SciencePG
Science Publishing Group

Ship Trajectory Data Compression Algorithms for Automatic Identification System: Comparison and Analysis

Le Qi[1, 2], Yuanyuan Ji[3, 4, *]

[1]School of Navigation, Wuhan University of Technology, Wuhan, China

[2]Hubei Key Laboratory of Inland Shipping Technology, Wuhan, China

[3]College of Information Science Technology, Dalian Maritime University, Dalian, China

[4]Ira A. Fulton Schools of Engineering, Arizona State University, Tempe, USA

Email address:
leqiem@hotmail.com (Le Qi), yyjem@outlook.com (Yuanyuan Ji)
[*]Corresponding author

Abstract: With the development of Internet of Things (IoT) technology and its vast applications in ship transportation systems, such as the Automatic Identification System (AIS), a large quantity of ship trajectory data have been recorded and stored. Nowadays ship transportation has also entered the age of big data, which can support IoT applications in Intelligent Transportation System (ITS), e.g. traffic monitoring, fleet management and traffic safety enhancement. However, the redundancy of ship trajectory data considerably reduces the effectiveness and efficiency of large scale traffic data storage, mining and visualization. Therefore, compression processing of the data becomes a very important issue for these applications. Because ship trajectory is a type of vector data, employing the vector data compression algorithms is an efficient way to solve the data redundancy problem. In this paper, the pseudo-code of five typical vector data compression algorithms for ship trajectory data compression is introduced. The performances of these algorithms were tested by the compression experiments of actual ship trajectories in the Qiongzhou Strait. The results show that ships' speeds and rate of turns, the requirement of real time processing can affect the option of the most appropriate algorithm, and the algorithm selection in different applications is suggested. The results and conclusions lay the foundation for the future development of ship transportation intelligentization.

Keywords: Automatic Identification System, Big Data, Data Compression Algorithms, Ship Trajectory

1. Introduction

Maritime transport plays a vital role in global supply chains. The *50 years of Review of Maritime Transport 1968–2018* published by United Nations Conference on Trade and Development (UNCTAD) notes that shipping carries the vast majority of international trade with its share ranging between 80 and 90 per cent of trade. In terms of trade value, the shipping share around 60 to 70 per cent of trade. 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 environmental protection, which need academia to develop supporting models and methods of analysis [1, 2].

With the development of information technology and its vast applications in transportation, trajectory data becomes easy to be obtained and have been widely used in road, railway and air traffic researches and practical applications. In marine transportation, ships' trajectories are becoming one of the main data sources for studying the characteristics of ship traffic behaviors, which will be an important basis for supporting the research and application of maritime transport in the future. At present, with the popularity of Automatic Identification System (AIS), a large number of ship trajectory data has been recorded and stored. Nowadays, ship transportation has also entered the age of big data [3-6].

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 intelligent of marine transportation becomes one of the most important research topics. Nonetheless, the AIS equipment of a ship generally publishes a message within every 2 s to 6 min, which makes the trajectory data from the AIS notably large. Because the AIS has a high frequency of information, the redundant problem of trajectory data from the AIS is serious [7-9]. This problem makes it difficult to be used in research and actual applications. Therefore, the ship trajectory data compression becomes particularly important [10, 11].

To compress the ship trajectory data, several methods have been proposed. Because ship trajectory compression quality significantly depends on the threshold selection, an adaptive Douglas Peucker algorithm with automatic thresholding for AIS-based ship trajectory compression was proposed [12-14]. Second, the reconstruction approach can also be used to reduce the volume of the data. Some algorithms were proposed to reconstruct a ship's trajectory by AIS data. In these algorithms, not only the navigational behavior could be clearly shown in the reconstructed trajectory, but also the data volume would decrease [15-17]. Third, the semantic trajectory compression method, which is used for the movement trajectories in an urban environment, has been used for data reduction techniques applied on AIS data [18, 19]. Besides, there are also many other methods used to compress ship trajectory data, e.g. clustering method [20, 21], piecewise linear segmentation method [22], directed acyclic graph method [23], direction-preserving trajectory simplification method [24], improved sliding window algorithm [25], etc.

