A Method for Identifying Patterns of Movement of Trajectory Sets by Using the Frequency Distribution of Points

Vanessa Barbosa Rolim
Computer Science Department
Santa Catarina State University (UDESC)
Joinville, Santa Catarina, Brazil
E-mail: nessabrolim@gmail.com

Marilia Ribeiro da Silva
Felipe Flamarion da Conceição
Cláudio César Gonçalves de Faria Filho
Fernando José Braz
Informatic Department
Federal Institute Catarinense (IFC)
Araquari, Santa Catarina, Brazil
E-mail: {marilia.ribeirods, felipeflamari, claudio.cesar.faria.filho}@gmail.com
E-mail: fernando.braz@ifc-araquari.edu.br

***Abstract*—This work aims to provide a method to identify movement patterns of trajectory sets. The proposal, considering the frequency distribution of points, identifies a set of frequent regions, which can be used to compose a global frequent region. This region represents the area where the set of trajectories executes its movements. Besides, the set of central points, for each individual frequent area, composes the reference trajectory.**

Keywords–trajectory; cluster; movement patterns; similarity.

I. INTRODUCTION

A trajectory can be comprehended as a set of points — represented by latitude, longitude and the time it was recorded. Studies about trajectories of mobile objects have assisted on understanding behaviors and patterns of people and animals mobility, transportation facilities, nature phenomena, and even sports.

One of the areas of study of trajectory is the similarities. That is, when different trajectories share common characteristics or are very close to an established pattern, we say they are similar. Some of characteristics looked on analysis of similarities between trajectories are: length, shape, speed, acceleration, movement, distance between different trajectories, semantic.

Therefore, tasks like clusters of data are useful and can generate knowledge on pattern discovery. In terms of trajectories, the patterns of movement are classified like: moving together pattern, sequential pattern and, periodical patterns [1][2].

According to Zheng [1] a group of objects that move together by a determined period of time show a moving together pattern. Also, if trajectories have a moving together pattern, then, they have shape similarity too.

Several works, like described in Section II, describe methods to find trajectories' similarities using data mining concepts. Besides that, the main approach about these methods is to find frequent trajectories, based in algorithm of association rules like the Apriori, and find clustering patterns based on Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and K-means.

Therefore, this paper proposes a method (*TRAJREF*) to find clusters of trajectories considering frequent coordinates, which generate a reference trajectory.

This work is organized as follows: Section II shows a revision of related works. Section III shows concepts and definitions. Section IV describes the methodology adopted. Section V relates the experiments. Section VI presents the results. Section VII concludes the article and presents perspective of future works.

II. RELATED WORK

Distance functions are important to manage and manipulate trajectories data. They are presented in steps that involve the preprocessing, like the compression and the trajectory discretization. Also they can be used as measures of similarity.

A comparative study made by Wang et al. [3], presents a quantitative analysis of six distance functions used in similarity measurement. The functions are based in measures like Euclidian Distance, Dynamic Time Warping, Edit Distance and Longest Common Subsequence.

Hung et al. [4] proposes an algorithm of clustering trajectories to identify routes and movement behaviors together of an user. The propose consists in a framework of trajectory pattern mining, called *Clustering and Aggregating Clues of Trajectories (CACT)*.

Trajectory Community Discovery using Multiple Information Sources (TODMIS) [5] is a mining framework to discover trajectories communities. According to the author, a group is different of a community. A group is one set of related objects by spatial proximity. A community is a set of objects that have a interaction or a relationship shared by proximity or resemblance.

Shaw and Gopalan [6] developed a method that find frequent trajectories using clustering based sequential patterns mining, called *Clustering Based Sequential Mining* (CBM).

Trajectory Clustering (TRACLUS) is an algorithm that transforms a trajectory in a set of segments [7]. This algorithm is used by *partition-and-group* framework that groups trajectories or similar segments. After the segmentation, the

 ISBN: 978-1-61208-539-5

framework groups the trajectories or the segments that have a together movement pattern similar.

