DB-SMoT: A Direction-Based Spatio-Temporal Clustering Method

Jose Antonio M. R. Rocha
Depto de Pesca, UFRPE, Brazil and
Informatics Center, UFPE, Brazil
Email: manso@nti.ufrpe.br
Valeria C. Times
Informatics Center, UFPE, Brazil
Email: vct@cin.ufpe.br

Gabriel Oliveira and
Luis O. Alvares
Instituto de Informatica
UFRGS, Brazil
Email: (goliveira,alvares)@inf.ufrgs.br

Vania Bogorny
Depto de Informatica e Estatistica
UFSC, Brazil
Email: vania@inf.ufsc.br

***Abstract*—Existing works for semantic trajectory data analysis have focused on the intersection of trajectories with application important geographic information and the use of the speed to find interesting places. In this paper we present a novel approach to find interesting places in trajectories, considering the variation of the direction as the main aspect. The proposed approach has been validated with real trajectory data associated to oceanic fishing vessels, with the objective to automatically find the real places where vessels develop fishing activities. Results have demonstrated that the method is very appropriate for applications in which the direction variation plays the essential role.**

I. Introduction and Motivation

The increasing use of mobile devices such as cell phones and GPS receivers emerges the need for new methods to analyze the trajectory data generated by these devices.

The data generated by moving objects are normally available as sample points, in the form (tid, x, y, t), where tid is the object identifier and x, y is the geographic location of the moving object at time t. Trajectory sample points have very little or no semantics, what makes their analysis very complex from the application point of view.

Recently, Spaccapietra [1] introduced a new model to reason about trajectories, which is called *stops* and *moves*. This model is specially interesting to add semantic information to raw trajectories. In general, stops are the most important parts of a trajectory from an application point of view, while moves are the movements of the moving object between stops. This model supports a rich number of applications, such as bird migration, where stops can be the countries where birds feed or rest; traffic management, where stops can be traffic lights, roundabouts, speed controllers, traffic jams, and so on.

So far, two different methods have been developed to instantiate the model of stops and moves, called IB-SMoT [2] and CB-SMoT [3]. The first method, IB-SMoT (Intersection-based Stops and Moves of Trajectories) generates stops based on the intersection of trajectory sample points with relevant geographic object types that are specially interesting for the application. This intersection should respect a minimum time threshold for the subtrajectory be considered as a stop. This method is interesting in applications where it is important to find places in which the moving objects have stayed for a minimal amount of time, as for instance, tourism, recreational activities in a park, urban planning, etc.

The method CB-SMoT (Clustering-Based Stops and Moves of Trajectories) is a clustering method based on the speed variation of the trajectory. This method first evaluates the trajectory sample points and generates clusters in places where the trajectory speed is lower than a given threshold for a minimal amount of time. In a second step, the method matches the clusters with a set of user defined relevant geographic places that are important for the application. This method is interesting for applications in which the speed plays an essential role, such as traffic management.

Recently, a new application scenario has emerged for the trajectories of fishing vessels using longlines of type pelagic (also called surface longlines) [4]. Fishing vessels have been monitored via GPS in several places in the world once fishing is under environmental control. However, it is difficult to control whether vessels are fishing only in allowed areas. First, there might not be enough relevant geographic information on the sea to analyze vessel trajectories. Therefore, the method IB-SMoT does not attend the need to control fishing places in order to automatically find the fishing stops. The speed-based method could be used to compute fishing stops, but then we have another problem that fishing vessels have low velocity when they leave the harbor, because they are heavily loaded with oil, water, and food, and also, on the route back, they might have low speed when loaded with caught fishes. So the speed-based method can generate semantically wrong stops.

Apart from the fishing activity application, there are some cases in which the existing methods to instatiate the model of stops and moves are not really the most approapriate. In bird migration, for instance, while birds feed or rest, the speed is very low, but how to differentiate such activities of birds? At the city center, for instance, how to know if a tourist is eating at a restaurant or shopping if the city centre geographic information is not available? At both places the speed of the trajectory will be very low. These applications have lead to the need of new methods to compute stops of trajectories in order to enrich them with context information. During

