



Article AIS Trajectories Simplification Algorithm Considering Topographic Information

Wonhee Lee D and Sung-Won Cho *D

Maritime Safety and Environmental Research Division, Korea Research Institute of Ships and Ocean Engineering, 32, Yuseong-daero 1312 beon-gil, Yuseong-gu, Daejeon 34103, Korea

* Correspondence: gungnir@kriso.re.kr; Tel.: +82-10-8593-1552

Abstract: With the development of maritime technology and equipment, most ships are equipped with an automatic identification system (AIS) to store navigation information. Over time, the size of the data increases, rendering its storage and processing difficult. Hence, it is necessary to transform the AIS data into trajectories, and then simplify the AIS trajectories to remove unnecessary information that is not related to route shape. Moreover, topographic information must be considered because otherwise, the simplified trajectory can intersect obstacles. In this study, we propose an AIS trajectory simplification algorithm considering topographic information. The proposed algorithm simplifies the trajectories without the intersection of the trajectory and obstacle using the improved Douglas–Peucker algorithm. Polygon map random (PMR) quadtree was used to consider topographic information on the coast, and the intersection between topographic information and simplified trajectories was efficiently computed using the PMR quadtree. To verify the effectiveness of the proposed algorithm, experiments were conducted on real-world trajectories in the Korean sea. The proposed algorithm yielded simplified trajectories with no intersections of the trajectory and obstacle. In addition, the computational efficiency of the proposed algorithm with the PMR quadtree was superior to that without the PMR quadtree.

Keywords: AIS information; PMR quadtree; Douglas–Peucker algorithm; trajectories simplification; topographic information

1. Introduction

The International Maritime Organization (IMO) adopted the introduction of automatic identification systems (AISs) to enhance the navigation safety of ships and protection of the marine environment [1]. An AIS is a device that automatically transmits and receives data between ships and AIS base stations, including static information (such as Maritime Mobile Service Identity (MMSI) number, call sign, type, and length), dynamic information (such as position, time, speed over ground, and navigational status), voyage-related information, and short safety messaging [2]. AIS data can be used to generate traffic networks [3,4], predict ship behavior [5–8] and trajectory [9,10], support maritime search and rescue systems [11,12], detect fishing activity [13], create networks [14] and routes [15] for navigation, and detect abnormal behavior [16–19].

In previous studies, knowledge was extracted from AIS data, and researchers commonly emphasized the importance of data preprocessing. There are challenges in extracting knowledge from AIS data because it has limitations such as sensor error, volume of data, incompleteness, and noise [2,20–23]. Among these preprocessing techniques, reducing the number of AIS-data points to reduce storage and computation costs is significant. AIS data are stored for periods of 5–100 s on the coast, and more than 100 million data points are generated every day within areas with high ship traffic. Eliminating unnecessary data points without losing information on the data has been studied in two directions: trajectory simplification and event identification [18]. Trajectory simplification eliminates redundant



Citation: Lee, W.; Cho, S.-W. AIS Trajectories Simplification Algorithm Considering Topographic Information. *Sensors* **2022**, *22*, 7036. https://doi.org/10.3390/s22187036

Academic Editors: Peng Peng, Feng Lu, Yu Yang and Tianzhen Wang

Received: 7 August 2022 Accepted: 15 September 2022 Published: 17 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). data points for representing the shapes while maintaining the overall shape of the trajectory [24]. Event identification involves analyzing AIS data for identifying the ship's arrival and departure from ports, stop, and turn, and extracting the desired trajectory [25–30]. In this study, we focused on trajectory simplification.

AIS trajectories simplification has been studied using line simplification methodology [31]. Shi et al. [32], Ji et al. [33], and Qi and Ji [34] compared various algorithms for simplifying AIS trajectories, including the choosing interval points, limiting vertical distance, limiting angle, offset angle, grating, Douglas–Peucker (DP) [35], Opheim [36], and Visvalingam–Whyatt algorithms [37]. Among these, the DP algorithm has been demonstrated to have fewer errors and better performance compared to the others.

Most studies on AIS trajectories simplification are based on the DP algorithm [38]. In the DP algorithm, the threshold is the sole parameter that determines the shape of the simplified trajectory. Therefore, methods for determining the threshold have generally been studied. Etienne et al. [39] proposed a spatio-temporal DP algorithm to simplify the trajectories and reduce the computation time. Muckell et al. [40] proposed an extended spatial quality simplification heuristic (SQUISH-E) algorithm. The SQUISH-E algorithm simplifies the trajectory by considering the errors caused by AIS data removal. Zhang et al. [31] determined the threshold of the DP algorithm using the size of the minimum ship domain as the evaluation criterion. Li et al. [41] determined the threshold of the DP algorithm through numerous experiments to ensure a good balance between AIS trajectory simplification and visualization performance. Singh et al. [42] developed the scan–pick–move (SPM) algorithm based on the DP algorithm. The SPM algorithm has lower computational complexity compared to the DP algorithm. Zhang et al. [43] conducted a study to set the threshold value on the basis of ship specifications. Zhao and Shi [44] extracted the data points at which the angle changes in the trajectory using the course over ground (COG) information from the AIS data. Huang et al. [45] proposed a GPU-based parallelization framework for shortening the computation time of the DP and the kernel density estimation algorithms for trajectory compression and visualization. Wei et al. [46] simplified the trajectories by combining the sliding window technique that simplifies the speed and course data with the DP algorithm. They used a threshold of 0.8 times the length of the ship, as proposed by Zhang et al. [31].

Many studies have set a static value as a threshold, which causes the poor compression quality of the trajectory. Recently, an adaptive threshold method responding to the condition of each trajectory has been studied to solve this problem. Liu et al. [47] proposed an algorithm that automatically generates the threshold value of the DP algorithm according to the average distance of the trajectory. Tang et al. [48] proposed an algorithm that identifies the turning points that rotate more than a certain angle and used them to change the threshold. Ji et al. [49] proposed an adaptive grating algorithm that can dynamically generate an appropriate threshold for each trajectory.

