Application of adaptive douglas-peucker with acceleration algorithm in ship trajectory compression

Lichao Yang*
State Key Laboratory of Maritime Technology and Safety
School of Navigation, Wuhan University of Technology
Wuhan, Hubei, China
ylchy@whut.edu.cn

Jingxian Liu
State Key Laboratory of Maritime Technology and Safety
School of Navigation, Wuhan University of Technology
Wuhan, Hubei, China
ljxteacher@sohu.com

Yukuan Wang
Hainan Institute
School of Navigation, Wuhan University of Technology
Sanya, Hainan, China
wangyk@whut.edu.cn

***Abstract*—In this study, the Adaptive Douglas-Peucker with Acceleration (ADPA) algorithm is introduced for compressing Automatic Identification System (AIS) ship trajectory data. This novel algorithm, tailored for maritime trajectory analysis, dynamically calculates compression thresholds by considering both acceleration and distance to a baseline, thereby enhancing the compression process. Unlike traditional methods, the ADPA algorithm is particularly effective in handling complex trajectory patterns, including circular and semi-circular paths. Empirical analysis using AIS data from the Ningbo-Zhoushan area and the Yangtze River Estuary demonstrated the ADPA algorithm's capability, achieving a compression rate of over 50%, significantly higher than that of standard Douglas-Peucker algorithms. This advanced approach provides a more efficient and accurate means of processing large-scale maritime trajectory data.**

Keywords—Water traffic, Ship trajectory compression, Automatic identification system (AIS), adaptive Douglas-Peucker with an acceleration(ADPA), Sliding window

I. INTRODUCTION

In the dynamic and complex realm of maritime transportation, efficient and accurate analysis of vessel trajectories is crucial. Maritime paths, unlike the structured and predefined routes of urban traffic, are less distinct and heavily influenced by environmental factors. This makes extracting and analyzing these routes a significant challenge.

The original AIS data, composed of numerous similar trajectory points, contains a considerable amount of redundancy[1]. Hence, it is essential to compress the data as a preprocessing step before mining the AIS data. The Douglas-Peucker (DP) algorithm is regarded as one of the most effective methods for linear data compression and is widely used in the compression of trajectories[2]. However, the effectiveness of the traditional DP algorithm heavily depends on the setting of the threshold value. Thus, finding a suitable compression threshold is a key aspect of the DP algorithm[3].

Li et al.[4]proposed an adaptive Douglas-Peucker algorithm based on trajectory speed variations. Tang et al.[5] introduced the Adaptive-threshold Douglas-Peucker algorithm, an innovative approach wherein the pivotal points of each trajectory are identified based on a dynamically varying threshold rate. Zhou et al.[6]introduced a novel approach for compressing AIS trajectories, utilizing the Multi-Objective Peak Douglas-Peucker Algorithm. This method represents an advancement in trajectory compression techniques.

Although previous studies have addressed the threshold setting issue of the DP algorithm to some extent, such algorithms significantly lose their effectiveness when compressing circular or semi-circular trajectories. To address this issue, we propose a method combining the adaptive Douglas-Peucker with an acceleration algorithm (ADPA) and the sliding window algorithm for compressing ship trajectories. This method adaptively determines the compression threshold for each trajectory point by quantifying the acceleration and the change in distance to the baseline. Furthermore, it uses the sliding window approach to segment and compress the trajectory, enhancing the compression effectiveness for circular and semi-circular paths.

II. ADAPTIVE DOUGLAS-PEUCKER WITH ACCELERATION ALGORITHM

A. Trajectory compression threshold adaptation

In order to reduce the computing costs associated with the clustering algorithm, it becomes imperative to incorporate trajectory compression within the preprocessing of AIS-based vessel trajectories. Trajectory compression directly determines the accuracy of turn points clustering and extraction waypoints. However, it is difficult to determine the trajectory compression threshold. Therefore, we proposed an adaptive douglas-peucker with acceleration algorithm (ADPA). This algorithm dynamically generates the corresponding trajectory compression threshold by computing the acceleration of a segment of the trajectory and the distance to the baseline. The acceleration and distance to the baseline at a specific point within the trajectory effectively characterize the ship's motion status. Assessing the alteration in acceleration and the shift in distance from that particular point to the baseline enables the determination of the degree of change in the ship's motion state. Consequently, the trajectory compression threshold is derived by gauging the alteration in acceleration and the shift in distance from the point to the baseline. The threshold model is shown below.

