Automated Traffic Route Identification through the Shared Nearest Neighbour Algorithm

Maribel Yasmina Santos[1], Joaquim Silva[2], João Moura-Pires[3], Monica Wachowicz[4]

[1] Algoritmi Research Centre, University of Minho, Campus de Azurém, Portugal, maribel@dsi.uminho.pt
[2] School of Technology, Polytechnic Institute of Cávado and Ave, Portugal, jpsilva@ipca.pt
[3] Faculty of Science and Technology, New University of Lisbon, Portugal, jmp@di.fct.unl.pt
[4] Geodesy and Geomatics Engineering, University of New Brunswick, Canada, monicaw@unb.ca

Abstract

Many organisations need to extract useful information from huge amounts of movement data. One example is found in maritime transportation, where the automated identification of a diverse range of traffic routes is a key management issue for improving the maintenance of ports and ocean routes, and accelerating ship traffic. This paper addresses in a first stage the research challenge of developing an approach for the automated identification of traffic routes based on clustering motion vectors rather than reconstructed trajectories from AIS data sets. The immediate benefit of the proposed approach is to avoid the reconstruction of trajectories in terms of their geometric shape of the path, their position in space, their life span, and changes of speed, direction and other attributes over time. For clustering the moving objects, an adapted version of the Shared Nearest Neighbour algorithm is used. The motion vectors, with a position and a direction, are analysed in order to identify clusters of vectors that are moving to-

wards the same direction. These clusters represent traffic routes and the preliminary results have shown to be promising for the automated identification of traffic routes with different shapes and densities, as well as for handling noise data.

1 Introduction

The Automatic Identification System (AIS) is now fitted to all commercial ships and is proving a great advantage in tracking and identifying ships along coastal routes and in port waters, where Vessel Traffic Services (VTS) operators use the identification tags constantly in conjunction with their radar pictures. AIS is a unique program that provides a means for ships to electronically broadcast ship data at regular intervals including: ship identification, position, course, and speed. AIS uses Global Positioning Systems (GPS) in conjunction with shipboard sensors and digital VHF radio communication equipment to automatically exchange navigation information electronically. Ship identifiers such as the ship name and VHF call sign are programmed in during initial equipment installation and are included in the transmittal along with location information originating from the ship's global navigation satellite system receiver and gyrocompass. AIS is used by marine ships in coordination with VTS to monitor ship location and movement primarily for traffic management, collision avoidance, and other safety applications (Perez et al., 2009).

Extracting the traffic routes where ships are located is a necessary element of Maritime Domain Awareness to achieve an “effective understanding” of maritime activity and its impact on safety, security, the environment and the economy. Towards this objective the Shared Nearest Neighbour (SNN) algorithm is used with a proposed distance function to analyse the AIS data associated with the positions of the ships and identify the main traffic routes that were followed by them. We propose clustering motion vectors, which are geometrical primitives, having an explicit magnitude, here represented by the bearing associated to them.

The proposed approach has as benefit of avoiding the need to reconstruct the trajectories, a multipart process associated with complex spatiotemporal constructs that usually include characteristics like the geometric shape of the path, its position in space, the life span, and the dynamics of the movement, that is how speed, direction and other point-related attributes change over time (Rinzivillo et al., 2008).

The contribution of this paper is twofold: i) propose a clustering approach based on a density-based algorithm that automatically identifies

traffic routes from motion vectors; ii) adjust the input parameters of the clustering algorithm as well as the weight factor of the distance function according to the ships' movements.

This paper is organised as follows. In the next section, the current research work on clustering movement data is described. Section 3 presents the data set available for our analysis. In section 4, we describe the clustering algorithm used to automatically extract traffic routes. Section 5 discusses the obtained results. Finally, section 6 provides the main conclusions and insights for future research work.

2 Related Work

Clustering is the process of grouping a set of objects into clusters in such a way that objects having high similarity with each other are placed within a cluster, and they are as dissimilar as possible to objects in other clusters (Zaït and Messatfa, 1997; Grabmeier, 2002). Trajectory clustering algorithms group trajectories using similarity measures that are computed from the spatial attributes (e.g. proximity in geometric shape) as well the temporal attributes (e.g. speed variations) of moving objects (Little and Gu, 2001; Chen et al., 2005; Giannotti and Pedreschi, 2008; Miller and Han, 2009).

Meratnia and By (2002) proposed distinct trajectory clustering approaches for dealing with multidimensional time series and potential noise. The first approach is based on a spline algorithm that supports a symbolic representation of a trajectory obtained from position time series of a moving object. The spline representation of trajectories enables to derive re-discretised position time series of moving objects with synchronised and constant sampling rate. In this case, the clustering uses a naïve technique to define the trajectory similarity based on the shortest distances between the re-discretised positions of the two trajectories being compared. Two positions from different trajectories are considered similar if they are within a pre-defined threshold distance.

However, this naïve similarity definition has a non-transitive characteristic and the calculations involved in the computations of distance threshold similarity are time complex. To overcome these problems, Meratnia and By (2002) have also proposed two raster-based clustering approaches based on spatial homogeneous units and spatiotemporal units respectively. In the spatial unit clustering, the raster cell size defines the distance threshold to be used as similarity measure. The fundamental notion is to determine similar trajectories by assigning the trajectory visits to each cell.

