A Parallel-based Map Matching Approach over Urban Place Records

Tiago Brasileiro Araújo[1], Carlos Eduardo Santos Pires[1], Demetrio Gomes Mestre[2], Andreza Raquel Monteiro de Queiroz[1], Veruska Borges Santos[1], Thiago Pereira da Nóbrega[2]

[1]Universidade Federal do Campina Grande (UFCG), Campina Grande, PB – Brasil

[2]State University of Paraíba (UEPB), Campina Grande, Brazil

{tiagobrasileiro@copin,cesp@dsc,andreza.queiroz@ccc,veruska

.santos@ccc}.ufcg.edu.br, {demetriogm,thiagonobrega}@uepb.edu.br

Abstract. *In the Smart Cities scenario, to avoid the conflicting geospatial records between official and non-official sources, it is necessary to detect the inconsistencies regarding the geospatial data provided by them. To this end, the map matching task, i.e., the task of identifying correspondent features between two geospatial data sources, should be applied. For spatial Big Data, the map matching task is confronted with challenges related to volume and veracity of the data. In this sense, we propose a Spark-based map matching approach, called MATCH-UPS. To evaluate, real-world data sources of New York (USA) and Curitiba (Brazil) were applied. The results showed that MATCH-UPS improved the precision by 26% and reduced the execution time by one third.*

1. Introduction

Geographical coordinates and maps (digital or paper-based) are a common element of our daily life which provide a two-dimensional representation of geographical features in the real world, such as parks, bus stops, roads, rivers, buildings, and places. Such information is often referred to as geospatial or geographical data and plays an essential role in many governmental, economic and social domains, such as disaster response, urban planning, and tourism [Du et al. 2017, Harb and Becker 2018]. In this sense, the large amount of data collected by municipalities and map projects (e.g., Open Street Map) may be used to improve the efficiency of public transportation and infrastructure investments. However, since geospatial data sources are prone to inconsistencies and quality issues, it is important to assess the data quality before using them to take valuable strategical decisions. In fact, an analysis based on incorrect information can lead to wrong decisions [Christen 2012].

For Smart Cities scenario, the municipalities benefit of the data collected from multiple sources, such as official documentation, third parties information, and environmental sensors [Bojic et al. 2015, Fan et al. 2014]. To avoid the conflicting records between official and non-official sources it is necessary to detect the inconsistencies regarding the geospatial data provided by them. To this end, the Map Matching task, i.e., the task of identifying correspondent features between two geospatial data sources [Du et al. 2017], can be applied. For instance, map matching can be used to identify geospatial records (e.g. parks, bus stops, and roads) from distinct data sources that refer to the same real-world place. It is an essential pre-processing step for data integration,

change detection, data updating, and data comparison [Fan et al. 2014]. Map matching is a powerful task to assist several projects interested in extract information from geospatial data (for instance, control of deforestation, fires and floods) and approaches related to smart cities [Bojic et al. 2015]. Several works report the necessity to apply the map matching task to solve diverse problems related to urban mobility and environment in different countries, such as USA [Pi et al. 2018], Chile [Arriagada et al. 2019], Sweden [Bouguelia et al. 2018], and Italy [Interdonato and Tagarelli 2017]. In this context, the present work has been conceived by analyzing the scenario and related issues addressed by an international cooperation project called EUBra-BIGSEA. This project aims to develop a cloud platform for data management and exploitation. In particular, the services are able to empower data analytics to support the development of data processing applications.

In the context of spatial Big Data, the map matching task is confronted with two main challenges: *volume*, as it handles a large amount of geospatial records (such as buildings, parks and bus stops of a megalopolis); and *veracity*, related to the reliability of the data provided by the source [Christen 2012]. The desktop GIS software typically takes hours to process (or compare) these massive geospatial data. Such a consumption of time is unacceptable for many applications, particularly for real-time policy decisions such as predicting which areas would be damaged by natural disasters. In this light, blocking techniques and parallel computing are applied in order to minimize the problems related to the large volume of data. Blocking techniques group similar records into blocks and perform comparisons of the records within each block. The map matching in parallel aims to reduce the overall execution time by distributing the comparisons between geospatial records among the various resources (e.g., computers or virtual machines) of a computational infrastructure. Regarding the veracity, the map matching task can be applied in order to assess the quality of data sources. For instance, by identifying conflicting and inconsistent records between them [Fan et al. 2014]. Problems related to veracity often occurs due to the data are collected through crowd-sourcing. For this reason, the OSM has often been denounced due to its heterogeneity in quality from the beginning of its development and needs to be evaluated by comparing with authority data [Fan et al. 2016].

