

文章编号:1671-1637(2020)04-0227-10

基于在线有向无环图的船舶轨迹压缩算法

张远强^{1,2,3}, 史国友¹, 李松²

(1. 大连海事大学 航海学院, 辽宁 大连 116026; 2. 宁波大学 海运学院, 浙江 宁波 315211;
3. 塔斯马尼亚大学 澳大利亚海事学院, 塔斯马尼亚 朗塞斯顿 TAS 7250)

摘要:为了解决船舶轨迹数据的压缩问题,提出了一种船舶轨迹在线压缩算法;使用多次滑动推算船位判断方法清洗船舶轨迹,使用在线有向无环图在干净轨迹上建立压缩路径树并输出采样点;为了提高轨迹队列和路径树在内存中的查询速度,使用哈希表对其进行管理;为了验证提出算法的效果,比较了真实船舶自动识别系统数据与方向保留算法、道格拉斯-普克算法的压缩时间和误差,采用可视化方法分析了原始轨迹、清洗轨迹和压缩轨迹。试验结果表明:在压缩时间方面,方向保留算法和道格拉斯-普克算法的压缩时间分别约为提出算法的1.1、1.3倍,说明提出的算法比其他2种算法的处理时间更短;提出的算法在压缩过程中保留了时间信息,平均同步欧氏距离误差在任何压缩率下都能保持在10 m以下,最大同步欧氏距离误差在压缩率为1%时仅有127 m,而其他2种算法的平均同步欧氏距离误差和最大同步欧氏距离误差不受控制,会随机变化;在垂直距离误差方面,提出的算法与道格拉斯-普克算法在压缩率不小于5%的条件下,都能保证垂直距离误差小于20 m,而方向保留算法的垂直距离误差会随机变化;在显示效果方面,提出的算法能有效清除轨迹噪声点,压缩轨迹能够较好地代表原始轨迹的宏观交通流情况。可见,提出的算法能更高效地保留原始轨迹的形状和时间信息。

关键词:船舶自动识别系统;船舶轨迹;轨迹压缩;压缩路径树;压缩率;平均同步欧氏距离误差
中图分类号:U675.7 **文献标志码:**A **DOI:**10.19818/j.cnki.1671-1637.2020.04.019

Compression algorithm of ship trajectory based on online directed acyclic graph

ZHANG Yuan-qiang^{1,2,3}, SHI Guo-you¹, LI Song²

(1. Navigation College, Dalian Maritime University, Dalian 116026, Liaoning, China;
2. Faculty of Maritime and Transportation, Ningbo University, Ningbo 315211, Zhejiang, China;
3. Australian Maritime College, University of Tasmania, Launceston TAS 7250, Tasmania, Australia)

Abstract: To solve the problem of ship trajectory data compression, an online ship trajectory compression algorithm was proposed. An estimation method of multiple sliding reckoning the ship position was used to clean the ship trajectory, and the online directed acyclic graph was used to build the compression path tree on the clean trajectory and to output the sampling points. The Hash table was used to manage the trajectory queue and path tree in memory in order to improve

收稿日期:2020-02-19

基金项目:国家自然科学基金项目(51579025)

作者简介:张远强(1982-),男,贵州贵阳人,大连海事大学工学博士研究生,从事船舶导航与监控系统研究。

导师简介:史国友(1968-),男,安徽桐城人,大连海事大学教授,工学博士。

引用格式:张远强,史国友,李松.基于在线有向无环图的船舶轨迹压缩算法[J].交通运输工程学报,2020,20(4):227-236.

Citation: ZHANG Yuan-qiang, SHI Guo-you, LI Song. Compression algorithm of ship trajectory based on online directed acyclic graph[J]. Journal of Traffic and Transportation Engineering, 2020, 21(4): 227-236.

their query speeds. To verify the effect of the proposed algorithm, the compression times and errors of real ship automatic identification system, direction-preserving algorithm and Douglas-Peucker algorithm were compared, and the original trajectory, clean trajectory and compression trajectory were analyzed by the visual method. Experiment result shows that in terms of the compression time, the compression times of direction-preserving algorithm and Douglas-Puck algorithm are about 1.1 and 1.3 times of the proposed algorithm, respectively, indicating that the execution time of the proposed algorithm is shorter than the other two algorithms. The proposed algorithm retains the time information in the compression process, and the average synchronized Euclidean distance error is less than 10 m at any compression rate. The maximum synchronized Euclidean distance error is only 127 m when the compression ratio is 1%, while the average synchronized Euclidean distance errors and the maximum synchronized Euclidean distance errors of the other two algorithms can not be controlled, and change randomly. In terms of perpendicular distance error, both the proposed algorithm and Douglas-Puck algorithm can guarantee that the perpendicular distance errors less than 20 m when the compression rate is not less than 5%, while the perpendicular distance error of direction-preserving algorithm changes randomly. In terms of display effect, the proposed algorithm can effectively remove the noise points of trajectory, and the compression trajectory can better represent the macroscopic traffic flow of original trajectory. Therefore, the proposed algorithm can preserve the shape and time informations of original trajectory more efficiently. 1 tab, 15 figs, 30 refs.

Key words: ship automatic identification system; ship trajectory; trajectory compression; compression path tree; compression ratio; average synchronized Euclidean distance error

Author resumes: ZHANG Yuan-qiang, male, doctoral student, zhangyuanqiang2016@163.com; SHI Guo-you(1968-), male, professor, PhD, sgydmu@dlmu.edu.cn.