Aung et al. [8] proposes two algorithms of frequent routes discovery, based in data mining algorithm *Apriori*. These algorithms are called *Apriori Based Approximate Frequent Route Miner (A-1)* and *Divide and Conquer Frequent Route Miner (A-2)*. Besides, he used the function of Fréchet distance like similarity measure. This measure is used in curve measures and in the resolution of computation problems.

Considering that if trajectories have similar forms, they also have similar deflection angles between their segments, Melo et al. [9] proposes a method to find shape similarity between trajectories applying statistics correlation in deflection angular vectors. The deflection angle is measured between the extension of previous trajectory segment and the relation of the next segment. This way, to find the deflection angle, it is used the difference between azimuths. Thus, by considering geodesic angles measures, this method shows that even if the trajectories are in totally distinct geographic positions, the shape similarity can be detected.

The previous proposals present methods and algorithms to find sets of similar trajectories. Segmentation, distance and shape matching are the tools to discover the cluster of similar trajectories.

This paper presents a proposal to find similar trajectories considering distance between coordinates points and a trajectory reference composed by a set of frequent points.

III. Basic Concepts and Definitions

This section presents the definitions related to the proposal. The concepts are based in Zheng et al. [10], Fontes and Bogorny [11].

Coordinate *(c): is a tuple (x, y), such that x is a latitude and y is a longitude.*

Point *(p): is a tuple (c, t), such that c is a coordinate and t represents the instant of time when the coordinate was captured.*

Trajectory *(T): is made by a vector of points and can be defined by $T = [p_1, p_2, \ldots, p_n]$, such that p_1 is the initial point, p_n is the final point and n is the number of points.*

Sub-trajectory *(S): is a vector $S = [p_i, p_{i+1}, p_{i+2}, \ldots, p_f]$, such that p_i is the initial point of segment and p_f is the final point of segment, and $0 < p_i < p_f$ e $p_i < p_f < p_n$, n is the number of points of trajectory, that is, $S \subset T$.*

Frequent Point *(fp): is a tuple (p, n), such that p is a point and n is its sample rate, that is, the point repetitions p in a set of trajectories.*

Frequent Area *(FA): is a vector that contains the frequent points fp delimiters for a radius with a distance d, defined as the input parameter of the algorithm and recalculated after the points frequent distribution. Therefore, the FA is a cluster of frequent points. So, a $FA = [fp_i, fp_{i+1}, fp_{i+2}, \ldots, fp_{i+n}]$.*

Candidate Points *(cp): is the frequent point fp that represent the center of a frequent area FA, so a $cp \in FA$. In this way, the candidate point cp is the frequent point fp with higher sample rate and the larger amount of neighbors, therefore the cp represents the cluster geometric center. A candidate point can be represented as $cp = \max\{(fp_i) | fp_i \in FA$ and $0 < i < n)\}$, where n is the vector length FA.*

Reference Trajectory *(RT): is a virtual trajectory T, that is a building trajectory with a set of candidate points cp. Like this, a reference trajectory to be can represented as $TR = [cp_i, cp_{i+1}, cp_{i+2}, \ldots, cp_{i+n}]$.*

IV. Proposed Method

The clusterization of similar trajectories can be realized according to various parameters, such as distance between points, trajectories, trajectories segments, and others [12]. This work aims to propose a trajectories clustering method based in frequency of points in a given area.

The proposal considers that a set of similar trajectories describes a movement into a neighborhood of a reference trajectory. The reference trajectory can be defined considering the frequency of coordinate points into a given radius.

The reference areas are determined according to the frequency that the points in the neighborhood of a reference point occurs. To reduce the computational cost of various trajectories, all points outside an existing area — that are not inside the neighborhood of a candidate point according with a predetermined distance — it is considered a new candidate point.

When a point is inside of two or more areas, it should analyze if this point could be a better candidate point than the points that were previously selected. This analysis occurs only in areas where the point is located. If this area has more points, it is considered a frequency area. Those points that are not in this area are grouped in new areas.

Once every point has been analyzed, and there are not points into two different areas, one reference trajectory is built by using the candidates points.