Although many studies have proposed AIS trajectories simplification methods based on the DP algorithm, no studies have considered topographic information. The reasons for not considering the information are twofold: First, there was no need to consider topographic information because it simplified the trajectory at sea without islands or obstacles. Second, the threshold of the DP algorithm was set to be small, and the number of simplifications was minimized. Since the AIS data represent the locations that the ship passed, a simplified trajectory to avoid obstacles can be generated if the threshold is set small even without considering the topographic information. However, depending on the user-defined threshold of the trajectory simplification algorithm, the intersection of the trajectory and obstacle occurs, or many unnecessary data points remain because the trajectory is not sufficiently simplified. Therefore, we propose the obstacle Douglas–Peucker (ODP) algorithm that considers topographic information.

In the ODP algorithm, obstacle detection is combined with the DP algorithm. To detect obstacles, the intersection of line segments and polygon (ILP) algorithm is used to determine the intersection of a trajectory and obstacle because topographic information

and trajectories are represented by polygons and lines, respectively. The DP algorithm has a recursive characteristic to repeatedly perform simplification. Thus, in areas such as coastal areas with many obstacles, the ILP algorithm further increases the computational burden. To address this challenge, we utilize a grid-based technique. Typically, grid-based methods, such as the K-dimensional tree, binary tree, quadtree, octal tree (octree), and R-tree, are used to represent complex obstacle fields [50,51]. The size of the grid must be significantly small to represent complex marine environments with the binary or K-dimensional tree [52]. However, these methods generate inefficient data storage, representing all areas with redundant information in a large terrain. In particular, this is critical because the sea, which contains large land masses over several kilometers, is the scope of the study. The quadtree is widely used in the modeling of complex and large-scale environments such as land [53,54], sea [55–57], earthquakes ground motion [58], flood [59], and tsunami [60]. Therefore, we utilized a quadtree as the spatial data structure to efficiently store topographic information and represent the free space.

The contributions of this study are summarized as follows: (1) A novel algorithm that simplifies the AIS trajectories considering topographic information is proposed. (2) To perform the ODP algorithm in a time frame, the ILP algorithm with the polygon map random (PMR) quadtree is proposed. (3) The practical restrictions for navigation on the coast are considered, and the effectiveness is demonstrated through testing in real-world environments.

The remainder of this paper is organized as follows. In Section 2, an AIS trajectory simplification algorithm considering topographic information is proposed. In Section 3, the results are confirmed and analyzed by applying the proposed algorithm to AIS trajectories. Finally, the conclusions and future plans are discussed in Section 4.

2. Methods

The ODP algorithm comprises three phases: (1) data preprocessing that removes outliers among the AIS data points and generates trajectories; (2) detecting the intersection of line segments and the polygon using the PMR quadtree; (3) simplifying the AIS trajectories using the DP algorithm considering topographic information.

2.1. AIS Data Preprocessing

Raw AIS data are difficult to use for analysis because the information from several ships is mixed. Therefore, data classification is essential. AIS data contain static information, such as MMSI number, IMO number, ship name, ship type, length, width, draft, and dynamic information, such as latitude, longitude, speed, COG, speed over ground (SOG), ship heading, and date (format: yyyy-mm-dd hh:mm:ss). Using the MMSI number, the ship data were classified using the MMSI. Then, each ship data was divided based on the berthing or anchoring, which means the ship was stopped for a certain period; this is called a voyage.

For each voyage data, preprocessing was performed. First, AIS data were chronologically sorted to detect outliers, which means abnormal location information caused by a system malfunction or data transmission error. Subsequently, data with SOG of three knots or less were removed owing to the characteristics of the ship. When the ship is berthing or anchoring, the SOG is nearly zero. Because the voyage with removed data did not equal time intervals, it needed to be set with the same time interval. To have the same time interval for the routes, resampling was performed. The resampling criterion was the AIS data time interval. In this study, the time interval was set to 1 s, which is the minimum time difference between two consecutive data points.

2.2. ILP Algorithm Using Quadtree

We propose an ILP algorithm that efficiently reduces the computational burden by using the PMR quadtree. A simple method to check the intersection of line segments and polygons is to determine whether they are crossed for all line segments. Because this method is time-consuming and inefficient, previous studies have proposed methods to improve it; however, the proposed methods were difficult to apply for spatial data of significantly large sizes [52–61]. Therefore, in this study, we propose an efficient ILP algorithm using the PMR quadtree.

The tree is a data structure in which each internal node has a group of several subnodes. Since the number of sub-nodes is set to the powers of 2 (2, 4, 8...), they are referred to as a binary tree, quadtree, octal tree (octree), and so on. Among these trees, the quadtree is widely used for topographic visualization and spatial processing. There are various types of quadtrees depending on the input data format and splitting method. Generally, there is a point region (PR) quadtree, used when the input data are points, and a polygon map (PM) quadtree, used when the input data are lines. Since topographic information is polygonal information composed of lines, a PM quadtree is suitable. However, the PM quadtree has the limitation of being time-consuming for the quadtree creation because points are included in the split conditions. To overcome this limitation, a PMR quadtree that sets the splitting condition as a line is proposed. Because the PMR quadtree subdivides the tree by inserting the input data one by one, the tree structure is rapidly created. Using PMR quadtree, the execution time per input data is similar as the size of the input data increases. Moreover, the CPU cost is smaller than that of other tree methods. In the previous study, it was confirmed that the PMR quadtree yielded results similar to the those of other tree methods, while producing the results at least five times faster compared to other tree methods [62].

The PMR quadtree checks the condition of the node related to line segments constituting the topographic information and divides the existing node into four sub-nodes to generate a grid map. The hyperparameters of the PMR quadtree are the splitting threshold and maximum depth. The splitting threshold is the maximum number of line segments required to divide the current node into sub-nodes; the maximum depth is a parameter for limiting the number of divisions of the initial node. When a line segment is input, if the related node exceeds the splitting threshold and the depth of the node does not exceed the maximum depth, the node is split into four sub-nodes. A node that is no longer divided is called a leaf node. Figure 1 depicts an example of creating a PMR quadtree, where the splitting threshold was set to 1 and the maximum depth was set to 4. The line segments constituting the polygon were input in the order of A-H. The initial node exceeds the splitting threshold of 1 when line segment B is input. As shown in Figure 1a, the initial node is split into northeast (NE), northwest (NW), southeast (SE), and southwest (SW) nodes. When line segment C is input, the number of line segments in the SW node exceeds the splitting threshold. Therefore, it is divided as shown in Figure 1b. Following this method, the line segments D–G generate the quadtree, as shown in Figure 1c, and the line segment H divides the nodes in Figure 1c as shown in the gray areas in Figure 1d.