They claim that this raster-based approach is advantageous in relation to the previous approach due to its fewer computations, independency from individual trajectories, and easier generalisation of information. The spatiotemporal unit clustering follows the same principle, but in this case, the similarity measure is computed by the number of trajectories hits per cell during a certain time interval. Unfortunately, all the three approaches have not been fully implemented and there are not experimental results.

Giannotti et al. (2007) developed an extension of the sequential pattern mining paradigm to analyse trajectories of moving objects. Trajectory patterns are descriptions of frequent behaviours both in space (e.g. regions visited during the movements) and in time (e.g. the duration of the movements). Every trajectory pattern integrates individual trajectories that visited the same sequence of places with similar travel times. In this approach the notions of regions of interest (that emerge from the analysed space) and the typical travel time between regions (that also emerge from the input trajectories) are used to obtain a sequence of spatial regions that are most visited. This means that the individual trajectories integrated in a pattern are not necessarily simultaneous, as the authors only require that the trajectories visit the same sequence of places with similar transition times and not at the same time. This approach requires the definition of potentially useful spatial regions that guide the extraction of trajectory patterns from source trajectory data.

Lee et al. (2007) proposed a partition and group framework that partitions a trajectory into a set of line segments and then group similar line segments into a cluster. The authors pointed out the advantage in discovering similar sub-trajectories from a trajectory database, since smaller portions of the trajectories can be identified. In the partition and group framework, the approach starts with a partitioning algorithm that splits the several trajectories into a set of sub-trajectories based on the minimum description length principle. After that, a density-based clustering algorithm is used for grouping these line segments. A distance function was defined to set the density parameter of the line segments, based on the perpendicular distance, the parallel distance and the angle distance of the line segments. Related to partitioning, the partition process should verify two properties: preciseness and conciseness. Preciseness means that the difference between a trajectory and its set of sub-trajectories should be as small as possible. Conciseness means that the number of trajectory partitions should be as small as possible.

For automated clustering of trajectories using Nearest Neighbour Clustering, Vlachos et al. (2002) identified the several issues that a distance function should address, including how to deal with different sampling rates and speeds; detect similar motions in different space regions; be ro-

bust to noise and outliers; deal with trajectories with distinct number of positions; and allow efficient computation of similarity. To cope with these issues, they proposed a non-metric similarity function based on the Longest Common Subsequence (LCSS) and demonstrated that the function performs well for noisy signals. Chen et al. (2005) introduced a novel distance function, Edit Distance on Real sequence and proved that it was more robust to noise and more accurate than other popular distance functions, including LCSS-based functions.

To support the visual analysis and exploration of large number of trajectories, Rinzivillo et al. (2008) proposed a progressive clustering approach where a simple distance function with a clear meaning is applied on each step. A distance function that incorporates all the characteristics of trajectories would be complex and very difficult to interpret. As an alternative, the authors proposed a library of well interpretable distance functions, where each function is based on a subset of trajectory attributes. The progressive clustering means that the outcomes can be used as the input to further clustering, where a different distance function can be chosen. The authors argue that this approach can increase the performance of the clustering and the user ability to interpret the results.

The authors developed a library with four distance functions that compare the trajectories according to some selected spatial and spatiotemporal properties to assess similarity between trajectories. They applied this approach to a dataset of GPS-tracks of cars using a density-based clustering algorithm, where the input parameter that defines the minimum number of neighbours of a core point was tuned according to the density of the data. This parameter was adjusted in order to obtain a small set of coherent and well interpretable clusters. After the largest clusters were examined, described, and excluded for further analysis, the procedure was repeated for the remaining data with a smaller parameter value. This procedure can be repeated until a complete and clear understanding of the dataset properties related to another distance function is achieved.

3 The available AIS dataset: MARIN Data Set

The data set analysed in this paper was collected by The Netherland Coastguard, and includes tracking data of shipping movements collected by AIS base stations. The Maritime Research Institute (MARIN) in Netherlands receives the data for use in safety assessment studies for maritime transportation management. MARIN has anonymised a week of AIS data, which has been used in this research. This data set contains raw tracking

data with a time interval of 60 seconds between readings. The data set includes attributes such as *Ship ID*, *Ship Type ID*, *Ship Type*, *Main Ship Type*, *Ship Size ID*, *Latitude*, *Longitude*, *Heading Rate of Turn*, *Speed Over Ground* and *Parse time*. *RecordID* is the field that provides the unique identification of each record. The database model with the subset of attributes used in this research work is depicted in Fig. 1.

