A Study of Vessel Trajectory Compression Based on Vector Data Compression Algorithms

Yuanyuan Ji, Wenhai Xu(✉), and Ansheng Deng

Information Science Technology College, Dalian Maritime University, Dalian 116026, China
yyjem@outlook.com, x_wenhai@163.com, asddmu@hotmail.com

Abstract. With the development of information technology and its vast applications in vessel traffic, such as the popular Automatic Identification System (AIS), a large quantity of vessel trajectory data has been recorded and stored. Vessel traffic has also entered the age of big data. However, the redundancy of data considerably reduces the availability of research and applications, and how to compress these data becomes a problem that needs to be solved. In this paper, several classical vector data compression algorithms are summarized, and the ideas of each algorithm and the steps to compress vessel trajectories are introduced. The vessel trajectory compression experiments based on the algorithms are performed. The results are analyzed, and the characteristics of each algorithm are summarized. The results and conclusions lay the foundation for the selection and improvement of the algorithms in vessel trajectory compression. Through the study of this paper, a systematic theoretical support for the compression of vessel trajectories is provided, which could guide practical applications.

Keywords: Vessel trajectory · Big data · Data compression algorithms · AIS

1 Introduction

Maritime transport plays a vital role in global supply chains. The Review of Maritime Transport published by United Nations Conference on Trade and Development (UNCTAD) in 2017 notes that over 80% of global trade by volume and more than 70% of its value being carried on board ships and handled by seaports worldwide, the importance of maritime transport for trade and development cannot be overemphasized. Ocean shipping will remain the most important mode of transport for international merchandise trade. However, maritime transport is facing many challenges to ensure a high level of efficiency, safety and

Supported by "the Fundamental Research Funds for the Central Universities" (No. 3132016021).

W. Abramowicz and R. Corchuelo (Eds.): BIS 2019 Workshops, LNBIP 373, pp. 473–484, 2019.
https://doi.org/10.1007/978-3-030-36691-9_40

environmental protection, which need academia to develop supporting models and methods of analysis [1].

Vessel trajectories are one of the main data sources for studying the characteristics of vessel traffic behaviors, which is an important basis for supporting the research and application of maritime transport. With the development of information technology and its vast applications in traffic, trajectory data becomes easy to be achieved and have been widely used in road, railway and air traffic researches and practical applications [5,15,19]. The researches and applications based on vessel trajectory data are less than the others, but with the popularity of Automatic Identification System (AIS), a large number of vessel trajectory data has been recorded and stored. Vessel traffic has also entered the age of big data [6,9,14]. Furthermore, increasingly numerous methods, theories and technologies of big data, knowledge mining and machine learning have been proposed. Therefore, how to take full use of the data to promote the development of the marine intelligent traffic system becomes one of the most important research topics [3,13,20–23,25]. Nonetheless, the AIS equipment of a vessel generally publishes a message within every 2 s to 6 min, which makes the trajectory data from the AIS notably large [2,8,17]. Because the AIS has a high frequency of information, the redundant problem of trajectory data from the AIS is highly serious. This problem makes it difficult to be used in research and actual applications. Therefore, the vessel trajectory data compression becomes particularly important [24]. There many methods to compress trajectory data, including clustering method, semantic trajectory compression method, context-aware method, piecewise linear segmentation method, directed acyclic graph method, offline direction-preserving trajectory simplification method, etc. [11,18]. It is noteworthy that as a type of vector data, vessel trajectory data can be compressed by the vector data compression algorithms. Generalized vector data compression should include the storage compression and re-sampling of the vector data [4,12]. The concept of storage compression reduces the amount of vector data by converting the data type or file type. The concept of resampling is to extract subset from set which is a collection of the points that compose the vector graphics. Subset should reflect the original data set within a certain accuracy as much as possible and should ensure that the points of subset are as little as possible. In this paper, the key study is of the re-sampling, which is the vector data compression in a narrow sense.

