The Parallel and Precision Adaptive Method of Marine Lane Extraction Based on QuadTree

Zhuoran Li[1,2], Guiling Wang[1,2](✉), Jinlong Meng[1], and Yao Xu[2]

[1] Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data,
North China University of Technology,
No. 5 Jinyuanzhuang Road, Shijingshan District, Beijing 100144, China
wangguiling@ict.ac.cn

[2] Ocean Information Technology Company,
China Electronics Technology Group Corporation (CETC Ocean Corp.),
No. 11 Shuangyuan Road, Badachu Hi-Tech Park,
Shijingshan District, Beijing 100041, China

Abstract. Extracting the marine lane results from the ocean spatial big data is a challenging problem. One of the challenges is that the quality of the trajectory data is quite low, and the trajectory data quality is extremely different in different areas. A parallel and precision adaptive method of marine lane extraction based on QuadTree is proposed to meet this challenge. The method takes advantage of several methods including average sampling, interpolation, removing noise, trajectory segmentation, and trajectory clustering based on GeoHash encoding through the MapReduce parallel computing framework. The preprocessing phase can effectively simplify the big data and improve the efficient of data processing. Based on the QuadTree data structure, a parallel merge filtering algorithm is proposed and implemented used Spark framework. The algorithm performs grid merging on the sparse grid regions, and obtaining a new grid result with different size. The sliding local window filtering algorithm based on the QuadTree is proposed to obtain the marine lane grid set data. Applying the Delaunay triangulation method on the grid data, the multi-precision marine lane results are effectively extracted. The experimental results show that the proposed method can automatically extract multi-precision marine lane using the trajectory data near the coast with high and low grid precision.

Keywords: AIS data · Precision adaptive · Marine lane extraction

1 Introduction

Road-related geographic data is an important part of national basic geographic information and intelligent transportation. It has great application value in smart city building, intelligent navigation, traffic control, and Internet map services. Recent years, with the development of technologies such as mobile sensors and cloud computing, massive trajectory data (also known as crowdsourcing trajectory data) are collected from a large number of vehicles (such as automobiles, ships, etc.). The geographical information of the road is cheaper and faster than the traditional way of acquiring geographic information. However, the data volume of the crowdsourcing trajectory is

H. Gao et al. (Eds.): CollaborateCom 2018, LNICST 268, pp. 170–188, 2019.
https://doi.org/10.1007/978-3-030-12981-1_12

large and there a lot of noisy data that can't reflect the real location of the vehicles. Due to these problems, it is very challenging to extract the geographic information from such crowdsourcing trajectory data and the challenges have attracted more and more researchers' attention.

In the field of urban transportation, the source trajectory data using road geographic information extraction can be collected from the Global Positioning System (GPS) terminal equipment or GPS acquisition center of land vehicles. In the field of maritime traffic, vessel trajectory data is collected from the terminal equipment of the vessel Automatic Identification System (AIS) or the AIS data collection center. Compared with the traditional methods which depend on manual measurement or the high-resolution remote sensing images to extract road information, the acquisition of trajectory data such as GPS and AIS data is cheaper and has higher performance. Since the trajectory data can be classified and analyzed according to different kinds of vehicles, the factual and detailed road information for different kinds of vehicles can be extracted, and the road changing can be reflected in time. Therefore, if the geographic information of the shipping lanes on the sea can be accurately extracted from the crowdsourcing trajectory data, it will have a great prospect.

The source trajectory data has the characteristics of large scale, high noise, and uneven sampling frequency distribution. For example, the original data collected by the global vessel trajectory data for one year is TB level, and almost every vessel's trajectory has wrong sampling data. The sampling frequency of the track points in the offshore area ranges from 5 s to 100 s and in the far-sea area the sampling interval is large ranges from 2 min to 10 min. These pose a challenge for accurately extracting fine marine lane information and it has important research significance.

Compared with vehicles on land such as automobiles, vessels' trajectories are more severely affected by harsh climatic conditions at sea. AIS data's density and quality are unevenly distributed. Therefore, compared with GPS data of vehicles on land, vessel AIS data has higher noise, and the density of AIS data is significantly different in different regions, which poses greater challenges for the extraction and updating of the marine lane. The density and quality of AIS data collecting by the vessel's sensors in the offshore and near shore areas are very different, and the vessel track points in the offshore areas are naturally more densely distributed than far-sea areas. So the fineness of extraction in the offshore areas is also higher than far-sea areas. The distribution of ship trajectory points more densely, and the fineness of the marine lane is higher. Therefore, it is impossible to use a uniform precision to extract the marine lane, and designing a method to uniformly extract the marine lane of different precision in the different areas is necessarily. This paper focuses on solving the difficult problems of unified extraction of the marine lane with different precision from the large-scale, high-noisy, and density uneven data. Aiming at this problem, a QuadTree data structure is proposed. Based on the data structure, a parallel adaptive precision merging and filtering algorithm is designed. It can be used for large-scale crowdsourcing trajectory data and different precision of marine lane recognition and extraction.

Note that the marine lane is an area with attributes such as width and depth. The paper mainly focuses on the plane attribute of the marine lane, and the extracted marine lane information only includes the plane boundary data.

Section 2 of the paper introduces related work. In Sect. 3, the basic concepts throughout the paper are introduced and the problems to be solved are described. Section 4 describes the preprocessing of ship trajectory data. Section 5 describes the basis algorithm for this paper. Including the parallel and precision adaptive marine lane extraction algorithm and so on. The Sect. 6 is the experiment and evaluation. Finally, the paper summary and prospects in the Sect. 7.