In the literature, there are several map matching approaches [Fan et al. 2016, Fan et al. 2014, Du et al. 2017, Alarabi et al. 2017]. However, most of them address the problem of road network matching [Fan et al. 2014]. In this work, we concentrate on two open areas not addressed by the previously mentioned works concomitantly: map matching (involving points and polygons) and parallel computing, applying Apache Spark. Overall, the contributions of our work are the following: i) our approach, called Matching of Urban Places with Spark (MATCH-UPS), proposes the parallelization of the map matching approach proposed in [Fan et al. 2014]. The novel approach applies the Spark framework to parallel the execution of the map matching task; ii) we propose the application of the semantic attributes (e.g., names of the places) and context information (e.g., neighbouring streets) for the matching task in order to assist the comparison of geospatial records (i.e., polygons or points). This information provides additional evidence used to assist the definition of the corresponding places (i.e., match) in the map matching task; and iii) we propose the application of the blocking technique based on the geographical similarity (from the coordinates) to group similar geospatial records into the same block.

2. Background

Matching Geospatial Data. In map matching, it is necessary to compare the geospatial records of one data source with the geospatial records of the other one. To determine the similarity of a record pair, geospatial and semantic attributes (e.g., geolocations, place names, addresses, or postcodes) can be compared [Christen 2012]. If the records of a record pair present various similar attribute values, these records have high chances of being considered similar. According to the type of geospatial data (in this work, points and polygons), the geospatial information can be explored in different ways. To compare the geospatial information of two points (i.e., latitude and longitude of each point), the distance between them is commonly evaluated [Fan et al. 2014, Xavier et al. 2016]. The closer the points are, the greater the chances of them being considered as correspondents. Regarding the polygons, the geospatial information (i.e., a set of latitude and longitude for each polygon) can be evaluated based on the overlapping area of the polygon pair. To improve the efficacy of the map matching results, semantic attributes (e.g., names, descriptions, and annotations) of record pairs can also be compared beyond the evaluation based on the geospatial attributes [Christen 2012]. For instance, the names of the buildings (semantic attribute) are linguistically compared (using linguistic measures such as Jaccard or Levenshtein) to determine the similarity degree between them.

To determine whether a record pair is a correspondence or not, the similarity values can be combined (e.g., given by average or weighted average) or used individually to compound a rule of classification. The most two commonly classifiers applied in this context are: threshold-based and rule-based [Christen 2012]. The former combines the similarity values (e.g., geospatial similarity and semantic similarity) and determines as a correspondence the record pairs that have a final similarity value higher than a given threshold. Related to the latter, a record pair can be classified not only as match or non-match. Thus, a record pair is classified as one of the predetermined classification profiles (for instance, match, possible match, or non-match) according to the rule, taking into account all the similarity values (e.g., geospatial similarity and semantic similarity). In this work, we applied rule-based and threshold-based classifiers according to the number of attributes assessed (consequently, the number of similarity values) in each scenario (i.e., involving polygons or points), as will be described in Section IV.

3. Related Work

The survey [Xavier et al. 2016] highlights several works which propose map matching approaches in the context of geographical data. In the work [Fan et al. 2016], a polygon-based approach is proposed to match roads and urban blocks provided by Open Street Map (OSM) and official data sources. The algorithm represents the urban blocks (i.e., elements of urban planning) as polygons (e.g., surrounding buildings) and lines (e.g., surrounding streets). Thus, the algorithm is able to match urban blocks evaluating their spatial topologies based on the polygons. To determine the similar polygons (which model the urban blocks), the algorithm evaluates the overlapping areas between the polygons.

Similar to [Fan et al. 2016], the work [Fan et al. 2014], which inspired our work, proposes a method for matching building footprints (i.e., polygons), in order to assess the quality of the data provided by OSM. The similarity between polygons is defined by the percentage of overlapping area between them, using 30% as the threshold for

considering a polygon pair as a match. However, the work [Fan et al. 2014] presents limitations when the same building is represented as two disjoint polygons (commonly in scenarios where the polygons have a small area) in OSM data and authoritative data [Du et al. 2017]. Regarding context-based map matching, the work [Zhang et al. 2014] affirms that in cases when there is ambiguity in the correspondences, the similarity of geographical features depends on the context. This work proposes a triangulation-based approach to define the neighbourhood of urban places, considering a continuous influence from the closest places.