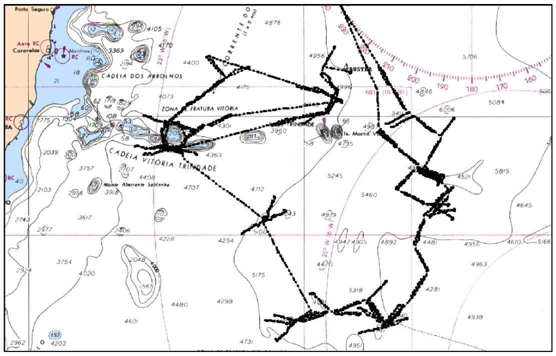

Fig. 1. Fishing vessel trajectory with longline of type pelagic

the fishing activity the main characteristic that identifies the fishing stops is the variation of the direction. This direction-based movement significantly differs from movements of a vessel either going to a fishing area or on the route back, when the course is smoother, as we can see in Figure 1, in a real trajectory of a fishing vessel. In a bird migration application a bird can move in aleatory directions when it is feeding, as a tourist could not move at a restaurant or move in aleatory directions while shopping.

In this paper we propose a novel clustering method to discover stops of trajectories, called DB-SMoT (Direction-Based Stops and Moves of Trajectories), considering direction as the main threshold to find the clusters. We evaluate the proposed method with real trajectories of tunas fishing vessels and compare the results with the method CB-SMoT. The results are evaluated by a user that is a specialist in fishing and that knows, for some trajectories, where the fishing activities have occurred.

The reminder of the paper is organized as follows: In Section 2 we present the basic concepts and some related works. Section 3 presents the proposed approach to discover interesting places in trajectories based on the direction change. Section 4 presents a list of experiments, while Section 5 discusses the results and Section 6 concludes the paper and suggests directions of future works.

II. Basic Concepts and Related Works

In this section we present the definitions of trajectory, stops and moves, according to the general definition of Spaccapietra. This definition is dependent on the particular application that the user is interested in.

Definition 1: A *sample trajectory* is a list of space-time points $\langle p_0, p_1, \ldots, p_N \rangle$, where $p_i = (x_i, y_i, t_i)$ and x_i, y_i, $t_i \in \mathbf{R}$ for $i = 0, \ldots, N$ and $t_0 < t_1 < \cdots < t_N$.

A. Stops and Moves

The exact information aggregated to the trajectory is related to the application context, but essentially the trajectory is partitioned in smaller pieces, called *stops* and *moves*, and the information added gives a more meaningful understanding to trajectories.

Spaccapietra [1] has not specified what information can be aggregated to a stop/move, however he defined some characteristics about them:

Stop: A stop is a part of a trajectory, such that (i) the user has explicitly defined this part of the trajectory to represent a stop; (ii) the temporal extent is a non-empty time interval; (iii) the traveling object does not move, as far as the application view of this trajectory is concerned; (iv) all stops in a same trajectory are temporally disjoint, i.e. the temporal extents of two stops are always disjoint.

Move: A move is a part of the trajectory, such that: (i) the part is delimited by two extremities that represent either two consecutive stops, or t_{begin} and the first stop, or the last stop and t_{end}, or $[t_{begin}, t_{end}]$ (case when the trajectory has no stops); (ii) the temporal extent $[t_{begin}, t_{end}]$ is a non-empty time interval; (iii) the spatial range of the trajectory for the $[t_{begin}, t_{end}]$ interval is the spatio-temporal line (not a point) defined by the trajectory, where t_{begin} is the initial point of the trajectory and t_{end} is the final one.

B. Clustering-Based Approaches

A cluster is a group of data that share a set of similar properties. Several clustering algorithms have been developed in the last few years, but very few algorithms have been developed for spatio-temporal data. The most well known density-based method is DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*), which has been the basis for several other clustering algorithms. DBSCAN requires little knowledge about the domain and needs only two parameters as input: a minimal number of points to build the cluster ($MinPts$) and a given distance around which the points are considered as neighbors (Eps). DJ-Cluster [5] is a density-based algorithm developed to find personal gazetteers in trajectories. It considers only the spatial properties of trajectories. Although it is the closest method to our approach, the baseline (important) places are specified by the user, and DJ-Cluster checks if these places occur in the trajectories, similarly to the method IB-SMoT.

ST-DBSCAN [6] (*Spatial-Temporal DBSCAN*) is a density-based clustering algorithm that deals with both space and time, but not for trajectories. It uses two distance metrics, $Eps1$ and $Eps2$, to define the similarity of spatial and non-spatial values, respectively.