The topographic information of the Korean sea consists of the coastline and more than 2500 islands. Figure 2 shows the Korean sea expressed using the PMR quadtree. The longitudinal and latitudinal boundaries of the quadtree were set as $125.1^{\circ}-129.8^{\circ}$ E and $33^{\circ}-37.7^{\circ}$ N, respectively. The splitting threshold was set to 2, and the maximum depth to 12. The area marked in Figure 2 is enlarged as shown in Figure 3.



Figure 1. Example of polygon map random (PMR) quadtree generation: (**a**) Quadtree generated by line segment B; (**b**) Quadtree generated by line segment C; (**c**) Quadtree generated by line segment D–G; (**d**) Quadtree generated by line segment H.



Figure 2. PMR quadtree visualization in Korean sea.



Figure 3. Detailed view of Zone 1 in Figure 2.

The line segments of topographic information overlapping boundaries are stored in the leaf node of the PMR quadtree. By using this characteristic, the intersection of a trajectory and an obstacle can be determined.

The overall procedure of the ILP algorithm using quadtree is described by Algorithm 1.

Algorithm 1 Edge acquisition in quads intersecting with line
Input: root, p, q
Output: edgeSet
1: function LineIntersectPoly(<i>p</i> , <i>q</i>)
2: GetEdges(<i>node</i> , <i>p</i> , <i>q</i> , <i>edgeSet</i>)
3: for $i = 1$ to N //N is the number of edges in <i>edgeSet</i>
4: If $edgeSet[i] \cap (p,q)$ then return true
5: end for
6: return false
7: end function
8: function GetEdges(<i>node</i> , <i>p</i> , <i>q</i> , <i>edgeSet</i>)
9: if <i>node</i> .child $\neq \emptyset$ and <i>node</i> .edges $\neq \emptyset$ then
10: <i>edgeSet</i> .append(<i>node</i> .edges)
11: else
12: $child = \{NE, NW, SE, SW\}$
13: for $i = 1:4$ do
14: if <i>node.child</i> [<i>i</i>].bound \cap (<i>p</i> , <i>q</i>) then
15: GetEdges(<i>node.child</i> [<i>i</i>], <i>p</i> , <i>q</i> , <i>edgeSet</i>)
16: end if
17: end for
18: end if
19: end function

An example of Algorithm 1 is shown in Figure 4. Figure 4 contains topographic information within the longitudinal and latitudinal boundaries of $126.165^{\circ}-126.195^{\circ}$ N and $34.235^{\circ}-34.265^{\circ}$ E, respectively. The total number of line segments was 600. Line segments A (126.1793° N and 34.2353° E)–B (126.1754° N and 34.2647° E) were provided. The boundaries of the quadtree (intersected quads) intersecting with the line segment A–B are shown as the region shaded in pink in Figure 4. The topographic edges contained in the quadtree intersecting with line segments A–B are shown in red. The ILP algorithm using quadtree reduced the number of line segments for checking intersections from 600 to 13. The proposed algorithm drastically improved efficiency. Subsequently, the proposed algorithm was used for the trajectory simplification algorithm considering the topographic information.

2.3. Trajectory Simplification Algorithm Considering Topographic Information

The DP algorithm is a method for approximating an existing shape to a line, which requires a threshold to simplify the line [35]. The existing DP algorithm simplifies the trajectory without considering the topographic information. However, if there are complicated coastlines or islands, topographic information must be considered. When the threshold is small, there is no issue with simplification. However, as the threshold increases, the trajectory is simplified as it intersects the obstacle. Therefore, we propose the DP algorithm that considers the topographic information, as shown in Algorithm 2.



Figure 4. Example of edges in quads intersecting with line.

Algorithm	2 ODP	algorithm
-----------	-------	-----------

Input: <i>trj</i> (trajectory), <i>eps</i> (threshold)	
Output: <i>s_trj</i> (simplified trajectory)	
1: function Obstacle_DP(<i>trj</i> , <i>eps</i> , <i>s</i> _ <i>trj</i>)	
2: $n = trj.size()$	
3: <i>index</i> , d_{max} = PerpendicularGeoDistance (<i>trj</i> , <i>trj</i> [1], <i>trj</i> [n])	
4: if $d_{max} > eps$ or LineIntersectPoly $(trj[1], trj[n])$	
5: result1 = Obstacle_DP $(trj[1], trj[1] + index)$	
6: result2 = Obstacle_DP $(trj[1] + index, trj[n])$	
7: $s_trj = [result1 result2]$	
8: else	
9: $s_{trj} = [trj[1] trj[n]]$	
10: end if	
11: end function	

The ODP algorithm proceeds as shown in Figure 5. As shown in Figure 5a, the distance between the generated line segment, connecting p_0 and p_{14} (the blue dashed line), and the point p_7 , which had the largest vertical distance (d_{max}) was compared with the threshold (ϵ). Because it is greater than ϵ , p_7 was fixed and two subtrajectories (p_0-p_7 and p_7-p_{14}) were generated as shown in Figure 5b. Figure 5b illustrates that, in the trajectory from p_0 to p_7 , the maximum vertical distance was greater than ϵ ; therefore, the trajectory was divided into two subtrajectories (p_0-p_3 and p_3-p_7). In the trajectory from p_7 to p_{14} , the maximum vertical distance was smaller than ϵ . However, the trajectory was divided into two subtrajectories (p_7-p_9 and p_9-p_{14}) owing to the obstacle. Furthermore, in Figure 5c, it can be observed that the trajectory from p_0 to p_3 intersects with the obstacle; therefore, it was divided into two subtrajectories (p_0-p_2 and p_2-p_3). In other trajectories, the maximum vertical distance was smaller than ϵ , and no intersection with the obstacle could occur; therefore, the trajectories are simplified. Finally, as shown in Figure 5d, the trajectory from



 p_0 to p_{14} was simplified to the trajectory from q_0 to q_5 . A simplified trajectory does not intersect with the obstacle.

Figure 5. Route simplification using the ODP algorithm: (a) Check the original trajectory p_0-p_{14} ; (b) Check the subtrajectories p_0-p_7 and p_7-p_{14} ; (c) Check the subtrajectories p_0-p_3 , p_3-p_7 , p_7-p_9 , and p_9-p_{14} ; (d) Simplified trajectory q_0-q_5 simplified from the original trajectory (grey).