In these terms, although the works previously discussed address the map matching task, none of them proposed approaches to address map matching and parallel computing simultaneously. Despite the work [Alarabi et al. 2017] apply parallel computing over the geospatial records management, it does not address the map matching task. Thus, applying parallelism/cloud computing in the context of geographical Big Data is treated as an open research area [Zhang et al. 2015, Xavier et al. 2016]. The work [Zhang et al. 2015] also highlights the lack of parallel-based map matching approaches. Therefore, our work emerges as a bridge between map matching and parallel computing since we propose a Spark-based approach that parallelizes the map matching task. In addition to assessing the geographical similarity of the geospatial records (provided by the overlapping area or the distance between the records), our work takes into account other attributes (e.g. place names) of the records and applies context-based information. Regarding context-based map matching, our work considers the streets as context information and applies rules (instead of triangularization) based on the distance between bus stops/streets to determine which bus stops are considered correspondents.

4. The MATCH-UPS Approach

4.1. Overview

Given two sets of geospatial records, represented by D_1 and D_2, the map matching task consists in identifying all correspondences between records. We denote the schema followed by the records of D as $E = (a_1, a_2, \ldots, a_n)$, in such a manner that each a_i corresponds to an attribute (e.g., name, category, and geographical coordinates). Therefore, the input data sources D_1 and D_2 contain a finite set of records denoted as a pair of $\langle attribute, value \rangle$: $r = [\langle a_1, v_1 \rangle, \langle a_2, v_2 \rangle, \ldots, \langle a_n, v_n \rangle]$. Let $sim(r_1, r_2)$ be the similarity measure between records r_1 and r_2, Φ_{max} the maximum threshold that defines whether r_1 matches r_2, and Φ_{min} the minimum threshold that defines the pair r_1 and r_2 as non-match. Thus, the map matching can classify the pairs of records as $Match(M) = \{(r_i, r_k) \mid r_i \in D_1, r_k \in D_2 \; and \; sim(r_i, r_k) \geq \Phi_{max}\}$, $Non - Match(NM) = \{(r_i, r_k) \mid r_i \in D_1, r_k \in D_2 \; and \; sim(r_i, r_k) \leq \Phi_{min}\}$ and $PotentialMatch(PM) = \{(r_i, r_k) \mid r_i \in D_1, r_k \in D_2 \; and \; \Phi_{min} < sim(r_i, r_k) < \Phi_{max}\}$.

In this work, we propose a Spark-based map matching approach, denoted as MATCH-UPS[1], for dealing with records represented by spatial geometries (e.g., polygons and points). Thus, the MATCH-UPS approach performs pairwise comparisons between geospatial records, determining the similarity between them. To classify a record pair, linguistic and geographical matchers are applied. A linguistic matcher explores the textual attributes of the record (e.g., name and description) to determine the linguistic similarity

[1]https://github.com/brasileiroaraujo/GeoMatch-MATCH-UPS-.

Table 1. Classification rules.

Rule	Classification
LS >*LST* and *GS* >*GST*	*Match*
LS <*LST* and *GS* <*GST*	*Non-Match*
(*LS* >*LST* and *GS* <*GST*) or (*LS* <*LST* and *GS* >*GST*)	*Potential match*

(*LS*) of a record pair. To this end, algorithms such as Jaccard and Levenshtein distance are used. A geographical matcher explores the coordinates of both geospatial records in order to identify the overlapping area or the distance between them. The proportion of the overlapping areas (for polygons) or the distance (for points) represent the geographical similarity (*GS*) between the records of a pair. Thus, the record pairs are categorized according to the classification rule show in Table 1 which considers the similarity values as well as a linguistic similarity threshold (*LST*) and a geographical similarity threshold (*GST*). The record pairs are classified into threes categories: *match*, *non-match*, and *potential match*.

4.2. Geospatial Polygons

Regarding the polygons (for instance, buildings, residential regions, parks, and forests), the similarities between the geospatial records can be measured through the linguistic and geographical matchers. As previously mentioned, the linguistic matcher can apply algorithms such as Jaccard and Levenshtein distance to determine the linguistic similarity between the records. To measure the geographical similarity, the overlapping area between the polygons of the geospatial records is evaluated. In this sense, works such as [Fan et al. 2014] consider that a pair of geospatial records with an overlapping area above 30% are classified as a match. Therefore, the geographical similarity threshold (*GST*) applied to polygons (records) pairs is 0.3 (30%). Equation 1 denotes the rule to measure the similarity of two polygons (p_1 and p_2):

$$sim(p_1, p_2) = min(\frac{overlappedArea(p_1, p_2)}{area(p_1)}, \frac{overlappedArea(p_1, p_2)}{area(p_2)}) \quad (1)$$