The only known work that considers trajectory direction change is [7], but clustering is over a set of trajectories. This approach is limited to trajectories of vehicles that are constrained to the road network. Vehicles cannot randomly move within the physical space, but must instead follow a set of constraints of the existing road network topology; vehicle movements follow well-understood traffic movement patterns; each vehicle is constrained by the movements of surrounding vehicles; vehicles generally travel in a single direction. The objective of this work is to form groups of vehicles for wireless communication between them, and direction is used to avoid including in the same cluster vehicles traveling in different directions. In our method, we do clustering of one single trajectory and we consider the variation of direction of the same moving object and not of a set of objects. While in [7] the trajectory is restricted to the road network we cluster only the trajectory itself without any external information such as a road network. We do not restrict the moving object to follow a set of constraints. We use the raw trajectory data as they are generated by the mobile device. The interestingness of our approach in relation to any other is exactly the fact that we find interesting places in single trajectories without the need of additional information.

Most of the works on trajectory clustering [8], [9], [10], [11] focus on trajectory similarity or dense regions, and considering a set of trajectories. Another problem is that most of existing approaches do not deal with the spatio-temporal properties of trajectories, dealing either with the spatial properties or the temporal ones. The works presented in [2] and [3] are the closest to our approach, since they respectively compute stops of trajectories based on the intersection of trajectories and geographic data and compute low speed clusters.

III. Direction-based Stops and Moves of Trajectories

The course of a given vessel is defined according to the direction of its movements that is given by the angle between this direction and the north direction, varying clockwise from 0 to 360 degrees. A fishing vessel follows a pattern when it is moving to a fishing area, basically moving in a given course. The variation of the vessel movement direction is low, i.e., there are only small corrections in the course. Imagine that a vessel is moving with course *north* to a fishing area. Only very small oscillations will happen while moving to a fishing region. For a vessel to change its course to more than 20 degrees, like 45 degrees or 90 degrees for instance, it would basically mean that the vessel has completely changed direction. Although a brutal course change could happen, for instance, the vessel can perform a complete turn to rescue a fisherman that fell down into the sea (i.e. in man overboard situation), this is not an usual movement.

When a vessel is fishing with pelagic longline the behavior of its movement changes completely for each fishing day. When fishing with longline the vessel has three typical activities that are developed according to biotic and abiotic oceanographic factors: set the longline, wait some time, and finally, haul the longline [12]. During this period the vessel has its movement influenced by the waves, the wind direction, movements to avoid the proximity of the longline from the propeller, etc.

Considering the direction of the trajectory as the main variable we use two main measures to characterize a fishing area: the frequent vessel direction variation and minimal time duration of this standard.

A. Definitions

This section presents some definitions to precisely specify the used concepts.

Definition 2: Direction Change.
Let $< p_{i-1}, p_i, p_{i+1} >$ be a subtrajectory. The direction change at p_i is the angle between the directions $\overline{p_{i-1}, p_i}$ and $\overline{p_i, p_{i+1}}$, denoted by $DC(p_i)$.

Definition 3: Candidate-cluster-point.
Let $< p_{i-1}, p_i, p_{i+1} >$ be a subtrajectory. The point p_i is a candidate-cluster-point with respect to $minDC$ if $DC(p_i) \geq minDC$.

The threshold $minDC$ specifies the minimum direction change at a point p_i in order to this point be considered as a candidate cluster point.

Definition 4: Connected-candidate-point.
Let $\langle p_i, p_{i+1}, p_{i+2}, \ldots, p_{i+n+1} \rangle$ be a subtrajectory. The point p_i is connected-candidate-point to p_{i+n+1} with respect to $minDC$ and $maxTol$ if p_i and p_{i+n+1} are candidate-cluster-points and $n \leq maxTol$.

The maximal tolerance threshold $maxTol$ specifies the maximum number of trajectory points with direction change less than the $minDC$ threshold that can be found consecutively in a cluster.

Definition 5: Trajectory cluster.
A cluster $C = \langle p_1, p_2, \ldots, p_n \rangle$ of a trajectory T with respect to $minDC$, $maxTol$ and $minTime$ is a non-empty subtrajectory of T formed by a set of contiguous time-space points such that:

1) $\forall p, q \in T$: if $p \in C$ and p is a connected-candidate-point to q with respect to $minDC$ and $maxTol$, then $q \in C$.
2) $\forall p, q \in C$: p is connected-candidate-point to q with respect to $minDC$ and $maxTol$.
3) $t_n - t_1 \geq minTime$, where $p_i = (x_i, y_i, t_i)$

B. The Algorithm DB-SMoT

The proposed method DB-SMoT finds clusters in single trajectories based on the direction change as defined in the previous section. Listing 1 shows the pseudo-code of the algorithm, which has as input a set of trajectories represented as sample points, the minimal direction change threshold, the minimal amount of time to generate a cluster and the maximal tolerance to evaluate the variation of the direction. We start by checking the number of points of the trajectory (line 10) and among every two points we compute the direction variation (line 12-14). Having computed the direction variation, the method *findClusters* (line 16) is called to find the clusters. For all subtrajectories which are not in clusters (stops), we generate the moves (line 17-25).

