Visuliazaiton of maritime trajectory based on edge bundling

Abstract—**In this paper, we propose a data visualization method for ship trajectories using the RDP and FDEB bundling algorithms. The RDP algorithm reduces the number of data points in the trajectories while retaining key features. The FDEB bundling algorithm simplifies the visualization by grouping similar trajectories together. An East China Sea shipping case study is reported using the proposed approach. The results show that our method can effectively reduce the complexity of the visualization while retaining key features of the data. The bundled visualization also clearly represents the main shipping routes and allows for a comparison of the frequency of routes. Our approach demonstrates a good balance between reducing data points and retraining key features while adjusting the parameters of the RDP algorithm and considering the trade-off between the data size and the visualization quality. Our approach can also be seen as a fruitful and effective tool for visualizing and analyzing ship trajectory data.**

Index Terms—**Computer Graphics**

I. Introduction

Maritime transport and navigation are important components of global trade and commerce in the context of globalization, facilitating the flow and exchange of people, goods, and resources worldwide. More than 90% of global goods are transported by sea, and efficient, reliable, and stable shipping routes are critical to the global supply chain and worldwide business and trade. Maritime routes are also important channels for the transport of energy resources such as oil and natural gas and are thus essential for driving global economic development.

In recent years, with the development of technologies such as GPS and remote sensing, it has become easier to obtain positioning data, such as latitude and longitude, for ship navigation. As a result, there is a growing demand for a visualization technique that is easy to interpret and display maritime routes. Moreover, as maritime traffic continues to increase, there is a need for a visualization approach that can effectively and intuitively display large quantities of maritime routes.

Therefore, the application of maritime route visualization technology has become increasingly important in maritime transportation and navigation. This technology provides an effective means of analyzing and visualizing maritime traffic data, aiding in the safe and efficient operation of ships and promoting the development of maritime transport and navigation. Moreover, the visualization of maritime routes can facilitate decision-making for maritime operators, enabling them to optimize routes and manage maritime traffic effectively.

The purpose of this study is to develop a universally applicable visualization technique for large-scale maritime routes to improve the layout and optimization of routes on maps. With the development of maritime transportation, the visualization of a large number of routes often leads to confusion and congestion, making the overall layout disorderly and difficult to explain and summarize.

To improve the visualization of a large number of maritime routes, we reviewed relevant literature on maritime visualization and identified commonly used techniques and main features. We proposed a new method for visualizing maritime routes by utilizing advanced layout algorithms to improve the optimization of route visualization. The significance of this study lies in providing a concise and clear way for the visualization of a large amount of route data. Through visualization, further improvements in route layout can be achieved, providing effective references for the design and implementation of maritime routes. In addition, our method also provides a visualization approach for the field of data visualization, demonstrating the potential of different layout algorithms in solving practical problems in visualization.

In conclusion, this study contributes to the development of maritime route visualization technology, providing a new method to improve the visualization of a large amount of maritime route data. The proposed approach has important practical implications for the design and optimization of maritime routes and has the potential to be applied to other areas of data visualization as well.

II. Related work

Many studies have been developed to visualize sea routes. One approach is to use 3D visualization techniques to display the routes in a visual way. For example, Huang et al [7] developed a 3D maritime navigation system that allows users to explore and analyze routes from different perspectives, using interactive 3D models of ships, ports, and other maritime features. Another approach is to use interactive maps that allow the user to manually zoom in and out interactively and thus explore shipping routes. For example, Karydis et al. [9] developed an interactive map-based system that allows mariners to visualize and plan routes using real-time data on weather conditions, vessel location, and other factors.

In addition to visualization techniques, several studies have investigated methods to optimize the layout and organization of maritime routes on a map. One approach is to use clustering techniques to group similar routes together and reduce visual clutter. For example, Kim et al. [10] developed a clustering algorithm to group routes based on similarities in departure and destination, ship type, and other factors. Another approach is to use force-directed algorithms to optimize the layout of routes on the map while minimizing overlap and crossover. For example, Wang et al. [12] developed a force-oriented algorithm that takes into account the distance between ports, channel rules, and other factors to optimize the route layout on the map.

