando por rota, obtivemos a quantidade total;
- A *distância percorrida* (em quilômetros) foi calculada a partir das coordenadas (latitude e longitude) do embarque e desembarque e, a partir desses dados, foi calculada a mediana da distância.
- Para a *velocidade*, usamos a distância percorrida (em quilômetros) e a duração da viagem (em hora);
- O *código do ônibus* onde a viagem foi realizada foi obtido através do seu identificador.

A etapa de pré-processamento dos dados inclui ainda uma atividade de filtragem. onde rotas com pouca quantidade de viagens por dia são excluídas. Para o estudo de caso conduzido, foram excluídas as rotas com menos de 10 viagens por dia. Este limiar pode ser ajustado para outras análises.

3. Análises e Resultados

3.1. Análise espacial de origens e destinos finais

A primeira análise proposta diz respeito a distribuição das origens e destinos finais das viagens. Ou seja, essa análise possibilita a visão de quais são os lugares da cidade mais demandados e em quais locais as pessoas mais embarcam. Essa análise é realizada de acordo com os horários escolhidos pelo pesquisador, gestor ou usuário. Assim, o usuário da ferramenta pode analisar a distribuição das origens e destinos de acordo com o dia e horário do seu interesse. Vale ressaltar que a análise é a de origens e destinos finais do passageiro, Isto é, todas as viagens intermediárias que um passageiro faz até chegar seu destino, passando por diferentes pontos ou terminais de ônibus, não são considerados. Assim, é possível observar, onde, de fato, os passageiros embarcaram em seu destino inicial e onde desembarcaram para o seu destino final.

Para a faixa de horário com mais viagens em Curitiba, das 6h às 8h, como mostra a Figura 1, o embarque mostra-se bem distribuído em bairros periféricos e distantes do centro, onde geralmente se encontram bairros residenciais. Constata-se também, pela Figura 2, que, no mesmo horário, o desembarque se mostra mais acentuado e concentrado na região central da cidade. Por outro lado, na faixa de horário das 17h às 19h, o desembarque se acentua na região periférica da cidade (Figura 4) e, o embarque, na região central (Figura 3). Isso indica que na faixa de horário da manhã as pessoas tendem a sair de seus bairros em direção a região Central e bairros comerciais da cidade para trabalhar, estudar, etc, e, a noite, voltam para as suas casas.

Observar a distribuição do destino das viagens mostra o grau da demanda de ônibus para determinadas regiões da cidade. Logo, saber que há muitas pessoas desejando ir para determinadas regiões, em algum horário do dia, auxiliará os gestores a determinar quando a oferta de ônibus deverá ser maior para os destinos mostrados.

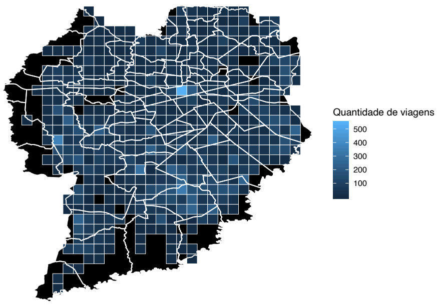


Figure 1. *Embarque* das viagens das 6:00 às 8:00

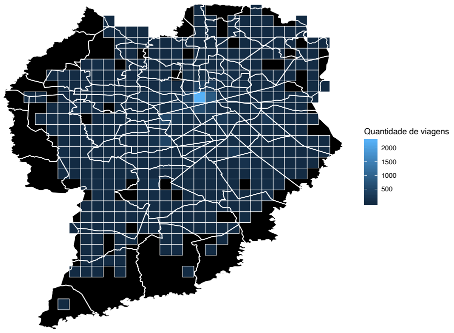


Figure 2. *Desembarque* das viagens das 6:00 às 8:00

3.2. Lotação dos ônibus por dia, horário e rota

Uma realidade do transporte público é a sua propensão a atingir o limite da lotação em determinados horários e itinerários. Em muitos momentos, devido a isso, os usuários demoram mais tempo do que planejaram para fazer uma viagem e os gestores precisam lidar com maiores desafios de gestão em seus sistemas de transporte.

Uma outra análise proposta pelo presente trabalho é prover um panorama sobre a lotação dos ônibus seja qual for a linha, o horário ou dia pretendido. Sendo assim, o usuário da ferramenta pode verificar a lotação dos ônibus de uma determinada rota em qualquer dia ou horário que for do seu interesse, sendo possível verificar a quantidade de passageiros em cada ônibus que esteve fazendo viagem para uma linha específica em qualquer horário pretendido. Ressalta-se que a análise não é realizada indicando a quantidade

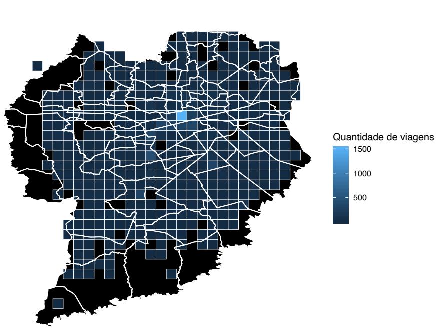


Figure 3. *Embarque* das viagens das 17:00 às 19:00

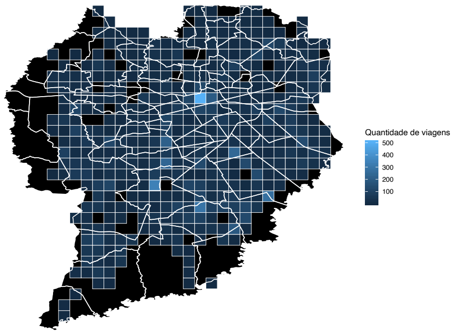


Figure 4. *Desembarque* das viagens das 17:00 às 19:00

de passageiros viajando naquele exato momento, mas sim aqueles que fizeram check-in na rota no horário espefificado.

Para Curitiba, em um exemplo de aplicação da ferramenta para a rota 303 na faixa de horário de 18h às 19:59min do dia 02/05/2017, observa-se na Figura 5 que, no geral, os ônibus seguem uma média de quantidade de passageiros parecida no decorrer do horário, com a excessão de dois ônibus que apresentam uma quantidade de passageiros transportados bem maior que os demais, chegando a ter mais de 230 passageiros embarcando nos ônibus durante a faixa de horário.

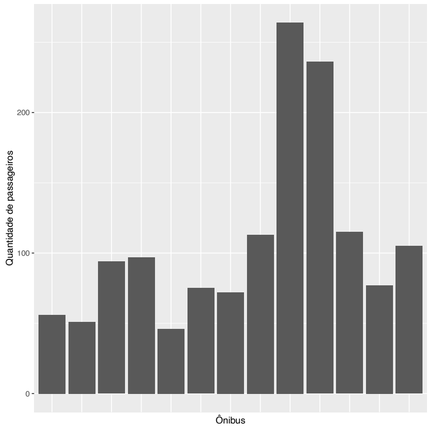


Figure 5. *Quantidade de passageiros* transportados nos ônibus na linha 303 de 18:00min às 19:59min do dia 02/05/2017

O gestor, ao observar a distribuição temporal dos ônibus e a quantidade de passageiros transportados, terá uma forma mais fácil de fiscalizar a lotação por horário, obtendo maior capacidade de planejar e dispor os resursos demandados para o oferecimento de um serviço de melhor qualidade.

3.3. Detecção de outliers (viagens lentas)

Um outlier é um valor que foge da normalidade e que pode causar anomalias nos resultados obtidos por meio de algoritmos e sistemas de análise. Em diversos cenários, os dados são tantos que realizar o processamento de todo o conjunto disponível é impraticável ou até mesmo indesejável. Assim, métodos capazes de selecionar aqueles dados com alto grau de distinção em meio a todo esse volume despertam grande interesse. [RODRIGUES 2018]

Diante disso, o presente trabalho sugere uma ferramenta de detecção de outliers objetivando buscar as viagens mais lentas realizadas. A intenção é dispor também ao usuário da ferramenta uma possibilidade de pré-processamento dos dados em função daqueles que fogem do comportamento esperado.

Nos dados de Curitiba-PR, podemos ver na Figura 6 que o padrão encontrado na relação entre distância percorrida e duração da viagem é diretamente proporcional. Isso quer dizer que, quanto maior for a distância da viagem, mais tempo o usuário demorará para chegar a seu destino. Nosso objetivo é analisar as viagens e, consequentemente, as rotas de ônibus que sistematicamente fogem desse padrão e apresentam velocidades baixas. Viagens e rotas de ônibus que são sistematicamente lentas podem ser alvo de ações estruturantes ou análises mais profundas quanto ao itinerário. O critério utilizado para definir uma viagem como lenta foi a sua distância da nuvem de dados da Figura 6 que concentra a maior quantidade de viagens.