4.3. Geospatial Points

To perform the map matching task over points records (for instance, bus stops, points of interest, and vehicle coordinates), linguistic and geographical matchers can be applied. In this work, we apply only a geographical matcher since the records provided by the data sources do not contain relevant linguistic attributes. The geographical matcher considers as *match* the record pairs with a distance (in meters) lower than a certain *GST* (also given in meters). Otherwise, the pair of records is considered as *non-match*. In this work, we apply a *GST* of 20 meters, since the work [Yang et al. 2014] proved experimentally that a pair of geospatial points with a distance lower than 20 meters can be considered as a match. On the other hand, if we consider only the geographical distance, two problems can be highlighted. Since several bus stops may be close to each other in the real world, a bus stop can be classified as match more than once with different representations of bus stops contained in the other data source. Moreover, the application of the geographical

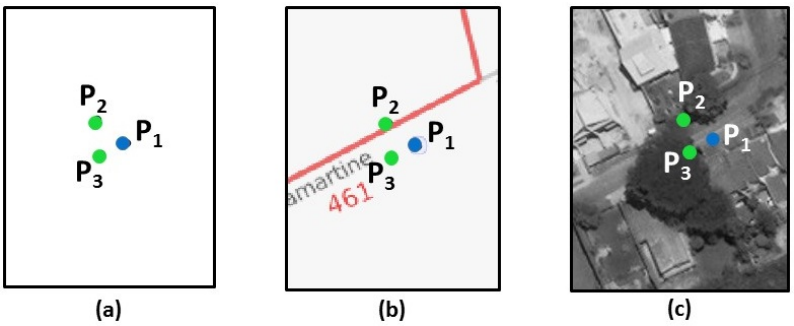

Figure 1. Applying the context information over bus stops data sources.

matcher can only reduce the reliability of the results since the bus stops may be close but might not represent the same bus stop. In order to minimize these problems, a context-based strategy was proposed.

Context-based Map Matching. Since the attributes of records are exploited in the map matching task in order to determine the similarity between a pair of records, the lack of comparable attributes reduces the confidence of the map matching results. The matching of urban areas is complex due to the ambiguities involved such as many-to-many correspondences, the positional discrepancy between geographical objects in two data sources, distinct map scales and objects with simplified descriptions or the absence of comparable attributes. This requires the application of context-based matchers [Zhang et al. 2014]. Hence, the map matching approaches can apply contextual information (e.g., surrounding geographical information such as streets, buildings, and urban blocks) to improve the accuracy of the results. The context information can be provided by external data sources in order to support the map matching task.

In this sense, we propose a context-based MATCH-UPS to compare geographical points (e.g., bus stops records). Considering that the only comparable attribute between two data sources is the georeferenced point (latitude and longitude), determining whether or not a pair of records (i.e., points) is a match turns out a challenging and imprecise task. For instance, Figure 1(a) depicts three geographical points (P_1, P_2 and P_3), where each point represents a bus stop. The bus stop P_1 (in blue) is provided by the municipality data source whilst bus stops P_2 and P_3 (in green) are provided by OSM. Since the distances (in meters) from bus stop P_1 to bus stops P_2 and P_3 are equivalent, it is not possible to determine which bus stop (P_2 or P_3) correspondends to bus stop P_1. Thus, the context of the bus stops (in this case, the surrounding streets) is applied to the map matching task in order to assist the matchers in the classification of pairs of records. In this case, the pair $\langle P_1, P_3 \rangle$ is considered a correspondence since the bus stops P_1 and P_3 are at located the same side of the street while the bus stop P_2 is positioned at the other side of the street, as illustrated in Figure 1(b) and 1(c).

4.4. An Efficient Approach for Map Matching Task

In this section, we describe the workflow of the MATCH-UPS approach, which combines blocking techniques and parallel computing to enhance the efficiency of the map matching task. The former groups similar records into blocks and perform comparisons within each block. In this work, we apply part of the geographical coordinates (i.e., blocking key) to block the records. In other words, records that share the same blocking key (based on the