It is noteworthy that, as a type of vector data, ship trajectory data can be compressed by the vector data compression algorithms, which can compress data very effectively for its advantages of easy implementation and low time complexity. Besides, it can also be used as a pre-processing of the above compression algorithms in practical applications. Generalized vector data compression should include the storage compression and re-sampling of the vector data [26, 27]. The concept of storage compression reduces the amount of vector data by converting the data type or file type. The concept of re-sampling is to extract subset B from set A which is a collection of the points that compose the vector graphics. Subset B should reflect the original data set A within a certain accuracy as much as possible and should ensure that the points of subset B are as little as possible.

At present, the most widely used algorithms for vector data are mainly the choosing interval points algorithm, limiting vertical distance algorithm, limiting angle algorithm, offset angle algorithm, Douglas Peucker algorithm, grating algorithm etc. [28-30]. The research on ship trajectory compression mainly focuses on the application and improvement of the Douglas Peucker algorithm, and some problems in the practical application of ship trajectory data

were effectively solved. However, many other vector data compression algorithms have not been applied to ship trajectory compression. More testing and analysis of these algorithms in ship 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 these algorithms in ship trajectory compression through experiments. The limiting angle algorithm has poor performance for vector data when the points are dense, so it is not been tested in this paper. The other algorithms are of five typical vector data compression algorithms, and many of the new vector data compression algorithms were proposed based on the five algorithms. Therefore, the five algorithms and their pseudo-code for ship trajectory data compression will be introduced. The data compression experiments for actual ship trajectories from AIS will be done, and the results will be used to analyze and compare the algorithms.

The remainder of this paper is organized as follows. Section 2 introduces the pseudo-code of the five algorithms for ship trajectories data compression. Section 3 presents the data compression experiments in which the performances of the algorithms are tested, and the results are analyzed and discussed. The study's conclusions are summarized in Section 4.

2. Compression Algorithms

Suppose a ship's trajectory is composed by a set of points in chronological order, which can be represented by set $A = \{p_1, p_2, …, p_n\}$. p is a point on the trajectory. The subscript represents the number of the point ordered by time. n is the total number of points on the trajectory. The ship sailed through each point in chronological order.

Let subset B stand for the compression result of set A. The pseudo-code of five typical vector data compression algorithms for ships' trajectories data compression is as follows.

2.1. Choosing Interval Points Algorithm (CIPA)

The basic idea of the choosing interval points algorithm is to retain a point in interval k points or an inter-equal distance d on the trajectory. Let k stand for the number of interval points. The pseudo-code of this algorithm is

Input: Set A, n, k
Output: Subset B
Initialize: $B \leftarrow \emptyset$
(1) $m = \lfloor (n-1)/k \rfloor$.
if n equals $mk + 1$ *then*
    (2) $B \leftarrow \{p_1, p_{k+1}, \cdots, p_{mk+1}\}$.
else
    (3) $B \leftarrow \{p_1, p_{k+1}, \cdots, p_{mk+1}, p_n\}$.
end
$\lfloor (n-1)/k \rfloor$ rounds to the nearest integer less than or equal to $(n-1)/k$. This algorithm supports real-time compression processing.

2.2. Limiting Vertical Distance Algorithm (LVDA)

The basic idea of the limiting vertical distance 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 $d_{threshold}$. If $d \geq d_{threshold}$, retain the middle point. Otherwise, if $d < d_{threshold}$, delete the middle point. After that step, select the second three consecutive points and repeat the process until all of the points on the trajectory are processed. The pseudo-code of this algorithm is

Input: Set A, n, $d_{threshold}$
Output: Subset B
Initialize: $B \leftarrow p_1, p_f \leftarrow p_1, p_s \leftarrow p_2, p_t \leftarrow \emptyset, d \leftarrow \emptyset$
for $i = 3$ to n *do*
    (1) $p_t \leftarrow p_i$
    (2) $d \leftarrow$ calculate the vertical distance from p_s to the line $p_f p_t$.
    if $d \geq d_{threshold}$ *then*
        (3) Add the point p_s to set B, $B \cup p_s$.
        (4) Update the p_f, p_s and p_t: $p_f \leftarrow p_s, p_s \leftarrow p_t, p_t \leftarrow \emptyset$.
        (5) Update the p_s and p_t: $p_s \leftarrow p_t, p_t \leftarrow \emptyset$.
    end
end

This algorithm supports the real-time compression processing.