At present, several of the most widely used classic compression algorithms for vector data are the choosing interval points algorithm, limiting vertical distance algorithm, limiting angle algorithm, offset angle algorithm, Douglas Peucker algorithm and grating algorithm [7,10,16]. The research on vessel trajectory compression mainly focuses on the application and improvement of the Douglas Peucker algorithm, and some problems in the practical application of vessel trajectory data were effectively solved. However, many other vector data compression algorithms have not been applied to vessel trajectory compression. More testing and analysis of these algorithms in vessel trajectory compression are needed. Moreover, different algorithms have different characteristics, which may

be highly effective in some specific data compression applications. Therefore, it is necessary to introduce the above vector data compression algorithms and to study the advantages and disadvantages of each algorithm in vessel trajectory compression through experiments. The remainder of this paper is organized as follows. Section 2 introduces the vessel trajectory data compression steps of the above five algorithms. Section 3 presents the data compression experiments in which the performances of the traditional algorithms are tested, and the results are analyzed and discussed. The study's conclusions are summarized in Sect. 4.

2 Compression Algorithms

Suppose a vessel's trajectory is composed by a set of points in chronological order, which can be represented by $A = P_1(x_1, y_1), P_1(x_2, y_2), ..., P_1(x_n, y_n)$, as shown in Fig. 1. P is a point on the trajectory, x is the abscissa and y is the ordinate. The subscript represents the number of the point ordered by time. n is the total number of points on the trajectory. The vessel sailed through each point in chronological order. Let subset B stand for the compression result of set A. The description of each algorithm is as follows.

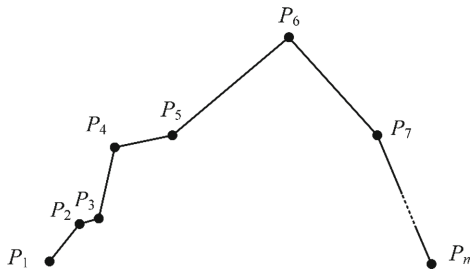


Fig. 1. Example of a vessel trajectory.

2.1 Choosing Interval Points Algorithm

The basic idea of this algorithm is to retain a point in interval k points or an inter-equal distance d on the trajectory. In addition, the first and last points should also to be retained. Let k stand for the number of interval points. The steps of trajectory data compression based on the choosing interval points algorithm are as follows.

(1) Calculate the number of intermediate points that need to be retain on the trajectory, $m = \lfloor (n-1)/k \rfloor$.
(2) Retain the points $P_{k+1}, P_{2k+1}, ..., P_{mk+1}$.
(3) Retain the first and last points P_1 and P_n.
(4) Take subset $B = P_1, P_{k+1}, ..., P_{mk+1}, P_n$.

Let $k = 2$ and $n = 8$. $\lfloor (n-1)/k \rfloor$ rounds $(n-1)/k$ to the nearest integers less than or equal to $(n-1)/k$. After the compression steps above, the result is shown in Fig. 2. The dashed line is the trajectory before compression, and the solid line is the compressed trajectory.

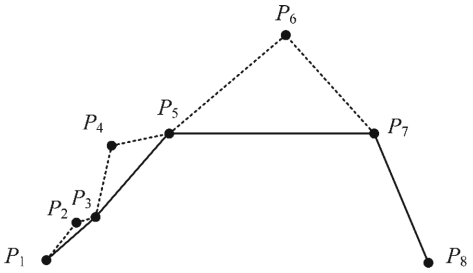


Fig. 2. Example of a vessel trajectory.

The choosing interval points algorithm is simple and easy to implement. This algorithm supports real-time compression processing. However, it is insensitive to the points where the curvature radius is small, such as P_6 in Fig. 2. This property may lead to a larger error when there are more twists and turns in the trajectory.