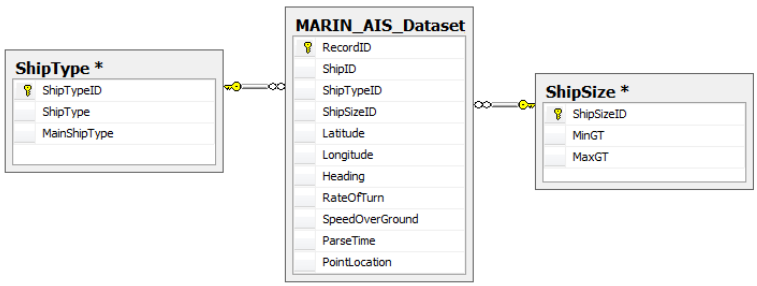


Fig. 1. A subset of the available attributes in the MARIN data set

For the week from 1 to 8 January 2009, 315,794 different records were collected. During the pre-processing, some duplicate readings were removed. The AIS data set includes information about different type of ships as the ones summarised in Table 1. Their spatial distribution is shown in Fig. 2.

Table 1. Sub set of the available ship types

ShipTypeID	Ship Type	Main Ship Type	Nr. Ships	Nr. Positions
5	CHEM IMO 2	Chemical	7	1 458
6	CHEM IMO 2 DH	Chemical	55	11 645
7	CHEM IMO 3	Chemical	5	1 456
8	CHEM IMO 3 DH	Chemical	5	993
10	CHEM DH	Chemical	3	920
11	CHEM WWR	Chemical	1	198
14	Oil crude oil DH	Oil	3	404
15	Oil product	Oil	1	315
16	Oil product DH	Oil	8	1 921
21	LPG semi pressured	LPG	12	2 452
22	LPG pressured	LPG	6	1 586
23	LPG remaining	LPG	1	130

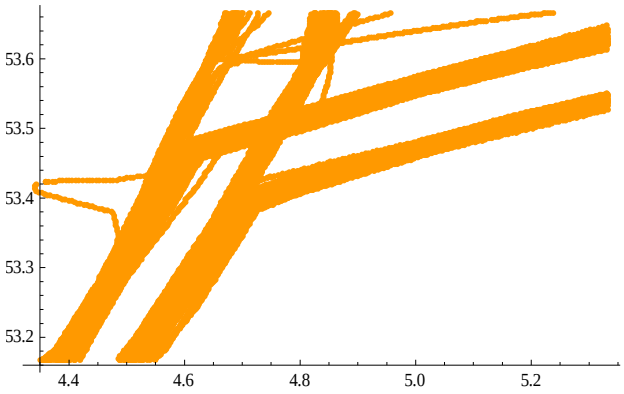


Fig. 2. Spatial distribution of the readings for the sub set of the ship types presented in Table 1

4. The SNN clustering algorithm

In our research, the SNN algorithm, previously proposed by Jarvis and Patrick (1973) and later improved by Ertoz et al. (2002), was used due to its capabilities of identifying clusters with convex and non-convex shapes, having different sizes and densities, as well as due to its ability to deal with noise. The similarity between motion vectors is obtained by looking at the number of nearest neighbours that two motion vectors share. Using this similarity measure, density is defined as the sum of the similarities of the nearest neighbours of a motion vector. Motion vectors with high density become the core vectors, while vectors with low density represent noise vectors. All groups of motion vectors that are strongly similar to core vectors will be included in the clusters.

The SNN algorithm has 3 parameters: *k*, *EPS* and *MinPts*. The number of neighbours that need to be analysed in each step of the clustering process is defined by *k*; *EPS* defines the value for the threshold density and *MinPts* defines the threshold that allows the classification of a motion vector as a core vector. The SNN algorithm first finds the *k* nearest neighbours of each motion vector of the data set. Then the similarity between pairs of motion vectors is calculated in terms of how many nearest neighbours the two motion vectors share. Using this similarity measure, the density of each motion vector is calculated as being the number of neighbours of the current motion vector with which the number of shared neighbours is equal or greater than *EPS* (density threshold). Next, the motion vectors are classified as being core vectors if their density is equal or greater than *MinPts*

(core point threshold). At this stage, the algorithm has all the information needed to build the clusters. The clusters start to be built around the core vectors. Motion vectors that do not integrate any cluster are classified as noise vectors. Since no input parameter is used to determine the number of clusters, the number of clusters emerges directly from the data and not from a number previously defined based on the domain knowledge of a user.

For the identification of the *k* nearest neighbours of a point, a distance function must be defined. In the original algorithm, this function is based on the Euclidean distance among points.

Since our approach deals with position readings and not with trajectories, the distance function must address the particular properties of a motion vector. For that, the distance function of the original SNN algorithm was redefined in order to accommodate the position and the bearing (heading) of a given motion vector. Also, the definition of weights for each one of these variables was also implemented in the SNN algorithm in order to be possible the identification of different types of clusters, which in turn, represent different types of movement.