A ferramenta concede ao usuário as seguintes possibilidades de detecção dos outliers:

- Observar a quantidade de viagens lentas por dia da semana, mostrando assim quais os dias da semana onde as viagens tendem a ser mais lentas (Figura 7);
- Escolher uma data específica e ver quais rotas apresentaram a maior quantidade de viagens lentas. A data escolhida foi 10/05/2017(Figura 8);
- Escolher uma rota específica e observar a quantidade de viagens lentas nos dias da semana. A rota escolhida foi a 370 (Figura 9).

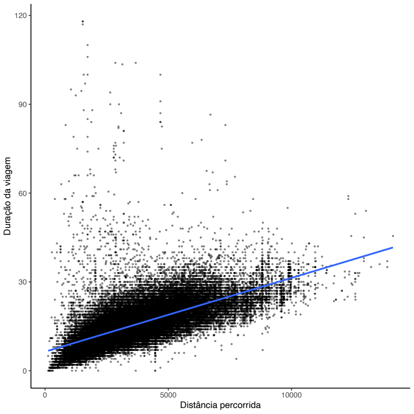


Figure 6. *Relação* da distância percorrida (km) e duração das viagens (minutos) de todos os dados da base de viagens.

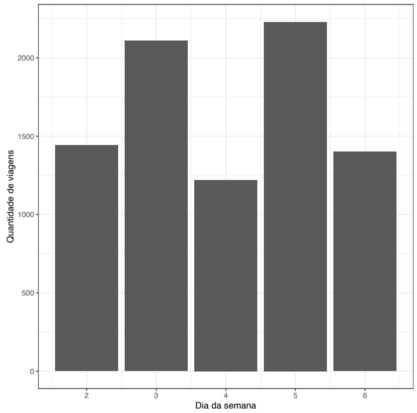


Figure 7. *Quantidade* de viagens lentas por dia da semana.

A ferramenta concede ao usuário uma maior capacidade de perceber onde há a necessidade de serem feitas correções, além de entender o funcionamento do transporte

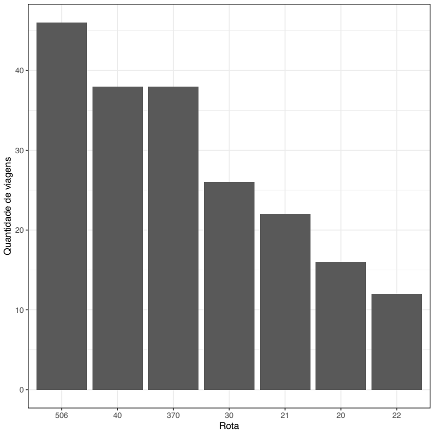


Figure 8. *Quantidade* de viagens lentas por rota através da data (10/05/2017)

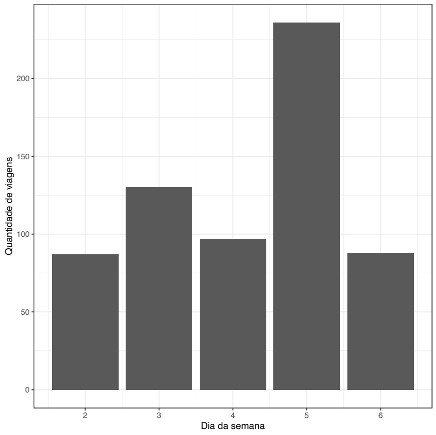


Figure 9. *Quantidade* de viagens lentas por dia da semana através da rota (370)

nos dias atípicos, ou seja, aqueles onde grandes eventos são realizados na cidade, exigindo uma dinâmica diferente da oferta de ônibus e da demanda de viagens em um horário específico.

4. Conclusão

Para a implementação da ferramenta, foi escolhida a linguagem de programação R tanto pela grande popularidade no desenvolvimento de análises, manipulação, quanto facilidade na codificação, legibilidade e reutilização em projetos de análises de dados. O desenvolvimento se deu pela escolha de três análises que pudessem ser aplicadas em qualquer contexto de pesquisa no transporte público.

O maior desafio encontrado foi conceber e desenvolver análises que fossem realmente relevantes para os gestores e aqueles que trabalham com pesquisa no transporte público, em vistas da base de dados disponível. Possíveis aprimoramentos da ferramenta incluem: expansão da detecção dos outliers para a demanda de locais de destino; visualizações mais avançadas para as análises espaciais; visualizações elaboradas para as análises de lotação dos ônibus; e análise da lotação dos ônibus em tempo real.

References

BRAZ, T. (2019). Inferring passenger-level bus trip traces fromschedule, positioning and ticketing data: Methodsand applications. *Universidade Federal de Campina Grande (UFCG).*

FAYYAD, U. M. (2020). “from data mining to knowledge discovery: an overview”. pages 1–34.

RODRIGUES, R. D. (2018). Detecção de outliersbaseada em caminhada determinística do turista. *Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto/USP.*

Variabilidade temporal do uso e cobertura da terra em escala global a partir de dados ESA CCI-LC

Lorena de Moura Joia Gomes[1], Isadora Haddad Ruiz[1], Gilberto Ribeiro de Queiroz[2], Thales Sehn Korting[2]. Douglas F. M. Gherardi[3]. Lênio S. Galvão[3]

[1] Divisão de Observação da Terra e Geoinformática – Instituto Nacional de Pesquisas Espaciais (INPE)
Caixa Postal 12.227 – 010 – São José dos Campos, SP – Brasil

{lorena.gomes; isadora.ruiz; Gilberto.queiroz; thales.korting;
douglas.gherardi; lenio.galvao}@inpe.br

Abstract. *Geospatial information increasingly allows data in different variations of time and space, promoting expansion of intergovernmental actions. In this context, the objective of this work was to analyze data from the ESA CCI-LC (Climate Change Initiative - Land Cover) product, from 2000 to 2015, using the Python programming language to verify land cover changes. The results indicated that classes such as water, permanent ice and sparse vegetation did not present significant variations in the analyzed period, while urban presents an increase and other classes of use (reduction and increase). In order to complement this analysis, we suggested new approaches to make the code more efficient for handling the CCI-LC product.*

Resumo. *Informações geoespaciais tem permitido cada vez mais o acesso a dados em diferentes escalas de tempo e espaço, o que amplia ações intergovernamentais. Neste contexto, o objetivo do trabalho foi analisar dados do produto global CCI-LC (Climate Change Initiative - Land Cover) da ESA, no período de 2000 a 2015, através da linguagem de programação Python, para verificar as mudanças de cobertura da terra. Os resultados indicam que classes como Água, Gelo permanente e Vegetação esparsa não apresentaram variações significativas de cobertura no período analisado. Por outro lado Urbano apresentou aumento de cobertura. De forma a complementar esta análise, o trabalho indica novas abordagens para tornar o código mais eficiente para manipulação do produto CCI-LC.*

1. Introdução

O monitoramento das mudanças no uso e cobertura da terra se tornou demanda crescente no meio técnico científico [Li et al. 2018], permitindo diagnosticar alterações nos ecossistemas naturais em escala global. Dessa forma, tecnologias de observação da terra, como o sensoriamento remoto orbital, permitem a coleta de dados em diferentes resoluções espaciais (regional e global) e temporais (mensal e anual) [Almeida et al. 2016].

Desde o ano 2000 diversos produtos de cobertura da terra (*Land Cover* - LC), com enfoque em áreas de pesquisas como suprimento de alimentos, mudanças climáticas e recursos hídricos, estão sendo desenvolvidos [Grekousis, Mountrakis e Kavouras 2015] Dentre eles, destaca-se o programa *Climate Change Initiative* (Iniciativa sobre Mudança Climática - CCI) da *European Space Agency* (Agência Espacial Europeia - ESA) [Grekousis, Mountrakis, Kavouras, 2015, Hua et. al. 2018]. Este programa foi criado, a partir da integração de dados de diferentes satélites, para atender demandas científicas sobre as mudanças climáticas globais, usando séries anuais de 1992 a 2015.

O presente trabalho tem por objetivo apresentar uma análise, em escala global, das mudanças de cobertura da terra comparando-se os anos de 2000, 2005, 2010 e 2015, a partir dos dados CCI-LC. Essas análises serão realizadas com ferramentas de software livre geoespaciais no ambiente *Python*. As mudanças ao longo do tempo e o código utilizado no processamento, serão disponibilizados para toda a comunidade no GitHub.