Foundation item: National Natural Science Foundation of China (51579025)

0 引 言

船舶自动识别系统(Automatic Identification System, AIS)能通过电子数据交换实现船舶间的自动识别和定位,安装有 AIS 设备的船舶根据自己的航向和航速变化情况以 2 s 到 3 min 的时间间隔自动播报本船动态信息^[1]。AIS 设备已广泛的安装在船上和岸上,数据可通过船站、岸站和卫星在全球进行传输,这产生了大量的以时序方式表示的空间轨迹点。这些时空数据点包含了大量的海上交通模式信息,在进行如海事交通模式识别^[2-4]、航线计划^[5]、航道设计^[6]、船舶异常检测^[7]等 AIS 数据的研究中有重要意义。但由于 AIS 轨迹的数据量大且接收时间长会需要大量的存储空间、造成高通信负荷和高处理耗时方面的问题。为了解决这些问题又不会影响 AIS 轨迹的正常使用,使用轨迹压缩算法删除 AIS 中的大量冗余数据是较好的选择。

为了解决 AIS 轨迹的压缩问题,Zhao 等^[8]基

于道格拉斯-普克(Douglas-Peucker, DP)算法实现了 AIS 轨迹的压缩;Zhu 等^[9]在 DP 算法的基础上综合考虑距离、速度和航向变化对轨迹变化的影响权重,使用综合误差阈值决定轨迹是否保留;张树凯等^[10-11]在 DP 算法上试验验证了不同距离阈值对压缩率的影响,并使用 AIS 数据统计的船舶邻域来决定 DP 算法中距离阈值的大小;Vries 等^[12]在误差边界决策方案的基础上提出了 2 阶段线性分割的压缩方法,其基本思想是使用欧式距离分别对原始轨迹进行速度压缩和位置压缩;Sánchez-Heres^[13]使用相同路径计划算法压缩 AIS 轨迹,该算法通过识别船舶的停止、直航和转向行为,分段对轨迹进行压缩;Singh 等^[14]对 4 种典型的离线压缩算法在船舶轨迹上的压缩误差和执行时间进行了比较,结果表明 DP 算法的压缩误差最小,扫描-选择-移动算法的执行时间最快。上述算法都是离线压缩算法,需要扫描整个航线的数据,会耗费大量的数据输入输出和排序时间,对 AIS 数据实时性要求较高,轨迹持续时间长的应用不

适合。根据轨迹的压缩是否需要完整的轨迹数据,可把压缩算法分为离线压缩和在线压缩2种^[15-16]。在线 AIS 轨迹压缩研究方面,高邈等^[17]使用滑动窗口实现 AIS 轨迹的在线压缩,此算法虽然能得到较高的压缩率,但是使用的是轨迹点与压缩轨迹之间的垂直距离作为误差进行判断,没有考虑到原始轨迹时间信息的保留。为了使压缩轨迹能够保留原始轨迹的时间信息,Gao 等^[18]使用原始点与压缩轨迹之间的时间距离作为误差进行判断,但该方法只考虑采样点的选择不会超出误差阈值的要求,而没有考虑到采样点数量和总体压缩误差的最小化。

对于轨迹压缩的研究,主要有两大类,一类是基于距离压缩的位置保留算法,另一类是基于方向压缩的方向保留算法。位置保留算法研究较多,典型的算法除 DP 算法外,还包括滑动窗口和自底向上算法^[19]、开窗时间比算法^[20]、局部最优开窗时间比算法^[21]、空间质量简化启发式算法^[22]和基于有向无环图的在线轨迹简化(Directed Acyclic Graph Based Online Trajectory Simplification, DOTS)算法^[23]等。各方法的区别主要在于如何快速的寻找最佳采样点得到最大压缩率和最小压缩误差。典型的方向保留算法有 Thresholds^[24],方向保留轨迹简化(Direction-Preserving Trajectory Simplification, DPTS)算法^[25]和方向保留轨迹简化加(Direction-Preserving Trajectory Simplification+, DPTS+)算法^[26]。

对于停止或低速行驶目标,由于方向不稳定,会导致方向保留算法不能有效地删除冗余点。AIS 数据中存在大量的停止轨迹,使用方向保留算法会导致大量的停止轨迹被保留,降低压缩率,因此,本文使用位置保留算法实现 AIS 轨迹的压缩。而选用 DOTS 算法对 AIS 轨迹进行压缩,主要是因为该算法与已有 AIS 轨迹压缩算法相比较,其在相同压缩率情况下能够尽量减少全局误差,简化轨迹考虑了时间信息的保留,且属于在线压缩方式,更为适合对 AIS 轨迹进行压缩。

1 DOTS 算法

DOTS 算法使用局部累积平方同步欧氏距离(Local Integral Square Synchronized Euclidean Distance, LISSSED)作为简化轨迹与原始轨迹之间的误差,LISSSED 可直接通过起点与终点的位置坐标和时间得到,方便实现轨迹误差的在线计算。假

设原始轨迹用 T 表示; p_k 为 T 上的第 k 个点,坐标为 (x_k, y_k, t_k) , x_k, y_k 和 t_k 分别为 p_k 的横坐标、纵坐标和时间; T' 是 T 的一条简化轨迹; p'_k 是 p_k 在 T' 上的同步时间点, p'_k 与 p_k 之间的长度称为同步欧氏距离(Synchronized Euclidean Distance, SED),用 $d(p_k, p'_k)$ 表示,则 T 上点 p_i 与 p_j 间的 LISSSED 为两点间所有点的 SED 平方和,用 $L(p_i, p_j)$ 表示,计算公式为^[27]

$$L(p_i, p_j) = \sum_{i \leq k \leq j} d^2(p_k, p'_k) = (c_1^2 + c_3^2)(j - i - 1) + (c_2^2 + c_4^2)(S_{t_{j-1}}^2 - S_{t_i}^2) + 2(c_1 c_2 + c_3 c_4)(S_{t_{j-1}} - S_{t_i}) + (S_{x_{j-1}}^2 + S_{x_i}^2) + (S_{y_{j-1}}^2 + S_{y_i}^2) - 2c_1(S_{x_{j-1}} + S_{x_i}) + 2c_3(S_{y_{j-1}} + S_{y_i}) - 2c_2(S_{x_{j-1}t_{j-1}} + S_{x_i t_i}) + 2c_4(S_{y_{j-1}t_{j-1}} + S_{y_i t_i}) \quad (1)$$