2.3. Offset Angle Algorithm (OAA)

The basic idea of the offset angle 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 $\angle p_1 p_2 p_3$. Next, compare it with a given threshold $\theta_{threshold}$. If $\angle p_1 p_2 p_3 < \theta_{threshold}$, retain the middle point p_2, otherwise delete p_2. The pseudo-code of this algorithm is

Input: Set A, n, $\theta_{threshold}$
Output: Subset B
Initialize: $B \leftarrow p_1, p_f \leftarrow p_1, p_s \leftarrow p_2, p_t \leftarrow \emptyset, \alpha \leftarrow \emptyset$
for $i = 3$ to n *do*
    (1) $p_t \leftarrow p_i$
    (2) $\alpha \leftarrow$ calculate the angle of $\angle p_f p_s p_t$.
    if $\alpha < \theta_{threshold}$ *then*
        (3) Add the point p_s to set B, $B \cup p_s$.
        (4) Update the p_f, p_s and p_t: $p_f \leftarrow p_s, p_s \leftarrow p_t, p_t \leftarrow \emptyset$.
        (5) Update the p_s and p_t: $p_s \leftarrow p_t, p_t \leftarrow \emptyset$.
    end
end

This algorithm supports real-time compression processing. 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 can be highlighted by increasing the distance between the three selected points.

2.4. Douglas Peucker Algorithm (DPA)

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. After that step, calculate the distance between the other points in the middle to this straight line. Next, discover the maximum distance d_{max} and the corresponding point p_i. Then, compare d_{max} with $d_{threshold}$. If $d_{max} < d_{threshold}$, delete all points between the first and last points. If $d_{max} \geq d_{threshold}$, retain the point p_i and divide the trajectory into two segments $\{p_1, …, p_i\}$ and $\{p_i, …, p_n\}$. Next, for each segment, repeat the above process until to the end. The pseudo-code of this algorithm is

Input: Set A, n, $d_{threshold}$
Output: Subset B
Initialize: $B \leftarrow \emptyset$
function dp_algorithm(point set C)
    Initialize: Set $D \leftarrow \emptyset$, m $\leftarrow$ the number of points in set C, Set $C_1 \leftarrow \emptyset$, Set $C_2 \leftarrow \emptyset$.
    if $m \leq 2$
        (1) *return C*.
    else
        (2) D $\leftarrow$ calculate the distances from points $p_2, \cdots, p_{m-1}$ to the line $p_1 p_m$, where $\{p_1, \cdots, p_m\} \in C$.
        (3) Find the max value in set D, $d_{max} \leftarrow max\{D\}$, and the corresponding point p_i.
        if $d_{max} \geq d_{threshold}$ *then*
            (4) $C_1 \leftarrow \{p_1, \cdots, p_i\}$, $C_2 \leftarrow \{p_i, \cdots, p_m\}$.
            (5) *return dp_algorithm(C_1) $\cup$ dp_algorithm(C_2)*.
        else
            (6) *return* $\{p_1, p_m\}$
        end
    end
end
(7) $B \leftarrow dp\_algorithms(A)$.
(8) $B \leftarrow unique(B)$.

Where unique(B) returns a copy of the subset B that contains only the sorted unique observations. This algorithm does not support real-time compression processing.

2.5. Grating Algorithm (GA)

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 is $2 \times d_r$. The pseudo-code of this algorithm is

Input: Set A, n, d_r
Output: Subset B
Initialize: B $\leftarrow p_1$, $p_f \leftarrow p_1$, $p_s \leftarrow p_2$, $p_t \leftarrow \emptyset$, $\theta_r \leftarrow \emptyset$, $\theta_l \leftarrow \emptyset$
for $i = 3$ to n *do*
    (1) $\theta_r \leftarrow \theta_l \leftarrow \arctan(d_r/p_f p_s)$.
    (2) $p_t \leftarrow p_i$.
    if p_t on the right of vector $\overrightarrow{p_f p_s}$ *then*
        if $\angle p_s p_f p_t < \theta_r$ *then*
            (3) $\Delta\theta_r \leftarrow \theta_r - \angle p_s p_f p_t$.
            (4) $p_s \leftarrow p_t$, $\theta_r \leftarrow \theta_l \leftarrow \arctan\{d_r/P_f P_s\}$.
            (5) $\theta_r \leftarrow \min\{\theta_r, \Delta\theta_r\}$.

else
      (6) $B \leftarrow B \cup p_s$.
      (7) $p_f \leftarrow p_s, p_s \leftarrow p_t$.
end
  else
    if $\angle p_s p_f p_t < \theta_l$ *then*
      (8) $\Delta\theta_l \leftarrow \theta_l - \angle p_s p_f p_t$.
      (9) $p_s \leftarrow p_t,\ \theta_r \leftarrow \theta_l \leftarrow \arctan\{d_r/P_f P_s\}$.
      (10) $\theta_r \leftarrow \min\{\theta_r, \Delta\theta_r\}$.
    else
      (11) $B \leftarrow B \cup p_s$.
      (12) $p_f \leftarrow p_s, p_s \leftarrow p_t$.
    end
  end
end

This algorithm supports real-time compression processing.