2 Related Work

At present, road map information mining from spatiotemporal trajectory big data has become a hot research topic in the research field of big data. There are many research results on extracting the road information from spatiotemporal big data. The GPS data and AIS data is widely used in pattern recognition, predicting route, and anomaly behavior detection, etc. Some researchers focus on extracting road center lines by trajectory data clustering. For the volume of the trajectory data is often very large, there are some works on using parallel computing technologies to simplify trajectory data processing. Edelsbrunner et al. [1] developed an optimal *O(n log n)* algorithm that constructs shapes. Brown [2] constructed the K-dimensional Euclidean Voronoi diagram of N points by transforming the points to K + 1-space. Zhang el al. [3] presented a more advanced method for detecting near miss ship collisions. Arguedas et al. [4] performed detection and discovery of such highlighting of frequent lines and breakpoints. He et al. [5] proposed nonlinear optimization method and the results showed that the average classification accuracy is 98.93%. Wu et al. [6] used AIS data to analyze navigational patterns along the waterways. Arguedas et al. [7] had general spatiotemporal characterization and statistical analyses of the traffic systems in some sea areas. Etienne et al. [8] perform a data mining on a huge quantity of mobile object's positions moving in an open space in order to deduce its behaviour. Unusual behaviours such as being ahead of schedule or delayed or veering to the left or to the right of the main route are detected. Vespe et al. [9] provided the basis for robust archives of data to extract main shipping intensities and routes. Pallotta et al. [10] reflected the knowledge discovery process of the Traffic Route Extraction and Anomaly Detection (TREAD) methodology. Wang et al. [11] attempted to tackle the big data issue caused by the AIS data for anomaly detection purposes. Ahmed et al. [12] represented a first comprehensive attempt to benchmark such map construction algorithms. Wang et al. [13] adopted a divide-and-conquer strategy for reconstructing road segments and road intersections separately from raw GPS trajectories. Wang et al. [14] thought GPS probe data can essentially provide information of a traffic condition of a given period, such as travel time estimation, as well as traffic congestion, which directly relates to the distance travelled by a vehicle in that period. Broach et al. [15] developed a multinomial logit (MNL) model to impute travel mode from GPS and accelerometer data. Winden et al. [16] investigated the automatic extraction of eight road attributes: directionality, speed limit, number of lanes, access, average speed, congestion, importance, and geometric offset and developed a supervised classification method (decision tree) to infer them. Costa and Baldo [17] presented a method based on the genetic algorithm for the generation of road maps from trajectories collected with a smartphone. Park et al.

[18] provided a methodology for integrating pedestrian facilities and obstructions information with an existing PND. Hu et al. [19] used area of interest (AOI) has been to describe one kind of POI collections, namely areas that attract and support various human interests and activities. Merry et al. [20] presented a GPU-accelerated implementation of the moving least-squares (MLS) surface reconstruction technique. Mistry et al. [21] built a hierarchy of cavities and protrusions for each polygon and used this hierarchy to check for matching between these geometric features of two polygons. Peethambaran and Muthuganapathy [22] presented a fully automatic Delaunay based sculpting algorithm for approximating the shape of a finite set of points S in R^2. Cheng [23] presented one kind of dynamic positive and negative feedback ACO which differs from existing ACO in two important aspects: (i) positive feedback inner-colony and negative feedback inter-colony, and (ii) parallel implementation on Hadoop, a framework built with iterative Map Reduce model. Aghabozorgi et al. [24] attacked the problem that several different techniques used to cluster time series and sequences by utilizing a novel incremental fuzzy clustering strategy in order to achieve the objective. LI et al. [25] proposed a method of heat factor similarity measurement based on the combination of distance and density of grid heat value. ANMED et al. [12] provided an evaluation and comparison of seven algorithms using four datasets and four different evaluation measures. Yang and Ai [26] presented a new approach to use vehicle trajectory lines to extract road boundary. Kuntzsch et al. [27] inferenced of traffic networks from GPS trajectories. Jiang et al. [28] proposed a thinning-algorithm-based method extract center lines to construct road network in Lujiazui, Shanghai with taxi trajectory data.

Many researchers have achieved many achievements about trajectory big data, but there are still some problems: (1) Most of the researches only extract the center line of the road structure, and do not accurately extract the internal and external boundary information of the road. (2) Most of the researches carry out boundary extraction for trajectory data under some marks of land things, it is not suitable for massive data with uneven distribution of density under a large range. Therefore, this paper focuses on the difficulty of large differences in density of unconstrained big ship data, and establishes a method to remove density differences and construct an effective precision adaptive model for extracting marine lane.

3 Definitions and Problem Description

The paper first gives a few basic concepts, then introduces the framework and working principle of the model.

Definition 1. Vessel Trajectory. Vessel Trajectory T_{vi} is a spatiotemporal point sequence of vessel v_i, which represents the sequence of positions of the vessel over a period of time. $T_{vi} = (v_i, <p_0, p_1, \ldots, p_N>)$, where $\mathrm{v_i}$ represents the maritime mobile service identify (MMSI) of the vessel, $p_j = <x_{i,j}, y_{i,j}, t_j>$ indicates the position of the vessel at a certain moment, t_j is the sampling time, $x_{i,j}$, and $y_{i,j}$ represents the latitude and longitude of the vessel v_i at t_j.

Definition 2. Marine Lane. Marine lane P_{lane} is a two-dimensional polygon representing the area where ships are allowed to sail. $P_{lane} = (p_0, p_1, ..., p_n)$, in which the set of vertices p_i constitutes a polygon P in a clockwise direction.

Definition 3. Empty Hole. Empty hole P_{empty} is a two-dimensional polygon representing the area where ships are not allowed to sail due to obstacles such as reefs and the Government controlled area inside the marine lane. The Empty hole is nested inside the marine lane and does not appear individually. $P_{empty} = (p_0, p_1, ..., p_n)$, in which the set of vertices p_i constitutes a polygon P in a clockwise direction.

Definition 4. Grid. Grid is a rectangular area on a map. By dividing the 2D geospatial through horizontal and vertical direction, the whole geographical area is divided into multi equal rectangular in size, each rectangular being called a grid. Grid can be described by *Grid = (Code, Dsy).*

Code: GeoHash code of grid. *Code* obtained by GeoHash encoding the location of the grid center including longitude and latitude, it is a string constituted by 0 or 1, and the coding length $|Code|$ represents precision that the grid up to.

Dsy: grid density. *Dsy* is the number of AIS points in one grid, $Dsy = |\{p|lllon_{grid} < p_{lon} < urlon_{grid}, lllat_{grid} < p_{lat} < urlat_{grid}, p\epsilon P\}|$, p indicates AIS points.

Definition 5. Parent Grid. Parent grid is the grid divided once in the latitude and longitude directions into four sub-grids. The divided grid is called the parent grid of the four sub-grids. For the parent *Grid = {Code, Dsy}*, which Code is the prefix of its any sub-grid's *Code*, that is $Code_{par} = subString(Code_{sub}, 0, |Code_{sub}| - 2)$, and the parent grid's density is the sum of four sub-grids' density, that is $Dsy_{par} = \sum Dsy_{sub}$.

Definition 6. Marine Lane Precision. The mean precision of the grid used to extract the marine lane is called the marine lane precision. $Precision_{lane} = \frac{\sum precision_g}{|G|}$, G indicates grid set used to extract marine lane, g is the grid in G.

The main steps of marine lane extraction from vessel AIS trajectory data with large density differences as follows: (1) Obtaining AIS data with specific conditions. (2) Simplifying grid trajectory data use noise data removing, missing data inserting, trajectory data segmentation, sample averaging four preprocessing method and clustering operation. (3) Extracting marine lane grid information extraction, using grid merging algorithm based on QuadTree and dynamic sliding window filtering algorithm to obtain effective marine lane information by grid. (4) Extracting marine lane by Delaunay triangulation and the lane extraction algorithm based on circumscribed circle radius.