Given the motion vectors $p_1(x_1, y_1, b_1)$ and $p2(x_2, y_2, b_2)$, the distance between them is calculated using Equation 1.

$$\begin{aligned} DistFunction(p_1, p_2) = \quad & w \times (\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} / mDist) + \\ & (1 - w) \times (\Phi(|b_1 - b_2|) / mBearing) \end{aligned} \tag{1}$$

with

$$\Phi(|b_1 - b_2|) = \begin{cases} |b_1 - b_2| & , |b_1 - b_2| \leq 180^o \\ 360^o - |b_1 - b_2| & , |b_1 - b_2| > 180^o \end{cases} \tag{2}$$

where x_i and y_i represent the position, and b_i represents the direction of movement (bearing). In this function, w is the weight assigned to the position and (1 - w) the weight assigned to the bearing, respectively. To normalise the obtained values, *mDist* is computed as the maximum difference between the values of any two motion vectors in the data set in analysis, according to the Euclidean distance. The *mBearing* value is a constant, with 180º assigned to it. The equation (1) can be rewritten by considering that $d_r(p_1, p_2)$ is the normalised Euclidean distance between p_1 and p_2 and $b_r(p_1, p_2)$ is the normalized bearing variation between p_1 and p_2:

$$DistFunction(p_1, p_2) = w \times d_r(p_1, p_2) + (1 - w) \times b_r(p_1, p_2) \quad (3)$$

The role of each input variable d_r and b_r and the weight w, in the Equation 3, can be better understood by looking at Fig. 3. It is worth to notice that the d_r values for neighbours is expected to be very small since *mDist* is usually a very large value when compared to the Euclidean distance between a point and its *k*-nearest neighbours. Consequently the calculated distance, *DistFunction*, between a point and its *k*-nearest neighbours is also expected to be very small. Fig. 3 plots 4 contour lines with the computed distance value *DistFunction* = 0.01 for different values of w = 0.94, 0.96, 0.98 and 0.99. The value of d_r matches the length of the vector (OA), from the centre of the axis to the contour line corresponding to the intended value of w, with the angle defined by the value of b_r. As can be observed in Fig. 3, when the bearing's variation becomes significant, smaller values of w will result in a larger penalty in the distance value. When w is smaller, $(1 - w)$ is larger and the bearing's influence becomes more important.

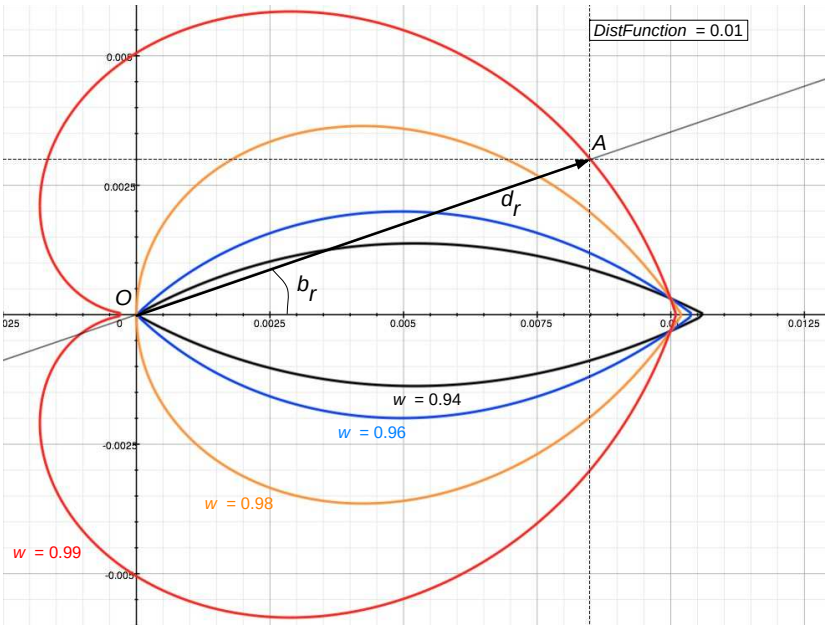
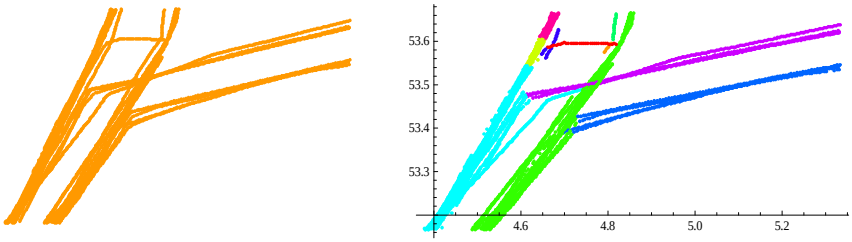


Fig. 3. Visual representation of the weights in the distance function

5. Results

The clustering process was carried out considering two spatial attributes: position and bearing. After some initial tests, the clusters started to emerge with *w* between 90% and 95%. These initial tests were carried out by selecting one particular type of ship and clustering the corresponding motion vectors. Starting with the ship type LPG (Liquefied Petroleum Gases), 4,168 records were available. The initial weight was *w=95%* and the SNN input parameters were *k*=10, *Eps*=3 and *MinPts*=7. These values were chosen in order to cluster motion vectors that are close to each other in terms of position and also pointing into similar directions. Fig. 4 illustrates the input data and the obtained clusters. It is worth to notice that the colours (Fig. 4 b)) are used just to identify different founded clusters and have no other additional meaning.



a) The trajectories of LPG b) The traffic routes of LPG

Fig. 4. The visual comparison between the trajectories and the traffic routes for LPG ships

After analysing the obtained results, it is possible to state that the clustering algorithm was able to aggregate the motion vectors into clusters that represent different traffic routes, although certain small clusters should be included in some of the identified routes as they follow the same alignment in terms of direction. These results call our attention to the need to tune the parameters of the clustering algorithm in order to improve the results.