2.2 Limiting Vertical Distance Algorithm

The basic idea of this algorithm is to select three consecutive points and calculate the vertical distance between the middle point and the straight line between the other two points. Let d stand for the vertical distance. Next, compare d with the distance threshold D. If $d \geq D$, retain the middle point. Otherwise, if $d < D$, delete the middle point. After that step, select the second three consecutive points and continue the pattern until all of the points on the trajectory are processed.

The limiting vertical distance algorithm is simple and easy to implement. This algorithm supports the real-time compression processing of the vessel trajectory data and is sensitive to the distance that a point deviates from the vessel's previous course. However, occasionally, the extreme point of curvature may be deleted, thereby leading to compression error.

2.3 Limiting Angle Algorithm

This algorithm is similar to the limiting vertical distance algorithm. The basic idea is to select three consecutive points, such as P_1, P_2 and P_3. Next, calculate the degree of the angle $\angle P_2P_1P_3$. Next, compare it with a given threshold *theta*. If $\angle P_2P_1P_3 < theta$, delete the middle point P_2, otherwise retain P_2. The steps

of trajectory data compression based on the limiting angle algorithm are similar to the limiting vertical distance algorithm above.

The limiting angle algorithm is simple and easy to implement. The algorithm supports the real-time compression processing of the vessel trajectory data, but it may delete the extreme point of curvature and result in compression error. It has high requirements for the curvature and point density of the trajectory, and it has not always been used in practical applications.

2.4 Offset Angle Algorithm

The offset angle algorithm is similar to the limiting vertical distance algorithm and the limiting angle algorithm. The basic idea of this algorithm is to select three consecutive points, such as P_1, P_2 and P_3. After that step, calculate the degree of the angle $\angle P_1P_2P_3$.

Next, compare it with a given threshold θ. If $\angle P_1P_2P_3 < \theta$, retain the middle point P_2, otherwise delete P_2. The steps of the trajectory data compression based on the offset angle algorithm are also similar to the limiting vertical distance algorithm above.

The offset angle algorithm is simple and easy to implement. This algorithm supports real-time compression processing of the vessel trajectory data, and it is sensitive to the course change of the vessel trajectory. However, when the point is dense or the course changes slowly, the algorithm may delete all the points on the curved segment and lead to compression error. To compensate for this defect, the course change is often highlighted by increasing the distance between the three selected points. Assuming that the middle point among the three points is P_j, the front and back points are P_{j-k} and $P_{j+k}, k \geq 1$. To a certain extent, the angle $P_{j-k}P_jP_{j+k}$ can reflect the course change from P_{j-k} to P_{j+k}. The value of k needs to be selected according to the data characteristics, related professional experience and application requirements.

2.5 Douglas Peucker Algorithm

The basic idea of the Douglas Peucker algorithm is to connect the first point P_1 and the last points P_n of the trajectory with a straight line, calculate the distance between the other points in the middle to this straight line, and discover the maximum distance d_{max} and the corresponding point P_i, then compare dmax with the maximum allowable error D_{error}. If $d_{max} < D_{error}$, delete all points between the first and last points. If $d_{max} \geq D_{error}$, retain the point P_i and divide the trajectory into two segments $P_1, P_2, ..., P_i$ and $P_i, P_{i+1}, ..., P_n$. Next, for each segment, repeat the above process until the end. The retained points and the first and last points constitute the compression result, subset B. Let the total number of track points be $n = 8$. The steps of trajectory data compression based on Douglas Peucker algorithm are as follows.

(1) Connect the points P_1 and P_8. Calculate the distances from $P_2, P_3, ..., P_7$ to the straight line P_1P_8. The distance from P_6 to the straight line is the maximum, and it is denoted by dmax, as shown in Fig. 3(a).

(2) If $d_{max} < D_{error}$, delete the points $P_2, P_3, ..., P_7$. The remaining points, P_1 and P_8, are not satisfied by the conditions for repeating step (1), and thus go to step (3). The compressed result is $B = \{P_1, P_8\}$. The compressed trajectory is a straight line P_1P_8, as shown in Fig. 3(b). If $d_{max} \geq D_{error}$, P_6 divides the trajectory into two segments, which are $P_1, P_2, ..., P_6$ and P_6, P_7, P_8. If each segment is a new trajectory, go to step (1), and process the two trajectories separately, as shown in Fig. 3(c).