In this paper, the problem of marine lane extraction is described as follows: Given AIS trajectory big data collected from the mobile sensors on a large number of vessels, we aim to extract the marine lane information in a certain area of a specific type of vessel. Usually, the vessel reports its position information at different times. These positions can constitute a complete dynamic trajectory sequence. Since the AIS trajectory data is collected by the sensors, there inevitably has error data, missing data and redundant data. And it is difficult to distinguish one vessel's multi trips in a total vessel trajectory sequence. To avoid these problems about data quality, four different preprocessing methods were proposed in this paper. However, the volume of the preprocessed data is still very large. It is difficult to perform marine lane extraction model on it. GeoHash encoding the data is a simplifying method for trajectory data, the neighbor points in position are uniformly simplified into one grid center point. All data points are encoded into GeoHash codes. Then we sum up the data points and can get a set *G* which is constituted by a large number of grid center points. In our method, we use the size of the grid area to indicate the grid precision of each grid, and use the number of original AIS points within the grid to represent the grid density of this grid. The encoded grid point with *Code* and *Dsy* two attributes can represent original data features effectively. Preprocessing and GeoHash encoding can improve data availability, data neatness, optimize memory usage, and improve efficiency of marine lane extraction model. The precision of marine lane depends on grid precision constituting marine lane. The larger the volume of data in a region, the higher the grid precision and the more significant the grid density in different regions after GeoHash encoding, the higher the precision of the extracted marine lane results and vice versa. However, in different conditions in the far sea and near shore regions, the volume of data collected in different regions is significantly different. If uniform high-precision grid parameters are used for extraction, there will be fine marine lane results in some regions which AIS data volume is proper to the parameters, but other regions are difficult to form effective marine lane. Then in some regions that not proper to the certain uniform parameters, the marine lane produce incomplete phenomena such as channel fracture or loss.

In order to solve the above problems, we firstly merge the sparse grids of different lower densities by the grid merging algorithm based on QuadTree, and then uses the sliding window filtering algorithm to obtain the marine lane grid information. The marine lane extraction algorithm is used to extract the marine lane results. The high-density grid is using to extract a high-precision marine lane. Conversely, the low-density grid is using to extract a low-precision marine lane, and overall the marine lane precision is adaptive.

Figure 1 is an overall architecture figure of parallel and precision adaptive model for marine lane extraction. The data storage layer is responsible for storing all raw AIS data and intermediate process data as well as marine lane extraction results. The core algorithm layer includes AIS data extraction under specific conditions, data preprocessing, and data clustering by Map Reduce parallel computing framework. It can be used once or multiple times median filtering to remove the meaningless isolated grid that may exist in the grid result. The marine lane information extraction process uses the Spark memory distributed computing framework to avoid the memory limitation problem. The marine lane grid information is obtained by the grid merge algorithm based on QuadTree and dynamic sliding window filtering algorithm based on merged QuadTree. Then triangulate the marine lane grid by Delaunay and extract the results by boundary extraction algorithms based on triangle circumcircle radius. The data display layer is responsible for visualizing the upper and lower results of the intermediate and final results, observing the data exception, and providing parameter guidance for the algorithm model.

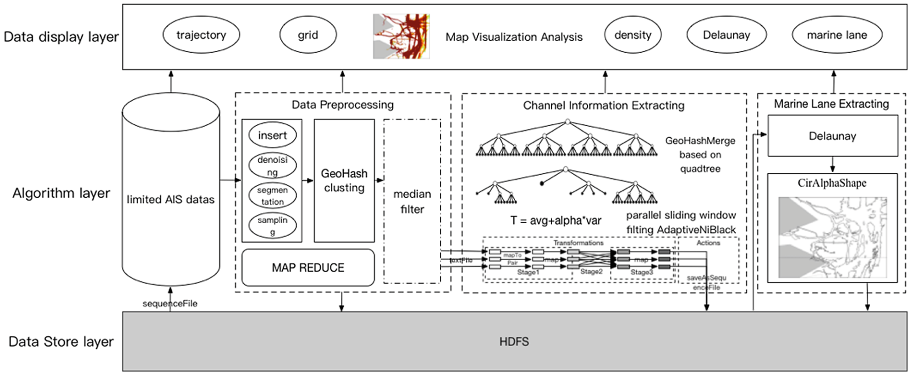


Fig. 1. Framework for marine lane extraction.

4 Trajectory Data Preprocessing

The data collection process is affected by the environment of the vessel, the quality of the telecommunication equipment, and the telecommunication process environment. Therefore, the data inevitably have problems such as data missing and data error and so on. In addition, when the moving vessel is berthed or anchored, the collected data is redundant in the marine lane extraction model. In this paper, we design the missing data inserting, noise data removing, trajectory data segmentation and sampling data averaging four methods to preprocess the original data using the Map Reduce parallel computing framework. The preprocessing can improve the data quality and optimize the extraction results.

The data preprocessing process is described as follows:

(1) Data split, dividing the SequenceFile file in the HDFS storing the original data into m ($m \gg n$, n represents the number of data nodes) data blocks Split, each block is processed by a Datanode.

(2) Map stage, program read data row-by-row, and obtain four attributes of v, x, y, t for each record of data. The field v is the key, and the tuple (x, y, t) is the value, and output form is $<v, (x, y, t)>$.

(3) Reduce stage, each Reduce processes data with the same key v as follows: First, setting the time threshold t_z, the distance threshold d_z and the trajectory number threshold n_z, and sorting the data having the same v by time t. Second step, add p_i in the array list arr_tra, if the distance d between the p_i and p_{i+1} is less than d_z, and the time interval Δt_i is less than t_z, then $i = i + 1$ (add the p_{i+1} to arr_tra), continue to circle this step, otherwise, continue to the third step. Third step, calculate the length N of array list arr_tra, if N is less than n_z, confirming points in the array list arr_tra as noise points, clearing the list, and $i = i + 1$, continue to the second step. Otherwise, confirming the array list arr_tra as a normal trajectory and compute the average distance d between two adjacent points in the array list, execute the fourth step. Fourth step, compute the distance d between the p_j and p_{j+1}, if d_j is less than d, or the difference degree between the longitudes of the two

points is greater than 300 (the two point near east longitude 180° and west longitude 180° respectively, the threshold value can be less than 360° and as large as possible, we assume it to 300), then $j = j + 1$, continue this step. Otherwise, $s = d_j/d + 2$, insert s points between the p_j and p_{j+1}, and add them in the list, then $j = j + 1$, continue to circle this step until traversal list, continue the fifth step. Fifth step, add the elements in the list, and clear the list, then $i = i + 1$, execute the second step until the same v data is traversed. The preprocessed data is output with the field v as the key, the tuple (x, y, t) as the key value, and the form is $<v, (x, y, z)>$.