The method *findClusters*, detailed in Listing 2, starts by checking the variation of the direction among every two points of the trajectory (line 5). While the variation passes the minimal direction change threshold (MinDirChange) the points are added to the cluster (line 6-7). When a point does not variate its direction (line 8) we check the maximal tolerance, in order to verify if the point that has not changed the direction was noise or if the direction change has ended (line 9-17). After adding the points that have enough direction variation to the cluster, we check if it passes the minimal time duration constraint (line 19-23). In positive case, we add the cluster to a list of *AllClusters*.

Listing 1. DB-SMoT pseudo-code

```
INPUT
 T               //Trajectory sample points
 MinDirChange    //Minimal direction change
 MinTime         //Cluster minimal time
 MaxTol  //Maximal tolerance
OUTPUT:
 S               //Set of Stops
 M               //Set of Moves
METHOD:
 n = sizePoint(T)
 //PRE-METHOD: evaluating direction change
FOR i from 2 to n DO
        variation[i-1] = calculateVariation(i,i-1)
ENDFOR
//CLUSTERING
Clusters = findClusters(T,MinDirChange,MinTime,MaxTol)
//FINDING MOVES
FOR i from 1 to n DO
        IF ( p_i is not in a Stop )
                Move = Move + {p_i}
        ELSE
                M = M + {Move}
                Move = {}
        ENDIF
ENDFOR
ENDMETHOD
```

Listing 2. findClusters pseudo-code

```
1  METHOD findClusters
2  i=1; n = sizePoint(T); clusterOpened = false
3  AllClusters = {}; Cluster = {}
4  WHILE (i <= n) DO
5   IF ( variation[i] > MinDirChange )
6            Cluster = Cluster + {p_i}
7                  clusterOpened = true
8   ELSE
9    IF (clusterOpened) //if there is a cluster
10    //check direction change of next points
11    lastIndex = lookAhead(MaxTol,MinDirChange)
12   IF (lastIndex <= i+MaxTol)
13    // add the points to the cluster
14          FOR j from lastIndex to i DO
15                  Cluster = Cluster + {p_j}
16          ENDFOR
17          i=lastIndex
18   ELSE //close the cluster
19          IF( time(Cluster) > MinTime )
20                  AllClusters = AllClusters +{Cluster}
21          ENDIF
22          Cluster = {}
23          clusterOpened = false
24   ENDIF
25  ENDIF
26 ENDIF
27 i++
28 ENDWHILE
29 RETURN AllClusters
30 ENDMETHOD
```

The complexity of the DB-SMoT algorithm is *O(P)*, where *P* the number of trajectory points, since each point of the trajectory is analyzed only once.

IV. Experimental Results

We implemented the proposed method in the Weka data mining toolkit, which we are extending to both preprocessing and mining trajectory data. We evaluate the proposed method with real trajectories of fishing boats, while in future works we will analyze trajectories of migration birds. For each trajectory there is a report, that by the Brazilian law has to be provided by every fishing vessel. Among other information, we have

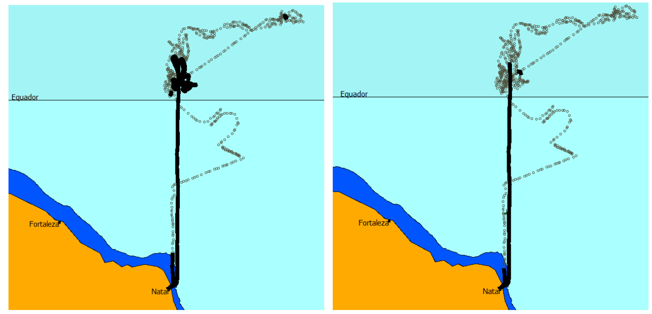

Fig. 2. (left) CB-SMoT with 60% average speed of the trajectory and 2 hours minimun time, (right) CB-SMoT with 40% average speed of the trajectory and 2 hours minimun time

the fishing period, the average fishing time for settings and haulings, the number of settings and haulings and the amount of captured fish. Because of space limitations we show the results for only two fishing trajectories. In these trajectories a GPS point is collected every 30 minutes. We concentrate on the analysis of the discovered clusters, trying to identify if the clusters correspond to the fishing areas.