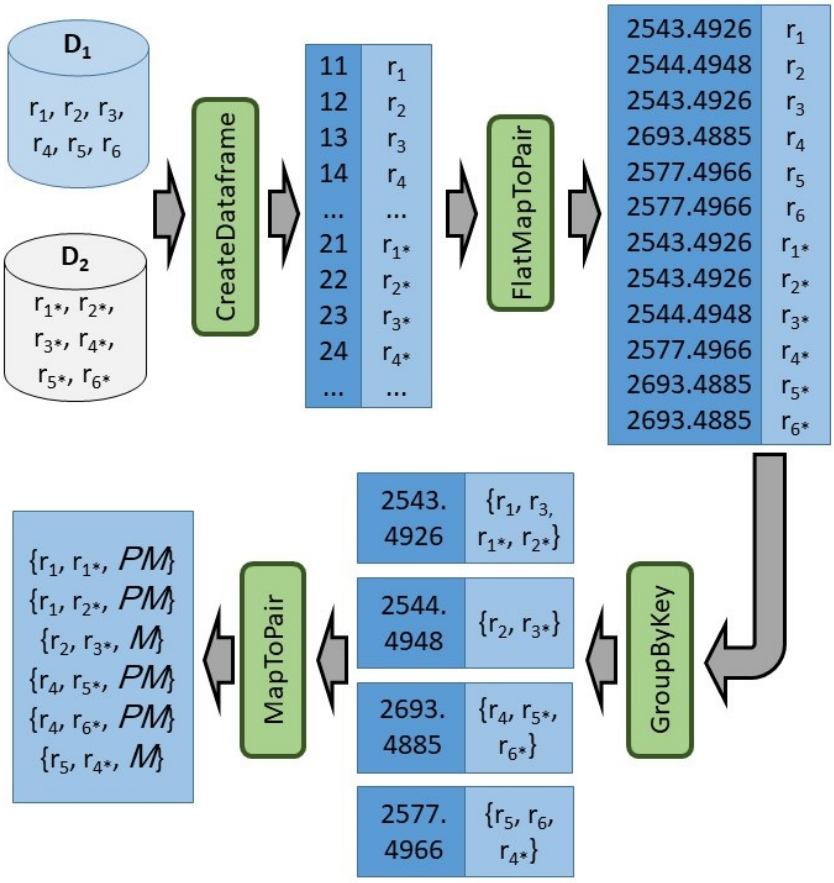

Figure 2. The MATCH-UPS workflow.

geographical coordinates) are grouped into the same block. Since a pair of records located at a long distance from each other has few chances to be similar, the blocking technique prevents that this pair of records from being compared. Regarding the map matching task in parallel, the Spark framework is applied to reduce the overall execution time by distributing the comparison between geospatial records among the available resources. Figure 2 depicts the whole workflow of the MATCH-UPS approach.

The proposed approach receives as input a set of records provided by two data sources D_1 and D_2. To a better understanding, the records (r) provided by D_2 are marked by the symbol "*". Initially, the records (r) are mapped as a key-value pair (i.e., FlatMapToPair operator), such that the key is formed by the data source ID (which provides the record) concatenated with the record ID whilst the value is the record itself. For instance, record r_1 provided by D_1 is mapped to the pair $\langle 11, r_1 \rangle$. These key-value pairs are stored in a DataFrame structure. Posteriorly, the first "n" numbers (in this example, $n = 4$ was applied) of the latitude and longitude coordinates are extracted from each record (if the record is a polygon, the coordinates are extracted from the centroid point). These numbers are concatenated in order to generate the blocking keys. Therefore, each record generates a blocking key composed of the first four numbers of the latitude and longitude coordinates. In the example, the first four numbers of the latitude ("*2543*") and longitude ("*4926*") coordinates of the record r_1 are concatenated to generate the pair $\langle 2543.4926, r_1 \rangle$. In the next step, all records sharing a

common blocking key are grouped into the same block (i.e., GroupByKey operator). For this reason, the records r_1, r_3, r_1* and r_2* are grouped into the same block since all records share the key "*2543.4926*". The blocks determine which records should be compared (by the matchers). Notice that each block is sent to an available resource, where the comparisons between records are performed. Finally, after comparison, the record pairs considered as a match or potential match are joined to compose the MATCH-UPS output (i.e., MapToPair operator). In Figure 2, the output generated for the example is $\{r_1, r_1*, PM\}, \{r_1, r_2*, PM\}, \{r_2, r_3*, M\}, \{r_4, r_5*, PM\}, \{r_4, r_6*, PM\}$ and $\{r_5, r_4*, M\}$.

Load balancing. In the map matching task, the most costly step (in terms of computational cost) is the comparison step, where the matchers are applied to compare the attribute values of each record pair. First of all, it is important to highlight two points: i) all records contained in a block are sent together to a specific resource (e.g., node) to be compared; and ii) since the blocks can have different number of records, the blocks can generate a different amount of comparisons between records. Due to these two points, the comparison step may suffer from load imbalancing problems. Load imbalancing occurs when some nodes execute comparisons for a long time while other nodes remain idle [Araújo et al. 2016].