(4) The preprocessed data is saved in a different text file by v.

(5) Finally, the preprocessed data is obtained. After using the above data preprocessing based on Map Reduce, the overall process was shortened from 40 h to 2 h and 30 min under the cluster experimental conditions described in Table 1.

In addition, we use the GeoHash coding clustering method to simplify the data. GeoHash coding is a classic method of encoding geographic data. This method recursively divides the entire geographical range into multi grid longitude and latitude direction, and obtains a grid map. Each grid corresponds to a GeoHash code, and then falls. Data points in the same raster correspond to the same encoding, and finally the center point of the grid represents all the data in the entire grid. The data set has been simplified by GeoHash clustering.

The GeoHash coding clustering is described as follows:

(1) Data split, dividing data files named by MMSI in HDFS into m' ($m' \gg n$, n represents the number of data nodes) data blocks, each block is processed by one Datanode.

(2) Map stage, Geohash encoding the longitude x and the latitude y of each record of data are extracted as the key, the value corresponds the key is 1, and the form is $<code, 1>$.

(3) Reduce stage, counts grid density as dsy, calculate the center point latitude and longitude C_x and C_y of its corresponding area C, with C_x, C_y and dsy as keys, and $null$ as key value, the form is $<(C\_x, C\_y, dsy), null>$.

(4) Complete GeoHash clustering. All the data is saved by tuple $<C\_x, C\_y, dsy>$.

The distribution of the grid data maintains the original distribution characteristics. In order to make the neighbor grid density more smooth, using the modified median filtering algorithm based on the image processing fuzzy algorithm to process results. Median filtering algorithm removes isolated noise points and makes the density variation trend of all regions smoother and more uniform.

5 Precision Adaptive and Parallel Extraction Algorithm for Marine Lane Based on QuadTree

5.1 Precision Adaptive and Parallel Grid Merging

The trajectory grid data is stored in a QuadTree, and marine lane precision adaptive and parallel merge algorithm merges it automatically based on the traversal method from bottom to up. Grid merging algorithm obtain parent grid unit which has a bigger density value, substitute for the four sub-grids which have smaller density value than merge threshold value. It increasing difference of the local adjacent grid density, so that the overall trajectory grid data performs the apparent difference characteristics between the channel and the non-channel. Therefore, it can establish the precision adaptive marine lane extraction model to extract the marine lane result.

For the grid data processed in Sect. 4, we first create a QuadTree to store data. Each node corresponds to a grid, stores a GeoHash grid with Code and Density value two attributes, except that the root node does not store Code and Density value. The QuadTree is shown in Fig. 2. The principle of QuadTree establishment is from the initial geographical range, and the undivided range is no coding and taken as the root node. Then, the initial range is equally divided into 4 parts through the latitude and longitude direction, and the QuadTree is created. The four nodes of the first layer of QuadTree correspond to the divided geographical regions of the four directions of southwest, southeast, northwest and northeast in the order of coding 00, 01, 10 and 11 respectively. The establishment of the second layer is performed by dividing each divided range of the first layer as above method, and obtaining 16 nodes constructing the second layer. The establishment of subsequent levels as above method too. In general, a QuadTree's layers will not exceed 20 layers, because the grid deviation range of the 20th layer is within 8 meters, which can meet the accuracy requirements of all problem environments. This paper uses the QuadTree structure to store data with the following three advantages: (1) The QuadTree data structure corresponds the principle of GeoHash coding partitioning algorithm, which can reflect the hierarchical relationship and neighbor relationship between geographic grids. (2) In general, creating a QuadTree starting from the entire world geographic range, up to 20 times division can meet the accuracy requirements of all applications, so the QuadTree layer is usually no more than 20 layers, each node of the QuadTree is at most four sub-nodes, the effective nodes avoid the local traversal search spend much time compared the traditional prefix

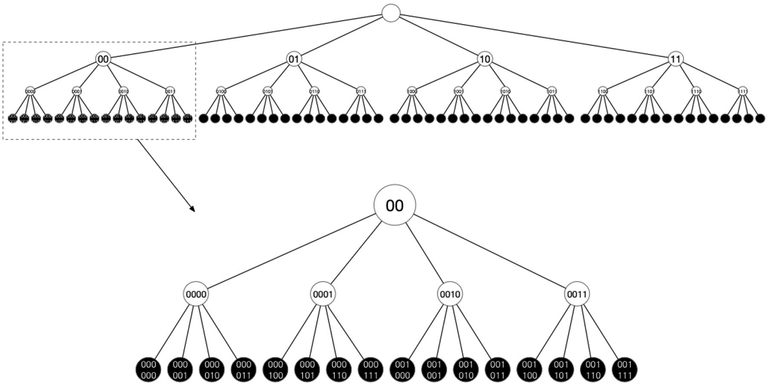


Fig. 2. A QuadTree with 3 layers, each leaf node has coding and density by integer value, not leaf node has coding and 0 density value, while root node has not coding and 0 density value.

tree, so the search efficiency is efficient. (3) For a node, which parent node coding and neighbor nodes coding can be quickly computed through its coding, so its parent and neighbor node can be quickly searched by itself. The algorithm has a lot of search for upper node and surrounding nodes, so using the QuadTree structure improves the efficiency of model.

The core idea of the precision adaptive merging algorithm is that the grid with bigger density value in a certain area has higher grid precision. On the contrary, the grid with smaller density value, the lower the grid precision of the grid. First set parameters of the algorithm: the highest grid precision $bitnum_{max}$ and the lowest grid precision $bitnum_{min}$, and the merge density threshold dsy_t, and then the entire geographic range can be according to the highest grid precision divided into equal sized grids, each grid containing a different number of AIS points, corresponding to the grid density values of the grid. The merging process unit is four sub-grids that belong to same parent grid. If the grid density value of the four sub-grids are lower than the merging density threshold dsy_t, then the four sub-grids are combined into one parent grid, that is, modify the parent grid density value to the sum of the four sub-grids, otherwise, the merge condition is not satisfied and the merging is not performed. The algorithm traverse tree by layer, and judges merging from the highest precision layer. After the first layer traversal is completed, the secondary high-precision layer grid is merged by the method above, and the QuadTree is merged layer by layer. The merge process is completed when the grid has reached the level of the set minimum grid precision. Figure 3 shows a grid which satisfies the merging condition and performs the merge operation. The data structure changes are shown in Fig. 3.