In the ODP algorithm (4th line in Algorithm 2), the perpendicular distance of a point from a straight line on Earth is computed. As shown in Figure 6, a triangle comprising points A, P, and H on the Earth is called a spherical triangle. In Equations (1) and (2), *a* and *h* are distances (km); A and H are angles (rad); l_{PH} and l_{AP} are the distances (km) between points P and H, and points A and P, respectively; R is Earth's radius (km); θ_{AH} and θ_{AP} are the azimuth angles (rad) between points P and H, and points A and P, respectively. The sine rule of a spherical triangle is formulated as Equation (1) [63]. Because the angle H is 90°, sin H = 1, and it can be written as Equation (2) considering the unit of distance. The angle A is computed using the difference between the azimuth angles θ_{AH} and θ_{AP} .

$$\sin a \sin H = \sin h \sin A \tag{1}$$

$$l_{PH} = R \cdot \arcsin\left(\sin\frac{l_{AP}}{R}\right)\sin(\theta_{AH} - \theta_{AP})$$
(2)



Figure 6. Spherical triangle for perpendicular distance.

In Equations (3) and (4), θ is the azimuth (rad), l is the distance (km), ϕ is the longitude, λ is the latitude, and $\Delta\lambda$ is the latitudinal difference between the two points. Equation (3) is used for calculating the azimuth, and Equation (4) is the haversine distance equation for calculating the distance between two points on Earth.

$$\theta = \arctan 2(\sin \Delta \lambda \cos \phi_2, \cos \phi_1 \sin \phi_2 - \sin \phi_1 \cos \phi_2) \Delta \lambda$$
(3)

$$l = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos\phi_1 \cos\phi_2 \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$
(4)

2.4. Evaluation Metric

A trajectory can be evaluated using length loss, compression rate, and violations. The length loss can be computed using the length of the original trajectory and the simplified trajectory according to Equations (5) and (6).

$$L = \sum_{n=1}^{N-1} d_n,$$
 (5)

$$LengthLoss = \left(1 - \frac{L_S}{L_o}\right) \cdot 100,\tag{6}$$

where d_n denotes the haversine distance between two adjacent points on one trajectory. L is the total length of the trajectory. L_O is the total length of the original trajectory, and L_S is the total length of the simplified trajectory. The length loss can be computed using the ratio of L_O and L_S .

The compression rate denotes how many points on the original trajectory were removed and can be expressed as Equation (7).

$$CompressionRate = \left(1 - \frac{N_S}{N_o}\right) \cdot 100,\tag{7}$$

where N_O denotes the number of points on the original trajectory, and N_S denotes the number of points on the simplified trajectory.

The violations denote whether the simplified trajectory intersects the boundary of the obstacle. The violations can be expressed as Equation (8).

$$violations = \sum_{n=1}^{N-1} LIP(p_i, p_{i+1}),$$
(8)

where *LIP* denotes Algorithm 1 *LineIntersectPoly* and p_i is a point on the trajectory. When the line of the trajectory intersects the boundary of the obstacle, *LIP* is true and 1. Otherwise, *LIP* is false and 0. After *LIP* is performed on all line segments on the trajectory, violations are the summation of all the *LIP* results.

3. Results

To verify the effectiveness of the ODP algorithm, experiments were conducted on trajectories 1 and 2 in the Korean sea, as shown in Figure 7. The AIS trajectory set consists of 10 trajectories with diverse departure and arrival. A sensitivity analysis on the threshold (ϵ) of the ODP algorithm was conducted. In addition, we compared the performance of the ODP algorithm with that of the existing DP algorithm. Finally, the computation time was compared depending on whether the PMR quadtree was applied or not.



Figure 7. AIS trajectories in Korean sea for verifying the performance of the proposed algorithm.

3.1. Comparison of Quadtree and Uniform Method

In the quadtree method, many nodes are generated in areas with many obstacles; otherwise, nodes are generated sparsely. Generally, the quadtree method can be compared with the uniform method. Table 1 shows the simplification using the quadtree and uniform methods for Trajectory 1. The number of edge comparisons means the number of intersections between the line segment of the obstacle and the trajectory when ODP is progressed. The resolution of the uniform method should be set to 1024, based on the minimum node size of the quadtree. The number of nodes generated by the uniform method is about 19 times more than when generated by the quadtree method. If the resolution is set to 256, a similar number of nodes can be generated. However, the number of edge comparisons increases greatly when performing ODP. Therefore, the quadtree method intensively creates nodes in areas with obstacles, which can reduce the number of edge comparisons while reducing the total number of nodes obtained from the uniform method.

3.2. Comparison of AIS Trajectory Simplification Results between the ODP and DP Algorithm

The ODP and DP algorithms were applied to all trajectories for comparison. Among them, trajectories 4 and 10 were selected for sensitivity analysis because the environment of these trajectories has many obstacles. Table 2 shows the results of the sensitivity analysis of the threshold for trajectory 4. In the DP algorithm, the threshold is the maximum perpendicular distance from the simplified line segment to the original trajectory. Since the maximum perpendicular distance is similar to the radius of the ship domain, the size of the ship domain can be used to determine the threshold [31]. The ship domain can be set with various sizes and shapes depending on the navigation situation. In recent studies, the ship domain was up to 1 km with a complex shape [64]. Therefore, sensitivity analysis was performed by setting the threshold up to 1 km.

Table 1. Comparison of the results obtained using the quadtree and uniform methods (Trajectory 1).

Method	The Number of Nodes	Minimum Node Size (km)	The Number of Edge Comparisons
Quadtree	55,534	0.407	95
Uniform (256 × 256)	65,536	1.629	2534
Uniform (512 × 512)	262,144	0.814	355
Uniform (1024 × 1024)	1,048,576	0.407	95

Table 2. Comparison of the results obtained using the ODP and DP algorithms for trajectory 4.