Edge bundling techniques are also widely used in maritime route visualization. Edge Bundling is a technique used to visualize a large number of edges. In a network with a large number of edges, traditional edge visualization techniques (e.g., direct drawing of edges) often result in overly dense images with crossed edges. Edge bundling techniques make the image more legible by bundling close edges together while preserving the information of the original edge data.

- Force-directed edge bundling (FDEB) [4]
  Force-directed edge bundling (FDEB) is a visualization algorithm for improving the layout of graphic edges. It reduces overlap and congestion by applying a physical model to the edges. The algorithm is based on a continuum mechanics model that simulates the interaction of edges by applying forces and then uses edge bundling techniques to bundle the edges together to create clear, readable layouts. FDEB algorithms have been widely used in large-scale graphs and have been shown to improve the visualization quality and readability of graphs.
- Graph Bundling by Kernel Density Estimation(GB-KDE) [8]
  Graph Bundling by Kernel Density Estimation (GB-KDE) algorithm is a common algorithm for optimizing the layout of graph edges. It uses kernel density estimation to cluster edges and bundle similar edges together to form sharper and more readable edge layouts. the GB-KDE algorithm can bundle edges quickly and efficiently in large-scale graphs, and can adaptively adjust the degree of bundling to accommodate different edge densities and distributions.
- Skeleton-based edge bundling (SBEB) [6]
  Skeleton-based edge bundling (SBEB) algorithm is a graph skeleton-based edge layout optimization algorithm. It uses a skeleton to represent the shape of the graph and uses edge bundling techniques to bundle adjacent edges together to form a clear layout.SBEB algorithm is a simple and effective edge layout algorithm that is particularly suitable for graphs with complex shapes and can significantly improve the readability and visualization quality of the graph.

In summary, these techniques and methods are visualization algorithms for improving edge layouts in large-scale graphics and have been widely used and validated. In this study, we will combine these techniques and methods to propose a new maritime route visualization algorithm to optimize the layout and readability of routes.

III. Methodology

As one of the largest inland bays in China, the Bohai Sea is a very important maritime transportation hub where a large number of ships sail every day. Therefore, the study of maritime route visualization in Bohai Bay is representative and important. Through the route visualization of Bohai Sea, we have researched a set of generally applicable maritime route visualization methods.

We take the route data of Bohai Bay as an example to explore the visualization under a large number of routes. The original data of ship navigation is the latitude and longitude coordinates recorded at a high frequency, and each csv file represents the navigation process of a ship. Therefore, we consider a file as a ship's route from departure to destination.

A. Data processing and analysis techniques

Although the data of the Bohai route can accurately and minutely reflect the positioning of the ship's voyage at different time periods, its data volume is too large, which adds a huge workload to the storage and calculation of visualization. For example, the average number of latitude and longitude information sent for a ship voyage is about 6000. Thus, the amount of raw data is so large that it becomes an important part of the work to consider the compression and simplification of the data.

In the actual navigation, the ship's route is often tortuous and changeable, and if all the position data are visualized directly as the points of the route, it will not only lead to an excessive amount of data but also make it difficult to show the route direction and changes visually. Therefore, we need to pre-process the raw data and extract the key points to better show the ship's route.

To solve this problem, we used the Rmer-Douglas-Peucker (RDP) algorithm [2] [11], a simple and effective data compression algorithm for reducing the number of points in a curve or a fold line. The basic idea of the RDP algorithm is to find a simple curve or fold line that approximates the original line, which consists of only a few key points, thus reducing the number of points in the original line to a certain degree, while preserving as much information as possible about the shape and inflection points of the original line. By applying RDP algorithm to ship's latitude and longitude data, we can significantly reduce the size of the data while retaining the key features of the ship's trajectory, reduce the workload of processing the data, make the data more concise and efficient, and facilitate further analysis and visualization.

The following are the specific execution steps of the RDP algorithm [3].

1) Select the start and end points from the given line segment.
2) Calculate the perpendicular distance from each intermediate point to the line segment using the following formula:
   $$d = \frac{|(\vec{p_1} - \vec{p_0}) \times (\vec{p} - \vec{p_0})|}{|\vec{p_1} - \vec{p_0}|}$$
   where $\vec{p_0}$ and $\vec{p_1}$ are the coordinates of the start and end points, and $\vec{p}$ is the coordinate of the intermediate point. 3. Find the point with the maximum perpendicular distance as the current segmentation point.