For instance, in Figure 2, four blocks were generated $b_1 = \{r_1, r_3, r_1*, r_2*\}$, $b_2 = \{r_2, r_3*\}$, $b_3 = \{r_4, r_5*, r_6*\}$ and $b_4 = \{r_5, r_6, r_4*\}$, which generate 4, 1, 2 and 2 comparisons, respectively. Assuming that there are two nodes available, a scheduler can send randomly the blocks b_1 and b_3 to the first node and blocks b_2 and b_4 to the other node. As a result, the task will suffer from the load imbalancing problem since one node will perform six comparisons (4 + 2) while the other node will perform three (1 + 2) comparisons. To minimize the load imbalancing problem, the greedy load balancing technique proposed in [Araújo et al. 2016] is applied. This load balancing technique takes into account the amount of comparisons to be performed in each block to guide the distribution of the blocks among the nodes. To this end, the blocks are ordered according to the amount of comparisons: b_1, b_3, b_4 and b_2. Posteriorly, the top block is removed from the stack and sent to the node with fewer comparisons already allocated, applying a greedy algorithm. Therefore, the blocks b_1 and b_2 are sent to a node and the blocks b_3 and b_4 to the other node. Thus, the load imbalancing is minimized since one node will perform five comparisons (4+1) while the other node will perform four (2+2) comparisons.

5. Evaluation

In this section, we evaluate the effectiveness and efficiency of the MATCH-UPS approach using a cluster infrastructure with five nodes. Each node has 1 core, an Intel(R) Xeon(R) 1.0GHz CPU, 7GB memory, and runs the 64-bit Debian GNU/Linux OS with a 64-bit JVM and Apache Spark 2.0[2]. For the evaluation, we use real-world data sources[3] of Curitiba (Brazil) and New York (USA), provided by Open Street Map[4] and their respective municipalities. Table 2 depicts the number of geospatial records contained in each data source as well as the number of duplicate records (i.e., correspondences) present in each

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

[3]Available in the project's repository.

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

Table 2. Data sources characteristics.

Pairs of Datasets	Municipality	OSM	Duplicates
Curitiba (Parks/Squares)	682	16,189	682
New York (Parks/Squares)	2,008	1,264,799	-
Curitiba (Bus Stops)	6,982	736	6,982
New York (Bus Stops)	3,365	74,140	-

pair of data sources. The data source provided by the Municipality of Curitiba was considered as the gold standard since this data source was cleaned and assessed by the Federal University of Technology (located in Curitiba) and the Institute of Research and Urban Planning of Curitiba (IPPUC[5]). However, we do not have enough information about the quality of the data source provided by the municipality of New York to consider it as the gold standard. Therefore, the cells of the New York data sources do not have value for the "Duplicates" column.

Effectiveness Results. This experiment evaluated the effectiveness of the MATCH-UPS approach over the data sources of Curitiba since there is a gold standard to validate the results. In this sense, three metrics are used to measure the effectiveness: recall, precision and F-measure. It is important to highlight the MATCH-UPS approach applies the same rule, proposed in [Fan et al. 2014], to determine whether or not a pair of geospatial records is considered a match. Therefore, the effectiveness results of the MATCH-UPS approach (without applying the blocking technique) are exactly the same as the results achieved by the approach proposed in [Fan et al. 2014].

Figures 3 (a) and (b) depict the effectiveness of the MATCH-UPS approach over the data sources that store the squares/parks and bus stops of the Curitiba. The goal of this experiment is to evaluate the impact of the blocking technique and context information on the effectiveness of the proposed approach. Regarding the squares/parks data sources, the following MATCH-UPS variations were evaluated: i) MATCH-UPS based on [Fan et al. 2014]; and ii) MATCH-UPS applying the blocking technique. Similarly, concerning the bus stops data sources, the following MATCH-UPS variations were evaluated: i) MATCH-UPS based on [Yang et al. 2014]; ii) MATCH-UPS applying the context information; and iii) MATCH-UPS combining the application of context information and blocking technique.

Based on the achieved results, it is possible to infer that the application of the blocking technique described in Section IV does not affect significantly the MATCH-UPS effectiveness. This behavior demonstrates that the blocking techniques just discarded comparisons with low chances of resulting in matches. Thus, the application of the blocking technique emerges as a useful step to perform the map matching task. Furthermore, we can highlight the 26% increase in the precision results achieved by the MATCH-UPS when the context information was applied, as shown in the Figure 3(b). It occurs due to the fact that this information assists the proposed approach to better classify the pairs of geospatial records. Concerning the recall metric depicted in Figure 3(b), the proposed approach presents low values since the intersection rate between the OSM data source and the municipality data source is only 10%, as described in [Araújo et al. 2017]. In other