(3) The points that have not been deleted through the above steps are the compressed results, which constitute the subset B. By adjusting the maximum allowable error D_{error}, let the number of points in subset B is 4, and then the result is shown in Fig. 3(d).

Compared with the above algorithms, this algorithm has better global characteristics, and it can retain the extreme point of curvature. After the compression, the spatial structural characteristics of the trajectory can be preserved well. However, the algorithm requires that the data of the entire trajectory must be obtained before the process, which means that it does not support real-time compression processing.

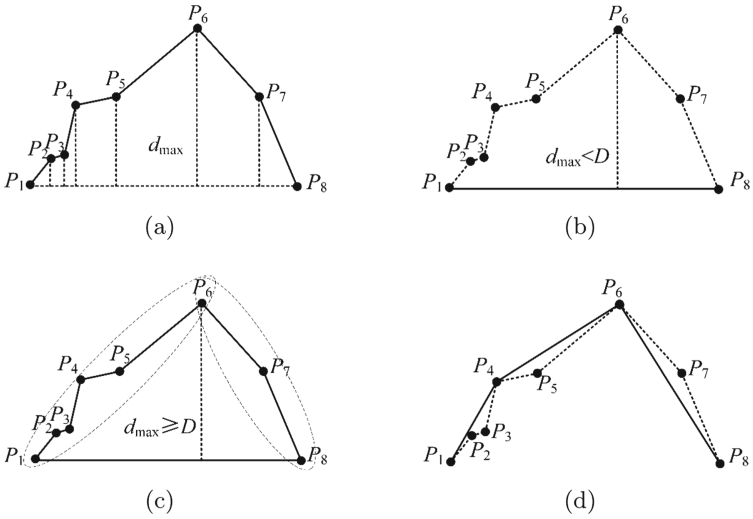


Fig. 3. Schematic diagram of the compression process based on the Douglas Peucker algorithm: (a) the d_max from P_6 to P_1P_8; (b) the compression result P_1, P_8, if $d_{max} < D_{error}$; (c) the two new segments divided by P_6, if $d_{max} \geq D_{error}$; (d) the compression result when the number of points in subset B is 4.

2.6 Grating Algorithm

The basic idea of the grating algorithm is to define a fan-shaped region and judge whether the point on the trajectory is inside or outside the region. If it

is inside, delete the related point. Otherwise, retain the related point. Let the caliber of the fan-shaped region be d. The steps of trajectory data compression based on grating algorithm are as follows.

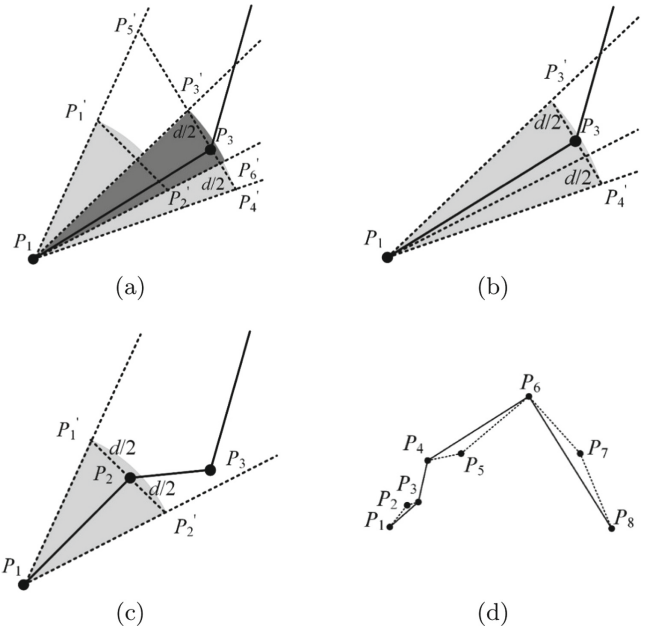


Fig. 4. Schematic diagram of the compression process based on the grating algorithm: (a) fan-shaped region $P_1'P_1P_2'$; (b) delete P_2 and the new fan-shaped region is $P_3'P_1P_4'$; (c) the overlapping area of $P_1'P_1P_2'$ and $P_3'P_1P_4'$; (d) compression result.