In order to find the set of reference areas and the trajectory reference, this work proposed the algorithm *TRAJREF*. The algorithm receives a set of trajectories *trajs*, and the maximum distance *d* that a point *p1* of a trajectory *a* must have from another point *p2* of a trajectory *b*. The algorithm returns the areas with higher frequency of points through a list of tuples, in which the first element of the tuple is the *cp*, and the second element is the list of next points that are into the area that encircles the *cp*. Also, the algorithm returns a reference trajectory compose by *cp*'s.

Figure 1 presents the steps proposed in order to find the reference trajectory.

1) **Identify frequent areas:** Go through every points of *trajs*, noting its relations with the *cp*'s that were already identified. The first point contained in *trajs* will be the first *cp*. Each new analyzed point must calculate the distance from *cp*, if this distance is smaller than *d*, this point must be concatenated in the list of near points from *cp*. If the distance is bigger than *d*, this point turns in *cp*. Figure 1(a) presents a set of trajectories (*trajs*), and Figure 1(b) shows the set of *cp*'s, and their neighborhood areas defined by the radius *d*.
2) **Verify *cp* auxiliary list:** Once that every points have been covered, it must analyze the points contained in *cp* auxiliary list. This verification intent to identify

 ISBN: 978-1-61208-539-5

the existence of a *cp* with more near points than the already known *cp*. Therefore, it must compare the distance *d* between the auxiliary list point with the frequency area points that this point is already contained. In case the area of auxiliary point has more points than frequency areas already known, then it must create a new area, which the center is the auxiliary CP. All points that aren't contained in the new frequency area must be inserted in new frequency areas centered in the first point that isn't contained in previous frequency area. Figure 1(c) represents two frequency areas that contain the same point. That point remains to both areas, and is recorded in an auxiliary list.

3) **Exclusion of irrelevant areas:** After verifying all points of the trajectories and identifying the best *cp*'s, the goal is to exclude areas that doesn't have frequent points into a distance *d*. The result of this process is presented by the Figure 1(e).
4) **Building a reference trajectory:** The reference trajectory is composed of a set of *cp*'s.
5) **Return:** The algorithm must return the frequency areas found and the reference trajectory.

V. EXPERIMENTS

This section presents the tools and materials used to develop the methodology. It also details the framework *partition-and-group* and the implementation of algorithm TRACLUS to compare and analyze the results.

A. Trajectories Data

To realize the experiments, one of the goals was reducing the capture and pre-processing cost of trajectories. This way, it was used simulated trajectories on MyMaps. MyMaps is a platform of Google that allows creating and sharing customized maps. Besides, it allows importing and exporting files on *kml* format, compatible formats of Quantum GIS.

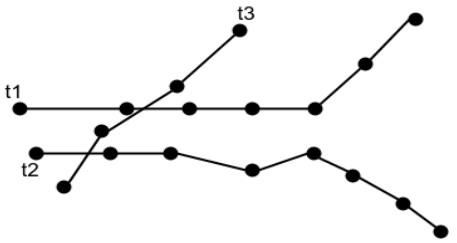
The simulated trajectories represent routes in a region of Itaum neighborhood in the city of Joinville - Santa Catarina - Brazil. Figure 2 shows the simulations on MyMaps; it contains 7 trajectories with starts and finals points close to each others.

B. Proposed Method

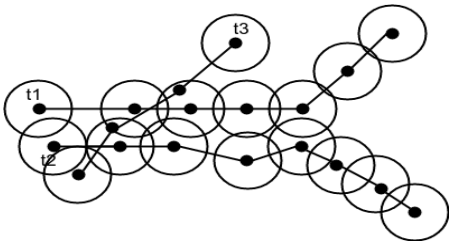
The environment to execute the experiments includes the programming language Python 2.7, the packages 'numpy' version 1.12.0rc2, 'utm' version 0.4.1 and 'heapq' version 8.4. The experiments consider three different values to distance (*d*): 15, 20 and 30 meters. Those values represent the radius that define the neighborhood area around the *cp* points. The reference trajectories were represented in a map in order to compare with the results obtained by using the TRACLUS proposal.