3) The RDP algorithm includes an important parameter ε, the distance threshold. This threshold controls the smoothness and accuracy of the resulting line segment. In practice, the optimal threshold can be chosen by experimenting with different values.If the maximum perpendicular distance is less than the given threshold ε , stop the segmentation and output the start and end points. Otherwise, use the current segmentation point as a new start or end point, and go back to step 2 [1].

The steps for running the RDP algorithm are shown in the following diagram.

B. Visualization techniques

FDEB (Force-Directed Edge Bundling) is a technique for visualizing network data that simplifies edge structures by grouping similar edges into bundles while preserving their topology [5]. It models edges as elastic springs that attract or repel each other based on their similarity and proximity, generating smooth and natural-looking curves that enhance the visualization's readability and aesthetics. FDEB has been widely applied in many domains, demonstrating its effectiveness and flexibility in revealing hidden patterns and relationships in complex network data. The FDEB algorithm has excellent visualization of complex network data, which can effectively reduce the complexity of the edge structure and improve the readability and aesthetics of the image. Therefore, we choose to use the FDEB algorithm to visualize the route data, so as to present the route patterns and characteristics of ships more clearly. With the bundling technique of FDEB algorithm, we can bundle similar routes together to form smooth and natural curves, while maintaining the topology of the routes to improve the aesthetics and readability of the visualized images. On this basis, we can optimize the overall layout and further improve the visualization by adjusting details such as the strength and parameters of the bundle to better understand the ship's route patterns and characteristics.

In our work, we use the RDP algorithm to process the raw data of the ship routes, then we construct the XML file by treating each CSV file as a route. In the XML file, points and edges are generated to form the complete route. Then, we visualize the routes using the edge bundling algorithm FDEB.

D3.js is a popular JavaScript library that is widely used for data visualization and web-based graphics [14]. It provides a set of tools for creating interactive and dynamic data visualizations, including support for data binding, scales, and transitions.The FDEB algorithm is implemented using d3.js as the programming language.

Specifically, the FDEB algorithm bundles the approaching edges together by means of a force-directed model. We first define a force model for each point such that the closer the neighboring points are to each other, the stronger the interaction forces are, and the farther the distance, the weaker the interaction forces are. Next, we define elastic and repulsive forces between edges to maintain the distance between edges and to prevent crossover and overlap between edges.

The following are the specific implementation steps of the FDEB algorithm [13] [8]

1) Compute the affinity matrix
Calculate the pairwise similarity between edges using the following affinity measure:
$$w(i,j) = \exp\left(-\frac{d(i,j)^2}{2\sigma^2}\right)$$
where $d(i,j)$ is the Euclidean distance between the edge endpoints, and σ is a parameter controlling the smoothness of the edge bundling.
2) Compute the sparsified affinity matrix [**?**] Apply a sparsification technique, such as thresholding or spectral clustering, to the affinity matrix to reduce its computational complexity and memory requirements.
3) Compute the edge paths Construct a set of geodesic paths connecting each edge to its nearest neighbors, using the Dijkstra's algorithm or any other suitable shortest path algorithm.
4) Compute the edge bundles For each edge, compute a weighted average of its neighboring paths, using the sparsified affinity matrix as the weight matrix. The resulting weighted average represents the position of the bundled edge.
5) Smooth the edge bundles Apply a smoothing function, such as B-spline interpolation or cubic splines, to the edge bundles to create a visually appealing and smooth edge bundling effect.

Finally, we optimize the overall layout by varying the strength of the bundle to obtain a more concise and easily interpretable route map. Our results show that the FDEB algorithm can effectively reduce the crossover and overlap of ship routes, thus greatly enhancing the readability and ease of interpretation of the images.

IV. Results

The image below shows the original data and the bundled image. Figure 2 shows the original data, and Figure 3 shows the image after bundling with FDEB after reducing the number of points by RDP. The comparison shows that the bundled image is clearer and more concise than the original image so that the main routes can be easily identified, and the different shades of the image depending on the degree of bundling can also be used to show the number of the route passes in the comparison.