(1) Connect P_1 and P_2, and pass P_2 to make a straight-line perpendicular to P_1P_2. Take two points P_1' and P_2' on the line and let $P_2P_1' = P_2P_2' = d/2$. Then, $P_1'P_1P_2'$ constitutes a fan-shaped region, as shown in Fig. 4(c).
(2) Extend PP_1' and PP_2' to judge whether P_3 is inside the extended fan-shaped region or not. If P_3 is inside, delete P_2. Then, connect P_1 and P_3, and make a straight-line perpendicular to P_1P_3. Take two points P_3' and P_4' on the line and let $P_2P_1' = P_2P_2' = d/2$, as shown in Fig. 4(b).
(3) The line $P_3'P_4'$ intersects line P_1P_1' at point P_5' and intersects line P_1P_2' at point P_6'. If point P_3' or P_4' is located outside the extended region of the fan-shaped $P_1'P_1P_2'$, replace them with P_5' or P_6', respectively. As shown in Fig. 4(a), P_6' replaces P_4', the points P_3', P_1 and P_6' constitute a new fan-shaped region, which is the overlapping area of $P_1'P_1P_2'$ and $P_3'P_1P_4'$.
(4) Extend P_1P_3' and P_1P_6' to judge whether P_4 is inside the extended fan-shaped region or not. Repeat steps (1) to (3) until a new point is outside the newly constituted fan shape.

(5) If P_4 is outside the extended fan shape, P_3 should to be retained. Let P_3 be the new starting point (instead of P_1) and repeat steps (1)–(5) until the end of the trajectory. All the points that are retained (including the first and last points) constitute the compression result, subset B. The solid line, as shown in Fig. 4(d), is the compressed trajectory.
Although the grating algorithm is a little more complex than the above algorithm, it is also easy to implement. Besides, it is very sensitive to the curvature variation, and it supports real-time compression processing.

3 Compression Experiment

This section will introduce the compression experiments based on the compression algorithms above. The compression results will be compared and analyzed.

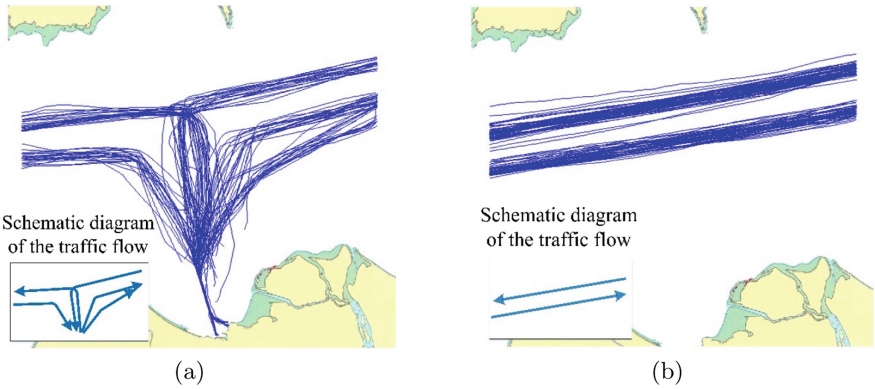
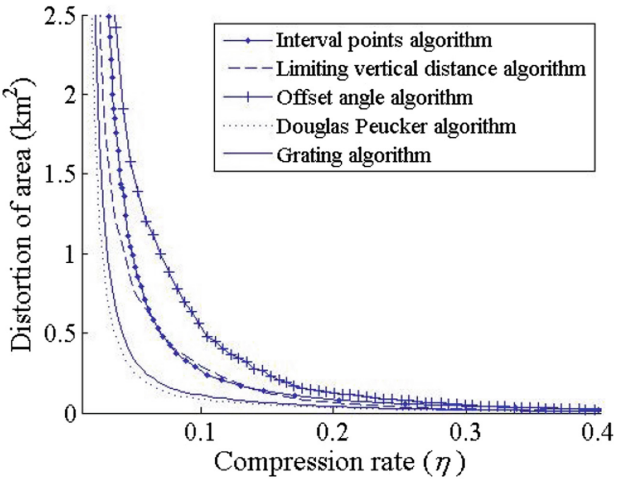