C. TRACLUS Proposal

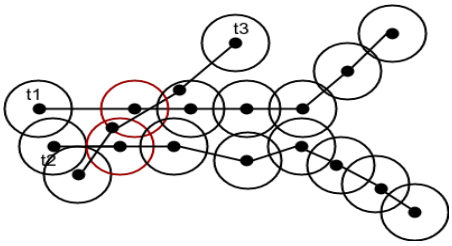
Experiments were made using the framework *partition-and-group* [7] to compare the results of purposed method in this article. The framework consists in two phases. In the partitioning phase, the trajectories are segmented using the minimum description length principle (MDL). The clustering phase considers the density to group similar segments in a cluster.



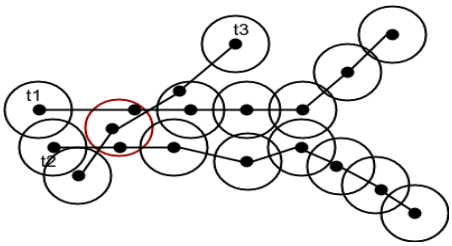
(a) Set of trajectories {t1, t2, t3}.



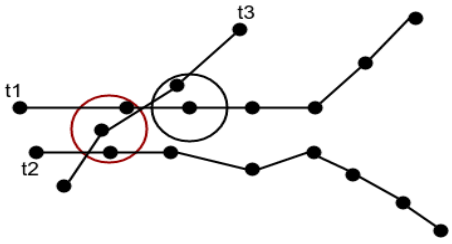
(b) Given a distance *d*, the points are grouped.



(c) If a point is in two areas, it must be considered a candidate point, and a new area is calculated.



(d) If the new area has more points, it is a frequent area.



(e) All areas without neighbors are discarded.

Figure 1. An example of the proposed algorithm.

The TRACLUS proposal was implemented using the package 'traclus_impl' version 0.999 [13] of python language. Besides the set of trajectories, TRACLUS considers another input values:

- ϵ: radius cluster [14][15];
- *min_neighbors*: minimal number of neighbor segments [7];

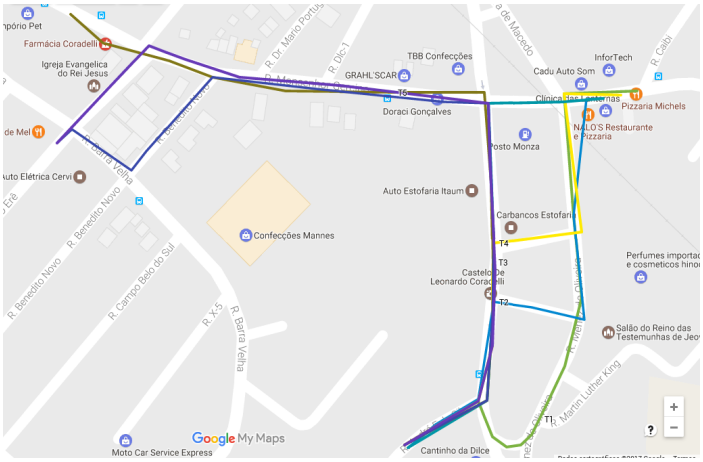


Figure 2. Simulated trajectories from Itaum neighborhood in Joinville - Santa Catarina - Brazil using MyMaps

- *min_num_trajectories_in_cluster*: minimal number of trajectories in order to compose a cluster [7];
- *min_vertical_lines*: minimal number of segments in order to compose a cluster [14].
- *min_prev_dist*: is a smoothing parameter, described by Lee et al. [7], in order to generate a representative trajectory, this concept is similar to the reference trajectory.

The experiments of the TRACLUS proposal were executed considering three values of ϵ 0.00015, 0.00020 and 0.00030. Table I presents the values. The ϵ values are the same to the d values of TRAJREF proposal, however, in a different scale. The other TRACLUS values remain stable during the three experiments.

TABLE I. VALUES VARIATIONS FOR THE INPUT PARAMETERS OF TRACLUS ALGORITHM.