The first trajectory is of the Seneca vessel that left the Brazilian coast and went for fishing during 22 days. This vessel was anchored at the harbor of Natal, from where it left to the fishing area, and 22 days after it returned to the same harbor. Although we know that for this vessel the fishing time was 6 hours we tested the methods using also 2 hours as minimum fishing time. We first applied the method CB-SMoT, which is based on the speed of the trajectory. We considered two hours as the minimum fishing time and speed as 60 % of the average speed of the trajectory in order to generate the stops, which should be the supposed fishing places. As can be seen in Figure 2 (left), with these parameters some fishing areas just above the equator where identified as clusters, but a stop from the departure of the vessel to the end of the continental shelf, and a very long one in the return of the vessel to the harbor was generated, what clearly does not correspond to fishing areas. In order to try do avoid this long cluster in the return of the vessel, we reduced the speed to 40% of the average speed of the trajectory. However, as shown in Figure 2 (right), the results were even worse, because we obtained clusters only on the move leaving and returning to the harbor, and almost nothing from the real fishing areas. What happened was that the vessel had a very good fishing period and it returned heavily loaded to the harbor at a very low speed. Figure 3 (left) shows the same trajectory but considering the CB-SMoT method with parameters 6 hours as the minimal time and 60% of the average speed of the trajectory. As can be seen, the result is very similar to Figure 2 (left).

Figure 3 (right) shows the same trajectory with stops computed using the direction based method DB-SMoT. We also considered 2 hours as the minimum time for a fishing place and considered 10 degrees as the minimal direction change and zero points as the maximal tolerance. In total, 25 stops were generated, most of them being over the fishing regions.

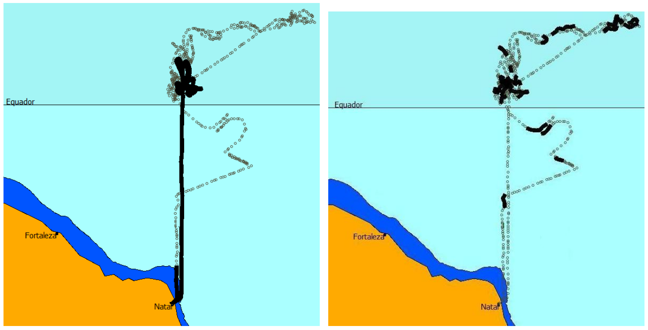

Fig. 3. (left) CB-SMoT with 60% average speed of the trajectory and 6 hours minimun time (right) DB-SMoT with 10 degrees minimum direction change, 2 hours minimum stop time and without maximal tolerance

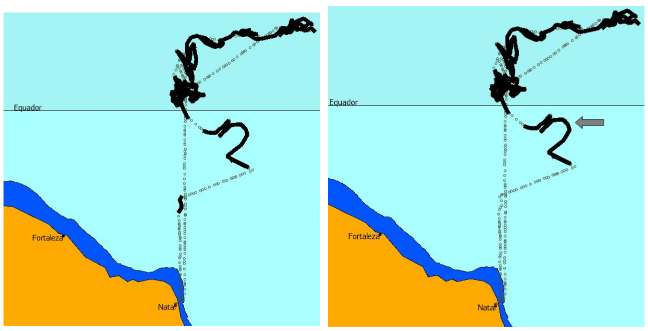

Fig. 4. (left) DB-SMoT with 10 degrees direction change, 2 hours minimum time, 5 points maximal tolerance (right) DB-SMoT with 10 degrees minimum direction change, 6 hours minimum time, 5 points maximal tolerance

With this method, no stop was generated near the harbor. One problem is that on the fishing area, several small stops are generated, which do not really cover the entire fishing region. Then we performed an experiment increasing the maximal tolerance to 5 points, getting the results shown in Figure 4 (left), which better cover the fishing areas, without having a stop when the vessel either leaves or arrives at the harbor.