2. Materiais e métodos

O produto empregado no estudo foram os mapas anuais de cobertura global produzidos pelo CCI-LC da *European Space Agency* derivados da integração de cinco sensores imageadores, com datas e resoluções espaciais respectivamente: AVHRR, 1992-1999, 1km; SPOT, 1999-2013, 1km; MERIS, 2003-2012, 300m; PROBA-V, 2014-2015, 1km. A resolução espacial do produto CCI-LC é de 300 metros, disponíveis em 24 mapas para o período de 1992-2015 em diversos formatos. Estes mapas apresentam 22 classes temáticas de cobertura da terra, estabelecidas com base na classificação do IPCC (*Intergovernmental Panel on Climate Change*) [Defourny et al. 2017]. O acesso aos dados e documentação é possível pela plataforma ESA/CCI Viewer[1].

Para o desenvolvimento do trabalho foram utilizados mapas anuais de cobertura de todo o planeta no formato GeoTiff referentes aos anos 2000, 2005, 2010 e 2015. O intervalo de 5 anos foi adotado para maximizar a detecção de ao menos uma mudança no intervalo, pois alterações de classes são detectadas se persistirem por mais de dois anos [Defourny et al. 2017] e o mapeamento teve início em 1992. Em sequência o processamento foi realizado na plataforma Jupyter do pacote Anaconda. Por ser uma linguagem de programação flexível e amplamente utilizada atualmente, as etapas de processamento consistiram em estruturar códigos de linguagem de programação *Python* capazes de validar os metadados quanto ao sistema de coordenada de referência e identificar as mudanças das classes temáticas de cobertura da terra nas matrizes. Para a manipulação e processamento dos dados foram utilizadas bibliotecas *GDAL*, *Numpy*, *Pandas*, *Matplotlib* e *Plotly.*

As imagens foram reclassificadas por meio do agrupamento das classes pré-existentes para ressaltar as grandes classes de interesse. Assim, as classes relacionadas a agricultura foram agrupadas na classe Agricultura (A), as de florestas foram agrupadas na classe Floresta (F), e assim sucessivamente para as demais categorias, reduzindo as 22 classes para 11 classes finais (Tabela 1).

A análise das mudanças percentuais das diferentes classes ao longo do tempo, foram realizadas em linguagem *Python*, por meio de uma matriz 3D, comparando as classes nos quatro anos estudados (2000, 2005, 2010 e 2015) e agrupando-as, para detectar e quantificar as alterações no uso e cobertura da terra ao longo dos anos estudados.

Tabela 1. Reclassificação do uso e cobertura da terra

Classes (adaptado)	*Classes de cobertura da terra*	*Classes CCI-LC*
0	Sem dados (N)	0
1	Agricultura (A)	10, 11, 12, 20, 30, 40
2	Floresta (F)	50, 60, 61, 62, 70, 71, 72, 80, 81, 82, 90, 100, 110, 160, 170
3	Prado, estepe e savana (G)	130

[1] http://maps.elie.ucl.ac.be/CCI/viewer/index.php

4	Áreas alagadas (L)	180
5	Urbano (U)	190
6	Vegetação arbustiva (B)	120, 121, 122
7	Líquens e musgos (M)	140
8	Vegetação esparsa (P)	150, 151, 152, 153
9	Área descoberta (D)	200, 201, 202
10	Água (W)	210
11	Gelo permanente (S)	220

3. Resultados

O percentual anual de cobertura para cada imagem analisada entre 2000 e 2015 para algumas classes apresentou pequena ou nenhuma variação ao longo do tempo, dentre elas, as áreas de S, cerca de 10%, de cobertura, áreas ocupadas por F (aproximadamente 8%), A (cerca de 4%) e D (média de 3%). As outras classes, como B e G ocupam em média aproximadamente 2% do globo e as demais classes ocupam uma pequena extensão, abaixo de 2%. Na Tabela 2 são apresentadas detalhadamente as porcentagens de ocupação de cada classe analisada na área, conforme os anos estudados (2000, 2005, 2010 e 2015).

Tabela 2. Porcentagens de cada classe para a área de estudo referente a cada ano estudado (2000, 2005, 2010 e 2015)

Classes de cobertura da terra	*Percentual anual (%)*			
	2000	*2005*	*2010*	*2015*
Sem dados (N)	0	0	0	0
Agricultura (A)	3,9297	3,9508	3,9535	3,9511
Floresta (F)	7,9451	7,9476	7,9578	7,9416
Prado, estepe e savana (G)	2,0960	2,1072	2,1091	2,1144
Áreas alagadas (L)	0,4200	0,4049	0,4004	0,3998
Urbano (U)	0,0701	0,0928	0,1053	0,1196
Vegetação arbustiva (B)	2,1057	2,0936	2,0922	2,0992
Líquens e musgos (M)	1,9301	1,9135	1,9111	1,9123
Vegetação esparsa (P)	0,5054	0,5054	0,5054	0,5054
Área descoberta (D)	3,0367	3,0280	3,0140	3,0054
Água (W)	67,5845	67,5794	67,5743	67,5744
Gelo permanente (S)	10,3768	10,3768	10,3768	10,3768

Como na análise não houve a remoção da classe W, esta foi identificada como predominante e ocupa a maior parte do globo (média de 67,58%), como já é de conhecimento geral, e a classe N não foi verificada em nenhum dos anos. Logo, a Figura 2 exibe os percentuais das áreas ocupadas por cada classe de cobertura da terra referente a cada ano, desprezando a apresentação da classe W e N já que não houve alterações ao longo dos anos.

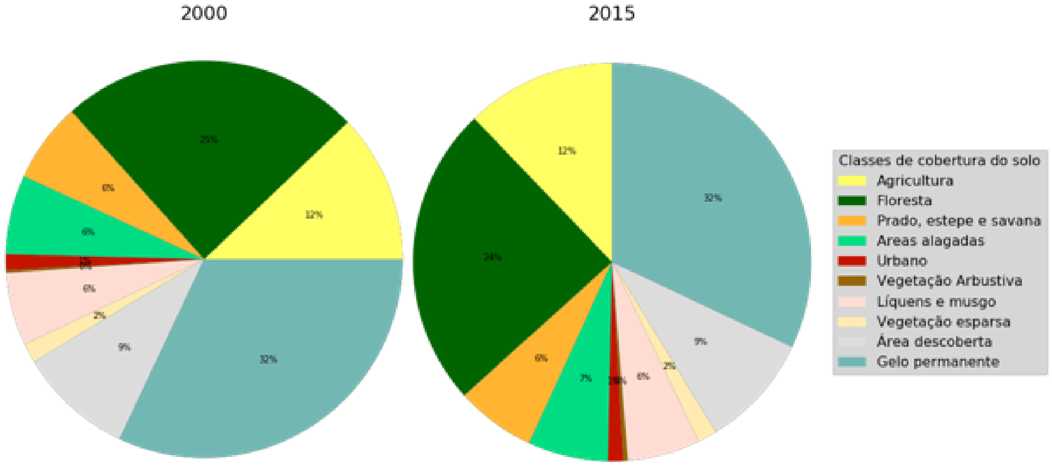


Figura 1. Percentual das classes de uso e cobertura da terra global referente aos anos 2000 e 2015

Observa-se que as porcentagens são quase regulares, indicando que as classes não sofreram grandes variações ao longo dos 15 anos analisados. No entanto, destacam-se apenas as classes de gelo permanente e vegetação esparsa que efetivamente permaneceram constantes ao longo dos anos estudados. Ainda sobre as mudanças entre os anos, a Figura 2 apresenta a variabilidade no período, indicando aumento e redução das classes entre os anos.

Como é possível observar, as oscilações apresentam baixa variabilidade entre -0,02 e 0,02. Entre as classes temáticas, a classe Urbano (U) e Prado, estepe e savana (G) são as únicas que apresentam aumento em todos os anos. As demais aumentam ou diminuem no período.