$$\theta_i = 0.5 * d_{ave} * \left(\left(1 - \frac{|a_i - a_{ave}|}{|a_{max} - a_{ave}|}\right) + \left(1 - \frac{|d_i - d_{ave}|}{|d_{max} - d_{ave}|}\right) \right) \quad (1)$$

Where θ_i denotes the threshold of point i, a_i indicates the acceleration at point i, a_{ave} expresses the average acceleration of the entire trajectory, a_{max} is the maximum

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00045

acceleration observed in the trajectory. Similarly, d_i indicates the distance from point i to the baseline, d_{ave} signifies the average distance of the trajectory from the baseline, d_{max} represents the maximum distance of the trajectory from the baseline. The baseline in this context is defined as the linear segment connecting the trajectory's starting and ending points.

The trajectory compression schematic iamge based on the original DP and the proposed ADPA algorithm is displayed in Fig. 1 The figure illustrates the outcome of applying the DP algorithm to trajectory compression, where a uniform threshold value for all trajectory points resulted in no points being eligible for compression. In contrast, utilizing the ADPA algorithm on the same trajectory, with its adaptive threshold values that vary for each point, prioritizing those with significant velocity changes, led to the selective compression of points, notably points P5 and P8.

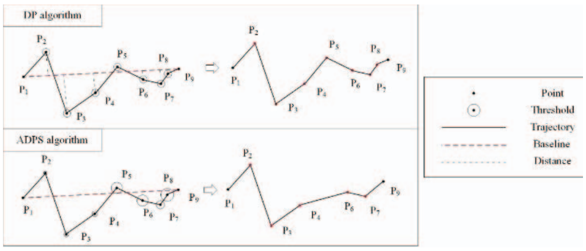

Fig. 1. Comparison of DP and ADPA Algorithms: A Schematic Representation

B. Sliding window & ADPA algorithm

The efficacy of the proposed ADPA algorithm notably decreases when it is employed for compressing trajectories that are circular or semi-circular in nature. This limitation stems from the baseline construction, which solely relies on the trajectory's start and end points, the method that is insufficient for circular or semi-circular forms. To address this issue, a sliding window algorithm has been adopted for the computation of the trajectory compression thresholds. The pseudo-code of the ADPS algorithm is described in TABLE I.