Fig. 5. Sample of vessel trajectories for the compression experiments: (a) the vessel trajectories of set C; (b) the vessel trajectories of set D.

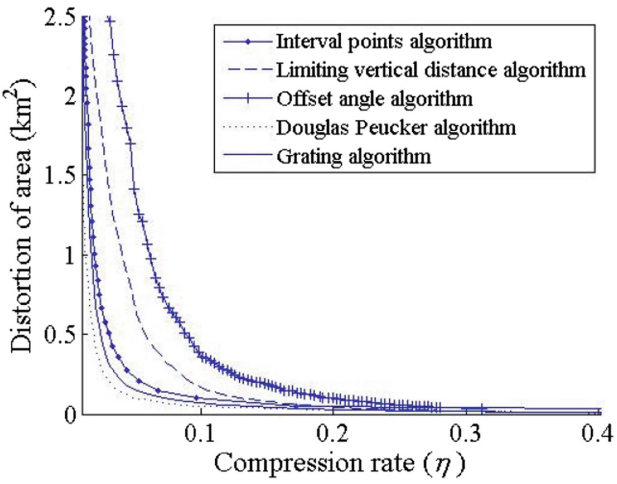
3.1 Vessel Trajectory Data

In this paper, the trajectory data sample for experiments is from the AIS. The vessel trajectory data are mainly obtained from the position report messages of the AIS. The update frequency of this massage is related to the speed of the ship and the rate of turn (ROT). Therefore, the sample data should include the AIS messages when vessels have different speeds and ROTs. Taking the trajectories of vessels in the Qiongzhou Strait as an example, which contains AIS data of vessels with different sailing conditions.

The data of one hundred trajectories in the Qiongzhou Strait with obvious speed and ROT variations were taken as the sample for the experiments, as shown in Fig. 5(a), and is called set C. The data of another one hundred trajectories without obvious speed and ROT variations were also taken as a sample for the experiments, as shown in Fig. 5(b), and is known as set D. Set C includes 29429 points, and set D includes 26425 points.



(a)



(b)

Fig. 6. Curves of the distortion of area with the compression rate: (a) Set C; (b) Set D.

3.2 Results and Discussion

The compression rate and compression error are the basic elements for evaluating a data compression algorithm. In this paper, the compression rate is defined as the ratio between the points' number of compressed trajectories and uncompressed trajectories. Suppose the number of trajectory points before compression is m and the number of compressed trajectory points is n. The compression rate is $\eta = n/m$.

The results shown in Fig. 6 indicate that when the compression rate becomes smaller, the distortion of area becomes larger. As the compression rate increases, the distortion of area will gradually decrease. In other words, the less points that are left after compression, the larger the distortion of area is, which is consistent with the regular pattern. When the compression rate is greater than 0.4, the distortion of area approaches zero for each compression algorithm. As the compression rate gradually decreases from 0.4, the distortion of area gradually increases. The extent and speed of the increase are different for each compression algorithm, which can reflect the differences of the algorithms' performances. Since each compression algorithm retains at least the first and last points, the compression rate is greater than zero. Therefore, this paper will analyze the compression results of the algorithms within the range of compression rate from 0.01 to 0.4.