[5]http://www.ippuc.org.br/

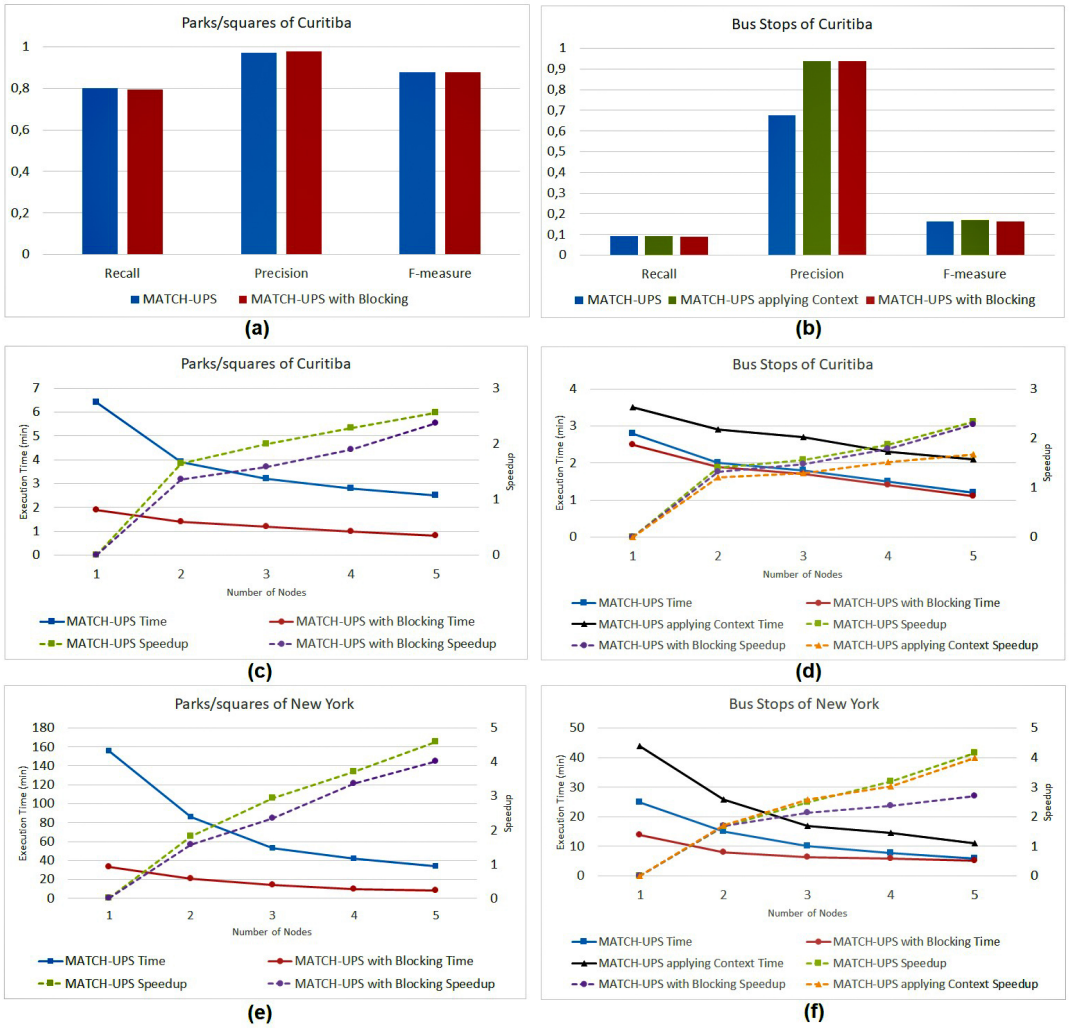

Figure 3. Evaluation of MATCH-UPS approach: (a) Effectiveness results for parks and squares of Curitiba; (b) Effectiveness results for bus stops of Curitiba; (c) Efficiency results for parks and squares of Curitiba; (d) Efficiency results for bus stops of Curitiba; (e) Efficiency results for parks and squares of New York; and (f) Efficiency results for bus stops of New York

words, the intersection between the OSM and the municipality data sources contains only 696 records (of 6,982 records stored in the municipality data source).

Efficiency Results. In this experiment, we evaluated the efficiency of the MATCH-UPS approach and its respective variations, similarly to the effectiveness experiment. It is important to notice that all variations of the MATCH-UPS approach use the load balancing technique described in Section IV. To measure the efficiency, the execution time (given by the average of three executions of each MATCH-UPS variation) and the speedup (which measures how much faster a process runs in parallel) are evaluated.

The efficiency results indicate that the application of parallel computing enhances the efficiency of the map matching task. Considering the MATCH-UPS approach (without blocking technique) for stand-alone mode (i.e., one node) as the baseline approach, it is possible to infer that the proposed Spark-based approach significantly reduces the execution time of the map matching task, as illustrated in Figures 3(c)-3(f). Although the

application of context information improves the effectiveness results, this strategy takes more time to be executed since it needs to process the context information besides to compare the geographical attributes of the records. Regarding the application of the blocking technique, it is important to highlight that the MATCH-UPS with the blocking technique achieved the best results regarding execution time for all experimental scenarios since the technique reduces the search space (i.e., number of comparisons between entities) of the map matching task, as depicted in Figures 3(c)-3(f). In Figure 3(c), with 1 node, the application of the blocking technique reduced the execution time by one third, without significant impact on the effectiveness (as shown in the Figure 3(a)).