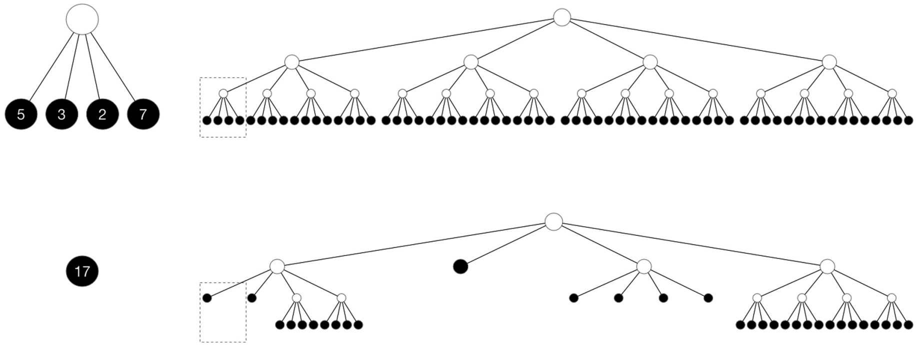


Fig. 3. A simple merge process of a QuadTree with 3 layers, merging start from bottom to top by layer, for one node if four sub-nodes' density value smaller than merge threshold density, then merge to their parent node, set parent node's density to sum of the sub-grids', until one sub-node not smaller than threshold or up to minimum merge layer.

Algorithm 1. Precision adaptive combining algorithm

```
Input: G: grid data
Output: MG--merged grid data
1: T = initGeohashTrie(G)              //create a QuadTree T
2: for node in T do                    // traverse the QuadTree T from bottom to top
3:     if node.depth in range(accuracy) then
4:         Q.add(node)                 //add a node not below the merge precision
5: for q in Q do:                      //traversing queue Q
6:     if (g.subnodes.dsy<maxDsy) then
7:         g.dsy = sum(g.subnodes)         //merged child nodes
8:         del(g.subnodes);                //delete child nodes
```

The process of the precision adaptive merge algorithm is shown in Algorithm 1:

Line 1 establishes a QuadTree T store the grid data.

Lines 2–4 traverse the QuadTree T from bottom to top by layer. If the node layer is not below the set minimum precision and not above the second highest precision range, the node is added to the queue Q.

Lines 5–8 are traversing the queue Q in reverse order, that is, starting from the sub-precision layer, determining whether the grid density values of the four sub-nodes of one node are smaller than the grid density threshold, and if true, set the density value of the node to the sum of all node density values and delete all child nodes, otherwise no operation.

When the grid precision is set high and the geographical range is large at the same time, the number of grid after initialization will more than 2^{40}. If a QuadTree is created to store these data, there will be high memory requirements and program efficiency become low. Aiming at these problems, this paper designs a parallel merging algorithm based on Spark memory-based distributed computing framework technology. The algorithm parallelism is to use the first N bits of GeoHash code as the key value. According to the geographic characteristics of the GeoHash common parent grid with the same prefix code, the algorithm is to divide the geographic range into 2 to N/2 powers. Then computing simultaneously for all divided regions in parallel.

5.2 Parallel Dynamic Sliding Window Filtering Algorithm

The general sliding window algorithm is mostly based on one-dimensional array or two-dimensional matrix data structure. This paper designs a new dynamic local sliding window filtering algorithm based on the above QuadTree structure. The main idea of the window filtering algorithm is to start from the center point of the first window in the upper left, search all the adjacent nodes in the window according to GeoHash code, and then perform local filtering equation in the window. The sliding of the window is the center point of the window. The center point is a sliding object, and the center point of the next window adjacent to it is searched to calculating and filtering until the traversal is completed. If the distance between the center point of the window and the boundary of the window is greater than the distance with the boundaries of the entire range, the boundary points in the window will be out of bounds. In order to avoid the problem of the out-of-bound boundary of the computed neighbor, the distance between selection of the center point with left and up boundary of the entire range is half of the width and height of the window. The range formed by the center point of the window aligned along the entire geographic extent, as the traversal range of the center point of the window.

The idea of local filtering within the window is based on the NiBlack binary filtering method. The filtering threshold is represented by T, and is retained if the center point grid density is greater than the threshold, otherwise discarded. The calculation formula of the filtering threshold T is as follows:

$$T = avg + alpha \times var \tag{1}$$

In Eq. (1), *avg* represents the average density value of all grids in the window, *var* represents the variance density value of all grids in the window, and *alpha* represents the variance correction factor.

When counting the *avg* and *var* of all grids in the window, because the sliding window is constituted of the lowest precision measured grid, the actual one grid may include 4^N higher precision sub-grids. Such grid needs to count all the density values of the actually included higher precision grid. Similarity, for the center point grid, if it is the lowest precision grid, just judging whether it is greater than threshold density value. Otherwise, that is the center point grid actually retains 4 ^{N} higher precision sub grids, needing compares every sub-grid with the threshold density value T.

Algorithm 2. AdaptiveNiBlack: local window filtering algorithm based on QuadTree

```
Input: MG--merged grid data
       wwidth--window width
       wheight--window height
Output: FG--filtered grid data
1: for node in MG do
2:     win = createWin(node,wwidth,wheight);  //initial sliding window
3:     avg = average(win)
4:     var = var(win)
5:     T = avg + alpha*var                    //calculate the threshold within the window
6:     for hNode in node do
7:         if hNode.dsy>T then
8:             FG.add(hNode)
```

The process of local filtering of the QuadTree window is shown in Algorithm 2:

The algorithm takes the current node as the center point of the window and creates a window according to the set window size (line 2).

According to the threshold formula, after the local threshold T in the window is calculated, the center point needs to be deeply traversed, and each child node is cyclically judged, and the node whose density value is greater than T is added to the filtered result set FG (lines 6–8).

When the grid precision is set high and the geographical range is global, the algorithm also has high memory requirements and low efficiency due to the large number of grids. Since the geographic range is divided according to the GeoHash coding principle, the data in different grids is locally filtered independently. Therefore, the GeoHash coding prefix is used as the key value to divide the data, and the Spark technique is used to simultaneously run the algorithm on the multi divided parts. Finally, all the filtering results are combined, which improves the efficiency of the algorithm and saves memory space.

5.3 Marine Lane Extraction Algorithm Based on Delaunay

Through the previous two processes, after obtaining the overall filtered grid result, it is equivalent to having obtained the grid points of all the channels. After the grid result is merged and filtered, there may still be uneven distribution on the density of the original data. Therefore, the fuzzy processing algorithm in the image processing technology can be repeatedly used to filter the grid result. That is use the mean value of the surrounding grid density value replaces the density value of the center grid, to achieve the purpose of removing clutter and smoothing the density distribution of the grid result.