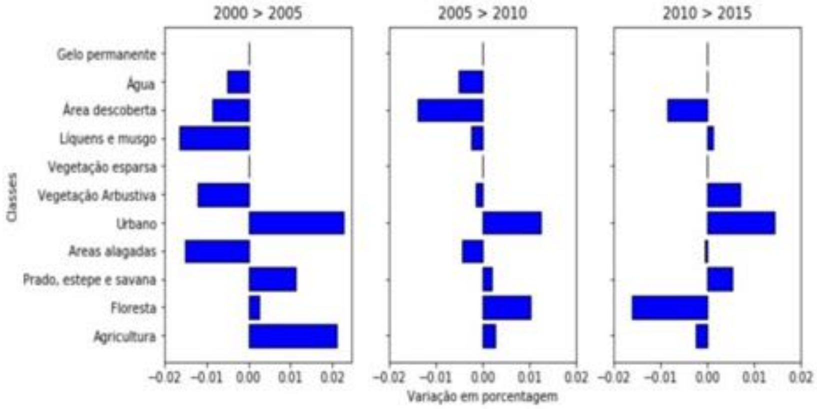


Figura 2. Percentual de variação das classes entre os anos (2000 para 2005, 2005 para 2010 e 2010 para 2015)

4. Discussão

Como já apontado anteriormente, a análise demonstrou que houve uma estreita variação entre quase todas as classes ao longo dos anos. No entanto, vale destacar que como a análise é global, ela não possui uma riqueza de detalhes. Consequentemente os resultados são dados gerais sobre o uso e cobertura da terra.

Em virtude da escala de análise e devido a resolução espacial média do produto CCI-LC, o nível de detalhamento para determinadas classes não foi garantido. Portanto é necessário atenção quanto a utilidade destes resultados para aplicações específicas.

Para uma melhor análise das variações, a Figura 2 exibe as transições entre os anos de 2000-2005, 2005- 2010 e 2010-2015. Pode-se verificar que a redução e o aumento de cobertura mostraram pouca variação ao longo dos anos, apresentando-se entre ± 0,02% da média. Os resultados desta análise indicaram, portanto, que a classe de Agricultura (A) apresentou redução apenas nos últimos cinco anos (2010 > 2015), assim como Floresta (F), que mostrou maior redução de 2010 para 2015 quando comparada aos demais anos.

A classe Urbano (U), por sua vez, aumentou em cobertura no período analisado, junto com a classe Prado, estepe e savana (G). Gelo permanente (S) e Vegetação esparsa (P) não mostraram mudanças significativas no período, apresentando oscilações em cobertura nos 15 anos analisados. Salienta-se que a classe Água (W) possui uma relação direta com as águas oceânicas. Para análises específicas sobre águas interiores seria necessário delimitar o processamento apenas para áreas continentais.

O período de 2000 a 2005 apresentou maiores variações na cobertura da terra, que se reduziu nos demais anos. No intervalo de 2005 a 2010 as variações de cobertura diminuíram para Líquens e musgos (M), Áreas alagadas (L) e Agricultura (A), ocorrendo aumento da classe Floresta (F) e Urbano (U). No período seguinte, 2010 a 2015, as variações reduziram ainda mais, não mostrando alterações nas classes de Água (W), com alguma alteração de cobertura na classe Áreas alagadas (L). Em contrapartida, Floresta (F) apresentou redução de cobertura.

5. Conclusão

O código elaborado baseou-se na linguagem de programação *Python* para observar mudanças na cobertura da terra, a partir do produto CCI-LC, e está disponível para consulta no seguinte endereço: https://github.com/ser-347/esa-cci-sankey. Entretanto, é necessário aperfeiçoamento para análise específica da acurácia do produto, pois podem ocorrer transições impossíveis nas mudanças de classes. Além disso, outros produtos de *Land Cover* podem complementar as observações e estudo sobre os dados CCI-LC, aplicando-os em diferentes escalas espaciais e temporais.

Para esta aplicação as classes foram adaptadas e apresentaram boa representação da cobertura global. Para futuras aplicações, recomenda-se a replicabilidade da análise para um período maior de observação. Um exemplo disso seria avaliar a cobertura da terra em 1992, 2002 e 2015, que possivelmente resultará em variações maiores entre as classes.

Referências

Almeida, C.A., Coutinho, A.C., Esquedo, J.C.D.M, Adami, M., Venturieri, A., Diniz, C.G., Dessay, N., Durieux, L., Gomes, A.R. (2016) “High spatial resolution land use and land cover mapping of de Brazilian Legal Amazon in 2008 using Landsat-5/TM and MODIS data.” Acta Amazonica, v. 46, n.3, p. 291-302.

Defourny, P., Santoro, M., Kirches, G., Wevers, J., Boettcher, M., Brockmann, C., Lamarche, C., Bontemps, S. (2017) “Land Cover CCI Product User Guide Version 2.0. UCLGeomatics”, Louvain-la-Neuve, Belgium, p. 1-105.

Grekousis, G.; Mountrakis, G.; Kavouras, M. (2015) “An overview of 21 global and 43 regional land-cover mapping products.” International Journal of Remote Sensing, DOI: 10.1080/01431161.2015.1093195.

Hua, T.; Zhao, W.; Liu, Y; W, S.; Yang, S. (2018) “Spatial Consistency Assessments for Global Land-Cover Datasets: A Comparasion among GLC2, CCI LC, MCD12, GLOBCOVER and GLCNMO.” Remote Sensing, n. 10, p. 1- 18, DOI:10.3390/rs10111846.

Li, W.; MacBean, N.; Ciais, P.; Defourny, P.; Lamarche, C.; Bontemps, S.; Houghton, R.A.; Peng, S. (2018) “Gross and net land cover changes in the main plant functional types derived from the annual ESA CCI land cover maps (1992–2015).” Earth System Science Data, v. 10, p. 219–234.

QPlanner: Módulo para Planejamento de Voo no *Software QGIS*

Frederyco Augusto Pereira Elleres[1], Carlos Rodrigo Tanajura Caldeira[1], Mayara Ortega Caldeira[1], Alan José Salomão Graça[2]

[1]Instituto Ciberespacial – Universidade Federal Rural da Amazônia (UFRA)
Belém – PA – Brasil

[2]Departamento de Engenharia Cartográfica – Universidade do Estado do Rio de Janeiro (UERJ)
Rio de Janeiro – RJ – Brasil.

fredelleres@gmail.com, {carlos.caldeira, mayara.caldeira}@ufra.edu.br, alan.salomao@eng.uerj.br

Abstract. *If the photography is to satisfactorily serve its intended purposes, the photographic mission must be carefully planned and faithfully executed according to the "flight plan." The objective of this paper was to develop a flight planning plugin for GIS QGIS. QGIS and Python documentation were used to develop the plugin through code implementation, where we used Python-based programming techniques, called PyQGIS. From this, there is an experimental flight planning plugin produced by QGIS, where it requests the user the essential parameters for the calculations in an automated way, obtaining as products the ground surface area covered by a block of photographs (polygon), the point clouds, where each point represents the central coordinate of the photograph taken during flight and the results of calculations of the variables required for flight planning. Given the results obtained by the plugin, the products and the time taken to generate them were compared with the mechanically performed operations. Concluding the great advantage of the plugin over the tutorial procedure, helping to save time and minimize the risk of mistakes.*

Resumo. *O planejamento de voo é uma das etapas primárias, no que diz respeito aos produtos fotogramétricos, fazendo com que o mesmo, deva ser bem planejado e executado. Dessa forma, o objetivo desse artigo, consistiu em desenvolver um módulo de planejamento de voo utilizando a plataforma de Sistema de Informação Geográfica QGIS. Foram usados manuais, métodos e bibliotecas do QGIS e Python, para desenvolver o complemento por meio da implementação de códigos, onde utilizou-se de técnicas de programação, com base na linguagem Python, chamada de PyQGIS. A partir disso, dispõe-se de um módulo experimental de planejamento de voo projetado para o QGIS, onde o mesmo solicita ao usuário os parâmetros essenciais para os cálculos de forma automatizada, obtendo como produtos a poligonal da área de sobrevoo, a nuvem de pontos, em que, cada ponto representa a coordenada central da fotografia capturada durante o voo e os resultados dos cálculos das variáveis necessárias para o planejamento de voo. Diante dos resultados obtidos pelo complemento, comparou-se, os produtos e o tempo levado para gerá-los, com as operações feitas mecanicamente. Concluindo-se a grande* *vantagem do complemento sobre o procedimento manual, contribuindo para economizar tempo e minimizar o risco de erros grosseiros.*

1. Introdução

A principal tarefa do levantamento aerofotogramétrico é o registro tridimensional de recursos naturais e artificiais no solo [Eisenbeiss e Sauerbier, 2011]. Para o sucesso de um levantamento aerofotogramétrico é necessário um bom planejamento de voo, determinando as ferramentas a serem utilizadas e o orçamento que precisará ter para subsidiar o aerolevantamento. Com isso, existe a necessidade de automatizar o processamento do plano de voo para obter dados precisos sobre a quantidade de fotos a serem tiradas e a área a ser mapeada. À medida que um maior número de usuários não-especialistas têm realizado voos fotogramétricos com RPAs (*remotely piloted aircraft*), há uma popularização do uso de APIs (*Application Programming Interface*) como por exemplo *Pix4D Capture* (https://www.pix4d.com/product/pix4dcapture), *DJI Go* (https://www.dji.com/br/goapp), *Drone Deploy* (https://support.dronedeploy.com), entre outros destinados ao planejamento da missão e cobertura fotogramétrica.