TABLE I. TABLE ADPA ALGORITHM

Algorithm 1 ADPA algorithm

```
Input: $T_i^j = (lon_i^j, lat_i^j, t_i^j, v_i^j)$, $i = 1,2,\dots,n$, $j = 1,2,\dots,m$, $WS = k$, $k \in N$
    ▻/* $(lon_i^j, lat_i^j)$ indicates the $i^{th}$ coordinate point in the $j^{th}$ trajectory,
    $t_i^j$ and $v_i^j$ represent the time and speed of point $(lon_i^j, lat_i^j)$, $WS$
    denotes the window size threshold for the sliding window algorithm. */
Output: $TT_i^j$
    ▻/* $TT_i^j$ is the compressed trajectories*/
//Calculate $\theta^j$//
for j = 1: m do
    if len ($T^j$) < WS then
        for i = 1: n do
            $a_i^j = |v_{i+1}^j - v_i^j| / |t_{i+1}^j - t_i^j|$;
            $a_{ave}^j = \sum_{i=1}^n a_i^j / (n-1)$;
            $a_{max}^j = max(a_i^j)$;
            $d_i^j = \frac{|(lon_n - lon_1)(lat_1 - lat_i) - (lon_1 - lon_i)(lat_n - lat_1)|}{\sqrt{(lon_n - lon_1)^2 + (lat_n - lat_1)^2}}$
            $d_{ave}^j = \sum_{i=1}^n d_i^j / (n-1)$;
            $d_{max}^j = max(d_i^j)$;
            $\theta_i^j = \frac{1}{2} * d_{ave}\left(\left(1 - \frac{|a_i^j - a_{ave}^j|}{|a_{max}^j - a_{ave}^j|}\right) + \left(1 - \frac{|d_i^j - d_{ave}^j|}{|d_{max}^j - d_{ave}^j|}\right)\right)$
        end for
    //sliding window//
    else
        for i = 1: (n-WS) do
            for p = i: $(i + WS)$ do
                $a_{i_p}^j = |v_{i+1}^j - v_i^j| / |t_{i+1}^j - t_i^j|$;
                $a_{ave_p}^j = \sum_{i=1}^n a_i^j / (n-1)$;
                $a_{max_p}^j = max(a_i^j)$;
                $d_{i_p}^j = \frac{|(lon_n - lon_1)(lat_1 - lat_i) - (lon_1 - lon_i)(lat_n - lat_1)|}{\sqrt{(lon_n - lon_1)^2 + (lat_n - lat_1)^2}}$
                $d_{ave_p}^j = \sum_{i=1}^n d_i^j / (n-1)$;
                $d_{max_p}^j = max(d_i^j)$;
                $\theta_{i_p}^j = 0.5 * d_{ave}\left(\left(1 - \frac{|a_i^j - a_{ave}^j|}{|a_{max}^j - a_{ave}^j|}\right) + \left(1 - \frac{|d_i^j - d_{ave}^j|}{|d_{max}^j - d_{ave}^j|}\right)\right)$
            end for
        end for
    end if
end for
$\theta^j = \{\theta_1^j, \theta_2^j, \dots, \theta_n^j\}$
//Generate $TT_i^j$//
for j = 1: m do
    for i = 1: n do
        if $d_i^j > \theta_i^j$ then
            $TT_i^j \leftarrow i$;
        else
            delet i;
        end if
    end for
end for
```

Fig. 2 and Fig. 3 respectively illustrate the compression process of a circular trajectory using the DP algorithm and the ADPA algorithm. In the DP algorithm, the baseline is defined using only the start and end points of the trajectory, with the distances of individual trajectory points to this baseline being compared to a predefined threshold. However, this methodology proves inadequate for compressing circular trajectories, often failing to achieve any significant reduction in trajectory complexity. Conversely, the ADPA algorithm enhances this process by integrating a sliding window method. This addition facilitates the division of circular trajectories into several segments, each of which is compressed independently. Figure 3 exemplifies this, where the application of the ADPA algorithm on a circular trajectory results in a reduction of trajectory points from 13 to 9.

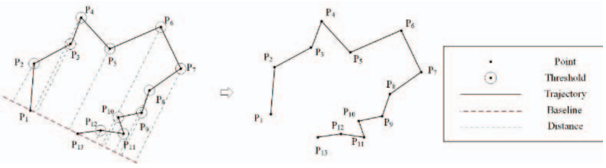

Fig. 2. Circular Trajectory Compression Schematic - DP Algorithm

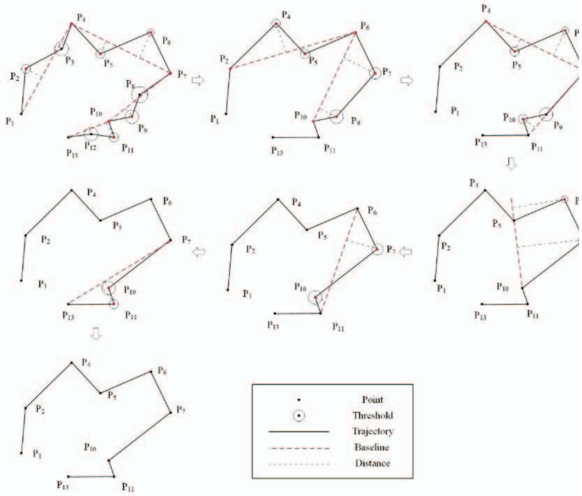

Fig. 3. Circular Trajectory Compression Schematic - ADPA Algorithm

III. EXPERIMENTAL CASE STUDY

In order to ascertain the efficacy of the ADPA algorithm, AIS data from May 2019 pertaining to the area around Ningbo-Zhoushan and the Yangtze River Estuary were selected for compression analysis. The designated maritime area was enclosed by the coordinates at Points A, B, C, and D. The precise coordinates for these points are enumerated in Table II. The initial data set consisted of 14,641 maritime trajectories, as depicted in Figure 4. To maintain data integrity, any entries with apparent anomalies, including irregular speeds or land-based coordinates, were rigorously filtered out.