Finally, it is necessary to correctly extract the marine lane formed by the adjacent grid from these grid results, and the boundary of the empty hole may exist inside the region where the adjacent grid is located. In this paper, based on the Delaunay triangulation method, the CirAlphaShape marine lane extraction algorithm is designed. The constructed Delaunay triangulation is used to determine the boundary set of the triangle by using the triangle circumscribed circle radius as the judgment condition, and then the marine lane result is extracted from the boundary edge set based on the algorithm.

Algorithm 3. CirAlphaShape: Marine lane extraction algorithm

```
Input:   FG--filtered grid data
         thresCir--circumscribed circle radius threshold
Output: P--marine lane result
D = delaunay(FG)              //FG triangulation
for d in D do
    if d.circum>thresCir then
        E.add(d.edges)        //d side added to E
P = polygonize(E)             //E form a set of polygon
```

6 Experiment and Performance Analysis

6.1 Data Set and Experimental Environment

The data used in this experiment is the AIS data of all cargo ships from June 2016 to July 2016 including China, Malaysia, Singapore, Indonesia and other important ports. AIS data includes vessel's name, call number, MMSI, IMO, ship type, captain, ship width and other static information and accuracy, latitude, longitude, direction, speed and other dynamic information as well as status, destination, ETA for the voyage data, the four columns of MMSI, time, longitude and latitude are used in this paper. The data examples are shown in Table 1. From June 2016 to July 2016, the total amount of AIS data collected by global cargo ships was 510G. After pre-processing, the total data volume was 364G. After encoding and clustering, the data volume was reduced to 5.5G.

The experimental environment of this paper runs in Hadoop cluster environment. The specific configuration of the cluster is shown in Table 2. The data preprocessing process was developed by Map Reduce parallel computing framework. The marine lane information filtering module was developed by Spark memory distributed computing framework. The marine lane extraction process used Python programming language implemented a single process program in the CentOS release 7.0 system.

Table 1. AIS data information table.

MMSI	Longitude	Latitude	Time
412351810	121.000671	30.449981	2016-06-10 12:00:43
412351810	121.000671	30.452728	2016-06-10 12:02:30
…	…	…	…
412351810	121.055603	31.045989	2016-06-10 16:10:58

Table 2. Hadoop cluster configuration table.

IP	Role	CPU	Memory
10.61.2.13	Slave3	Intel Xeon E5620 2.40 GHz	32G
10.61.2.14	Slave4	Intel Xeon E5620 2.40 GHz	32G
10.61.2.17	Slave5	Intel Xeon E5620 2.40 GHz	32G
10.61.2.111	Master	AMD operon(tm) 6128	64G
10.61.2.112	Slave1	AMD operon(tm) 6128	64G
10.61.2.113	Slave2	Intel Xeon E5620 2.40 GHz	32G
10.61.2.123	Slave6	Intel Xeon E5620 2.40 GHz	32G
10.61.2.124	Slave7	Intel Xeon E5620 2.40 GHz	32G
10.61.2.125	Slave8	Intel Xeon E5620 2.40 GHz	32G

6.2 Analysis of Results

The original AIS data is affected by the telecommunication conditions, and there are problems such as data missing, data error, data redundancy, etc. It is difficult to accurately extract the marine lane results directly using original data from the model. After the pre-processing, the preprocessed trajectory data is clean and usable. It is stored in HDFS by vessel. It is impossible to directly extract the boundary of the channel area where a large number of vessels are sailing together. It is necessary to further abstract the data into grid data by GeoHash encoding clustering. This paper compared the parallelization method and the single process method to complete the time performance of the pre-processing module. The result showed that the parallelization method can increase the time performance by more than 5 times on average compared with the single process method.

In order to verify the precision adaptive performance of the proposed method, this paper used the OSTU-based marine lane extraction and the traditional NiBlack-based marine lane extraction for experimental comparison. The OSTU method belongs to the global optimal threshold method, and the optimal threshold is calculated from the overall grid density which can divide all elements into two parts most apparently. The traditional NiBlack is based on the matrix dynamic local window threshold method, which dynamically filters the grid of the same precision to obtain the marine lane information. The AdaptiveNiBlack method performs dynamic local window threshold filtering algorithm based on QuadTree to obtain marine lane information with different grid precision. The Fig. 4 is an experimental comparison of three different methods for marine lane extraction of the same data set.

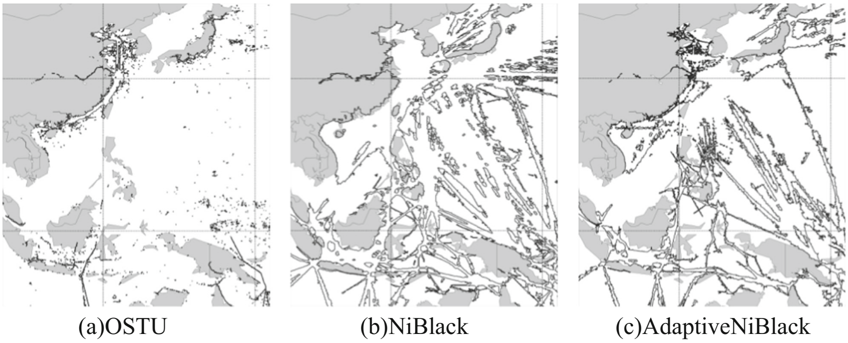


Fig. 4. Marine lane extraction result using same AIS data including five important ports Qingdao, Shanghai, Jakarta, Singapore and Sauron by OSTU, NiBlack and AdaptiveNiBlack.