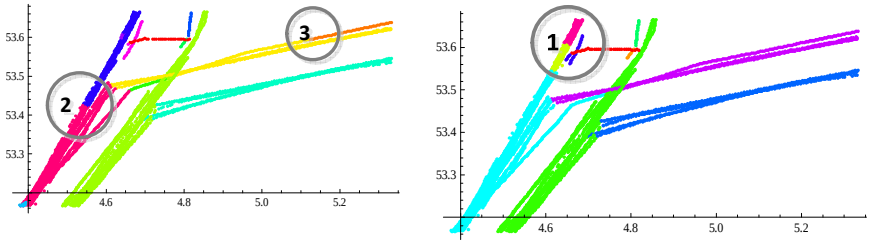
In order to evaluate the impact of parameter selection in computing the traffic routes, we have i) given more weight to the bearing value (maintaining the SNN input parameters); ii) changed the SNN input parameters (maintaining the weight in the distance equation); iii) tuned the weight and the SNN input parameters accordingly to the results obtained in i) and ii).

This process of tuning the parameters is a usual procedure in clustering tasks, since the algorithm always seeks to fit the data under analysis. The majority of existing clustering techniques are dependent on multiple pa-

rameters that may be difficult to tune, mainly in real-life applications (Bouguessa, 2011).

In this work, the input parameters of the algorithm need to be adjusted to the spatial distribution and density of the data, and the weight of the distance function must allow a proper identification of routes, as both the position and the bearing of the motion vectors influence the results. Rinzivillo et al. (2008) followed a similar approach, for clustering trajectories, tuning the minimum number of neighbours according to the density of the data.

Starting with the changes in the weight of Equation 1, more weight was given to the bearing value trying to join clusters that are more aligned with respect to direction. The weight of *w=90%* was adopted, maintaining the SNN input parameters. The obtained results are presented in Fig. 5a), while Fig. 5b) shows the results represented previously in Fig. 4b), in order to facilitate the comparison between the two clustering results.



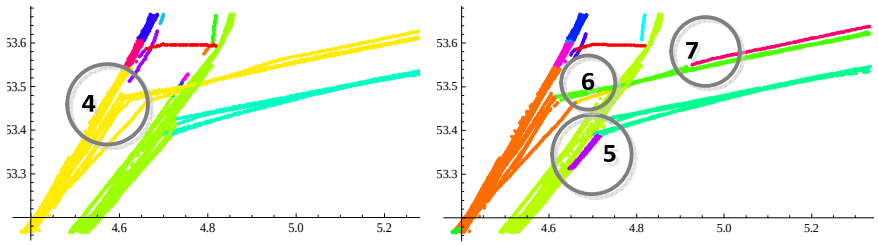
a) Clustering results with *w=90%*, *k=10*, *Eps=3* and *MinPts=7*

b) Clustering results with *w=95%*, *k=10*, *Eps=3* and *MinPts=7*

Fig. 5. Clustering results with *w=90%* and *w=95%*

The overall result shows some improvements, joining clusters that were previously disjoint (like the ones marked with **1** in Fig. 5b)), but the identification of traffic routes was not completely achieved. The algorithm was not able to join all the motion vectors that follow the same alignment in terms of direction and that are close to the main identified traffic routes (cases **2** and **3** in Fig. 5a)).

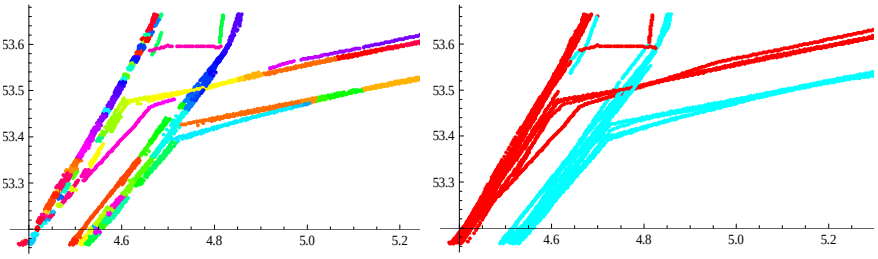
When the SNN input parameters were changed, starting by the *MinPts* input parameter, we have decreased or increased the original value. Decreasing *MinPts* to 6 has avoided the identification of all routes as the algorithm is able to joint motion vectors that follow different directions, joining different routes (like case **4** in Fig. 6a)). Increasing this number has led to the appearance of more clusters (as the cases **5**, **6** and **7** in Fig. 6b)), as more similar motion vectors are needed in the neighbourhood of a given motion vector to both be part of the same cluster.