		ODP Algorithm			DP Algorithm	
ϵ (km)	Length Loss (%)	Compression Rate (%)	Number of Violations	Length Loss (%)	Compression Rate (%)	Number of Violations
None	100.0	100.0	-	100.0	100.0	-
0.1	99.62	97.85	0	99.62	97.85	0
0.3	99.49	98.88	0	99.49	98.88	0
0.5	99.43	99.17	0	99.18	99.21	1
0.7	99.26	99.38	0	99.01	99.42	1
1.0	99.26	99.38	0	98.92	99.46	2

We set thresholds for ODP and DP to 0.1, 0.3, 0.5, 0.7, and 1.0 km. The length of trajectory 4 is 242.49 km, and the number of points on trajectory 4 is 2420. In Table 2, the number of violations represents the number of times the simplified trajectory intersects the obstacle. Because the ODP algorithm is a modified DP algorithm, their simplification results were similar. As the threshold increased, several data points in trajectory 4 were removed, and consequently, the distance of the trajectory reduced. The simplified trajectories generated using the ODP algorithm are shown in Figure 8. In all the results, the intersection of the trajectory and obstacle was not observed.

Table 3 shows the results of the sensitivity analysis of the threshold for trajectory 10. The length of trajectory 10 is 513.06 km, and the number of points on trajectory 10 is 5122. Similar to the results for trajectory 4, it was confirmed that, as the threshold increased, several data points in trajectory 4 were removed, and thus, the distance of the trajectory also reduced. The simplified trajectories obtained using the ODP algorithm are illustrated in Figure 9, which shows no intersection of the trajectory and obstacle.



Figure 8. AIS trajectory simplification results of the ODP algorithm for trajectory 4: (a) $\epsilon = 0.1$; (b) $\epsilon = 0.3$; (c) $\epsilon = 0.5$; (d) $\epsilon = 1.0$.

	Table 3.	Comparison	of the results	obtained	using the	ODP and 1	DP algorithms	for trajectory	y 10
--	----------	------------	----------------	----------	-----------	-----------	---------------	----------------	------

		ODP Algorithm			DP Algorithm	
ϵ (km)	Length Loss (%)	Compression Rate (%)	Number of Violations	Length Loss (%)	Compression Rate (%)	Number of Violations
None	100.0	100.0	-	100.0	100.0	-
0.1	99.61	98.03	0	99.61	98.03	0
0.3	99.49	98.97	0	99.49	98.98	0
0.5	99.44	99.14	0	99.44	99.16	1
0.7	99.33	99.32	0	99.3	99.36	1
1.0	99.26	99.39	0	96.85	99.51	2



Figure 9. AIS trajectory simplification results of the ODP algorithm for trajectory 10: (a) $\epsilon = 0.1$; (b) $\epsilon = 0.3$; (c) $\epsilon = 0.5$; (d) $\epsilon = 1.0$.

3.3. Comparison with Other Methods

To verify the usefulness of the ODP algorithm, it was compared with other simplification algorithms, including the DP algorithm and the algorithm considering ship behavior [46]. Table 4 shows the results of the ODP and other algorithms. In areas with many obstacles, the DP algorithm and Wei et al. [46]'s algorithm show that violations occur frequently. The ODP algorithm shows no violations in all cases. Since the ODP algorithm is a DP-based algorithm, it was confirmed that the ODP results were similar to the DP results. The compression rate of ODP was slightly higher than that of the DP algorithm because the trajectory with obstacles was not simplified. For the same reason, the length loss of ODP was slightly lower than that of the DP. In terms of length loss and compression rate, ODP was generally better than the algorithm of Wei et al. [46].

Figure 10 shows the simplified trajectories generated using the ODP and DP algorithms for trajectory 4 with $\epsilon = 1.0$ km. Because the DP algorithm and the method of Wei et al. [46] remove the data points without considering the topographic information, the simplified

trajectory intersects the obstacles. As can be observed from Table 4, the larger the threshold, the more violations the obstacle and trajectory cross. On the other hand, as shown in Figure 10b–d, the ODP algorithm performed trajectory simplification without the trajectory intersecting the obstacle. Since the ODP algorithm has the property of DP, it is better than the method of Wei et al. [46] in the perspective of length loss and compression rate.



Figure 10. Results obtained from the ODP and DP algorithms for trajectory 4 ($\epsilon = 1.0$): (a) Overview; (b) Zone 1 in (a); (c) Zone 2 in (a); (d) Zone 3 in (a) [46].

Figure 11 shows the intersections of the simplified trajectory 10 and obstacles. As can be observed from Table 4, the larger the threshold, the more violations the obstacle and trajectory cross. Figure 11a–d shows the enlarged visualizations of the areas containing intersections of the trajectory and obstacle. From these results, it can be concluded that the DP algorithm and the method of Wei et al. [46] yield simplified trajectories intersecting the obstacles, and the ODP algorithm yields simplified trajectories that do not intersect with the obstacle. Similar to the results for trajectory 4, the ODP algorithm is better than the method of Wei et al. [46] from the perspective of length loss and compression rate. From these results, it can be concluded that the ODP algorithm is more practical and suitable for navigation safety.

		DP			Wei et al. [46]			ODP	
Trajectory	Length Loss (%)	Compression Rate (%)	Number of Violations	Length Loss (%)	Compression Rate (%)	Number of Violations	Length Loss (%)	Compression Rate (%)	Number of Violations
1	0.74	99.48	1	1.04	98.77	3	0.72	99.47	0
2	0.71	99.62	1	0.67	98.73	1	0.46	99.60	0
3	0.98	99.61	1	0.45	98.83	1	0.56	99.59	0
4	1.08	99.46	2	1.13	98.35	2	0.74	99.38	0
5	0.40	99.56	1	0.48	98.79	1	0.37	99.55	0
6	0.58	99.51	1	0.60	97.65	1	0.55	99.48	0
7	0.46	99.50	1	0.55	98.43	2	0.45	99.48	0
8	0.38	99.57	2	0.45	99.12	1	0.36	99.55	0
9	0.64	99.51	1	0.79	99.01	1	0.61	99.48	0
10	3.15	99.51	5	0.58	98.42	2	0.74	99.39	0

Table 4. Results of ODP and other algorithms.



Figure 11. Results obtained from the ODP and DP algorithms for trajectory 2 ($\epsilon = 1.0$): (a) Overview; (b) Zone 1 in (a); (c) Zone 2 in (a); (d) Zone 3 in (a) [46].