其中

$$c_1 = \frac{x_i t_j - x_j t_i}{t_j - t_i}$$

$$c_2 = \frac{x_j - x_i}{t_j - t_i}$$

$$c_3 = \frac{y_i t_j - y_j t_i}{t_j - t_i}$$

$$c_4 = \frac{y_j - y_i}{t_j - t_i}$$

$$S_{x_i} = \sum_{k=1}^i x_k$$

$$S_{y_i} = \sum_{k=1}^i y_k$$

$$S_{t_i} = \sum_{k=1}^i t_k$$

$$S_{x_i^2} = \sum_{k=1}^i x_k^2$$

$$S_{y_i^2} = \sum_{k=1}^i y_k^2$$

$$S_{t_i^2} = \sum_{k=1}^i t_k^2$$

$$S_{x_i t_i} = \sum_{k=1}^i x_k t_k$$

$$S_{y_i t_i} = \sum_{k=1}^i y_k t_k$$

当压缩误差阈值为 ϵ 时,如果 $L(p_i, p_j) < \epsilon$,则可用 p_i 与 p_j 之间的直线段代表原始轨迹。用图 1 对 LISSSED 的含义进行解释,图中有 6 个点 $p_1 \sim p_6$,其连线为原始轨迹, $p'_1 \sim p'_6$ 为其在简化轨迹上对应的同步时间位置, p_1 与 p'_1 、 p_6 与 p'_6 为同一

点,则 p_k 与 p'_k 之间的长度为 $d(p_k, p'_k)$,如图 1 中对应点之间的虚线,有 $L(p_i, p_j) = \sum_{1 \leq k \leq 6} d^2(p_k, p'_k)$ 。

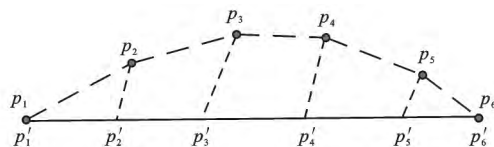


图 1 压缩误差尺度

Fig. 1 Compression error measure

在采样点选取方面,DOTS 使用在线有向无环图建立一个具有 $K+1$ 层的轨迹压缩路径树, $K=1,2,3,\dots,V_K$ 为第 K 层存储轨迹点的层。图 2 有 11 个点,使用 DOTS 算法建立的压缩路径树见图 3,共有 5 个层, V_1 层只有一个点 p_1 。假设当前层为 V_{K+1} ,上一层为 V_K ,如果 $L(p_i, p_j) < \epsilon$,则 p_i 可作为 p_j 的备选母节点,如果有多个备选母节点,选择累积平方同步欧氏距离(Integral Square Synchronized Euclidean Distance, ISSSED)最小的点作为 p_j 的母节点,ISSSED 为 p_1 到 p_j 点的所有 LISSSED 之和。如 p_9 和 p_5 、 p_9 和 p_6 的 LISSSED 都能满足误差条件,但由于 (p_1, p_5, p_9) 组合比 (p_1, p_6, p_9) 组合的 ISSSED 误差要小,所以 p_9 选择 p_5 作为母节点。ISSSED 用 I 表示,为

$$I = \sum_{p_K \in T'} L(p_K, p_{K+1}) \quad (2)$$

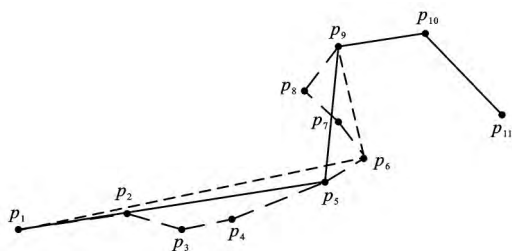


图 2 DOTS 算法形成的简化轨迹

Fig. 2 Simplification trajectory built by DOTS algorithm

在层结束插入条件方面,如果 p_i 与 p_j 满足 $d(p_i, p_j) > \eta \epsilon$,则将 p_i 插入到 R_K 队列中,其中 η 为误差倍数,可设置为 2, R_K 用于存储 V_K 层中满足 η 倍误差阈值的点,当 V_K 层中的所有点都被插入到 R_K 中时,则结束 V_{K+1} 层的插入。如图 3 中,在插入 V_2 层时,当发现 $d(p_1, p_7) > 2\epsilon$ 时,将 p_1 插入 R_1 中,而 V_1 层只有一个点 p_1 ,于是结束 V_2 层的插入。

从顶层到 V_K 层,如只有一个点与 V_{K+1} 层连接,则使用图 4 的解码和输出程序将该点作为最终压缩点输出,唯一与 V_{K+1} 层连接的点称为唯一活跃点。如图 3 中在结束 V_3 层插入时,输出 p_1 为采样点并删除 V_1 层。在结束 V_4 层插入时,输出 p_5 和