a) *MinPts=6* b) *MinPts=8*

Fig. 6. Clustering results with *w=95%*, *k=10*, *Eps=3*

The other input parameter that was changed was the *k* value. A *k* value of 8 and 12 was considered. The results clearly show that decreasing *k* splits the analysed trajectories into very small clusters, as less neighbours are compared with the motion vector under analysis (Fig. 7a) and increasing *k* excessively joint motion vectors in the same clusters avoiding the identification of routes (Fig. 7b)).



a) *k=8* b) *k=12*

Fig. 7. Clustering results with *w=95%*, *Eps=3*, *MinPts=7*

The results obtained so far allowed us to verify that increasing the weight of the bearing variable improves the results by joining clusters that follow the same direction, although in some cases imposes the creation of more clusters if the directions are not fully aligned. However, this excessive number of clusters can be controlled if we impose the verification of more neighbours of a motion vector, by increasing the *k* value. This leads us to change the weight and the input parameters in order to improve the overall results. In this case, the weight of 90% was considered, as well as the *k* value was increased to 12 (maintaining all the others SNN input parameters). The obtained results were very promising as the main traffic routes were identified through the extraction of 9 clusters (Fig. 8).

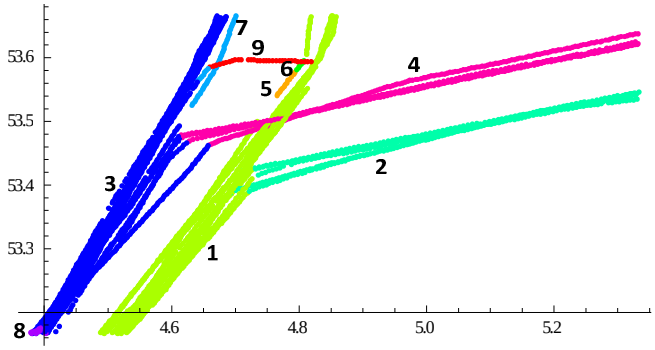


Fig. 8. Clustering results with *w=90%*, *k=12*, *Eps=3* and *MinPts=7*

The small clusters that naturally emerged due to the distribution of the motion vectors can be eliminated in a post-processing stage, as the number of motion vectors that integrate the clusters that represent routes and the other clusters is very different. For example, Fig. 9 presents a histogram with the number of motion vectors per cluster, calling our attention to the huge difference between the number of motion vectors integrated in clusters number 1, 2, 3 and 4 when compared with clusters 5, 6, 7, 8, and 9.

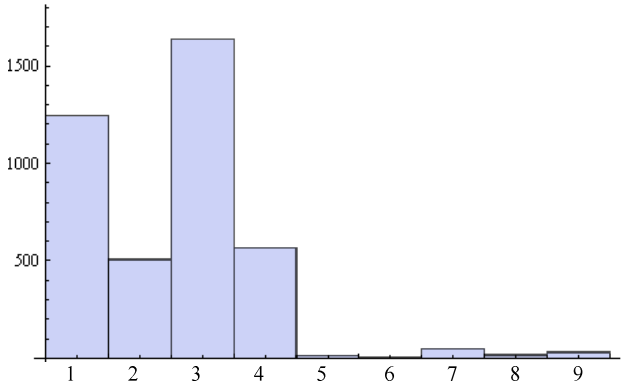


Fig. 9. Histogram with the number of motion vectors per cluster

If we look into the interval of the bearing values inside each cluster, we can see aligned clusters, with a small variation in the bearing values per cluster. Fig. 10 depicts these results. In this figure we can see cluster number 0, representing the motion vectors classified as noise by the SNN algorithm, with a huge variation in the bearing value.

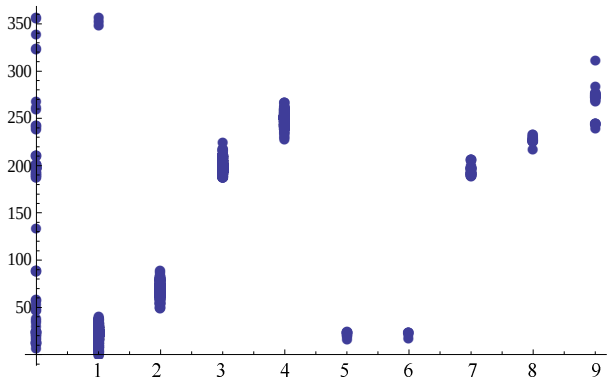
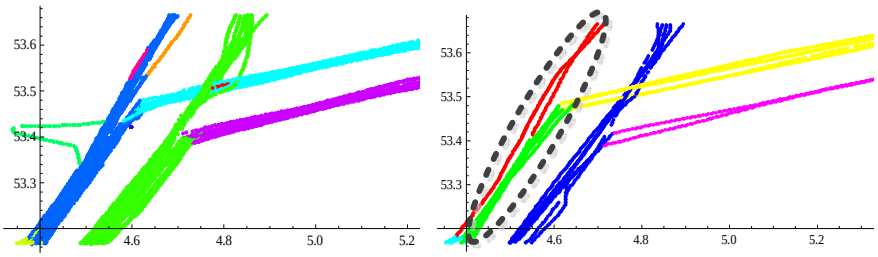


Fig. 10. Bearing values per cluster

In order to verify if the obtained results and the used parameters are independent of the type of ship and also independent from the number of motion vectors under analysis, other types of ships were analysed. The ship type Chemical was selected. Due to the number of points available for this type (16,670), a sample data set with 7,500 records was used. Also, the ship type Oil was analysed, with all the 2,640 records associated to it. In both cases, the clustering process was able to automatically identify the main routes (Fig. 11). Again, only the small clusters need to be discarded in a post-processing stage of the routes identification process. For the Oil ship type, it is worth to mention that if more data were available, the green and the red clusters would be one (dashed ellipse in Fig. 11b)), as happened in the other results presented so far.