3.4. Computation Efficiency of the ODP Algorithm with PMR Quadtree

Tables 5–7 show the computation times of the ODP algorithm with and without the PMR quadtree for trajectories 1, 4, and 10. A typical ILP algorithm was used in the ODP algorithm without the PMR quadtree. The computation time for each threshold presented in Tables 5–7 is the average computation time for 10 repetitions. For $\epsilon = 1.0$ of Tables 5–7, it is shown that the ODP algorithm with the PMR quadtree reduced the computation time by about 89% compared to that without; the ODP algorithm with the PMR quadtree is superior to that without. Furthermore, it can be seen that the ODP algorithm with the PMR quadtree yielded a more robust performance in terms of the computation time than that without. Therefore, we can conclude that the proposed method is worth introducing in practice because it has the most reliable performance.

Table 5. Comparison of the computational efficiencies of the ODP algorithms with and without the PMR quadtree (Trajectory 1).

ϵ (km)	ODP with PMR Quadtree	ODP without PMR Quadtree	Ratio (%) (with/without)
0.1	6.4	169.3	3.78
0.2	6.2	115.1	5.39
0.3	5.8	87.1	6.66
0.4	5.6	75.1	7.46
0.5	5.4	66.6	8.11
0.6	5.2	59.5	8.74
0.7	5.2	55.2	9.42
0.8	5.0	50.4	9.92
0.9	5.0	45.8	10.92
1.0	4.9	43.8	11.19

Table 6. Comparison of the computational efficiencies of the ODP algorithms with and without the PMR quadtree (Trajectory 4).

ϵ (km)	ODP with PMR Quadtree	ODP without PMR Quadtree	Ratio (%) (with/without)
0.1	6.9	180.1	3.83
0.2	6.4	127.7	5.01
0.3	6.0	99.4	6.04
0.4	5.8	78.6	7.38
0.5	6.0	68.3	8.78
0.6	5.4	62	8.71
0.7	5.6	55.7	10.05
0.8	5.2	51.6	10.08
0.9	5.0	47.4	10.55
1.0	5.1	46.1	11.06

ϵ (km)	ODP with PMR Quadtree	ODP without PMR Quadtree	Ratio (%) (with/without)
0.1	6.7	174.2	3.85
0.2	6.2	118	5.25
0.3	5.8	89.8	6.46
0.4	5.6	76.1	7.36
0.5	5.4	66.3	8.14
0.6	5.2	59.7	8.71
0.7	5.2	55.8	9.32
0.8	5.1	51.3	9.94
0.9	4.9	45.4	10.79
1.0	4.9	43.4	11.29

Table 7. Comparison of the computational efficiencies of the ODP algorithms with and without the PMR quadtree (Trajectory 10).

4. Conclusions

The AIS trajectory stores ship location information based on a time series. The longer the storage time, the larger the size of the data. This renders it difficult to store and process the data. Therefore, AIS trajectories simplification plays a pivotal role in the study of maritime fields such as weather routing and safety navigation.

In this study, the ODP algorithm based on the DP algorithm was proposed. The proposed algorithm simplifies AIS trajectories through the data preprocessing that removes outliers and creates trajectories, checking the intersections of line segments and polygons using the PMR quadtree, and the DP algorithm, considering topographic information. We conducted experiments on real-world trajectories in the Korean sea to verify the performance of the proposed algorithm. The experimental results indicated that the ODP algorithm yielded simplified trajectories with no intersections of the trajectory and obstacle, unlike the DP algorithm. In addition, the ODP algorithm with the PMR quadtree was up to 30 times faster than that without the quadtree. This result suggests that the performance of the proposed algorithm is more effective in the Korean sea with complex topography.

Future studies can focus on extending our method in consideration of various information on ships. The proposed method considered only the ship location based on AIS trajectories. It is difficult to generate a simplified trajectory with a high compression ratio and low distortion when the trajectory contains the rotation due to obstacles. To solve this problem, it would be interesting to extend our method by considering the speed and direction of the ship and extracting more meaningful AIS data points.