3. Compression Experiment

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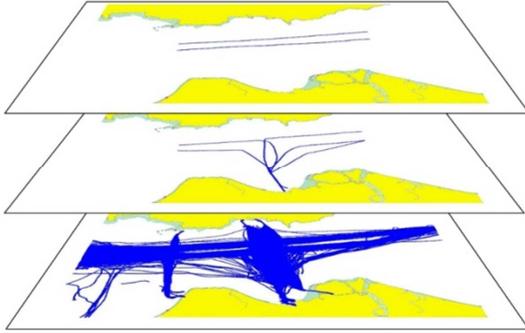

Figure 1. *Ship trajectories in the Qiongzhou Strait.*

The data sample for experiments is from the position report messages of AIS. The update frequency of the message is related to the ships' speed and the rate of turn (ROT). Therefore, the sample data should include the AIS messages when ships have different speeds and ROTs. Taking the trajectories of ships in the Qiongzhou Strait as an example, as shown in the bottom layer of Figure 1, it contains AIS data of ships with different navigational states. Five representative trajectories with obvious variations in speed and ROT were taken as the samples for the first experiments, as shown in the middle layer of Figure 1. And another two representative trajectories without obvious variations in speed and ROT were also taken as the samples for the second experiments, as shown in the top layer of Figure 1.

Firstly, compress the five representative trajectories by the choosing interval points algorithm, limiting vertical distance algorithm, offset angle algorithm, Douglas Peucker algorithm and grating algorithm. When the data compression ratio is 20, the compressed trajectories and the mean of the points' displacements are shown in Figure 2(b)~(e). All the compressed trajectories can reflect the spatial distribution of

the raw trajectories. However, there are also some different between the compression results.
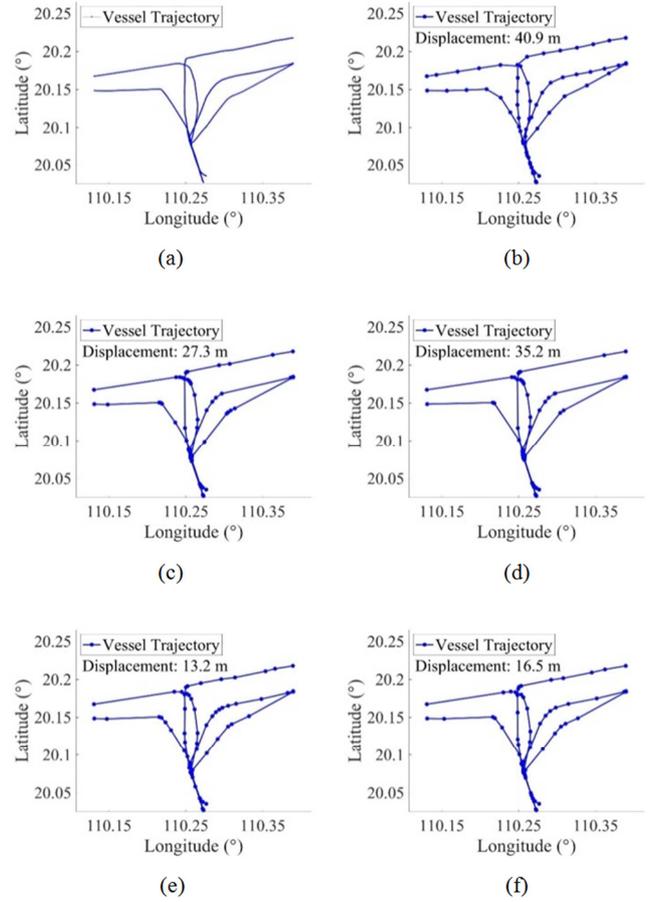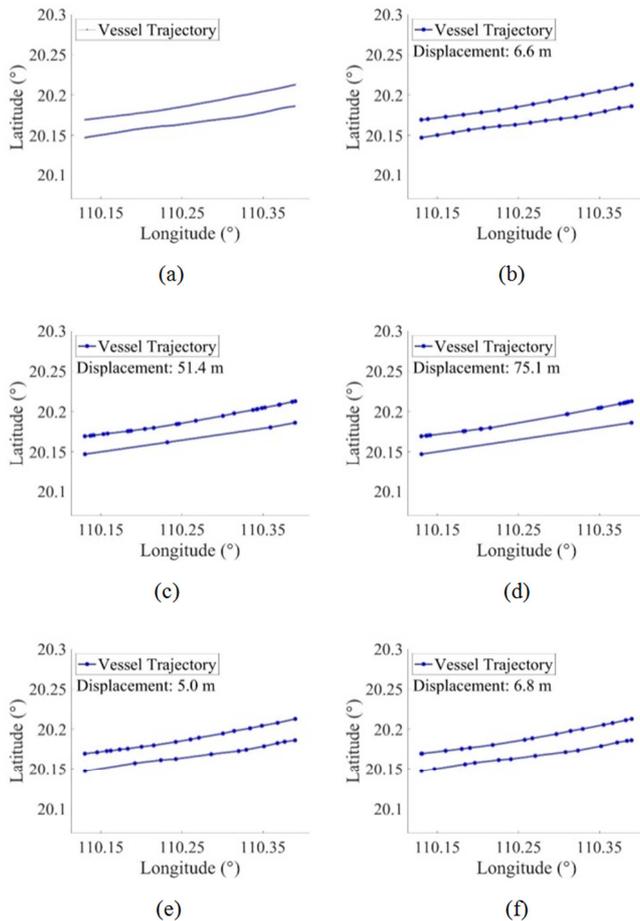