The vessel trajectory data compression error caused by the offset angle algorithm is the largest, but it can be optimized by changing the interval parameter according to the data's characteristics and professional experience. The compression errors caused by the choosing interval points algorithm and limiting vertical distance algorithm are smaller than the offset angle algorithm. The choosing interval points algorithm is more suitable for the trajectories that are approximately straight lines. The limiting vertical distance algorithm is more suitable for the trajectories whose structures are complex. The performance of the Douglas Peucker algorithm and grating algorithm are better than other algorithms in the experiments. The error caused by the Douglas Peucker algorithm is minimal, but it does not support real-time processing. Although the error caused by the grating algorithm is a little larger than the Douglas Peucker algorithm, it support real-time processing. Therefore, when it needs to compress historical vessel trajectory data, the Douglas Peucker algorithm is recommended, and when it needs to compress vessel trajectory data in real time, the grating algorithm is recommended.

4 Conclusions

This paper introduces several classic vector data compression algorithms, as well as their ideas and implementation steps for vessel trajectory compression. Through the compression experiment, the performances of each algorithm are compared and analyzed. The result shows that the performance of the Douglas Peucker algorithm and grating algorithm are better than the other algorithms. The Douglas Peucker algorithm is suitable for historical data compression. The grating algorithm is suitable for real-time data compression. Furthermore, the advantages and disadvantages of each algorithm in compressing vessel trajectory data are also analyzed, which provides a basis for the selection and improvement of the algorithms in the future. The research described in this paper provides theoretical support for data compression in the application of vessel traffic research and practical applications based on AIS data.

Acknowledgments. This research was supported by "the Fundamental Research Funds for the Central Universities" (No. 3132016021). The authors thank the researchers who participated in the data processing and provided language assistance.

References

1. Bell, M.G., Meng, Q.: Special issue in transportation research part b-shipping, port and maritime logistics. Transp. Res. Part B: Methodol. **93**(PB), 697–699 (2016). https://doi.org/10.1016/j.trb.2016.09.003
2. Bole, A.G., Dineley, W.O., Wall, A.: Chapter 5 - automatic identification system (AIS). In: Radar and ARPA Manual, Oxford, 3rd edn, pp. 255–275 (2014). https://doi.org/10.1016/B978-0-08-097752-2.00005-2
3. Borkowski, P.: The ship movement trajectory prediction algorithm using navigational data fusion. Sensors **17**(6), 1432 (2017). https://doi.org/10.3390/s17061432
4. Chen, F., Ren, H.: Comparison of vector data compression algorithms in mobile GIS. In: 2010 3rd International Conference on Computer Science and Information Technology, vol. 1, pp. 613–617 (2010). https://doi.org/10.1109/ICCSIT.2010.5564118
5. Clements, J.C.: The optimal control of collision avoidance trajectories in air traffic management. Transp. Res. Part B: Methodol. **33**(4), 265–280 (1999). https://doi.org/10.1016/S0191-2615(98)00031-9
6. Dittmar, C.: Die nächste evolutionsstufe von AIS: big data. In: Gluchowski, P., Chamoni, P. (eds.) Analytische Informationssysteme, pp. 55–65. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-47763-2_4
7. Gudmundsson, J., Katajainen, J., Merrick, D., Ong, C., Wolle, T.: Compressing spatio-temporal trajectories. Comput. Geom. **42**(9), 825–841 (2009). https://doi.org/10.1016/j.comgeo.2009.02.002
8. Ifrim, C., Iuga, I., Pop, F., Wallace, M., Poulopoulos, V.: Data reduction techniques applied on automatic identification system data. In: International KEYSTONE Conference on Semantic Keyword-Based Search on Structured Data Sources, pp. 14–19 (2017). https://doi.org/10.1007/978-3-319-74497-12
9. Isenor, A.W., St-Hilaire, M.O., Webb, S., Mayrand, M.: MSARI: a database for large volume storage and utilisation of maritime data. J. Navig. **70**(2), 276–290 (2017). https://doi.org/10.1017/S0373463316000540
10. Ji, H., Wang, Y.: The research on the compression algorithms for vector data. In: 2010 International Conference on Multimedia Technology, pp. 1–4 (2010). https://doi.org/10.1109/ICMULT.2010.5631153
11. Lever, R., Hinze, A., Buchanan, G.: Compressing GPS data on mobile devices. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006. LNCS, vol. 4278, pp. 1944–1947. Springer, Heidelberg (2006). https://doi.org/10.1007/11915072_102
12. Li, Y., Zhong, E.: A new vector data compression approach for WebGIS. Geo-Spat. Inf. Sci. **14**(1), 48–53 (2011). https://doi.org/10.1007/s11806-011-0431-1
13. Mao, S., Tu, E., Zhang, G., Rachmawati, L., Rajabally, E., Huang, G.B.: An automatic identification system (AIS) database for maritime trajectory prediction and data mining. In: Proceedings of ELM-2016, pp. 241–257 (2018). https://doi.org/10.1007/978-3-319-57421-9_20
14. Moffitt, K.C., Vasarhelyi, M.A.: AIS in an age of big data. J. Inf. Syst. **27**(2), 1–19 (2013). https://doi.org/10.2308/isys-10372