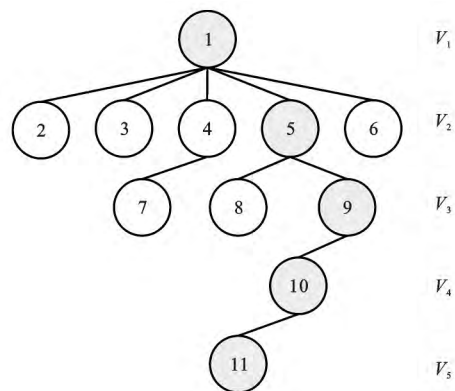


图 3 DOTS 算法形成的压缩路径树

Fig. 3 Compression path tree built by DOTS algorithm

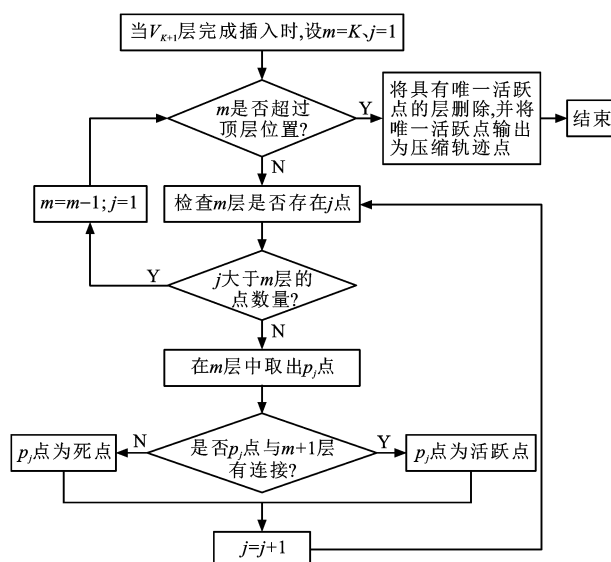


图 4 解码和输出流程

Fig. 4 Process of decoding and output

p_9 为采样点并删除 V_2 和 V_3 层。这样可保证内存不会一直增加。

DOTS 算法使用了一种增量式构建压缩路径树的方法实现轨迹的在线压缩, p_j 在选择母节点时,即保证与母节点的压缩误差在一个允许范围内,又保证母节点的选择具有全局压缩误差的最小值。

2 船舶轨迹压缩算法建立

DOTS 算法只给出单个轨迹的 V_{K+1} 层建立和最终输出点的处理方法,要运用到船舶的轨迹压缩上,还需要解决坐标转换、轨迹清洗和散乱的待压缩轨迹在内存中的管理 3 个问题,以形成一套完整的船舶轨迹压缩算法。

2.1 坐标转换

AIS 数据中的船位使用的是经纬度表示的地理坐标,为了能够使用 DOTS 算法,需要将地理坐标

转换为平面坐标,即

$$\begin{cases} R = \cos(\varphi)a/\sqrt{1-E^2\sin^2(\varphi)} \\ q = \text{Intan}\left(\frac{\pi}{2} + \frac{\varphi}{4}\right) - \frac{E}{2}\ln\left[\frac{1+E\sin(\varphi)}{1-E\sin(\varphi)}\right] \\ x = R\lambda \\ y = Rq \end{cases} \quad (3)$$

式中: φ, λ 分别为轨迹点的纬度和经度; x, y 分别为墨卡托横、纵坐标; R 为基准纬度圈半径; a 为地球椭球长轴半径; q 为等量纬度; E 为椭球第一偏心率。

2.2 轨迹清洗

由于使用不当、信号传输问题及 AIS 系统本身, AIS 数据会产生一些噪声^[28-29]。如果不删除噪声数据,将影响试验结果。为了保证异常轨迹点被删除的同时尽量保留正常的原始轨迹点,并且算法还应具有在线清洗的功能。本文设计一种多次滑动推算船位判断的方法对轨迹进行清洗。如图 5 所示,有 5 个轨迹点 $p_1 \sim p_5$, p_3 是待校验点。有 3 个滑动窗口 W_1, W_2 和 W_3 , 每个窗口都包含 3 个点。在 3 个窗口中,分别使用式(4)计算第 3 个点的推算位置。假如第 3 个点的推算位置和接收值之间的距离大于预设距离阈值,则 p_3 的噪声计数加 1,当 p_3 的噪声计数等于 3 时, p_3 可作为噪声被清除。对于整个轨迹的开始 2 个点和最后 2 个点,假如它们使噪声计数增加,则可作为噪声被删除。轨迹清洗算法至少需要收到 5 个轨迹点,当 p_3 判断为正常点后, p_1 输出给压缩算法。使用多次推算轨迹判断 p_3 是否为噪声点,目的是为了将部分转向或变速剧烈的轨迹点作为噪声去除。轨迹清洗算法考虑了船舶的速度变化对推算位置的影响,对于短时间内更新的轨迹,可以认为船舶是匀加速的运动

$$\hat{p}_{k+1} = A p_k \quad (4)$$

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \frac{\Delta t^2}{2} \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$p_k = (x_k, y_k, v_{x_k}, v_{y_k}, a_{x_k}, a_{y_k})^T$$

式中: \hat{p}_{k+1} 为轨迹的推算状态向量; A 为状态转移矩阵; Δt 为轨迹之间的间隔时间; p_k 为上一个轨迹的状态向量; x_k, y_k 分别为 k 点沿 x, y 轴的位置; v_{x_k}, v_{y_k} 分别为 k 点沿 x, y 轴的速度; a_{x_k}, a_{y_k} 分别为 k 点