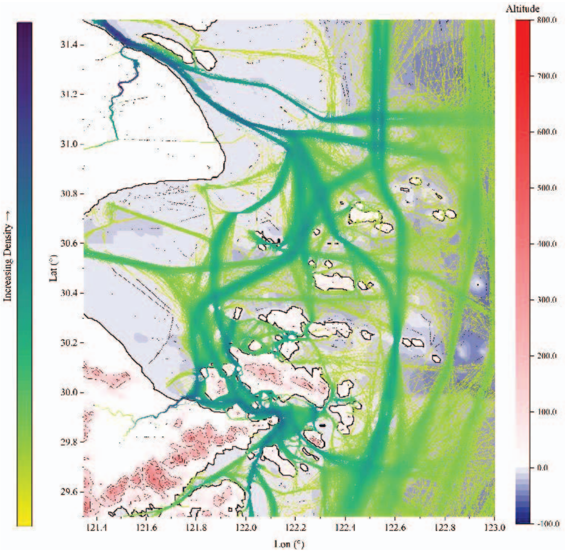

Fig. 4. Raw AIS Vessel Trajectory Plot

TABLE II. GEOGRAPHIC COORDINATE

Point	**Coordinate**	
	latitude	***longitude***
A	29°30'00.10"N	121°19'48.05"E
B	31°29'59.99"N	121°19'48.05"E
C	31°29'59.99"N	122°59'59.93"E
D	29°30'00.10"N	122°59'59.93"E

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

First, we sequentially selected AIS data by date. Then, we traversed the selected AIS data using the Maritime Mobile Service Identity (MMSI) numbers. Finally, we applied the APDA algorithm for data compression. The original dataset contained a total of 52,417,608 trajectory points, which was reduced to 26,541,028 points post-compression, achieving a compression rate of 50.63%. Detailed daily compression results are tabulated in Table III. Furthermore, to demonstrate the ADPA's compression efficacy, trajectory scatter plots and polyline diagrams for pre- and post-compression data from May 4th were generated, as depicted in Fig. 5 and Fig. 6.

TABLE III. TRAJECTORY COMPRESSION RESULTS

Date of Data	**Number of Raw Data Trajectory Points**	**Number of Compressed Trajectory Points**	**Compression Ratio (%)**
05-01	1203675	650273	54.02
05-02	1821798	915907	50.27
05-03	2256009	1083938	48.05
05-04	1871273	927526	49.57
05-05	1609480	812961	50.51
05-06	1731345	867298	50.09
05-07	1638952	796730	48.61
05-08	1818955	904864	49.75
05-09	1680551	846212	50.35
05-10	1667049	820902	49.24
05-11	1876772	922764	49.17
05-12	1742060	865936	49.71
05-13	2120463	1027346	48.45
05-14	1581279	813837	51.47
05-15	1616939	855369	52.90
05-16	1710894	901228	52.68
05-17	1195862	722218	60.39
05-18	1936007	936992	48.40
05-19	1978806	1018151	51.45
05-20	1822739	919555	50.45
05-21	1808059	920831	50.93
05-22	1931921	976682	50.55
05-23	1599565	834869	52.19
05-24	1717624	884206	51.48
05-25	1461538	764608	52.32

Date of Data	**Number of Raw Data Trajectory Points**	**Number of Compressed Trajectory Points**	**Compression Ratio (%)**
05-26	999040	550203	55.07
05-27	1759207	822262	46.74
05-28	1717777	871142	50.71
05-29	1862478	931049	49.99
05-30	1484598	760473	51.22
05-31	1194893	614696	51.44