V. Discussion

In the process of implementing the algorithm, we should pay attention to adjusting the parameters of the algorithm and use reasonable parameters. To ensure that the picture can express the actual meaning to the maximum extent. In DRP algorithm, if the parameters are larger, the important details may be lost and the actual meaning will be lost. Therefore, it is necessary to find a balance between reducing the amount of data and preserving the key points of the ship. In the FDBE process, if the bundle parameter is too high, the image may be distorted, and if the bundle parameter is too low, the simplification effect may not be achieved.

For example, when the parameter epsilon in the DRP algorithm is set too high, such as epsilon=10000000 in the figure below, the algorithm will basically only keep the first two points, and the middle navigation route is missing, thus, the image will be processed as a two-point line problem, which greatly reduces the interpretability of the image, and makes the image missing the meaning of the route to be expressed. Therefore, in the visualization process, we need to experiment several times and use reasonable parameters to achieve a balance between simplicity and interpretability.

In addition, the size of the data is an issue that needs to be taken into account when using the FDEB algorithm. Since the FDEB algorithm needs

In addition, the size of the data is an issue that needs to be taken into account when using the FDEB algorithm. Since the FDEB algorithm needs to calculate the similarity between each pair of edges, too large a data size can lead to a significant increase in the running time of the algorithm and may not even generate visualization results. Therefore, in practical applications, it is necessary to balance the relationship between data volume and visualization results and adjust the data size according to the specific situation.

It should be noted that when adjusting the data volume size, the quality of the visualization effect should be guaranteed. If the data volume is too small, the characteristics and trends of the data cannot be accurately reflected, which may lead to distortion of the visualization results; if the data volume is too large, the clarity and readability of the visualization effect may be affected. Therefore, it is necessary to reasonably adjust the data volume size according to the actual application scenario, combining the characteristics of the algorithm and the features of the data, in order to obtain the best visualization effect. to calculate the similarity between each pair of edges, too large a data size can lead to a significant increase in the running time of the algorithm and may not even generate visualization results. Therefore, in practical applications, it is necessary to balance the relationship between data volume and visualization results and adjust the data size according to the specific situation.

It should be noted that when adjusting the data volume size, the quality of the visualization effect should be guaranteed. If the data volume is too small, the characteristics and trends of the data cannot be accurately reflected, which may lead to distortion of the visualization results; if the data volume is too large, the clarity and readability of the visualization effect may be affected. Therefore, it is necessary to reasonably adjust the data volume size according to the actual application scenario, combining the characteristics of the algorithm and the features of the data, in order to obtain the best visualization effect.

VI. Conclusion

In conclusion, the visualization scheme developed in this study provides a simple and intuitive overview of the Bohai Bay route by studying the ship's trajectory and path