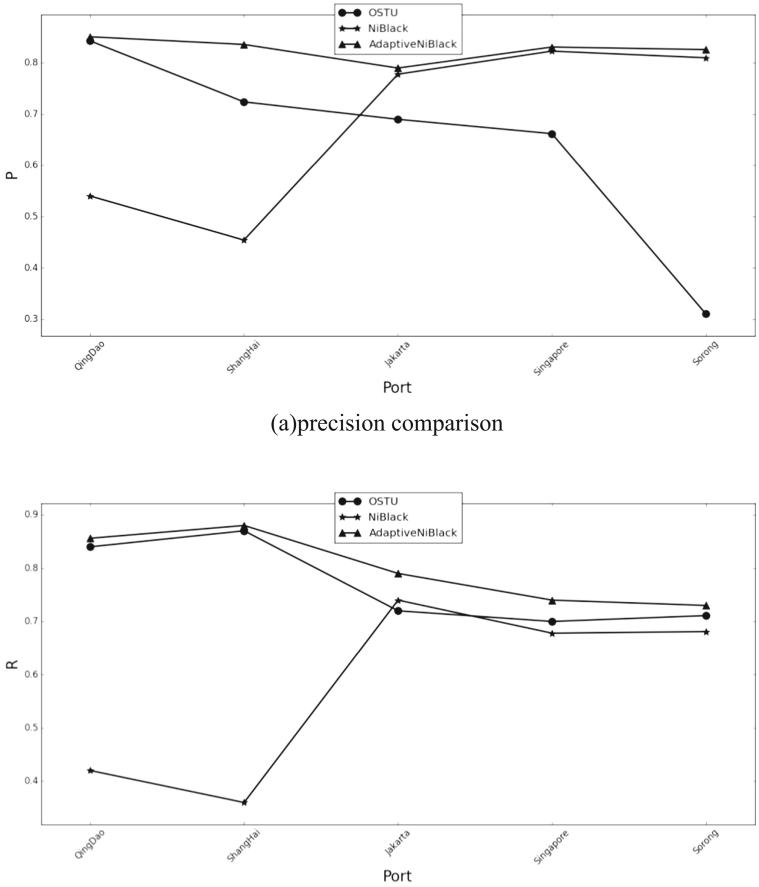
In this paper, the extracted marine lane grid set data is superimposed and compared with the marine lane grid set result extracted by the local single precision algorithm for qualitative evaluation. Two commonly evaluation indicators, namely precision and recall rate, were used to evaluate the experimental results.

$$precision = \frac{grid_{ext} \cap grid_{std}}{grid_{ext}} \tag{2}$$

$$recall = \frac{grid_{ext} \cap grid_{std}}{grid_{std}} \tag{3}$$

The precision and recall indicators calculated as in Eqs. (2) and (3) above, where $grid_{std}$ represents the standard marine lane grid set, $grid_{ext}$ indicates the extraction marine lane grid set.

Quantitative evaluated and analyzed of five important ports in Qingdao, Shanghai, Jakarta, Singapore and Sauron respectively, it can be found in Qingdao, Shanghai and others offshore areas $P_{AdaptiveNiBlack} > P_{OSTU} > P_{NiBlack}$, $R_{AdaptiveNiBlack} > R_{OSTU} > R_{NiBlack}$, in Jakarta, Singapore, Sauron, etc. In the far-sea area, $P_{AdaptiveNiBlack} > P_{NiBlack} > P_{OSTU}$, $R_{AdaptiveNiBlack} > R_{NiBlack} > R_{OSTU}$. Overall, the AdaptiveNiBlack method can adaptively extract more accurate marine lane results in the offshore and near shore areas simultaneously (Fig. 5).

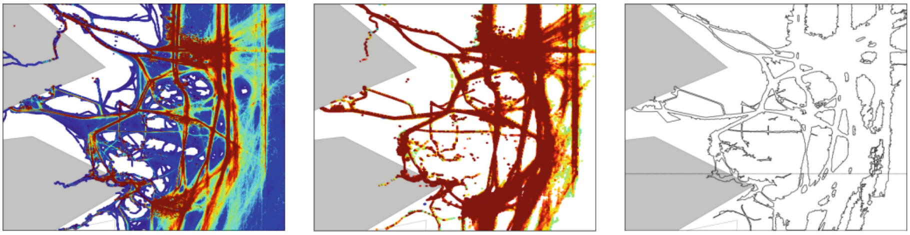


(a)precision comparison

(b) recall comparison

Fig. 5. The precision and recall in the five ports Qingdao, Shanghai, Jakarta, Singapore and Sauron by OSTU, NiBlack and AdaptiveNiBlack.

The AdaptiveNiBlack algorithm consists of four parameters {$\mathrm{dsy}_{\mathrm{max}}$, $\mathrm{dsy}_{\mathrm{min}}$, $\mathrm{dsy}_{\mathrm{m}}$, wsize}, as shown in Fig. 6. Figure 6(a) is a trajectory grid result after preprocessing in the East China Sea region, and Fig. 6(b) is a parallel merge filter and smoothed result when the parameter value is {200, 5, 20, 5 * 5}. And Fig. 6(c) is the marine lane results used extraction algorithm to extract.



(a) (b) (c)

Fig. 6. Marine lane result of east sea of China, (a) is density plot after grid, (b) is density plot after AdaptiveNiBlack filter and once median filter, (c) is marine lane result used CirAlphaShape.

In this paper, we use the weighted optimization method combined with professional experience to find the optimal combination of multiple parameters. As shown in Table 3, when the parameter value is $\{dsy_{max}, dsy_{min}, dsy_{m}, wsize\} = \{200, 5, 20, 5 * 5\}$, the precision rate of 89.7 and the recall rate of 89.3 can be achieved in the East China Sea region.

Table 3. Quantitative analysis evaluation table.

Number	Precision	Recall	Parameter list
1	63.4	48.3	$dsy_{max} = 100$, $dsy_{min} = 5$ $dsy_{m} = /$, wsize = 5 * 5
2	78.1	56.5	$dsy_{max} = 200$, $dsy_{min} = 5$ $dsy_{m} = /$, wsize = 5 * 5
3	85.5	82.4	$dsy_{max} = 100$, $dsy_{min} = 5$ $dsy_{m} = 20$, wsize = 5 * 5
4	89.7	89.3	$dsy_{max} = 200$, $dsy_{min} = 5$ $dsy_{m} = 20$, wsize = 5 * 5
5	83.2	86.8	$dsy_{max} = 200$, $dsy_{min} = 5$ $dsy_{m} = 20$, wsize = 7 * 7
6	79.7	83.8	$dsy_{max} = 200$, $dsy_{min} = 0$ $dsy_{m} = 20$, wsize = 5 * 5
7	76.7	80.1	$dsy_{max} = 200$, $dsy_{min} = 5$ $dsy_{m} = 10$, wsize = 5 * 5

In the parameter list of Table 3, dsy_{max} indicates the maximum density value of grid, which is to limit the threshold density value be wrong when the grid with the maximum density, that is, if the grid density value is larger than the dsy_{max}, it is set to dsy_{max}. Dsy_{min} represents the minimum density value for preliminary filtering of a large number of grids with extremely small density values, if the grid density is less than this dsy_{min}, it is realized a noise grid and set to 0. Dsy_{m} represents the merging judgement density value, dsy_{m} determines the merging depth of the QuadTree's nodes. *Wsize* represents the size of dynamic sliding window. By using multiple different sets of parameters to perform experiments, the effectiveness of the precision adaptive parallel extraction algorithm is verified, and has high accuracy and completeness.