O progresso da Fotogrametria, nos últimos anos, deve-se em grande parte às contribuições da tecnologia da informação, uma vez que cálculos matemáticos da fotogrametria analítica podem ser programados com resultados adequados, principalmente em termos de economia de tempo e maior precisão nos trabalhos fotogramétricos [Amorim et al., 2006]. Em decorrência desses avanços tecnológicos, a maioria dos processamentos fotogramétricos se dão através de imagens digitais e recursos de visão computacional, onde fotografias podem ser compreendidas como um conjunto regular de *pixels*, os quais são descritos por sua geometria e radiometria, mas também podem ser especificados por outros elementos, como o ângulo determinado por um pixel e a distância focal [Berveglieri, 2014].

Nesse contexto da Fotogrametria Digital para a extração de feições terrestres, o planejamento de voo é uma etapa crucial da missão para a orientação das imagens aéreas e a geração subsequente dos produtos fotogramétricos. A maioria dos planos de voo inclui um conjunto de especificações detalhadas que descrevem os materiais, equipamentos e procedimentos a serem usados no projeto [Wolf et al., 2014]. No universo dos serviços de planejamento de voo, *softwares* proprietários, mesmo com pagamento por licenças de uso, ainda assumem uma posição privilegiada na preferência dos usuários devido a grande maioria dos *softwares* livres não apresentarem uma plataforma amigável e nem manuais detalhados para a entrada de *scripting* no aplicativo [Paixão Junior e Tavares Junior, 2019]. Desenvolver um complmento para planejamento de voo em uma plataforma *FOSS* (*Free and Open Source Software*) integrando o *software QGIS* a linguagem *Python*, com uso do *PyQGIS* no *plugin Builder* [Sherman, 2016], foi uma alternativa proposta para facilitar a conexão entre o *scripting* de plano de voo e os *códigos* internos do *QGIS 3.4*.

O presente trabalho traz consigo a criação da ferramenta *QPlanner,* um módulo de planejamento de voo gratuito, para atuar como um complemento plataforma livre *QGIS*. Entre suas atribuições estão a otimização do trabalho do usuário em calcular e gerar as informações para o levantamento fotogramétrico, contribuindo para o aprimoramento do *QGIS*, aumentando suas funcionalidades e finalidades de integração SIG/Fotogrametria.

2. Metodologia

A fim de permitir a aquisição de imagens de forma automatizada e precisa para o processamento fotogramétrico, elementos como a trajetória de voo aeronave deve ser calculada com antecedência, utilizando-se para isso parâmetros do projeto, como o tipo de objeto a ser documentado, a sobreposição das imagens, o sensor da câmera, especificações da aeronave bem como sua autonomia de tempo, e restrições de voo (devido a requisitos de segurança), são parâmetros relevantes de um projeto fotogramétrico [Eisenbeiss e Sauerbier, 2011]. Para a elaboração do módulo, adotou-se os parâmetros propostos por Eisenbeiss (2018), Figueiredo e Figueiredo (2018) e Wolf et al. (2014) pertinentes ao planejamento de voo como: Altura de voo; Distância focal; Dimensões do pixel e do quadro da câmara; Superposição lateral e longitudinal; Velocidade da aeronave; Coordenadas da área de sobrevoo; e Projeção. Apresenta-se ao final um diretório de saída com os resultados calculados.

A interface operacional do módulo é editável, e foi elaborada na extensão *Qt Designer* do *QGIS* a partir de um arquivo de interface pré-definido no *plugin Builder* [Sherman, 2016]. Para a atualização da interface em cada modificação feita no *Qt Designer*, utilizou-se outro *plugin*, denominado de *Reloader* [Sherman, 2016]. Em seguida, inseriu-se as bibliotecas *PyQgis* necessárias e recomendadas pelo Sherman (2016), tais como: *PyQt.QtCore, PyQt.QtGui, PyQt.QtWidgets, utils* e a *core*. Além de uma biblioteca *python* denominada *numpy* (https://qgis.org/pt_BR/site/). Dessa maneira o arquivo escrito em *python* na extensão do *Builder* contém todas as variáveis, formulações e comandos que o *QGIS* necessita para interpretar e gerar os produtos do levantamento fotogramétrico. Com essas bibliotecas é possível executar os processos para geração do planejamento de voo, seguindo a esquematização do fluxo de trabalho como mostra a Figura 1.

Com as bibliotecas inseridas, continuou-se com o desenvolvimento do *scripting*, em que, nessa etapa fez-se a adaptação das equações para determinar cada variável necessária para o plano de voo como mostra a Tabela 1.

Tabela 1. Discriminação das variáveis e suas respectivas equações

Nº	*Variáveis*	*Equações*
1	Denominador da escala	$den_{escala} = \frac{H_v}{f}$
2	GSD	$GSD = den_{escala} * d$
3	Escala da foto	$EF = \frac{f}{H_v}$
4	Lado longitudinal	$GL = den_{escala} * qX$
5	Lado transversal	$GT = den_{escala} * qY$
6	Aerobase	$B = GL * \left(1 - \frac{PE}{100}\right) * 100$
7	Avanço de base	$Av = \left(1 - \frac{PE}{100}\right)$
8	Distância entre faixas	$W = GL * \left(1 - \frac{PS}{100}\right)$

9	Distância longitudinal	$L = \sqrt{(Longitude\ 1 - Longitude\ 2)^2}$
10	Distância transversal	$Q = \sqrt{(Latitude\ 1 - Latitude\ 2)^2}$
11	Área total	$AT = L * Q$
12	Quantidade de modelos por faixa	$Nm = int\left(\frac{L}{B} + 1\right)$
13	Quantidade de fotos por faixa	$Nf = Nm + 1$
14	Quantidade de faixas por bloco	$Ns = int\left(\frac{Q}{W} + 1\right)$
15	Área incremental	$Anm = B * W$
16	Modelo estereoscópico	$Am = (GL - B) * GT$
17	Total de fotografias	$Nt = int\left(\frac{At}{Anm}\right)$
18	Total de fotografias capturadas	$Ntf = Nf * Ns$
19	Intervalo entre exposição	$I_e = \frac{B}{v}$

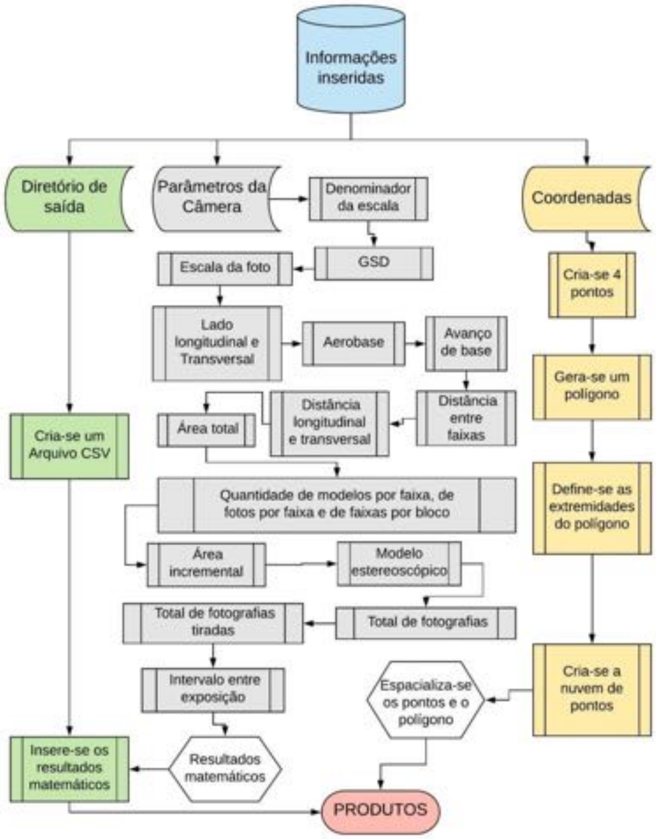

Informações inseridas
Diretório de saída
Parâmetros da Câmera
Denominador da escala
Coordenadas
GSD
Escala da foto
Cria-se 4 pontos
Lado longitudinal e Transversal
Aerobase
Avanço de base
Gera-se um polígono
Cria-se um Arquivo CSV
Área total
Distância longitudinal e transversal
Distância entre faixas
Quantidade de modelos por faixa, de fotos por faixa e de faixas por bloco
Define-se as extremidades do polígono
Área incremental
Modelo estereoscópico
Total de fotografias tiradas
Total de fotografias
Intervalo entre exposição
Espacializa-se os pontos e o polígono
Cria-se a nuvem de pontos
Insere-se os resultados matemáticos
Resultados matemáticos
PRODUTOS

Figura 1. Fluxograma dos processos que módulo executa.