Figure 2. *The five representative raw trajectories and compression results based on these algorithms, when the data compression ratio is 20. (a) raw trajectories; (b) CIPA; (c) LVDA; (d) OAA; (e) DPA; (f) GA.*

Figure 2(b) shows the compression result by the choosing interval points algorithm. The points are evenly distributed but some details on the corners are ignored. In contrast to this, the points on the corners are preferentially retained by the offset angle algorithm, as shown in Figure 2(d). However, the points on other segments are ignored. Therefore, the distribution of the points in Figure 2(b) and Figure 2(d) are not proper. Besides, the mean of the points' displacements of the two results are larger than others, which is 40.9 m for the choosing interval points algorithm and 35.2 m for the offset angle algorithm.

The compression result by the limiting vertical distance algorithm, as shown in Figure 2(c), retains more details of the raw trajectories than that by the offset angle algorithm, but the improper distribution of points still remains and many points are concentrated on the corners. The mean of the points' displacements is 27.3 m, which is less than the above two algorithms.

Figure 2(e) and Figure 2(f) show the compression results by the Douglas Peucker algorithm and the grating algorithm. The performances of the two algorithms are better than the above algorithms. The points are properly distributed, which

means that the spatial structural characteristics of the trajectories and details on the corners are preserved well. The mean of the points' displacements caused by the Douglas Peucker algorithm is minimum, which is 13.2 m. The point displacement caused by the grating algorithm is a litter larger than it, which is 16.5m.

Figure 3. The two representative raw trajectories and compression results based on these algorithms, when the data compression ratio is 20. (a) raw trajectories; (b) CIPA; (c) LVDA; (d) OAA; (e) DPA; (f) GA.

Secondly, compress the two representative trajectories, by the choosing interval points algorithm, limiting vertical distance algorithm, offset angle algorithm, Douglas Peucker algorithm and grating algorithm. When the data compression ratio is 20, the compressed trajectories and the mean of the points' displacements are shown in Figure 3 (b)~(e). All the compressed trajectories can reflect the spatial distribution of the raw trajectories. However, there are also some different between the compression results.

For the choosing interval points algorithm, as shown in Figure 3(b), some detail still cannot be retained. However, as there are no obvious corner on the two trajectories, the compression result is better than the result shown in Figure 2(b). The mean of the points' displacements reduces to 6.6 m. On the contrary, for the offset angle algorithm, because the feature points on the local segments have been ignored, the improper distribution of the points is enlarged. As shown in Figure 3(d), there are just two points are retained on lower trajectory. The mean of the points'

displacements rises to 75.1 m.