For this vessel trajectory, the average time of each fishing activity was around 6 hours. Considering 6 hours as the minimal fishing time, with 10 degrees minimal direction change and 5 points as maximal tolerance, we get the results shown in Figure 4 (right), covering very well the fishing areas. The only problem is the big turn at the beginning of the fishing area (represented with a pointer in Figure 4 (right)) that in fact connects two small fishing regions.

Figure 1 shows an image of a trajectory over the background information of sea depth, where there are shallow sea mounts with only 50 meters depth. At these places there is a large amount of sunlight that penetrates the water column reaching the substrate layer, facilitating primary production and as a consequence being attractive to plankton, which in turn attract many small fishes and finally attracting big fishes searching for food. As a result, these regions are good fishing areas.

Figure 5 shows an example of clusters computed with the method CB-SMoT on a second trajectory, using 60% of the trajectory mean speed as the minimum speed for a cluster

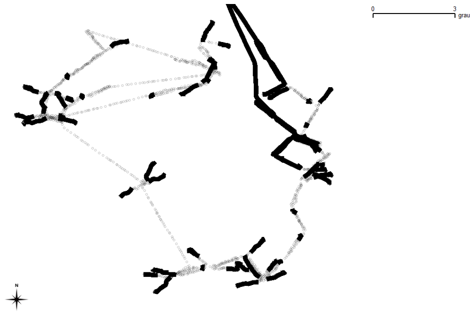

Fig. 5. CB-SMoT method, 60 % average speed, and 2 hours minimum time

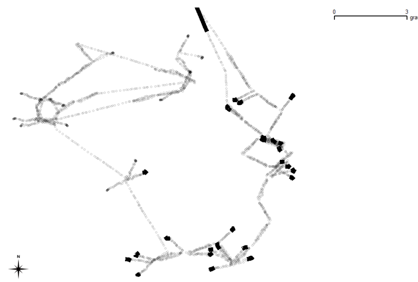

Fig. 6. CB-SMoT with 30 % average speed, and 2 hours minimum time

(stop) and 2 hours as the minimun time for the stop duration, characterizing the real fishing stops average time. Again, as in the previous experiment, two long clusters were generated on the top right that do not correspond to fishing areas. Then we decreased the average speed to 30%, and still 2 hours as the minimun time for the stop duration. The result of this experiment is shown in Figure 6. It generated, as expected, less clusters, and they do not completely cover each fishing region. Furtheremore, the biggest cluster found does not correspond to a fishing area.

Better results were found with method DB-SMoT. Among several values for the minimal direction change parameter, the best results were obtained with 10 degrees. In Figure 7 we used 2 hours as the minimal stop duration, without considering the maximal tolerance. Again, with the direction-based method we do not have clusters in the displacement of the vessel from or to the harbor. However, with these parameters we have several small clusters over big fishing areas what is not a good result. We solve this problem considering the maximal tolerance. With this parameter set as 5 points we get a much better result, as shown in Figure 8, which according to the specialist user, covers very well the fishing places.

The maximal tolerance parameter extends the cluster in order to cover noisy points inside a fishing area, so it plays an essential role in the proposed method.

V. Discussion

In this paper we have shown some of several experiments performed on trajectories with different characteristics. For each trajectory a different fishing technique has been applied, and the proposed method has shown to cover different types of fishing trajectories. To our surprise, in general the fishing

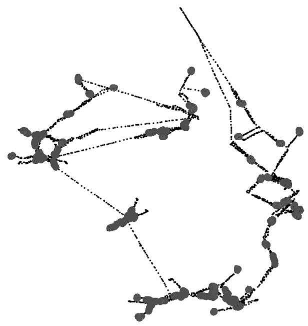

Fig. 7. DB-SMoT method, 10 degrees minimal direction change, 2 hours minimal stop duration and zero points as maximal tolerance

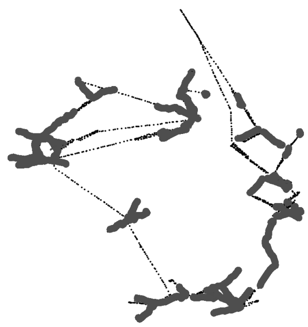

Fig. 8. DB-SMoT with 10 degrees minimal direction change, 2 hours minimal stop duration and 5 points maximal tolerance

activities correspond to small direction variation. This might be explained because GPS points are collected in an average of each 30 minutes or more. In future works we will collect GPS data of fishing vessels decreasing the interval between collected points. What we can conclude for the set of results is that the minimum direction change parameter around 10 degrees in general characterizes the fishing areas.