	Experiments		
	1	2	3
ϵ	0.00015	0.00020	0.00030
min_neighbors	2	2	2
min_num_trajectories_in_cluster	2	2	2
min_vertical_lines	2	2	2
min_prev_dist	0.0002	0.0002	0.0002

Considering the values presents in Table I, a cluster will be formed by at least two trajectories, segments with two or more neighbors. Besides, the ϵ variations, in this work, will be used to identify differences in the points sample rate belonging to a cluster.

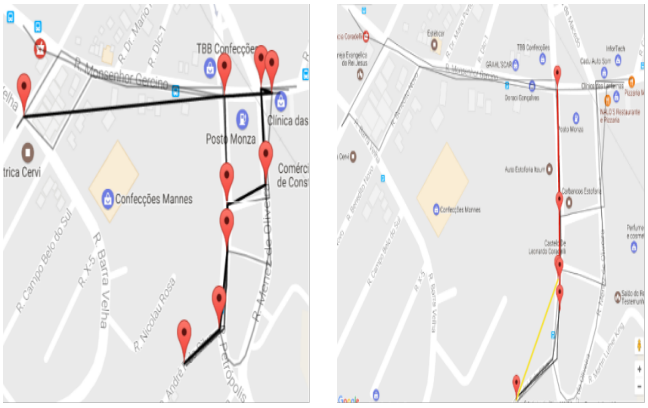
VI. Result Analysis

The analysis of the experiments considers the quantity of frequent points areas and the trajectory composed by those areas.

The variables considered to analyze the results of TRACLUS algorithm were the quantity of representative trajectories obtained, the clusters number and the segments belonging to each cluster number.

Three experiments were conducted considering three different values of d (15, 20 and 30), and ϵ — TRACLUS — (0.00015, 0.00020 and 0.00030). The obtained results were plotted in a map in order to facilitate the visualization and comparison (see Figures 3, 4 and 5) The reference trajectories are represented by the lines with *map icons*.

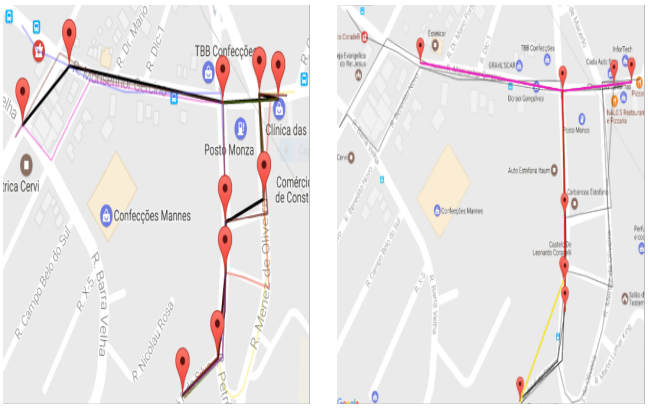
The Figure 3(a) shows the result of the proposed method, with 9 frequency areas (clusters) which form the reference trajectory. Figure 3(b) represents the TRACLUS results, with two representative trajectories, which segments are represented by yellow and red lines. TRACLUS proposal can not identify the cluster in the left side of the Figure 3(a), considering ϵ = 0,00015.



(a) (b)

Figure 3. Experiment 1 (d = 15 and ϵ = 0,00015) - (a) reference trajectory resulting by the proposed algorithm, and (b) representative trajectories by TRACLUS.

Figure 4(a) shows the result of the proposed method, with 10 frequency areas. Figure 4(b) shows the three representative trajectories of TRACLUS. The results are similar. The increase of ϵ value implies an increase in the number of clusters and representative trajectories. Besides, it is possible to verify that when the d value increase, the accuracy of the proposed method increases too.