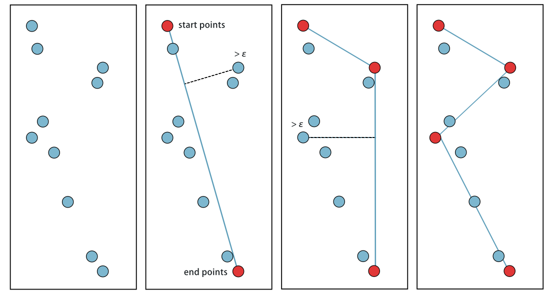

Fig. 1. RDP steps

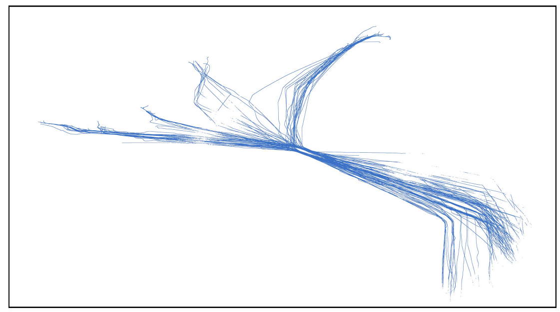

Fig. 2. Raw route map

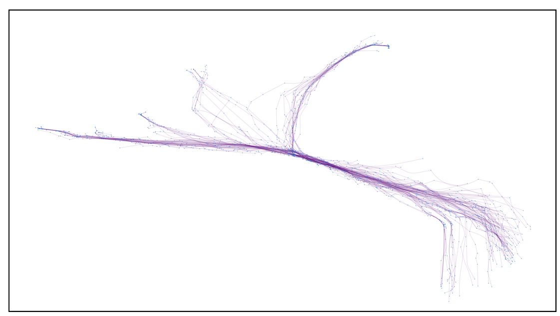

Fig. 3. Processed route map

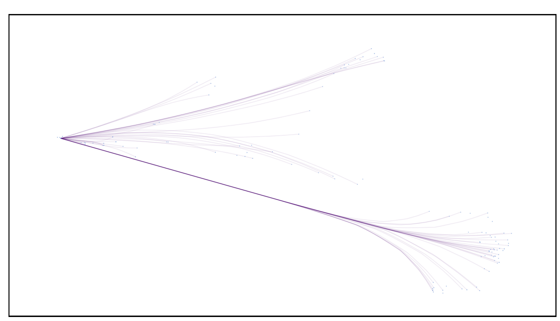

Fig. 4. Distortion effect figure

information. By employing the RDP algorithm and FDEB technique, we were able to significantly reduce the data size while preserving the key features of the ship's trajectory and enhancing the readability and aesthetics of the visualization.

The results of this study can provide more comprehensive and accurate information support for maritime management agencies, port management departments, and ship operators, enabling safe and efficient passage of waterways in the Bohai Bay region.

However, it is worth noting that this study has limitations. For instance, the accuracy of the trajectory data may be affected by various factors, such as weather conditions and technical issues. Additionally, while the FDEB technique can effectively simplify the edge structure and preserve topology, it may also result in some degree of distortion in the representation of the ship's trajectory.

Future work can focus on improving the accuracy and robustness of the trajectory data, exploring alternative visualization techniques, and integrating other types of data, such as weather and traffic information, to provide more comprehensive insights into the waterways in the Bohai Bay region. Overall, this study lays a foundation for further research in the field of waterway visualization, with potential applications in various domains, including transportation management, environmental monitoring, and urban planning.

REFERENCES

[1] Pau Cebrian and Juan Carlos Moure. Gpu-accelerated rdp algorithm for data segmentation. In *Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part I 20*, pages 234–247. Springer, 2020.

[2] David H Douglas and Thomas K Peucker. 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, 1973.

[3] Furkan Goz, Alev Mutlu, and Orhan Akbulut. Analysis of ramer-douglas-peucker algorithm as a discretization method. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2018.

[4] Danny Holten and Jarke J van Wijk. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.

[5] Danny Holten and Jarke J Van Wijk. Force-directed edge bundling for graph visualization. In *Computer graphics forum*, volume 28, pages 983–990. Wiley Online Library, 2009.

[6] Xiaolong Hu, Cong Gao, Jinyan Hu, and Qi Wu. Skeleton based edge bundling for visualization of large maritime data. *IEEE Transactions on Visualization and Computer Graphics*, 24(9):2463–2475, 2018.

[7] Yong Huang, Mingquan Zhou, Yanhui Xu, and Xiaofang Xie. 3d visualization of maritime navigation based on opengl. *International Journal of Control and Automation*, 7(2):349–358, 2014.

[8] Christophe Hurter, Ozan Ersoy, and Alexandru Telea. Graph bundling by kernel density estimation. 31(3pt1):865–874, 2012.

[9] Ioannis Karydis, Georgios Goulios, and John Vourdoubas. An integrated system for sea route planning and navigation based on electronic charts. *Journal of Navigation*, 69(6):1236–1252, 2016.

[10] Jeong-Hwan Kim, Sungwon Choi, Jaeuk Jung, and Ilkwon Yun. Clustering-based maritime route visualization using ais data. *Journal of Navigation*, 71(6):1416–1430, 2018.

[11] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, 1(3):244–256, 1972.

[12] Hui Wang and Yang Liu. A force-directed layout algorithm for shipping route maps. *Journal of Navigation*, 69(2):333–347, 2016.

[13] Hong Zhou, Panpan Xu, Xiaoru Yuan, and Huamin Qu. Edge bundling in information visualization. *Tsinghua Science and Technology*, 18(2):145–156, 2013.

[14] Nick Qi Zhu. *Data visualization with D3. js cookbook*. Packt Publishing Ltd, 2013.