O passo seguinte foi a programação de um método que possibilita criar e espacializar uma geometria de forma automática. Primeiramente definiu-se a geometria da camada do polígono a ser gerada, a partir de um provedor para os pontos fornecidos pelo usuário, ou seja, o código reconheceu as coordenadas e cria mais duas, formando assim um polígono quadrilátero dentro do *QGIS*, correspondente à área que será sobrevoada. Dessa forma adiciona-se uma nova camada ao painel de *layers* do *QGIS*.

Para o desenvolvimento da nuvem de pontos, representando as coordenadas do centroide de cada fotografia, utilizou-se às equações: 14, correspondente a aerobase [Wolf et al., 2014] para representar o espaçamento dado em X e 16 para o espaçamento em Y, equivalente a distância entre faixas. Em seguida, no *scripting* inseriu-se uma variável para obter a extensão da camada da poligonal, e uma nova camada de pontos vetoriais. Com a nuvem de pontos construída, dentro dos limites da área que será sobrevoada, a próxima etapa, compõe a adição dessas coordenadas da camada de pontos e, assim, uma atualização da extensão da camada foi necessária para que ao final a camada ao painel do *QGIS* pudesse ser inserida. Do *scripting* para o diretório de saída dos arquivos calculados, o produto é gerado no formato *Comma-separated values* (.csv), e assim, seleciona-se os dados calculados e determina-se uma nomenclatura para cada variável, com o intuito de facilitar a inserção das informações no CSV.

3. Resultados Preliminares

A interface operacional criada a partir da extensão *Qt Designer*, possibilita ao usuário inserir os parâmetros da câmera, onde o tamanho do pixel em X e Y está em micrômetros, a altura do voo em metros, tamanho do quadro (x, y) em milímetros e a velocidade da aeronave está em metros por segundo. Outras informações inseridas corresponderam com as coordenadas do quadro que será sobrevoado, onde essas devem ser inseridas no formato UTM. Além de ser solicitado a projeção que os arquivos serão gerados e o diretório de saída, com isso, facilitando os cálculos que o *scripting* realizará, deste modo, a interface pode ser vista na Figura 2.

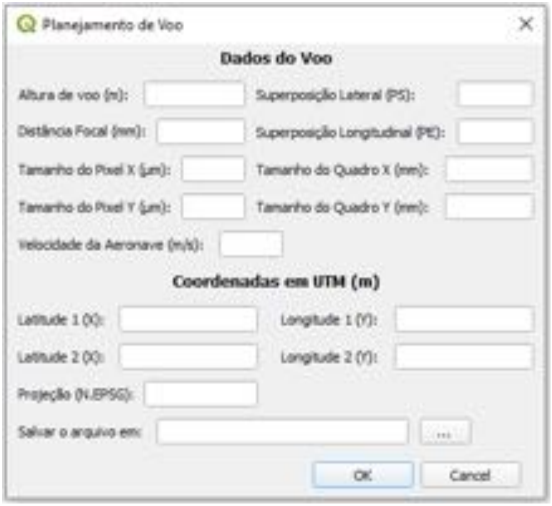


Figura 2. Interface operacional do módulo *QPlanner* para o QGIS.

O tempo de execução entre os cálculos realizados pelo módulo e manualmente foi de ±5 segundos e ±20 minutos, respectivamente, ou seja, a utilização do complemento, além da praticidade, tornou os cálculos 240 vezes mais rápido, com o processo sendo executado pela equipe desse projeto nos testes preliminares.

4. Considerações Finais

O presente trabalho conseguiu desenvolver um *scripting* para o planejamento de voo, dentro da plataforma do *software QGIS*, com uma interface de fácil manipulação e intuitiva, onde um usuário só precisa inserir as informações solicitadas. Dessa forma, seus produtos serão gerados de forma automática. Além disso, a instalação do módulo se torna bem simples, sendo baixado diretamente nos complementos do *software*.

Em suas funcionalidades o módulo consegue determinar as coordenadas do centro perspectivo de cada fotografia que será tomada durante o voo, e, com essas coordenadas gera-se uma nuvem de pontos para representar a localização de cada fotografia. Deste modo, fazendo com que o usuário consiga tomar decisões antes do voo, devido a geração de resultados matemáticos do plano de voo de forma rápida e precisa, além de poder com esses dados, gerar uma carta de auxílio para o voo.

Com isso, o módulo gerado, necessitará de adaptações e melhorias para facilitar o seu uso. Porém, nota-se que o objetivo do trabalho foi alcançado, gerado o plano de voo e exportando suas informações, o ambiente *QGIS*. Assim, essas informações geradas, são essenciais para a montagem de uma carta de voo, que em softwares comerciais são omitidas.

5. References

Amorim, A., Tommaselli, A. M. G. and Silva, I. (2006). Use of hybrid Stereopairs in Map Revision. In *Geodésia On-Line*, v. 10, p. 1–13. UFSC.

Berveglieri, A. (2014). “Localização automática de pontos de controle em imagens aéreas baseada em cenas terrestres verticais”. Tese de Doutorado, UNESP Presidente Prudente, Programa de Pós-Graduação em Ciências Cartográficas, 148 f.

Eisenbeiss, H. (2008). The autonomous mini helicopter: a powerful platform for mobile mapping. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, v. 37, p. 977–983. ISPRS.

Eisenbeiss, H. e Sauerbier, M. (2011). Investigation of UAV Systems and Flight Modes for Photogrammetric Applications. *The Photogrammetric Record*, v. 26, n. 136, p. 400–421.

Figueiredo, E. O. e Figueiredo, S. M. de M. (2018). Planos de voo semiautônomos para fotogrametria com aeronaves remotamente pilotadas de classe 3. *Circular Técnica*, v. 75, p. 1-56. Embrapa.

Paixão Junior, J. G. C. e Tavares Junior, J. J. R. (2019) Perfilagem Multifonte de Borda com Mapeador de Texturas em *Python*: Baía de Icó-mandantes, Reservatório de Itaparica – PE. In: *Anais Do XIX Simpósio Brasileiro De Sensoriamento Remoto*, Instituto Nacional de Pesquisas Espaciais – INPE.

Sherman, G. (2016). *The PyQGIS Programmer’s Guide*. Locate Press LLC, 1st edition.

Wolf, P. R., Dewitt, B. A. e Wilkinson, B. E. (2014). *Elements of Photogrammetry with Applications in GIS*, McGraw-Hill, 4th edition.

CLUSTERMAP: *PLUGIN* DE VISUALIZAÇÃO DE DADOS MULTIVARIADOS EM MAPAS COROPLÉTICOS

Tiago P. Silvano[1], Bryan M. Correa[1], Philipe Borba[2], Ivanildo Barbosa[1]

[1]Seção de Ensino de Engenharia Cartográfica - Instituto Militar de Engenharia (IME)
Rio de Janeiro – RJ – Brazil

{tiago.silvano,bryan.correa,ivanildo}@ime.eb.br

[2] Instituto de Geociências – Universidade de Brasília (UnB)
Brasília – DF – Brazil

philipe.borba@unb.br

Abstract. *The ClusterMap plugin was developed to be used in the QGIS environment to create clusters from multivariate numerical data regarding georeferenced features. Some clustering methods were embedded to enable user to visualize the spatial distribution of the features within each resulting cluster, as well as to analyze their behavior based on decision rules. It is also available a resource to suggest to the user the optimal number of clusters, in case he/she found useful. The user is free to combine any number of numeric variables, any number of clusters must be created as well as the clustering methods to use, and receive a new choropletic map, quality indicators for clusters, and decision rules.*

Resumo. *O plugin ClusterMap foi implementado para o ambiente QGIS com o objetivo de criar clusters a partir de um conjunto de variáveis, do tipo numérico, associadas a objetos georreferenciados. Foram implementados diferentes métodos de clusterização, permitindo ao usuário visualizar a distribuição espacial dos clusters formados e entender, por meio das regras de decisão, quais as características de cada um deles. Foi disponibilizada uma funcionalidade para sugerir ao usuário o número ótimo de clusters, caso ele julgue útil. O usuário possui a liberdade de testar a quantidade de variáveis, a quantidade de clusters e o método de clusterização, obtendo um novo mapa coroplético, indicadores de qualidade de clusters e regras de decisão.*

1. Introdução

O objetivo deste trabalho é descrever as funcionalidades de um *plugin*, desenvolvido para o ambiente QGIS, que efetua a clusterização a partir de um conjunto de dados georreferenciados, em formato vetorial, com variáveis numéricas, entregando ao usuário um mapa coroplético baseado nos grupos gerados. No rodapé desta página há um *link* para um vídeo que demonstra as funcionalidades da solução desenvolvida.