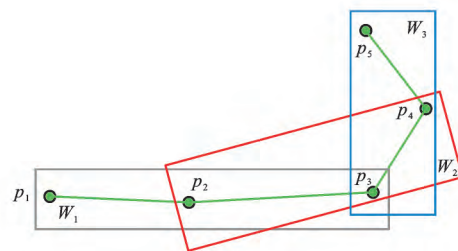


图5 多次滑动推算船位

Fig. 5 Dead reckoning by multiple sliding

沿 x, y 轴的加速度。

本文用于判断轨迹是否为噪声的距离阈值通过统计规律确定,图 6 为距离阈值与删除点数量之间的关系,可看出删除点的数量随距离阈值的变化规律是一个从剧烈变化到逐渐稳定的过程,当阈值在 30 m 以上时,删除点数量基本稳定,可判断大部分被删除点的推算位置与接收位置之间的距离远超 30 m,因此,本文将距离阈值设置为 30 m。

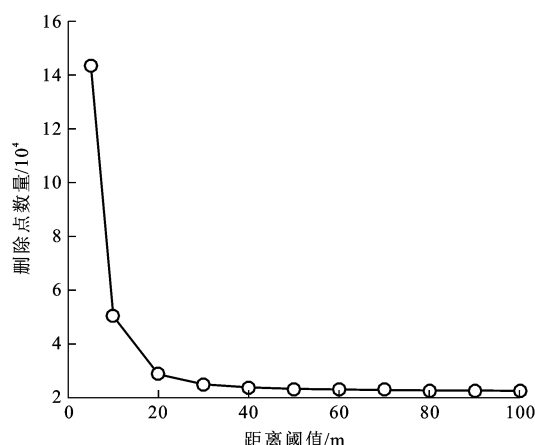


图6 距离阈值与删除点数量曲线

Fig. 6 Curve between distance threshold and deleted point number

2.3 DOTS 算法改进

由于原有 DOTS 算法只描述单条轨迹 V_{K+1} 层的建立,而实际接收到的 AIS 数据是按时间顺序排列的多条散乱船舶数据。为了适应船舶轨迹压缩,本文对原有算法的流程进行了适当修改,具体见图 7。当收到一个新点 p_j 时,在 V_K 层中循环的计算每个点到前一层的局部累积平方同步欧氏距离 $L(p_i, p_j)$ 和到第一个点的累积平方同步欧氏距离 I ,在满足条件 $L(p_i, p_j) < \epsilon$ 的点中选择 I 最小的点作为 p_j 的母节点,同时将 $L(p_i, p_j) > 2\epsilon$ 的点放入 R_K 中,如果 R_K 中包含 V_K 层中的所有点,则调用解码和输出程序算法,并结束 V_{K+1} 层的插入,否则等待新点的到来。与原算法流程不同的是原算法将母节点的选择放到 V_K 层循环后,这会对 V_K 层进行重

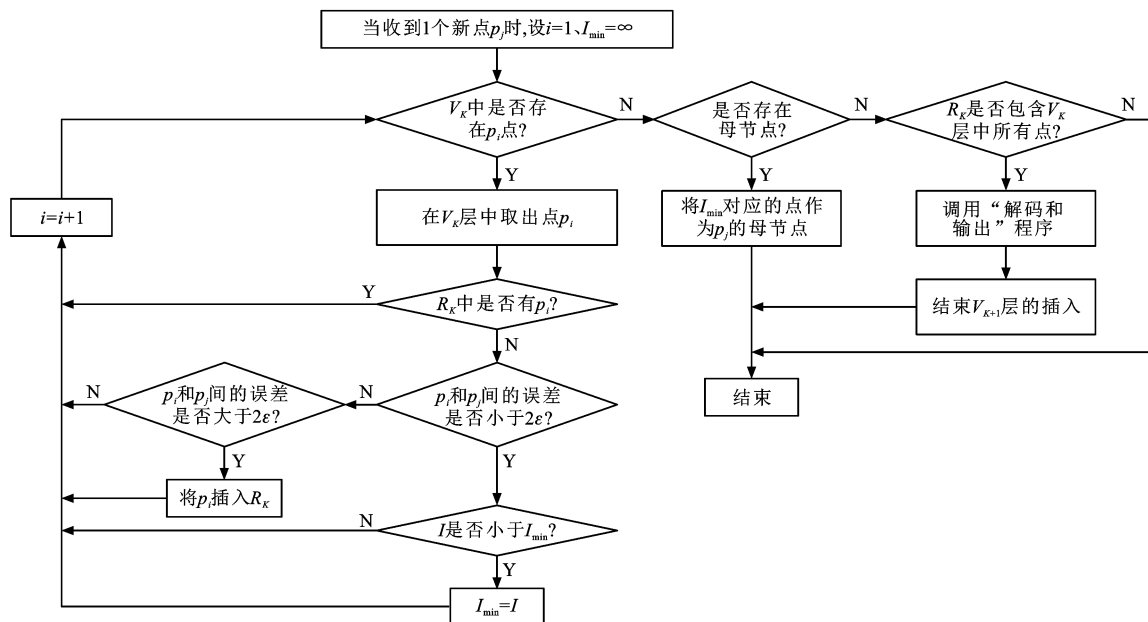


图 7 在线有向无环图流程

Fig. 7 Process of online directed acyclic graph

新扫描,而本文对 $L(p_i, p_j)$ 和 I 在每次循环中都进行计算和判断,则只需对 V_K 层做一次循环。

2.4 轨迹压缩处理流程

为了保证接收到的所有船舶轨迹都能被快速的处理,本文建立了图 8 所示的船舶轨迹压缩处理流程。在系统收到一个新的船舶轨迹点 p_o 后,会根据船舶的水上移动业务标识码 (Maritime Mobile Service Identify, MMSI) 将 p_o 放入对应的轨迹队