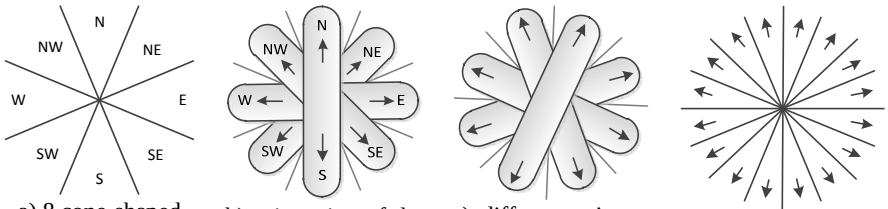


a) Ship Type Chemical b) Ship Type Oil

Fig. 11. Clustering results with *w=90%, k=12, Eps=3, MinPts=7*

The results obtained so far call our attention to the clusters shape, and their alignment, and whenever their represent traffic routes or not. In the context of this work, routes are considered pathways that are followed by a set of moving objects. Let $M_p \equiv \{(x_1, y_1, t_1), (x_2, y_2, t_2),..., (x_n, y_n, t_n)\}$ be the motion vectors present in a moving objects data set. T_r denotes a traffic

route if its motion vectors are aligned in such a way that the dispersion of the corresponding bearing values are under a specific threshold. This threshold is measured using the standard deviation and can be set to a specific value attending to the application domain under analysis. In this work, the threshold for the standard deviation was set to 22.5º having in mind a division of the space that considers cardinal directions with 8 cone-shaped regions (Fig. 12 a)). This approach allows the emergence of traffic routes with different orientations, as the intervals of the cone-shaped regions can emerge from the data under analysis (see Fig. 12 b) and c) for different orientations). This approach also allows the identification of narrowed pathways as we can define a cone-shaped division with 16 regions and set the standard deviation threshold to a lower value (Fig. 12 d))



a) 8 cone-shaped regions

b) orientation of the cone-shaped regions

c) different orientation of the cone-shaped regions

d) 16 cone-shaped regions

Fig. 12. Division of the space using Cardinal directions

For the results previously presented, Table 2 summarizes the calculated standard deviation for the clusters that integrate a high number of motion vectors. This table also shows the minimum and maximum bearing values of the motion vectors in each cluster.

Table 2. Standard deviation, minimum and maximum bearing values per cluster

	LPG			**Chemical**			**Oil**		
	σ	Min	Max	σ	Min	Max	σ	Min	Max
C_1	5.70	353º	41º	9.95	355º	57º	3.37	237º	264º
C_2	5.80	49º	90º	3.83	187º	214º	3.74	191º	216º
C_3	4.44	187º	224º	6.23	220º	261º	7.63	349º	40º
C_4	5.12	228º	267º	5.08	53º	87º	4.82	55º	80º
C_5							6.04	191º	214º

One important issue to consider in this work is the processing time that is needed to compute the routes. Clustering is a very demanding process in computational terms (Bhavsar and Jivani, 2009). In this paper we only showed the analysis of small data samples. For larger samples, the clustering process can take days, depending on the implementation. In this work

was used an implementation made in Mathematica (http:// http://www.wolfram.com/). To avoid this delay, pre-processing techniques are under analysis to limit the number of motion vectors that need to be clustered in order to identify the routes.

Although this work is out of the scope of this paper, Fig. 13 shows the time (in seconds) needed to cluster several data samples. The samples are all associated with the ship type Chemical, considering sample data sets with 100; 500; 1,000; 2,000; 4,000; 6,500 and 7,500 motion vectors. As depicted in this figure, the processing time presents a non-linear growth considering the number of motion vectors under analysis.