Para analisar múltiplas variáveis simultaneamente, os autores optaram por empregar a clusterização para agrupar os objetos de acordo com a similaridades de seus atributos, ou seja, objetos de um mesmo *cluster* são similares entre si, ao mesmo tempo que se distinguem de objetos de *clusters* vizinhos.

Link para vídeo demonstrativo: https://youtu.be/XU8C9e9WNd8

Linguagens como *Python,* R e *Weka,* embora realizem a tarefa de clusterização e visualização de dados, necessitam que o usuário tenha conhecimento em programação. O ambiente QGIS possui interface mais amigável e permite a visualização de mapas coropléticos para analisar a distribuição espacial de dados multivariados.

A motivação para este trabalho é a implementação de funcionalidades não encontradas, atualmente, em outros *plugins* como*:* ferramenta de análise do número ótimo de *clusters*, critério descritivos para cada *cluster* segundo modelo de Árvore de Decisão e método de avaliação da clusterização, bem como a visualização espacial dos *clusters.*

Alguns requisitos foram elencados a fim de aprimorar a experiência do usuário:

a) Os valores dos parâmetros de cada método são informados pelo usuário: método de agrupamento, métrica de distância, medidas de dissimilaridades, e número de *clusters*;
b) A camada gerada é carregada no QGIS automaticamente, categorizada pelo número do *cluster* associado a cada feição;
c) A fim de subsidiar a interpretação do significado implícito na formação de cada *cluster*, foram extraídas regras baseadas em árvores de decisão, que podem ser interpretadas pelo usuário ou utilizadas para a geração de legenda;
d) É possível calcular o número ótimo de *clusters*;
e) É possível calcular as larguras médias de silhueta, geral e por *cluster,* medidores de qualidade da clusterização;

A combinação dos valores dos parâmetros, a definição do número de classes e a interpretação dos resultados fica a cargo do usuário, de modo que o *plugin* seja apenas uma ferramenta de apoio, flexível para aplicações em diversas áreas de atuação.

2. Implementação do *Plugin*

O código implementado na linguagem *Python tem* como base a estrutura do *Processing Framework* do QGIS, buscando mesmos padrões e funcionalidades dos algoritmos de processamento do *QGIS.* As principais bibliotecas *Python* utilizadas nesse projeto são a biblioteca *Sklearn* para implementação dos métodos de agrupamento e classificação, *Numpy* para o processamento de *array* e gerenciamento de matrizes, *Matplotlib* com o objetivo de gerar gráficos bidimensionais para análise do usuário.

As interfaces foram customizadas com auxílio do *Qt Designer* e da biblioteca *PyQt,* conexão entre *Python* e o *Qt,* de forma a atender as funcionalidades do *plugin.*

3. Funcionalidades do *Plugin*

O *plugin* permite ao usuário a escolha de dois métodos de agrupamento, um não hierárquico (*k-means)* e um hierárquico aglomerativo. Os valores dos parâmetros são passados pelo usuário para cada método por intermédio da interface gerada pelo *Processing* do QGIS.

Caso o usuário opte pelo método *k-means,* o *plugin* permite a escolha da camada vetorial dentre aquelas pré-carregadas no QGIS. Depois que a camada é selecionada, os atributos de tipo numérico são apresentados para que o usuário possa selecionar quais serão utilizados no processo de agrupamento. Além disso, o usuário necessita selecionar o número de *clusters k* a serem gerados no processamento [Camilo e da Silva 2009]. Para auxiliar o usuário nesta escolha, foi implementada uma ferramenta de análise do número ótimo de *clusters* com os métodos *Elbow e Silhouette* [The scikit-yb developers 2019].

Por outro lado, caso o usuário selecione o método hierárquico, as mesmas funcionalidades da escolha da camada e atributos são mantidas. Em seguida, o usuário possui a opção de escolha das medidas de dissimilaridade entre *clusters Ward, Single Linkage, Complete Linkage* e *Average Linkage,* bem como as métricas *Euclidiana e Manhattan* para o cálculo das distâncias [Camilo e da Silva 2009]. Assim como no método *k-means,* o usuário necessita selecionar o número *k* de *clusters* a serem criados.

O principal produto gerado pelo *plugin* (*output)* é uma camada vetorial temporária com os mesmos atributos da camada selecionada para o processo de agrupamento, acrescido de um atributo que registra o número do *cluster* a que pertence cada feição. O *plugin* apresenta ao final do processamento as larguras médias de silhueta, geral e dividida por *cluster*, como indicador de qualidade da clusterização. Também são disponibilizadas as regras de decisão extraídas do resultado da etapa anterior, a fim de auxiliar o usuário na compreensão da semântica implícita na geração dos *clusters.*

4. Conclusão

Neste trabalho foi apresentado o *plugin* ClusterMap, utilizado para realizar clusterização e visualização de um conjunto de dados multivariados georreferenciados, em formato vetorial, com variáveis do tipo numérico. O *plugin* também disponibiliza ao usuário métricas para avaliação da qualidade da clusterização e um conjunto de regras de decisão que permitem ao usuário interpretar a lógica de formação de cada *cluster.*

É importante destacar alguns exemplos de análises que podem ser viabilizadas com aplicação do *plugin* desenvolvido: a comparação entre indicadores de composição demográfica, atividade econômica, a evolução do número de notificações de alguma patologia agregados por municípios [Silvano, Correa e Barbosa 2020] e estudo aplicado à agropecuária [Pena et al 2017].

O *plugin* está disponível no repositório oficial do QGIS. O código implementado, vídeo demonstrativo de uso do *plugin,* guia de instalação do ClusterMap, bibliotecas *Python* exigidas e documentação estão disponíveis em repositório no ambiente GitHub[1].

Referências

Camilo, C. O. e da Silva, J. C. (2009) “Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas: Relatório Técnico RT-INF_001-09”, Universidade Federal de Goiás.

Pena, M.G., Moreira, G.C.C., Guimarães, L.F.D., Laureto, C.R., Albuquerque, P.H.M., Carvalho, A.X.Y. e Basso, G.G. (2017) Clusterização Espacial e Não Espacial: Um Estudo Aplicado à Agropecuária Brasileira. Tendências em Matemática Aplicada e Computacional, 18, N.1, p. 69-84.

Silvano, T. P., Correa, B. M. e Barbosa, I. (2020) Análise da distribuição espacial de indicadores sociais e demográficos: uma abordagem baseada em mineração de dados. In. *Revista Brasileira de Cartografia*, v. 72. n. 1, páginas 67-80.

The scikit-yb developers (2019) “Clustering Visualizers”, https://www.scikit-yb.org/en/latest/api/*cluster*/index.html, Setembro.

[1]https://github.com/tiagoPrudencio/ClusterMap

DIMS-LapsusTerrae: Sistema de Gerenciamento de Informação de Desastres de Deslizamento de Terra

Lucas F. Dorigueto, Carlos H. T. Brumatti, Erick L. Figueiredo, Jugurta Lisboa-Filho

Departamento de Informática – Universidade Federal de Viçosa (UFV)
Viçosa, MG, Brasil

{lucas.dorigueto, carlos.h.tavares, erick.figueiredo, jugurta}@ufv.br

Abstract. *The article presents DIMS-LapsusTerrae, an implementation of the LapsusVGI framework, a platform that integrates Voluntary Geographic Information (VGI) with ISO and OGC standards, providing an architecture to be used in the implementation of Disaster Information Management Systems (DIMS) aimed at landslides.*

1. Introdução

Segundo o Banco Mundial (2016), desastres naturais são responsáveis por 520 bilhões de dólares em perdas, além de levar 26 milhões de pessoas a pobreza por ano. No Brasil, 59.4% dos 5.570 municípios não possuem nenhum plano de gestão de risco para desastres, e 15% já foram atingidos por deslizamento de terra (IBGE, 2018).