列 Q_N 中, Q_N 负责保存还未清洗的点; 然后调用轨迹清洗算法判断是否有干净点输出, 如有, 则使用 DOTS 算法将判断为干净的点 p_j 压缩到对应的船舶压缩路径树 B_N 中。轨迹队列和压缩路径树用一个哈希表管理, 哈希表使用 MMSI 作为关键值进行查询。因为 DOTS 算法在 V_{K+1} 层没有完成插入前, 会继续向后搜索, 但这些处于 ϵ 和 2ϵ 之间的新轨迹点又不能删除, 需要继续留在轨迹队列中, 因此, 需要有一个指针指示队列中当前的处理位置。当完成 V_{K+1} 层构建后, V_{K+2} 层需要从轨迹队列的开头开始处理。当完成 V_{K+1} 层的构建后, 将所有唯一活跃点输出到数据库。

轨迹压缩流程见图 9, 当接收到一个新轨迹点时, 将新收到的轨迹点转换为平面坐标, 然后根据 MMSI 在哈希表中查找是否存在新点对应的船舶, 如不存在则创建一个新的队列和树, 并将其插入到队列中。当轨迹队列中的点达到 5 个时, 使用轨迹清洗算法清洗轨迹, 如果是噪声点, 则继续等待接收新轨迹点。如果有干净点输出, 则使用在线有向无环图算法将该点插入到路径树中。插入后如有采样点输出, 则存入船舶轨迹管理数据库, 否则继续等待接收新轨迹点。

当一艘船舶的轨迹长时间未收到或程序结束时, 需要将未压缩的轨迹点压缩到路径树中, 并将路径树中各层的最终压缩点输出到数据库。方法是重复调用在线有向无环图算法将所有未压缩的轨迹点都插入到路径树中后, 然后根据最后一个点的位置

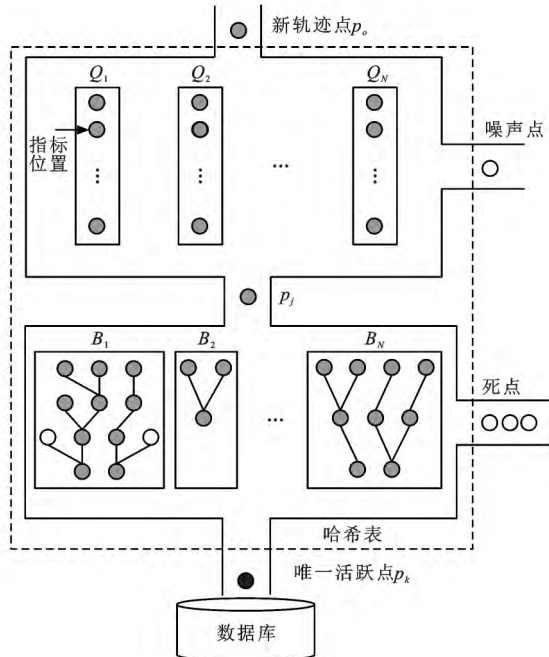


图 8 轨迹压缩流程

Fig. 8 Process of trajectory compression

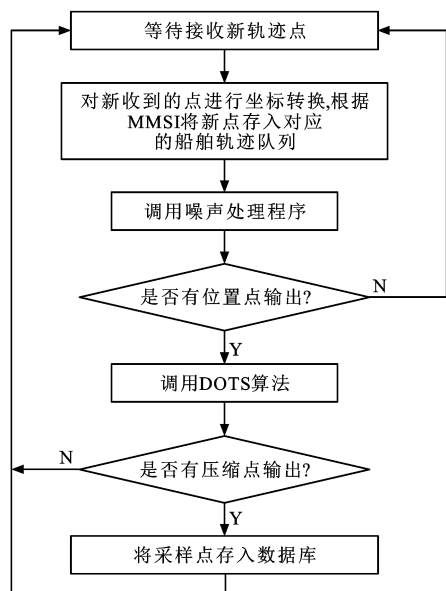


图9 船舶轨迹简化算法流程

Fig. 9 Simplification algorithm process of ship trajectory

反向查找连接路径。

3 试验结果分析

为了检验本文提出的船舶轨迹压缩算法的效果,使用宁波港接收到的共10 d的数据进行压缩,地理范围从(29°32' N, 121°44' E)到(30°02' N, 122°29' E)。与典型的DP算法^[30]和DPTS算法在相同压缩率下的压缩时间、平均SED误差(后面简称“平均误差”)和最大SED误差(后面简称“最大误差”)3个方面进行比较,还对3种算法在相同垂直误差下的压缩率进行了比较。为了避免误差分析的不准确,算法之间用于误差比较的轨迹是经过清洗以后的轨迹。另外,用可视化的方法,呈现压缩轨迹与原始轨迹在宏观交通流上的显示效果。为了保证算法可以在大多数电脑上运行,所有的试验均运行在同一台计算机上。

3.1 误差定义

在进行试验前,需给出如下误差定义。

定义1:设压缩率即压缩轨迹点数量与原始轨迹点数量之间的比值。

定义2:压缩时间即在相同压缩率下压缩相同轨迹需要的时间。

定义3:设 n 为压缩轨迹点数量,平均误差为所有压缩轨迹SED误差的平均值,用 \bar{e} 表示,即

$$\bar{e} = \frac{1}{n} \sum_{k=1}^n L(p_k, p'_k)$$

定义4:最大误差为所有压缩轨迹SED误差的

最大值,用 e_m 表示,即

$$e_m = \max_{1 \leq k \leq n} L(p_k, p'_k)$$

3.2 试验结果分析

图10为3种算法在压缩率均为10%的情况下的压缩时间比较,可以看出:DPTS算法压缩时间约为本文算法的1.1倍,DP算法约为本文算法的1.3倍,可见本算法比其他2种算法的处理时间更短;本算法的处理时间随数据量呈线性变化,表明算法执行过程稳定;在其他压缩率采样条件下的处理时间本文也进行过统计,DPTS和DP算法的压缩时间均约为本文算法的1.1和1.3倍。