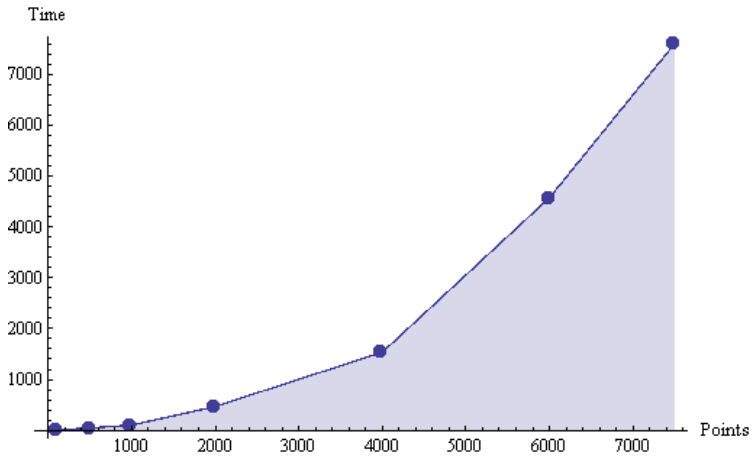


Fig. 13. Required processing time attending to the number of motion vectors

6. Conclusions and future work

This paper presented the analysis of movement data in order to identify the main routes in an AIS dataset. For the automated identification of the main traffic routes, a density-based clustering algorithm, the Shared Nearest Neighbour algorithm, was applied to the motion vectors of a ship. The proposed approach avoided the reconstruction of trajectories and the clustering of such trajectories. The obtained results showed that there is no need to reconstruct the trajectories in order to be able to identify the traffic routes. Another important advantage is that no background knowledge is needed to a-priori select the routes or regions of interest.

In our approach, the main requirement is the adjustment of the input parameters of the proposed clustering algorithm, mainly because they influence the obtained traffic routes. The algorithm seeks to fit the spatial distribution and density of the data under analysis.

As future work, we plan to investigate pre-processing strategies that limit the number of motion vectors that are needed in the clustering process in order to identify traffic routes. This will speed up the clustering process that is very expensive in computational terms. Post-processing heuristics are also needed to exclude the small clusters that do not represent routes in the data set under analysis.

Moreover, the identification of different metrics to measure the coherence of the obtained clusters is also envisaged. Those metrics could be used in the self-tuning of the clustering algorithm through the definition of heuristics that support the parameters tuning process. Afterwards, different data sets need to be analysed to test the metrics and the self-tuning process.

Acknowledgements

We would like to thank the Maritime Research Institute in The Netherlands, for making the data available for analysis under the MOVE EU Cost Action IC0903 (*Knowledge Discovery from Moving Objects*).

References

Bhavsar, H. and Jivani, A. (2009) The Shared Nearest Neighbor Algorithm with Enclosures (SNNAE), Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering, IEEE, pp. 436-442.

Bouguessa, M. (2011) A Practical Approach for Clustering Transaction Data, Proceeding of the 7th International Conference on Machine Learning and Data Mining, New York, August/September, LNAI 6871, Springer-Verlag.

Chen, L., Özsu, M. and Oria, V. (2005) Robust and fast similarity search for moving object trajectories, Proceedings of the 2005 ACM SIGMOD international conference on Management of data - SIGMOD '05, ACM Press, New York, New York, USA.

Ertoz, L., Steinbach, M. and Kumar, V. (2002) Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data, Proceedings of the Second SIAM International Conference on Data Mining, San Francisco.

Giannotti, F., Nanni, M., Pedreschi, D., and Pinelli, F. (2007) Trajectory Pattern Mining, Proceedings of the Knowledge Discovery in Databases (KDD'07) Conference, San Jose, pp. 330-339.

Giannotti, F. and Pedreschi, D. (2008) Mobility, Data Mining and Privacy: A Vision of Convergence. In: Giannotti, F. and Pedreschi, D. (Eds.): Mobility, Data Mining and Privacy, Springer-Verlag, pp. 1-11.

Grabmeier, J. (2002) Techniques of Cluster Algorithms in Data Mining, Data Mining and Knowledge Discovery, 6(4), pp. 303-360.

Jarvis, R. and Patrick, E. (1973) Clustering Using a Similarity Measure Based on Shared Near Neighbors, IEEE Transactions on Computers, C-22(11), pp. 1025-1034.

Lee, J.-G., Han, J. and Whang, K.-Y. (2007) Trajectory Clustering: A Partition-and-Group Framework, Proceedings of SIGMOD Conference (SIGMOD'07), Beijing, pp. 593-604.

Little, J. J. and Gu, Z. (2001) Video retrieval by spatial and temporal structure of trajectories, Proceedings of SPIE, The International Society for Optical Engineering, pp. 545-552.

Meratnia, N. and de By, R. A. (2002) Aggregation and comparison of trajectories, Proceedings of the 10th ACM international symposium on Advances in Geographic Information Systems, ACM, pp. 49–54.

Miller, H. J. and Han, J. (2009) Geographic Data Mining and Knowledge Discovery, 2nd edition, Taylor & Francis Group.

Perez, H. M., Chang, R., Billings, R., and Kosub, T. L. (2009) Automatic Identification Systems (AIS) Data Use in Marine Vessel Emission Estimation, Presented at the 18th Annual International Emission Inventory Conference. Baltimore.

Rinzivillo, S., Pedreschi, D., Nanni, M., Giannotti, F., Andrienko, N., and Andrienko, G. (2008) Visually driven analysis of movement data by progressive clustering, Information Visualization, 7, pp. 225-239.

Vlachos, M., Kollios, G. and Gunopulos, D. (2002) Discovering similar multidimensional trajectories, Proceedings 18th International Conference on Data Engineering, IEEE Computer Society, San Jose, CA, USA, pp. 673-684.

Zaït, M. and Messatfa, H. (1997) A comparative study of clustering methods, Future Generation Computer Systems, 13(2), pp. 149-159.