Similar to the compression result shown in Figure 2(c), the compression result by the limiting vertical distance algorithm is better than the offset angle algorithm but the improper distribution of the points is still exist. The mean of the points' displacements shown in Figure 3(c) is 51.4 m.

Figure 3(e) and Figure 3(f) show the compression results by the Douglas Peucker algorithm and grating algorithm. The points distribution property of the two algorithm is still better the other algorithms. The important points on local segments are retained, and the detail variations of the two trajectories are preserved well. The mean of the points' displacements caused by Douglas Peucker algorithm is 5.0 m, which is still minimum in the results. The mean of the points' displacements caused by the grating algorithm is similar to the choosing interval points algorithm, which is 6.8 m.

In summary, the performance of the offset angle algorithm is worse than the other algorithms. The performance of the limiting vertical distance algorithm is a little better than it. The performance of the choosing interval points algorithm is unstable, which is more suitable for the trajectories that are approximately straight lines. The performance of the Douglas Peucker algorithm and grating algorithm are better than other algorithms in the experiments. The data compression error caused by the Douglas Peucker algorithm is minimum, 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 supports real-time processing. Therefore, when it needs to compress historical ship trajectory data, the Douglas Peucker algorithm is recommended, and when it needs to compress ship trajectory data in real time, the grating algorithm is recommended.

4. Conclusions

Several classic vector data compression algorithms are introduced, as well as their pseudo-code for ship trajectory compression. Through the experiment of ship trajectory data (from AIS) compression, the performances of these algorithms are analyzed. The advantages and disadvantages of these algorithms in compressing ship trajectory data are compared. The result shows that the performances 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. When the trajectories are approximately straight line, the choosing interval points algorithm can be considered, because its performance will rise in this situation and it is suitable for both historical and real-time data compression. The research results provide a support for the selection of the algorithms in practical applications, e.g. ship traffic monitoring, safety enhancement and fleet management, based on AIS.

Acknowledgements

References

[1]   Bell, M. G., Meng, Q. Special issue in Transportation Research
Part B-Shipping, port and maritime logistics. Transportation
Research Part B: Methodological 2016, 93, 697–699. Maritime
Logistics.

[2]   United Nations Conference on Trade and Development
(UNCTAD), 50 years of Review of Maritime Transport, 1968–
2018, New York, 2018.

[3]   Moffitt, K. C., Vasarhelyi, M. A. AIS in an Age of Big Data.
Journal of Information Systems 2013, 27, 1–19.

[4]   Dittmar, C., Die nächste Evolutionsstufe von AIS: Big Data. In
Analytische   Informationssysteme: Business
Intelligence-Technologien und -Anwendungen, Gluchowski, P.,
Chamoni, P., Eds., Springer Berlin Heidelberg: Berlin,
Heidelberg, 2016, pp. 55–65.

[5]   Zhang, L., Meng, Q., Fwa, T. F. Big AIS data based
spatial-temporal analyses of ship traffic in Singapore port
waters. Transportation Research Part E: Logistics and
Transportation Review. 2017.

[6]   Isenor, A. W., 342 St-Hilaire, M. O., Webb, S., Mayrand, M.
MSARI: A Database for Large Volume Storage and Utilisation
of Maritime Data. Journal of Navigation 2017, 70, 276–290.

[7]   Bole, A., Wall, A., Norris, A. Chapter 5 - Automatic
Identification System (AIS). In Radar and ARPA Manual
(Third Edition), Third Edition ed., Bole, A., Wall, A., Norris,
A., Eds., Butterworth-Heinemann: Oxford, 2014, pp. 255–275.

[8]   Zhi Sang, L., Wall, A., Mao, Z., ping Yan, X., Wang, J. A novel
method for restoring the trajectory of the inland waterway ship
by using AIS data. Ocean Engineering 2015, 110, 183-194.

[9]   Qi, L., & Zheng, Z. (2016). Trajectory prediction of vessels
based on data mining and machine learning. J. Digit. Inf.
Manage, 14 (1), 33-40.

[10] Qi, L., & Zheng, Z. (2016). Vessel Trajectory Data Compression
based on Course Alteration Recognition. A A, 1 (2), 1.

[11] Ji, Y., Xu, W., Deng, A. A Study of Vessel Trajectory
Compression Based on Vector Data Compression Algorithms.
International Conference on Business Information Systems.
Springer, 2019, pp. 473–484.