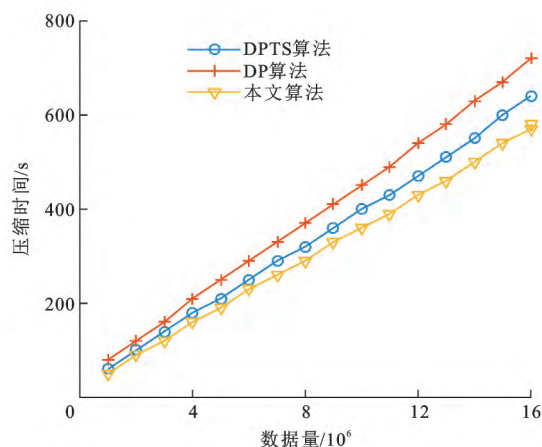


图10 压缩时间比较

Fig. 10 Comparison of compression times

图11为3种算法的平均误差比较,只显示平均误差小于50 m的情况,可以看出:与DP算法比较,在压缩率大于20%时,本算法与DP算法的平均误差都能小于5 m,在压缩率小于20%时,随着压缩率的减小,DP算法的平均误差增加很快;与DPTS算法比较,在压缩率大于80%时,本算法与DPTS算法的平均误差也能小于5 m,在压缩率小于80%时,

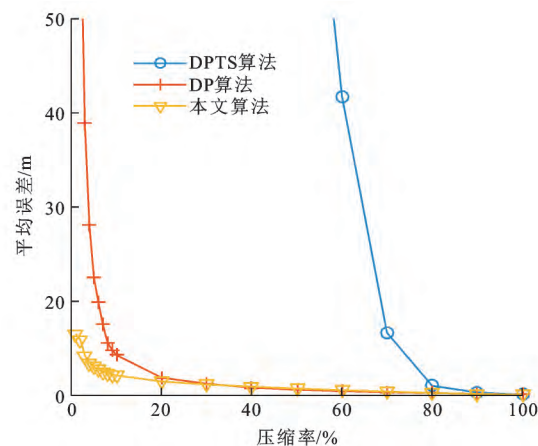


图11 平均误差比较

Fig. 11 Comparison of average errors

随着压缩率的减小, DPTS 算法的平均误差增加很快; 在任何压缩率下, 本算法的平均误差都能保证在 10 m 以下。

图 12 为最大误差比较, 可以看出: 本文算法的最大误差可以受到限制, 而 DPTS 和 DP 算法的最大误差不受限制。

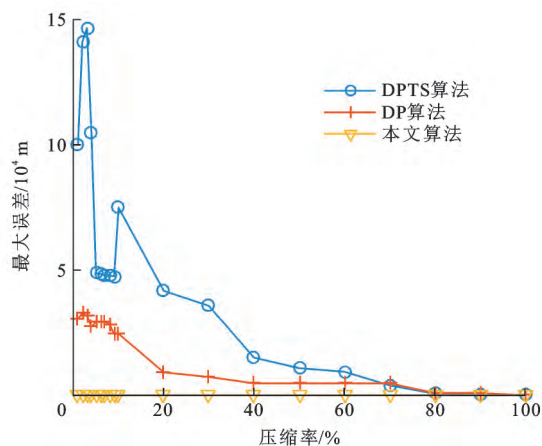


图 12 最大误差比较

Fig. 12 Comparison of maximum errors

表 1 为 3 种算法在部分压缩率情况下的最大误差, 可以看出本文算法可以很好地保留轨迹的时间信息。

表 1 不同压缩率的最大误差

Tab. 1 Maximum errors of different compression ratios

压缩率/%		1	10	30	60	90
最大误差/m	DPTS 算法	100 194	75 038	35 750	9 436	297
	DP 算法	30 523	24 614	7 323	4 795	866
	本文算法	127	25	20	17	15

图 13 为垂直距离误差比较, 垂直距离误差为原始轨迹点到简化轨迹垂直距离中的最大值。由图 13 可知: 在相同的垂直距离误差下, 本文算法与 DP 算法相当, 在 20 m 误差点, 本文算法对应的压缩率为 4%, DP 算法对应的压缩率为 5%, 即在压缩率不小于 5% 条件下, 本文算法和 DP 算法都能保证最大垂直距离误差小于 20 m; 与 DPTS 算法比较,

本文算法优势明显; 由于垂直距离可衡量简化轨迹和原始轨迹的形状相似度, 因此, 本文算法在保留原始轨迹的时间信息基础上, 也能如 DP 算法一样, 很好地保留原始轨迹的形状。

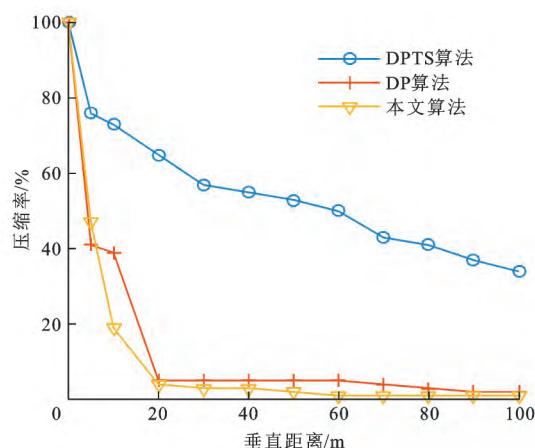


图 13 垂直距离误差比较

Fig. 13 Comparison of perpendicular distance errors

3.3 可视化分析

为了能够清晰显示轨迹清洗和压缩的效果, 使用一天的数据进行可视化分析。图 14 分别为原始轨迹、清洗轨迹和压缩轨迹; 由于存在船舶共用 MMSI、船位错误、时间错误等问题, 会造成 AIS 轨迹中存在噪声轨迹, 如果不加处理的将存在误差的轨迹按照时间顺序连接, 会呈现出图 14(a) 的效果, 错误的轨迹将重要的信息掩盖, 一些轨迹会穿越陆地, 致使正常的船舶历史航行记录无法被辨认; 图 14(b) 为经过轨迹清洗后的效果, 可看出经过清洗后的轨迹位于可航水域内, 能够反映宁波港水域的宏观交通流分布; 图 14(c) 是压缩率为 10% 的轨迹, 从宏观上看, 经过压缩后的轨迹与原始轨迹反映的宏观交通流趋势相同, 这会较大的提高轨迹数据处理的效率和缩短可视化的显示时间。