15. Montanino, M., Punzo, V.: Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. Transp. Res. Part B: Methodol. **80**, 82–106 (2015). https://doi.org/10.1016/j.trb.2015.06.010
16. Popa, I.S., Zeitouni, K., Oria, V., Kharrat, A.: Spatio-temporal compression of trajectories in road networks. GeoInformatica **19**(1), 117–145 (2015). https://doi.org/10.1007/s10707-014-0208-4
17. Tichavska, M., Cabrera, F., Tovar, B., Araña, V.: Use of the automatic identification system in academic research. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST 2015. LNCS, vol. 9520, pp. 33–40. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27340-2_5
18. de Vries, G., van Someren, M.: Clustering vessel trajectories with alignment kernels under trajectory compression. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS (LNAI), vol. 6321, pp. 296–311. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15880-3_25
19. Wang, P., Goverde, R.M.: Multi-train trajectory optimization for energy efficiency and delay recovery on single-track railway lines. Transp. Res. Part B: Methodol. **105**, 340–361 (2017). https://doi.org/10.1016/j.trb.2017.09.012
20. Wu, X., Mehta, A.L., Zaloom, V.A., Craig, B.N.: Analysis of waterway transportation in Southeast Texas waterway based on AIS data. Ocean Eng. **121**, 196–209 (2016). https://doi.org/10.1016/j.oceaneng.2016.05.012
21. Wu, X., Rahman, A., Zaloom, V.A.: Study of travel behavior of vessels in narrow waterways using AIS data-a case study in Sabine-Neches waterways. Ocean Eng. **147**, 399–413 (2018). https://doi.org/10.1016/j.oceaneng.2017.10.049
22. Zhang, L., Meng, Q., Fwa, T.F.: Big AIS data based spatial-temporal analyses of ship traffic in Singapore port waters. Transp. Res. Part E: Logist. Transp. Rev. (2017). https://doi.org/10.1016/j.tre.2017.07.011
23. Zhang, L., Meng, Q., Xiao, Z., Fu, X.: A novel ship trajectory reconstruction approach using AIS data. Ocean Eng. **159**, 165–174 (2018). https://doi.org/10.1016/j.oceaneng.2018.03.085
24. Zhang, S., Liu, Z., Cai, Y., Wu, Z., Shi, G.: AIS trajectories simplification and threshold determination. J. Navig. **69**(4), 729–744 (2016)
25. Zhang, S., Shi, G., Liu, Z., Zhao, Z., Wu, Z.: Data-driven based automatic maritime routing from massive AIS trajectories in the face of disparity. Ocean Eng. **155**, 240–250 (2018). https://doi.org/10.1016/j.oceaneng.2018.02.060