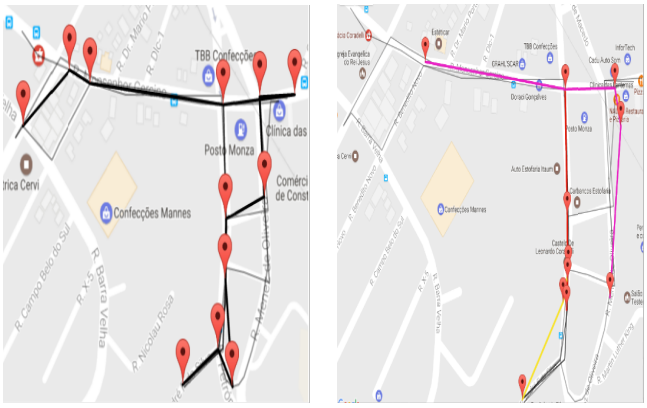
(a) (b)

Figure 4. Experiment 2 (d = 20 and ϵ = 0,00020) - (a) reference trajectory resulting by the proposed algorithm, and (b) representative trajectories by TRACLUS.

Figure 5(a) represents the result of the proposed method, with 12 frequency areas. The Figure 5(b) shows the three representative trajectories of TRACLUS, similar to the previous experiment. In this case, TRACLUS proposal identifies a new segment, represented by a pink line in the figure.

In order to compare the quantitative results, Tables II and III are presented.

 ISBN: 978-1-61208-539-5



(a) (b)

Figure 5. Experiment 3 (d = 30 and ϵ = 0,00030) - (a) reference trajectory resulting by the proposed algorithm, and (b) representative trajectories by TRACLUS.

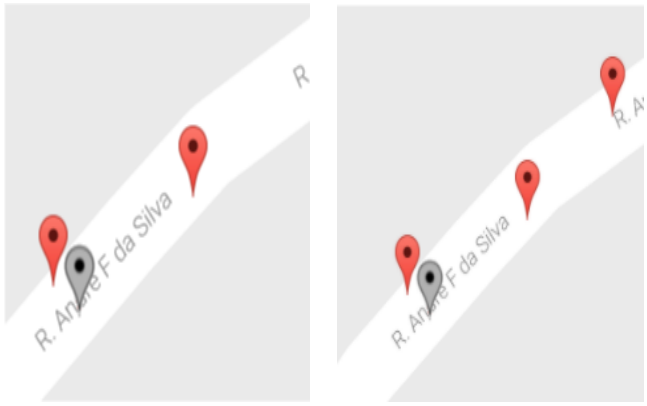
TABLE II. VALUES OF INPUT PARAMETERS - PROPOSED ALGORITHM.

	Experiment		
	1	2	3
n^o of reference trajectories	1	1	1
n^o *of clusters*	9	10	12

TABLE III. VALUES BY INPUT PARAMETERS - TRACLUS.

	Experiment		
	1	2	3
n^o of representative trajectories	2	3	3
n^o of points of the representative trajectories	2, 4	4, 2, 4	5, 3, 4
n^o *of clusters*	2	3	3
n^o *of segments in clusters*	3, 5	7, 3, 5	3, 5, 6

Figures 6(a) and 6(b) represent a *zoom* of the further south point of trajectories. It is a candidate point in all experiments, and will be named by *p1*.

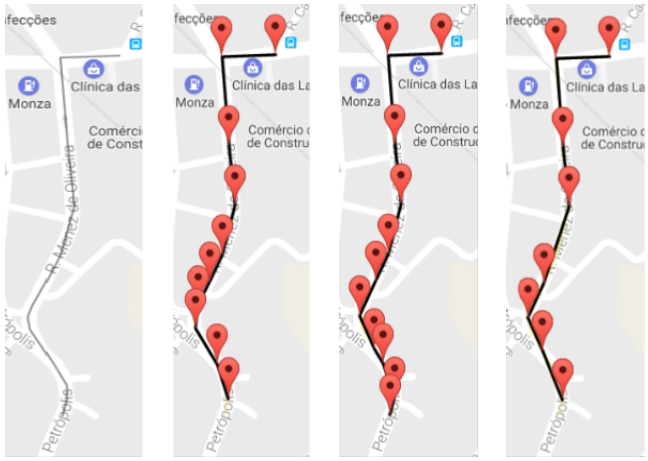


(a) (b)

Figure 6. Zoom in a candidate point - (a) frequent area where d = 20 - (b) frequent area where d = 30

It is possible to verify that when the maximal distance between points decreases, the quantity of points that belong to the frequent area *p1* also decreases. It is an expected behavior of the proposed algorithm and contributes to certify the results.