Concerning the speedup metric, the MATCH-UPS approach achieved better results when large data sources (e.g. data source of New York) are submitted. It occurs due to the fact that Spark initialization time dominates the MATCH-UPS execution time for small (and medium) data sources [Araújo et al. 2016]. On the other hand, the speedup results of MATCH-UPS for large data sources (illustrated in Figures 3(e) and (f)) denote the scalability of the proposed approach.

6. Conclusion

This article presents MATCH-UPS, a Spark-Based approach to perform map matching in parallel. Context information and a blocking (indexing) technique are applied to enhance the effectiveness and efficiency of the approach, respectively. It is important to highlight that the MATCH-UPS approach can assist several smart cities approaches and environment projects from different countries that face common Big Data challenges, supporting the citizens and changing the way they live. Furthermore, other map matching approaches (e.g., the approaches proposed in [Fan et al. 2014, Araújo et al. 2017]) can benefit from the Spark-based workflow and the blocking technique proposed in this work to enhance the efficiency of them. Based on the experimental experiments, the results show that the MATCH-UPS applying the blocking technique improves the efficiency without significant impact on the effectiveness results. Moreover, the combination of context information and blocking technique enhances the efficiency and effectiveness results.

In future work, we intend to execute the proposed approach over other large geospatial data sources. Furthermore, we aim to extend the proposed approach in order to match line records (e.g., streets and trajectories). Another open area is to propose a map matching approach able to deal with streaming geospatial data.

References

Alarabi, L., Mokbel, M. F., and Musleh, M. (2017). St-hadoop: A mapreduce framework for spatio-temporal data. In *International Symposium on Spatial and Temporal Databases*, pages 84–104. Springer.

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

Araújo, T. B., Pires, C. E. S., da Nóbrega, T. P., and Nascimento, D. C. (2016). A fine-grained load balancing technique for improving partition-parallel-based ontology matching approaches. *Knowledge-Based Systems*, 111:17–26.

Arriagada, J., Gschwender, A., Munizaga, M. A., and Trépanier, M. (2019). Modeling bus bunching using massive location and fare collection data. *Journal of Intelligent Transportation Systems*, 23(4):332–344.

Bojic, I., Massaro, E., Belyi, A., Sobolevsky, S., and Ratti, C. (2015). Choosing the right home location definition method for the given dataset. In *International Conference on Social Informatics*, pages 194–208. Springer.

Bouguelia, M.-R., Karlsson, A., Pashami, S., Nowaczyk, S., and Holst, A. (2018). Mode tracking using multiple data streams. *Information Fusion*, 43:33–46.

Christen, P. (2012). *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media.

Du, H., Alechina, N., Jackson, M., and Hart, G. (2017). A method for matching crowd-sourced and authoritative geospatial data. *Transactions in GIS*, 21(2):406–427.

Fan, H., Yang, B., Zipf, A., and Rousell, A. (2016). A polygon-based approach for matching openstreetmap road networks with regional transit authority data. *International Journal of Geographical Information Science*, 30(4):748–764.

Fan, H., Zipf, A., Fu, Q., and Neis, P. (2014). Quality assessment for building footprints data on openstreetmap. *International Journal of Geographical Information Science*, 28(4):700–719.

Harb, J. G. and Becker, K. (2018). Emotion analysis of reaction to terrorism on twitter. In *Proceedings of the SBC Brazilian Symposium on Databases*, pages 97–108.

Interdonato, R. and Tagarelli, A. (2017). Personalized recommendation of points-of-interest based on multilayer local community detection. In *International Conference on Social Informatics*, pages 552–571. Springer.

Pi, X., Egge, M., Whitmore, J., Silbermann, A., and Qian, Z. S. (2018). Understanding transit system performance using avl-apc data: An analytics platform with case studies for the pittsburgh region. *Journal of Public Transportation*, 21(2):2.

Xavier, E., Ariza-López, F. J., and Ureña-Cámara, M. A. (2016). A survey of measures and methods for matching geospatial vector datasets. *ACM Computing Surveys (CSUR)*, 49(2):39.

Yang, B., Zhang, Y., and Lu, F. (2014). Geometric-based approach for integrating vgi pois and road networks. *International Journal of Geographical Information Science*, 28(1):126–147.

Zhang, C., Zhao, T., and Li, W. (2015). *Geospatial semantic web*. Springer.

Zhang, X., Ai, T., Stoter, J., and Zhao, X. (2014). Data matching of building polygons at multiple map scales improved by contextual information and relaxation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 92:147–163.