Author Contributions: Conceptualization, W.L.; methodology, W.L.; software, W.L.; validation, W.L.; formal analysis, W.L. and S.-W.C.; investigation, W.L.; resources, W.L.; data curation, W.L.; writing—original draft preparation, W.L. and S.-W.C.; writing—review and editing, W.L. and S.-W.C.; visualization, W.L.; supervision, W.L. and S.-W.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Korea Research Institute of Ships and Ocean Engineering a grant from Endowment Project of "Development of Open Platform Technologies for Smart Maritime Safety and Industries" funded by Ministry of Oceans and Fisheries (1525012040).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors are grateful to the three anonymous reviewers for carefully assessing the manuscript and providing constructive comments, which significantly helped improve this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. International Maritime Organisation (IMO). *Guidelines for the Onboard Operational Use of Shipborne Automatic Identification Systems* (*AIS*); International Maritime Organisation (IMO): London, UK, 2002.
- Harati-Mokhtari, A.; Wall, A.; Brooks, P.; Wang, J. Automatic Identification System (AIS): Data reliability and human error implications. J. Navig. 2007, 60, 373–389. [CrossRef]
- 3. Filipiak, D.; Węcel, K.; Stróżyna, M.; Michalak, M.; Abramowicz, W. Extracting maritime traffic networks from AIS data using evolutionary algorithm. *Bus. Inf. Syst. Eng.* 2020, *62*, 435–450. [CrossRef]
- Wang, G.; Meng, J.; Han, Y. Extraction of maritime road networks from large-scale AIS data. *IEEE Access* 2019, 7, 123035–123048. [CrossRef]
- 5. Liu, T.; Ma, J. Ship Navigation Behavior Prediction Based on AIS Data. *IEEE Access* 2022, 10, 47997–48008. [CrossRef]
- 6. Murray, B.; Perera, L.P. An AIS-based deep learning framework for regional ship behavior prediction. *Reliab. Eng. Syst. Saf.* 2021, 215, 107819. [CrossRef]
- Suo, Y.; Ji, Y.; Zhang, Z.; Chen, J.; Claramunt, C. A Formal and Visual Data-Mining Model for Complex Ship Behaviors and Patterns. Sensors 2022, 22, 5281. [CrossRef]
- Zhang, Y.; Li, W. Dynamic Maritime Traffic Pattern Recognition with Online Cleaning, Compression, Partition, and Clustering of AIS Data. Sensors 2022, 22, 6307. [CrossRef]
- 9. Suo, Y.; Chen, W.; Claramunt, C.; Yang, S. A ship trajectory prediction framework based on a recurrent neural network. *Sensors* 2020, 20, 5133. [CrossRef]
- 10. Sørensen, K.A.; Heiselberg, P.; Heiselberg, H. Probabilistic Maritime Trajectory Prediction in Complex Scenarios Using Deep Learning. *Sensors* 2022, 22, 2058. [CrossRef]
- 11. Pelot, R.; Akbari, A.; Li, L. Vessel location modeling for maritime search and rescue. In *Applications of Location Analysis*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 369–402.
- 12. Wielgosz, M.; Malyszko, M. Multi-Criteria Selection of Surface Units for SAR Operations at Sea Supported by AIS Data. *Remote Sens.* **2021**, *13*, 3151. [CrossRef]
- 13. Ferreira, M.D.; Spadon, G.; Soares, A.; Matwin, S. A Semi-Supervised Methodology for Fishing Activity Detection Using the Geometry behind the Trajectory of Multiple Vessels. *Sensors* 2022, 22, 6063. [CrossRef] [PubMed]
- 14. Lee, J.S.; Lee, H.T.; Cho, I.S. Maritime Traffic Route Detection Framework Based on Statistical Density Analysis From AIS Data Using a Clustering Algorithm. *IEEE Access* 2022, *10*, 23355–23366. [CrossRef]
- 15. Yan, Z.; Xiao, Y.; Cheng, L.; He, R.; Ruan, X.; Zhou, X.; Li, M.; Bin, R. Exploring AIS data for intelligent maritime routes extraction. *Appl. Ocean Res.* **2020**, *101*, 102271. [CrossRef]
- 16. Karataş, G.B.; Karagoz, P.; Ayran, O. Trajectory pattern extraction and anomaly detection for maritime vessels. *Internet Things* **2021**, *16*, 100436. [CrossRef]
- 17. Rong, H.; Teixeira, A.; Soares, C.G. Data mining approach to shipping route characterization and anomaly detection based on AIS data. *Ocean Eng.* **2020**, *198*, 106936. [CrossRef]
- Sánchez-Heres, L.F. Simplification and event identification for ais trajectories: The equivalent passage plan method. *J. Navig.* 2019, 72, 307–320. [CrossRef]
- 19. Li, H.; Liu, J.; Liu, R.W.; Xiong, N.; Wu, K.; Kim, T.h. A dimensionality reduction-based multi-step clustering method for robust vessel trajectory analysis. *Sensors* 2017, 17, 1792. [CrossRef]
- 20. Last, P.; Bahlke, C.; Hering-Bertram, M.; Linsen, L. Comprehensive analysis of automatic identification system (AIS) data in regard to vessel movement prediction. *J. Navig.* **2014**, *67*, 791–809. [CrossRef]
- Iphar, C.; Napoli, A.; Ray, C. Detection of false AIS messages for the improvement of maritime situational awareness. In Proceedings of the Oceans 2015-mts/ieee washington, Washington, DC, USA, 19–22 October 2015; pp. 1–7.
- 22. Tu, E.; Zhang, G.; Rachmawati, L.; Rajabally, E.; Huang, G.B. Exploiting AIS data for intelligent maritime navigation: A comprehensive survey from data to methodology. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 1559–1582. [CrossRef]
- 23. Makris, A.; Kontopoulos, I.; Alimisis, P.; Tserpes, K. A comparison of trajectory compression algorithms over AIS data. *IEEE Access* **2021**, *9*, 92516–92530. [CrossRef]
- 24. Qi, L.; Zheng, Z. A measure of similarity between trajectories of vessels. J. Eng. Sci. Technol. Rev. 2016, 9, 17–22. [CrossRef]
- 25. Pallotta, G.; Vespe, M.; Bryan, K. Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction. *Entropy* **2013**, *15*, 2218–2245. [CrossRef]
- 26. De Vries, G.K.D.; Van Someren, M. Machine learning for vessel trajectories using compression, alignments and domain knowledge. *Expert Syst. Appl.* **2012**, *39*, 13426–13439. [CrossRef]
- 27. Patroumpas, K.; Alevizos, E.; Artikis, A.; Vodas, M.; Pelekis, N.; Theodoridis, Y. Online event recognition from moving vessel trajectories. *GeoInformatica* **2017**, *21*, 389–427. [CrossRef]
- 28. Zhao, L.; Shi, G. A trajectory clustering method based on Douglas-Peucker compression and density for marine traffic pattern recognition. *Ocean Eng.* **2019**, *172*, 456–467. [CrossRef]
- 29. Fikioris, G.; Patroumpas, K.; Artikis, A. Optimizing vessel trajectory compression. In Proceedings of the 2020 21st IEEE International Conference on Mobile Data Management (MDM), Versailles, France, 30 June–3 July 2020; pp. 281–286.