The minimum time parameter varies according to the fishing technique and the technical characteristics of the fishing vessel. It should be great enough to avoid direction change during a small time period, that does not correspond to fishing activity, but it should not be greater than the typical duration of a fishing activity in the considered trajectory.

In terms of efficacy, the results obtained with the method DB-SMoT in comparison to the fishing activity reports obtained for the evaluated fishing trajectories, DB-SMoT was able to identify around 90% of the the fishing places. The complexity of the method DB-SMoT is similar to the complexity of CB-SMoT, as explained in the previous section. The same occurs for the execution time. We compare our method to CB-SMoT, which is the only known clustering method for single trajectories.

VI. Conclusion and Future Works

In this paper we proposed a novel clustering method to find interesting places in single trajectories considering the direction variation of the movement as the main threshold. We have presented new concepts for trajectory clustering and proposed an algorithm which discovers interesting places (clusters) based on a minimum direction change for a minimal time.

We compared the proposed algorithm with the method CB-SMoT, which has the same objective as our approach, but that considers the variation of the speed as the main metric to discover interesting places in trajectories. Experiments were performed with real trajectory data of fishing boats, and the results have demonstrated to be very useful to automatically discover the fishing areas. As future works we are extending the method DB-SMoT to consider not only direction but the combination of speed and direction and testing the method in other application domains.

VII. Acknowledgments

The authors would like to thank CAPES, CNPq (307588/2008-4), and FAPESC (CP005/2009) which partially supported this research.

References

[1] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. de Macedo, F. Porto, and C. Vangenot, “A conceptual view on trajectories,” *Data and Knowledge Engineering*, vol. 65, no. 1, pp. 126–146, 2008.

[2] L. O. Alvares, V. Bogorny, B. Kuijpers, J. A. F. de Macedo, B. Moelans, and A. Vaisman, “A model for enriching trajectories with semantic geographical information,” in *ACM-GIS*. New York, NY, USA: ACM Press, 2007, pp. 162–169.

[3] A. T. Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares, “A clustering-based approach for discovering interesting places in trajectories,” in *ACMSAC*. New York, NY, USA: ACM Press, 2008, pp. 863–868.

[4] H. G. Hazin, F. H. V. Hazin, P. E. P. F. Travassos, F. C. de Carvalho, and K. Erzini, “Fishing strategy and target species of the brazilian tuna longline fishery,from 1978 to 2005, inferred from cluster analysis,” *Collective Volume of Scientific Papers. International Commission for the Conservation of Atlantic Tunas*, vol. 56, pp. 1942–1951, 2006.

[5] C. Zhou, D. Frankowski, P. J. Ludford, S. Shekhar, and L. G. Terveen, “Discovering personally meaningful places: An interactive clustering approach,” *ACM Trans. Inf. Syst.*, vol. 25, no. 3, 2007.

[6] D. Birant and A. Kut, “St-dbscan: An algorithm for clustering spatial-temporal data,” *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208–221, 2007.

[7] P. Fan, A. Mohammadian, P. C. Nelson, J. Haran, and J. Dillenburg, “A novel direction based clustering algorithm in vehicular ad hoc networks,” in *86th Annual Transportation Research Board Meeting*, 2007.

[8] J.-G. Lee, J. Han, and K.-Y. Whang, “Trajectory clustering: a partition-and-group framework,” in *SIGMOD Conference*, C. Y. Chan, B. C. Ooi, and A. Zhou, Eds. ACM, 2007, pp. 593–604.

[9] P. Laube and S. Imfeld, “Analyzing relative motion within groups of trackable moving point objects,” in *GIScience*, ser. Lecture Notes in Computer Science, M. J. Egenhofer and D. M. Mark, Eds., vol. 2478. Springer, 2002, pp. 132–144.

[10] P. Laube, S. Imfeld, and R. Weibel, “Discovering relative motion patterns in groups of moving point objects,” *International Journal of Geographical Information Science*, vol. 19, no. 6, pp. 639–668, 2005.

[11] H. Cao, N. Mamoulis, and D. W. Cheung, “Discovery of collocation episodes in spatiotemporal data,” in *ICDM*. IEEE Computer Society, 2006, pp. 823–827.

[12] J. C. Sainsbury, *Commercial Fishing Methods: An Introduction to Vessel and Gear*. Surrey: Fishing News (Books) Ltd, 1971.