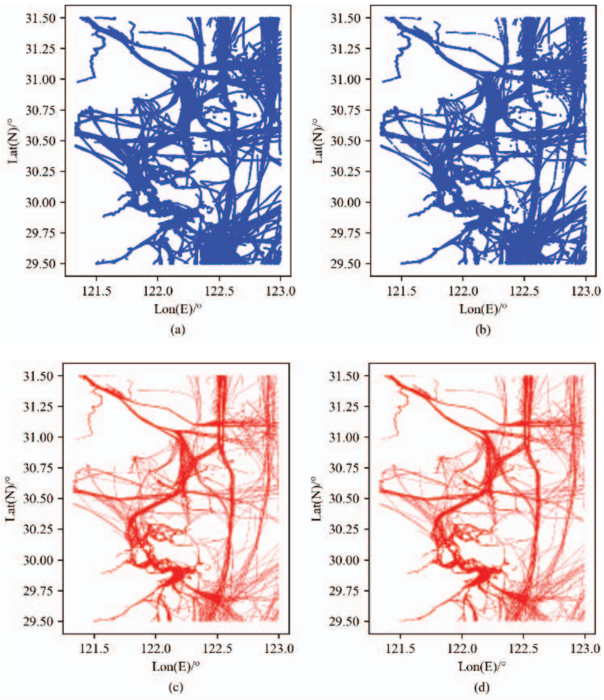

Fig. 5. The original vessel trajectories and compressed trajectories (2019-05-04), (a) the original point data, (b) the compressed point data, (c) the original trajectory data, (d) the compressed trajectories

IV. DISCUSSION

In this study, we developed the Adaptive Douglas-Peucker with Acceleration (ADPA) algorithm for preprocessing and compressing vessel trajectories from Automatic Identification System (AIS) data. The ADPA algorithm dynamically calculates compression thresholds based on acceleration and distance to a baseline, enabling more effective characterization of ships' motion changes. This approach notably outperforms the traditional Douglas-Peucker method, especially in compressing points with significant velocity changes, as evidenced in our empirical study on AIS data from the Ningbo-Zhoushan area and the Yangtze River Estuary. Here, the ADPA algorithm reduced the dataset from 52,417,608 to 26,541,028 trajectory points, achieving a 50.63% compression rate. However, its performance is less effective for circular or semi-circular trajectories and is influenced by AIS data quality and environmental factors, highlighting areas for future improvement.

V. CONCLUSIONS

In this study, we have developed the Adaptive Douglas-Peucker with Acceleration (ADPA) algorithm, a novel approach for efficiently preprocessing and compressing AIS-based maritime trajectories. This method represents a significant advancement in handling large volumes of maritime data, maintaining the integrity and usability of trajectory information.

The ADPA algorithm distinguishes itself through its dynamic threshold calculation for trajectory compression, adeptly managing complex trajectory patterns and improving upon traditional compression methods. Its adaptive nature enables selective compression of critical trajectory points, enhancing data processing efficiency and the accuracy of maritime traffic analysis.

However, the algorithm's reliance on AIS data quality and its limited consideration of environmental factors highlight areas for future improvement. Subsequent research will focus on enhancing the ADPA algorithm's robustness by integrating diverse data sources and accounting for environmental influences. This evolution aims to extend the algorithm's applicability, making it a more comprehensive tool for maritime navigation and safety analysis.

REFERENCES

[1] Yang, L., Liu, J., Liu, Z., & Luo, W. (2023). Grounding risk quantification in channel using the empirical ship domain. Ocean Engineering, 286: 115672.

[2] Muckell, J., Olsen, P. W., Hwang, J. H., Lawson, C. T., & Ravi, S. S. (2014). Compression of trajectory data: a comprehensive evaluation and new approach. GeoInformatica, 18: 435-460.

[3] Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: the international journal for geographic information and geovisualization, 10(2): 112-122.

[4] Li, H., Lam, JSL., Yang, Z., Liu, J., Liu, R. W., Liang, M., & Li, Y. (2022). Unsupervised hierarchical methodology of maritime traffic pattern extraction for knowledge discovery. Transportation Research Part C: Emerging Technologies, 143: 103856.

[5] Tang, C., Wang, H., Zhao, J., Tang, Y., Yan, H., & Xiao, Y. (2021). A method for compressing AIS trajectory data based on the adaptive-threshold Douglas-Peucker algorithm. Ocean Engineering, 232: 109041.

[6] Zhou, Z., Zhang, Y., Yuan, X., & Wang, H. (2023). Compressing AIS Trajectory Data Based on the Multi-Objective Peak Douglas–Peucker Algorithm. IEEE Access, 11: 6802-6821.