- 30. Feng, C.; Fu, B.; Luo, Y.; Li, H. The Design and Development of a Ship Trajectory Data Management and Analysis System Based on AIS. *Sensors* **2021**, *22*, 310. [CrossRef]
- 31. Zhang, S.K.; Liu, Z.J.; Cai, Y.; Wu, Z.L.; Shi, G.Y. AIS trajectories simplification and threshold determination. *J. Navig.* **2016**, *69*, 729–744. [CrossRef]
- Shi, W.; Cheung, C. Performance evaluation of line simplification algorithms for vector generalization. *Cartogr. J.* 2006, 43, 27–44. [CrossRef]
- Ji, Y.; Xu, W.; Deng, A. A Study of Vessel Trajectory Compression Based on Vector Data Compression Algorithms. In International Conference on Business Information Systems; Springer: Berlin/Heidelberg, Germany, 2019; pp. 473–484.
- 34. Qi, L.; Ji, Y. Ship trajectory data compression algorithms for Automatic Identification System: Comparison and analysis. J. Water Resour. Ocean. Sci. 2020, 9, 42. [CrossRef]
- 35. Douglas, D.H.; Peucker, T.K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica* **1973**, *10*, 112–122. [CrossRef]
- 36. Opheim, H. Fast data reduction of a digitized curve. *Geo-processing* 1982, 2, 33–40.
- Visvalingam, M.; Whyatt, J.D. Line generalization by repeated elimination of points. In *Landmarks in Mapping*; Routledge: London, UK, 2017; pp. 144–155.
- Willems, N.; Van De Wetering, H.; Van Wijk, J.J. Visualization of vessel movements. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2009; Volume 28, pp. 959–966.
- 39. Etienne, L.; Devogele, T.; Bouju, A. Spatio-temporal trajectory analysis of mobile objects following the same itinerary. *Adv. Geo-Spat. Inf. Sci.* **2012**, *10*, 47–57.
- 40. Muckell, J.; Olsen, P.W.; Hwang, J.H.; Lawson, C.T.; Ravi, S. Compression of trajectory data: A comprehensive evaluation and new approach. *GeoInformatica* 2014, *18*, 435–460. [CrossRef]
- Li, Y.; Liu, R.W.; Liu, J.; Huang, Y.; Hu, B.; Wang, K. Trajectory compression-guided visualization of spatio-temporal AIS vessel density. In Proceedings of the 2016 8th International Conference on Wireless Communications & Signal Processing (WCSP), Yangzhou, China, 13–15 October 2016; pp. 1–5.
- Singh, A.K.; Aggarwal, V.; Saxena, P.; Prakash, O. Performance analysis of trajectory compression algorithms on marine surveillance data. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 1074–1079.
- 43. Zhang, S.k.; Shi, G.y.; Liu, Z.j.; Zhao, Z.w.; Wu, Z.l. Data-driven based automatic maritime routing from massive AIS trajectories in the face of disparity. *Ocean Eng.* **2018**, 155, 240–250. [CrossRef]
- 44. Zhao, L.; Shi, G. A method for simplifying ship trajectory based on improved Douglas–Peucker algorithm. *Ocean Eng.* **2018**, *166*, 37–46. [CrossRef]
- 45. Huang, Y.; Li, Y.; Zhang, Z.; Liu, R.W. GPU-accelerated compression and visualization of large-scale vessel trajectories in maritime IoT industries. *IEEE Internet Things J.* 2020, *7*, 10794–10812. [CrossRef]
- 46. Wei, Z.; Xie, X.; Zhang, X. AIS trajectory simplification algorithm considering ship behaviours. *Ocean Eng.* **2020**, *216*, 108086. [CrossRef]
- 47. Liu, J.; Li, H.; Yang, Z.; Wu, K.; Liu, Y.; Liu, R.W. Adaptive douglas-peucker algorithm with automatic thresholding for AIS-based vessel trajectory compression. *IEEE Access* **2019**, *7*, 150677–150692. [CrossRef]
- 48. Tang, C.; Wang, H.; Zhao, J.; Tang, Y.; Yan, H.; Xiao, Y. A method for compressing AIS trajectory data based on the adaptive-threshold Douglas-Peucker algorithm. *Ocean Eng.* **2021**, 232, 109041. [CrossRef]
- 49. Ji, Y.; Qi, L.; Balling, R. A dynamic adaptive grating algorithm for AIS-based ship trajectory compression. *J. Navig.* 2022, 75, 213–229. [CrossRef]
- 50. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
- 51. Yap, P. Grid-based path-finding. In *Conference of the Canadian Society for Computational Studies of Intelligence;* Springer: Berlin/Heidelberg, Germany, 2002; pp. 44–55.
- 52. Zhang, Y.; Zhang, A.; Gao, M.; Liang, Y. Research on Three-Dimensional Electronic Navigation Chart Hybrid Spatial Index Structure Based on Quadtree and R-Tree. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 319. [CrossRef]
- Yahja, A.; Stentz, A.; Singh, S.; Brumitt, B.L. Framed-quadtree path planning for mobile robots operating in sparse environments. In Proceedings of the 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146), Leuven, Belgium, 20–22 May 1998; Volume 1, pp. 650–655.
- 54. Brondani, J.R.; de Lima Silva, L.A.; Zacarias, E.; de Freitas, E.P. Pathfinding in hierarchical representation of large realistic virtual terrains for simulation systems. *Expert Syst. Appl.* **2019**, *138*, 112812. [CrossRef]
- Shah, B.C.; Gupta, S.K. Speeding up A* search on visibility graphs defined over quadtrees to enable long distance path planning for unmanned surface vehicles. In Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, London, UK, 12–17 June 2016.
- 56. Thomas, M.; Geiger, C.; Kambhamettu, C. High Resolution Motion Estimation of Sea Ice Using an Implicit Quad-Tree Approach. ISPRS Workshop on High-Resolution Earth Imaging for Geospatial Information. *Citeseer*. 2007. Available online: https: //www.isprs.org/proceedings/XXXVI/1-W51/paper/Thomas_geiger_kambha.pdf (accessed on 10 September 2022).

- 57. Lee, W.; Choi, G.H.; Kim, T.W. Visibility graph-based path-planning algorithm with quadtree representation. *Appl. Ocean Res.* **2021**, *117*, 102887. [CrossRef]
- 58. Bielak, J.; Ghattas, O.; Kim, E. Parallel octree-based finite element method for large-scale earthquake ground motion simulation. *Comput. Modeling Eng. Sci.* **2005**, *10*, 99.
- Liang, M.I.; Ramaswamy, B.; Patterson, C.C.; McKelvey, M.T.; Gordillo, G.; Nuovo, G.J.; Carson, W.E. Giant breast tumors: Surgical management of phyllodes tumors, potential for reconstructive surgery and a review of literature. *World J. Surg. Oncol.* 2008, *6*, 117. [CrossRef]
- 60. Popinet, S. Quadtree-adaptive tsunami modelling. Ocean. Dyn. 2011, 61, 1261–1285. [CrossRef]
- 61. Zhou, C.; Chen, Z.; Li, M. A parallel method to accelerate spatial operations involving polygon intersections. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 2402–2426. [CrossRef]
- 62. Hjaltason, G.R.; Samet, H. Speeding up construction of PMR quadtree-based spatial indexes. VLDB J. 2002, 11, 109–137. [CrossRef]
- 63. Baselga, S.; Martínez-Llario, J.C. Intersection and point-to-line solutions for geodesics on the ellipsoid. *Studia Geophys. Geod.* **2018**, 62, 353–363. [CrossRef]
- 64. Pietrzykowski, Z.; Wielgosz, M. Effective ship domain–Impact of ship size and speed. Ocean Eng. 2021, 219, 108423. [CrossRef]