[12] Zhang, S. k., Liu, Z. j., Cai, Y., Wu, Z. l., Shi, G. y. AIS
Trajectories Simplification and Threshold Determination.
Journal of Navigation 2015, -1, 1–16.

[13] Zhao, L., Shi, G. A trajectory clustering method based on
Douglas-Peucker compression and density for marine traffic
pattern recognition. Ocean Engineering 2019, 172, 456–467.

[14] Liu, J., Li, H., Yang, Z., Wu, K., Liu, Y., Liu, W. Adaptive
Douglas-Peucker Algorithm with Automatic Thresholding for
AIS-Based Vessel Trajectory Compression. IEEE Access 2019,

PP, 1–1.

[15] Liang, M., Liu, W., Zhong, Q., Liu, J., Zhang, J. Neural
Network-Based Automatic Reconstruction of Missing Vessel
Trajectory Data. 2019, pp. 426–430.

[16] Zhang, L., Meng, Q., Xiao, Z., Fu, X. A novel ship trajectory
reconstruction approach using AIS data. Ocean Engineering
2018, 159, 165–174.

[17] Xiao, F., Ligteringen, H., van Gulijk, C., Ale, B. Comparison
study on AIS data of ship traffic behavior. Ocean Engineering
2015, 95, 84-93.

[18] Schmid, F., Richter, K. F., Laube, P. Semantic Trajectory
Compression. Advances in Spatial and Temporal Databases,
Mamoulis, N., Seidl, T., Pedersen, T. B., Torp, K., Assent, I.,
Eds., Springer Berlin Heidelberg: Berlin, Heidelberg, 2009, pp.
411–416. doi: 10.1007/978-3-642-02982-0_30.

[19] Ifrim, C., Iuga, I., Pop, F., Wallace, M., Poulopoulos, V. Data
Reduction Techniques Applied on Automatic Identification
System Data. Semantic Keyword-Based Search on Structured
Data Sources, Szyman´ ski, J., Velegrakis, Y., Eds., Springer
International Publishing: Cham, 2018, pp. 14–19.

[20] Yang, B., Li, Q. Efficient compression of vector data map based
on a clustering model. Geo-spatial Information Science 2009,
12, 13–17.

[21] de Vries, G., van Someren, M. Clustering Vessel Trajectories
with Alignment Kernels under Trajectory Compression.
Machine Learning and Knowledge Discovery in Databases,
Balcázar, J. L., Bonchi, Gionis, A., Sebag, M., Eds., Springer
Berlin Heidelberg: Berlin, Heidelberg, 2010, pp. 296–311.

[22] De Vries, G. K. D., van Someren, M. Machine learning for
vessel trajectories using compression, alignments and domain
knowledge. Expert Systems with Applications 2012, 39,
13426-13439.

[23] Cao, W., Li, Y. DOTS: An online and near-optimal trajectory
simplification algorithm. Journal of Systems and Software
2017, 126, 34–44.

[24] Deng, Z., Han, 447 W., Wang, L., Ranjan, R., Zomaya, A. Y.,
Jie, W. An efficient online direction-preserving compression
approach for trajectory streaming data. Future Generation
Computer Systems 2017, 68, 150–162.

[25] Gao, M., Shi, G. Y. Ship Spatiotemporal Key Feature Point
Online Extraction Based on AIS Multi-Sensor Data Using an
Improved Sliding Window Algorithm. Sensors 2019, 19.

[26] Image Compression. In Data Compression: The Complete
Reference, Springer New York: New York, NY, 2004, pp. 251–
512.

[27] Yang, B., Purves, R. S., Weibel, R. Variable-resolution
Compression of Vector Data. GeoInformatica 2008, 12, 357–376.

[28] Ji, H., Wang, Y. The Research on the Compression Algorithms
for Vector Data. International Conference on Multimedia
Technology, 2010, pp. 1–4.

[29] Gudmundsson, J., Katajainen, J., Merrick, D., Ong, C., Wolle,
T. Compressing spatio-temporal trajectories. Computational
Geometry 2009, 42, 825–841.

[30] Sandu Popa, I.; Zeitouni, K.; Oria, V.; Kharrat, A.
Spatio-temporal compression of trajectories in road networks.
GeoInformatica 2015, 19, 117–145.