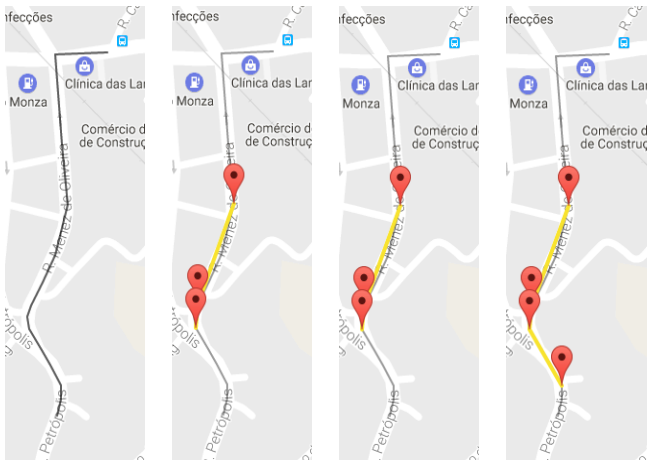
Figure 7 presents the results of the experiments considering two identical trajectories. The proposed algorithm (TRAJREF) finds several frequency areas. Those areas are overlapping the original trajectories, as expected.



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

Figure 7. Experiment with identical trajectories - (a) original trajectory (b) (d = 15) (c) (d = 20) (d) (d = 30).

Figure 8 shows the result of the experiment with TRACLUS proposal considering two identical trajectories. It is possible to note that the proposal returns a representative trajectory over a segment of the original trajectory. The algorithm does not find a representative trajectory matching the complete original trajectory. The partitioning step of the TRACLUS could be a reason for that.



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

Figure 8. Experiment with identical trajectories - (a) original trajectory (b) ($\epsilon = 0.00015$) (c) ($\epsilon = 0.00020$) (d) ($\epsilon = 0.00030$).

It is possible to note that the proposed algorithm (TRAJREF), is efficient to find the reference trajectory with precision and fidelity to the original trajectories presents in the database. Besides, found a greater sampling of points along the reference trajectory.

VII. Conclusion

The prime goal of this work was to present a method to find clusters of trajectories considering the frequency of occurrence of points based in an area defined by a given radius. After finding the clusters, the method returns a reference trajectory, based in the set of clusters.

Experiments allowed to conclude that the proposal (TRAJREF) presents similar results with TRACLUS proposal. How-

 ISBN: 978-1-61208-539-5

ever, the present proposal does not have a segmentation step, an expensive computational process.

The reference trajectory obtained by TRAJREF method presents a high level of fidelity to original trajectories of cluster. However, the propose method, still is not able to identify diferent clusters along reference trajectory, like TRACLUS algorithm. Like this, additional research is necessary.

The future works include the generation of several reference trajectories to representation of different clusters. The segmentation is another future point of research.

A very important point of research is to develop a additional step to prevent that points of the same trajectory are considered neighbor of candidate point.

Finally, it is possible to affirm that this method is suitable to be used in contexts where frequent points in a set of trajectories are a preponderant factor, as urban planning and public transportation.

REFERENCES

[1] Y. Zheng, "Trajectory Data Mining: An Overview," ACM Transactions on Intelligent Systems and Technology, vol. 6, no. 3, may 2015, pp. 29:1 – 29:41, URL: http://dx.doi.org/10.1145/2743025 [retrieved: out, 2016].

[2] Z. Feng and Y. Zhu, "A Survey on Trajectory Data Mining: Techniques and Applications," IEEE Access, vol. 4, apr 2016, pp. 2056 – 2067, ISSN:2169-3536, URL: http://ieeexplore.ieee.org/document/7452339/ [retrieved: out, 2016].

[3] H. Wang, H. Su, K. Zheng, S. Sadiq, and X. Zhou, "An Effectiveness Study on Trajectory Similarity Measures," in Proceedings of the Twenty-Fourth Australasian Database Conference, vol. 137. Australian Computer Society, Inc., 2013, pp. 13 – 22, ISBN: 978-1-921770-22-7, URL: http://dl.acm.org/citation.cfm?id=2525416.2525418 [retrieved: out, 2016].