7 Conclusion

Based on the data structure of QuadTree, this paper has implemented a parallelized method of precision adaptive extraction of marine lane, which effectively solved the problem that the vessel big data were affected by the environment and the sampling data density is large that resulting in poor marine lane results. The algorithm preprocesses the acquired vessel trajectory big data, and clusters the simplified trajectory data into grid data based on GeoHash coding method, and uses median filtering to smooth the grid result and eliminates discrete points. A parallel merge filtering algorithm has been proposed and implemented based on the QuadTree data structure and Spark technology parallelization. Finally, based on the Delaunay triangulation method, the marine lane results with different precisions have been effectively extracted. The experimental results have shown that the method can automatically identify the marine lane with different fineness and effectively extract the marine lane with different precision.

At the same time, there is still a lot of content that needs to be further improved: (1) This paper only extracts the geometric data of the channel, and other important information such as water depth and weight limit existed in the actual channel remains to be further studied. (2) This paper extracts marine lane of the main channel in a large range, due to the lack of strengthen processing for minor channel caused by minor channel losing, so how to strengthen the minor channel information and join our algorithm is the next research aims.

Acknowledgements. This work is supported by Beijing Natural Science Foundation No. 4172018, National Natural Science Foundation of China No. 61832004, No. 61672042, and University Cooperation Projects Foundation of CETC Ocean Corp.

References

1. Edelsbrunner, H., Kirkpatrick, D., Seidel, R.: On the shape of a set of points in the plane. IEEE Trans. Inf. Theory **29**(4), 551–559 (1983)
2. Brown, K.Q.: Voronoi diagrams from convex hulls. Inf. Process. Lett. **9**, 223–228 (1979)
3. Zhang, W., et al.: An advanced method for detecting possible near miss ship collisions from AIS data. Ocean Eng. **124**(1), 141–156 (2016)
4. Arguedas, V.F., Pallotta, G., Vespe, M.: Automatic generation of geographical networks for maritime traffic surveillance. In: International Conference on Information Fusion, pp. 1–8. IEEE (2014)
5. He, W., et al.: An Internet of Things approach for extracting featured data using AIS database: an application based on the viewpoint of connected ships. Symmetry **9**(9), 186 (2017)
6. Wu, L., et al.: Mapping global shipping density from AIS data. J. Navig. **70**(1), 67–81 (2017)
7. Arguedas, V.F., Pallotta, G., Vespe, M.: Maritime traffic networks: from historical positioning data to unsupervised maritime traffic monitoring. IEEE Trans. Intell. Transp. Syst. **PP**(99), 1–11 (2017)

8. Etienne, L., Devogele, T., Bouju, A.: Spatio-temporal trajectory analysis of mobile objects following the same itinerary. In: Advances in Geo (2010)
9. Vespe, M., Greidanus, H., Alvarez, M.A.: The declining impact of piracy on maritime transport in the Indian Ocean: statistical analysis of 5-year vessel tracking data. Mar. Policy **59**, 9–15 (2015)
10. Pallotta, G., Vespe, M., Bryan, K.: Traffic route extraction and anomaly detection from AIS data. In: COST MOVE Workshop on Moving Objects at Sea (2013)
11. Wang, X., Liu, X., Liu, B., et al.: Vessel route anomaly detection with Hadoop MapReduce. In: IEEE International Conference on Big Data. pp. 25–30. IEEE (2015)
12. Ahmed, M., Karagiorgou, S., Pfoser, D., et al.: A comparison and evaluation of map construction algorithms using vehicle tracking data. Geoinformatica **19**(3), 601–632 (2015)
13. Wang, J., Rui, X., Song, X., et al.: A novel approach for generating routable road maps from vehicle GPS traces. Int. J. Geogr. Inf. Syst. **29**(1), 69–91 (2015)
14. Wang, Y., Zhu, Y., He, Z., Yue, Y., Li, Q.: Challenges and opportunities in exploiting large-scale GPS probe data. Technical report. HPL-2011-109, HP Laboratories (2011)
15. Broach, J., Mcneil, N.W., Dill, J.: Travel mode imputation using GPS and accelerometer data from a multi-day travel survey. In: Transportation Research Board 93rd Annual Meeting (2014)
16. Van Winden, K., Biljecki, F., Van der Spek, S.: Automatic update of road attributes by mining GPS tracks. Trans. GIS **20**(5), 664–683 (2016)
17. Costa, G.H.R., Baldo, F.: Generation of road maps from trajectories collected with smartphone - a method based on Genetic Algorithm. Appl. Soft Comput. **37**, 799–808 (2015)
18. Park, S., Bang, Y., Yu, K.: Techniques for updating pedestrian network data including facilities and obstructions information for transportation of vulnerable people. Sensors **15**(9), 24466–24486 (2015)
19. Hu, Y., Gao, S., Janowicz, K., et al.: Extracting and understanding urban areas of interest using geotagged photos. Comput. Environ. Urban Syst. **54**, 240–254 (2015)
20. Merry, B., Gain, J., Marais, P.: Moving least-squares reconstruction of large models with GPUs. IEEE Trans. Vis. Comput. Graph. **20**(2), 249–261 (2014)
21. Mistry, S., Niranjan, U.N., Gopi, M.: Puzzhull: cavity and protrusion hierarchy to fit conformal polygons. Comput.-Aided Des. **46**(1), 233–238 (2014)
22. Peethambaran, J., Muthuganapathy, R.: A non-parametric approach to shape reconstruction from planar point sets through Delaunay filtering. Comput.-Aided Des. **62**(1), 164–175 (2015)
23. Cheng, X.: Parallel implementation of dynamic positive and negative feedback ACO with iterative MapReduce model. J. Inf. Comput. Sci. **10**(8), 2359–2370 (2013)
24. Aghabozorgi, S., Saybani, M.R., Teh, A., et al.: Incremental clustering of time-series by fuzzy clustering. J. Inf. Sci. Eng. **28**(4), 671–688 (2012)
25. Li, J., Chen, W., Li, M., et al.: The algorithm of ship rule path extraction based on the grid heat value. J. Comput. Res. Dev. **55**(5), 908–919 (2018)
26. Yang, W., Ai, T.: The extraction of road boundary from crowdsourcing trajectory using constrained Delaunay triangulation. Acta Geodaetica et Cartographica Sinica **46**(2), 237–245 (2017)
27. Kuntzsch, C., Sester, M., Brenner, C.: Generative models for road network reconstruction. Int. J. Geogr. Inf. Sci. **30**(5), 1012–1039 (2016)
28. Jiang, Y., Li, X., Li, X., et al.: Geometrical characteristics extraction and accuracy analysis of road network based on vehicle trajectory data. J. Geo-Inf. Sci. **14**(2), 165–170 (2012)