Sistemas de Gerenciamento de Informação de Desastres (DIMS) auxiliam os gestores em diferentes situações de emergência. Segundo Ryoo e Choi (2006), um dos principais obstáculos em DIMS é a falta de padronização nos dados, o que dificulta a interoperabilidade entre sistemas. Conforme um levantamento referente ao período de três anos realizado por Tavares et. al. (2018), foram identificados apenas dois sistemas que cobrem as três etapas em uma situação emergencial (pré, durante e pós evento), porém, nenhum dos dois sistemas adotam algum tipo de padrão em sua arquitetura, deste modo, há a necessidade da construção e aperfeiçoamento de sistemas que possam auxiliar no gerenciamento e diminuição dos danos causado por desastres.

Sistema de Informação Geográfica Voluntária (VGI) são sistemas colaborativos nos quais os usuários colaboram com dados que estão associados a uma localização espacial (Goodchild, 2007). Sistemas VGI também têm sido utilizados para auxiliar na tomada de decisões em situações de emergência, como o sistema de coleta de VGI Ushahidi, que foi utilizado após o terremoto que ocorreu no Haiti em 2010 (Ushahidi, 2019) e em diferentes ações humanitárias em alguns países africanos. Este trabalho apresenta um DIMS para desastres relacionados a deslizamento de terra, que se baseia no framework *LapsusVGI* (Dorigueto et al., 2020), uma plataforma que adota diferentes padrões ISO e OGC para garantir a interoperabilidade dos dados e também coleta VGI.

2. O Sistema *DIMS-LapsusTerrae*

O Sistema *DIMS-LapsusTerrae* é uma implementação do framework *LapsusVGI* cujo objetivo é prover suporte aos gestores na tomada de decisões e auxiliar a comunidade em momentos de emergências, possibilitando coleta de VGI.

Para a construção do sistema, utilizou-se a arquitetura Model-View-Controller (MVC), que provê boa manutenibilidade e portabilidade, permitindo que posteriormente o sistema possa ser portado para plataformas mobile sem alterações na estrutura do projeto. Para a construção das interfaces responsáveis pela interação com o usuário, foram utilizadas tecnologias como HTML, Bootstrap e Jquery, além da biblioteca Leaflet, que permite a visualização e extração das feições provindas do sistema mundial de mapeamento voluntário OpenStreetMap (OSM).

No back-end, o sistema foi desenvolvido na linguagem PHP, em conjunto com o SGBD MySQL, que possui estruturas voltadas para o gerenciamento de feições espaciais. Para possibilitar a disponibilização das futuras colaborações no sistema, o DIMS está integrado ao sistema GeoServer, para que os dados possam ser fornecidos a partir dos padrões Web Map Service (WMS) e Web Feature Service (WFS), deste modo, as colaborações podem ser utilizadas em outras plataformas que utilizem tais padrões, por exemplo, o QGIS.

A Figura 1 apresenta um esquema ER contendo entidades e relacionamentos presentes na aplicação. Vale ressaltar que somente colaboradores podem fazer contribuições VGI, já que os gestores foram associados às Informações Compartilhadas de Gerenciamento de Emergência (EMSI), que são mensagens para serem utilizadas entre organizações envolvidas em situações de emergência.

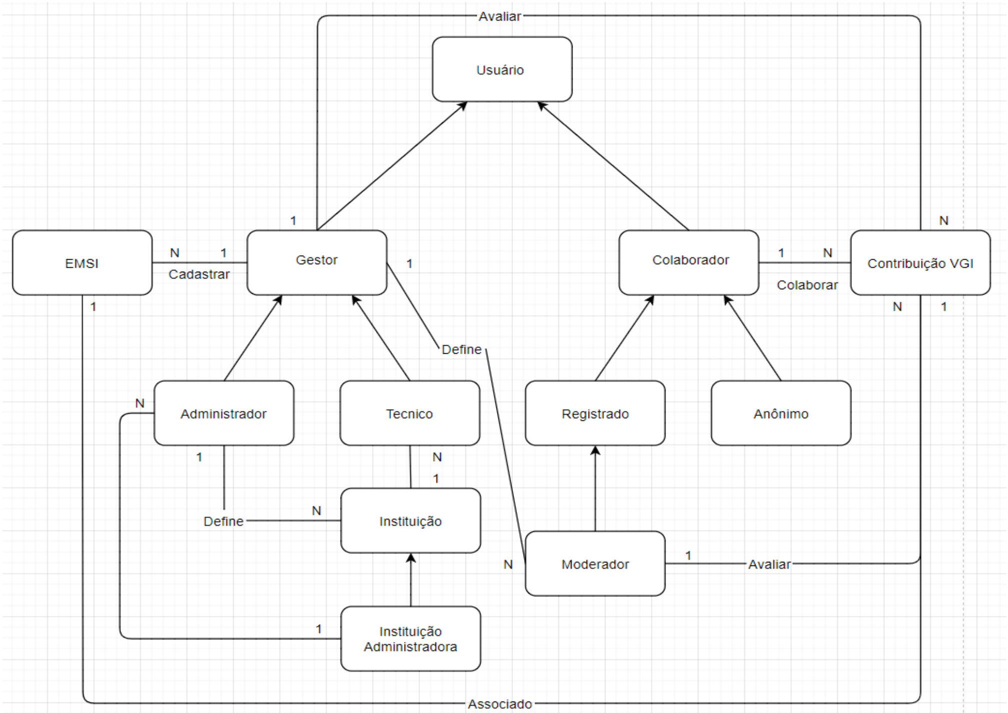


Figura 1. Parte do esquema conceitual *LapsusTerrae*

O sistema possui três tipos de usuários colaboradores e dois tipos de usuários gestores. Os colaboradores, registrado ou anônimo, possuem permissão apenas para contribuir com VGI no sistema, além de poder exportar os dados. Já o colaborador moderador, além das permissões mencionadas anteriormente, também pode aceitar ou rejeitar colaborações enviadas por outros colaboradores.

O usuário gestor pode ser administrador ou técnico. Ambos podem definir novos moderadores e cadastrar mensagens EMSI, que neste contexto, correspondem às ocorrências oficiais de autoridades ou ocorrências originadas de colaborações validadas de VGI. Devido à grande quantidade de atributos que são definidos nas mensagens EMSI, o modelo da mesma não será exibido neste artigo por questão de espaço.

Por fim, para auxiliar na organização, o sistema pode ser gerenciado a nível de instituição, onde haverá uma instituição administradora, por exemplo, defesa civil, que poderá gerenciar e contar com a ajuda de outras instituições, por exemplo, polícia militar e corpo de bombeiros, deste modo, cada gestor deve estar associado a uma instituição.

3. Conclusões

Este artigo apresenta o software *DIMS-LapsusTerrae,* uma implementação do framework *LapsusVGI*, plataforma VGI baseada em padrões ISO e OGC que fornece uma arquitetura para ser utilizada em DIMS voltado a deslizamentos de terra. Vale ressaltar que toda a tecnologia utilizada na construção da plataforma é gratuita e de fácil acesso, o que permite facilidade para qualquer entidade que queira efetuar modificações futuras.

Esta plataforma está sendo desenvolvida com o intuito de ser disponibilizada como software livre, deste modo, a aplicação pode ser encontrada juntamente com seu código fonte no endereço: http://www.dpi.ufv.br/projetos/lapsusVGI.

Agradecimentos

Projeto parcialmente financiado pela CAPES e Fapemig.

Referências

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

Banco Mundial (2019). Natural Disaster Force 26 Million People into Poverty and Cost 520bn in Losses Every Year, New World Bank Analysis Finds. https://www.worldbank.org/en/news/press-release/2016/11/14/natural-disasters-force-26-million-people-into-poverty-and-cost-520bn-in-losses-every-year-new-world-bank-analysis-finds.

Dorigueto, L. F. et al. LapsusVGI: um framework para sistemas de gerenciamento de informação sobre deslizamento de terra. Submetido ao GeoInfo 2020.

Instituto Brasileiro de Geografia e Estatística – IBGE (2019). MUNIC 2017: 45,6 dos municípios do país foram afetados por secas nos últimos 4 anos. https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/21636-munic-2017-48-6-dos-municipios-do-pais-foram-afetados-por-secas-nos-ultimos-4-anos.

Ryoo, J., Choi, Y. B. A comparison and classification framework for disaster information management systems. *International Journal of Emergency Management,* 3: 264-279

Tavares, J. F. et al. (2018). Systematic review on the use of groupware technologies in emergency management. *In proc. of the Third IFIP TC 5 DCITDRR Int. Conf. on Information Technology in Disaster Risk Reduction*, pages 22-35.

Ushahidi (2019). Ushahidi, https://www.ushahidi.com, December.

Index of authors