[4] C.-C. Hung, W.-C. Peng, and W.-C. Lee, "Clustering and Aggregating Clues of Trajectories for Mining Trajectory Patterns and Routes," The VLDB Journal, vol. 24, no. 2, apr 2015, pp. 169 – 192, ISSN: 1066-8888, URL: http://dx.doi.org/10.1007/s00778-011-0262-6 [retrieved: out, 2016].

[5] S. Liu, S. Wang, K. Jayarajah, A. Misra, and R. Krishnan, "TODMIS: Mining Communities from Trajectories," in Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management. ACM, 2013, pp. 2109 – 2118, ISBN: 978-1-4503-2263-8, URl: http://doi.acm.org/10.1145/2505515.2505552 [retrieved: nov, 2016].

[6] A. A. Shaw and N. Gopalan, "Finding frequent trajectories by clustering and sequential pattern mining," Journal of Traffic and Transportation Engineering (English Edition), vol. 1, no. 6, 2014, pp. 393 – 403, ISSN: 2095-7564, URL: http://www.sciencedirect.com/science/article/pii/S2095756415302890 [retrieved: nov, 2016].

[7] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory Clustering: A Partition-and-group Framework," in Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data. ACM, 2007, pp. 593 – 604, ISBN: 978-1-59593-686-8, URL: http://doi.acm.org/10.1145/1247480.1247546 [retrieved: nov, 2016].

[8] H. H. Aung, L. Guo, and K.-L. Tan, Mining Sub-trajectory Cliques to Find Frequent Routes. Springer Berlin Heidelberg, 2013, pp. 92 – 109, ISBN: 978-3-642-40235-7, URL: http://dx.doi.org/10.1007/978-3-642-40235-7_6 [retrieved: nov, 2016].

[9] A. A. de Melo, G. Scheibel, F. Baldo, and F. J. Braz, "A Method for Calculating Shape Similarity among Trajectory of Moving Object Based on Statistical Correlation of Angular Deflection Vectors," in GEOProcessing 2016, The Eighth International Conference on Advanced Geographic Information Systems, Applications, and Services, IARIA, Ed., Venice, Italy, apr 2016, pp. 63 – 68, ISBN: 978-1-61208-469-5, ISSN: 2308-393X, URL: https://thinkmind.org/index.php?view=article&articleid=geoprocessing_2016_4_10_30078 [retrieved: out, 2016].

[10] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining Interesting Locations and Travel Sequences from GPS Trajectories," in Proceedings of the 18th International Conference on World Wide Web. ACM, 2009, pp. 791 – 800, ISBN: 978-1-60558-487-4, URL: http://doi.acm.org/10.1145/1526709.1526816 [retrieved: jan, 2017].

[11] V. C. Fontes and V. Bogorny, "Discovering Semantic Spatial and Spatio-Temporal Outliers from Moving Object Trajectories," CoRR, 2013, URL: http://arxiv.org/abs/1303.5132 [retrieved: dez, 2016].

[12] B. Morris and M. Trivedi, "Learning Trajectory Patterns by Clustering: Experimental Studies and Comparative Evaluation," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 312 – 319, URL: http://ieeexplore.ieee.org/document/5206559/ [retrieved: jan, 2017].

[13] A. Polcyn. traclus_impl 0.999 : Python package index. [Online]. Available: https://pypi.python.org/pypi/traclus_impl [retrieved: jan, 2016]

[14] L. Schauer and M. Werner, "Clustering of Inertial Indoor Positioning Data," in 1st GI Expert Talk on Localization. Department of Computer Science of RWTH Aachen University, 2015, pp. 21 – 23, URL: http://www.cip.ifi.lmu.de/~schauer/publications/ClusteringIndoor.pdf [retrieved: jan, 2017].

[15] B. Liu. Ship Trajectory Clustering Using TraClus Algorithm. [Online]. Available: http://luborliu.me/doc/project_report_TraclusAlgorithm2014.pdf [retrieved: apr, 2014]

 ISBN: 978-1-61208-539-5