图 15 为使用 3 种算法对一段船舶航行轨迹进行压缩的效果, 原始轨迹有 520 个点, 经压缩后的最大误差都为 50 m, 使用 DOTS、DP 和 DPTS 算法压

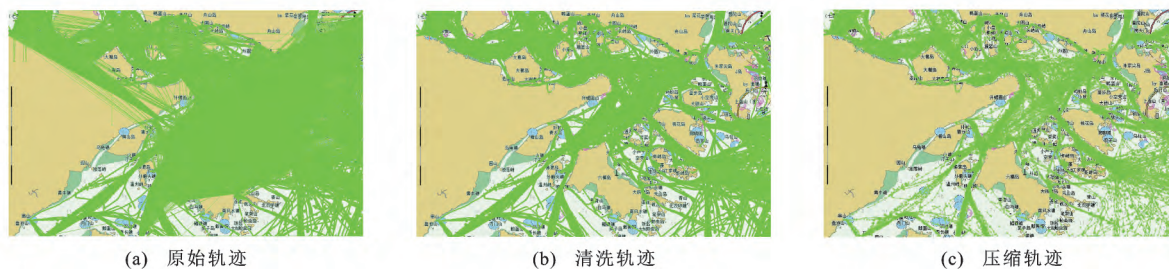


图 14 轨迹压缩可视化效果

Fig. 14 Visualization results of trajectory compression

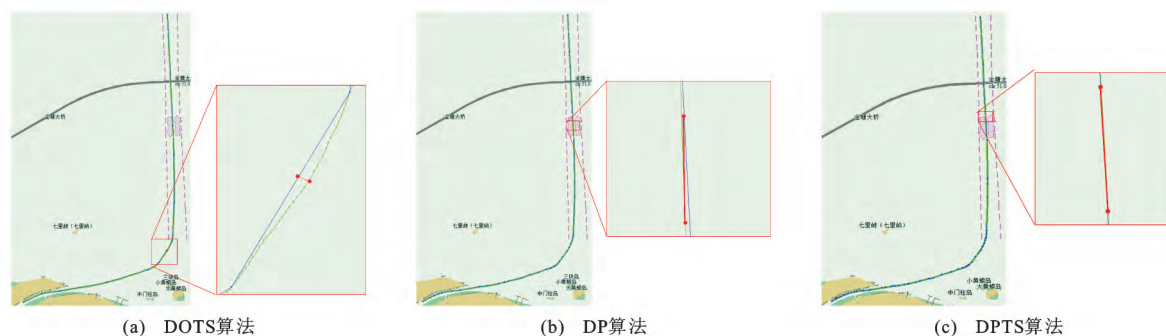


图 15 具有相同最大误差的压缩效果比较

Fig. 15 Comparison of compression effects with same maximum error

缩后的采样点数分别为 11、38 和 69 个;总体来看,3 种算法得到的简化轨迹与原始轨迹的形状基本相似,但从保留的采样点数比较,在相同最大误差下,DOTS 算法可以用最少的采样点数表示原始轨迹。

4 结 语

(1)本文设计的船舶轨迹在线压缩算法根据多次推算的船位判断轨迹点是否为噪声,可以避免将机动性过大的正常轨迹点作为噪声删除;使用在线有向无环图建立船舶轨迹压缩路径树并输出简化轨迹,在建立的轨迹压缩路径树中,点之间的连接保证简化轨迹的局部误差在一个限制范围内,全局误差为最小,且误差的计算同时考虑了位置信息和时间信息的保留。

(2)使用实际的 AIS 数据验证表明,本文的船舶轨迹在线压缩算法可以很好地删除原始轨迹中的噪声点,在处理时间上比方向保留算法和道格拉斯-普克算法更短,方向保留算法和道格拉斯-普克算法的处理时间约为本文算法的 1.1 倍和 1.3 倍,得到的简化轨迹能更好地保留原始轨迹的形状和时间信息。

(3)在试验中发现,基于方向保留的压缩算法不能很好地对船舶轨迹进行压缩的主要问题在于轨迹中存在大量的停止点,船舶在停止或低速航行时,方向极不稳定,会导致大量的停止点被保存。但方向保留算法对正常航行中的船舶轨迹保留效果更为理想,特别是船舶在改向幅度较大时,可保留重要的转向信息,因此,下一步的研究工作将是考虑在不影响压缩速度的情况下,如何结合方向保留算法和位置保留算法实现船舶轨迹的在线压缩。

参考文献:

References:

[1] HAMMOND T R, PETERS D J. Estimating AIS coverage

from received transmissions[J]. The Journal of Navigation, 2012, 65(3): 409-425.

[2] SHENG Pan, YIN Jing-bo. Extracting shipping route patterns by trajectory clustering model based on automatic identification system data[J]. Sustainability, 2018, 10(7): 1-13.

[3] 牟军敏,陈鹏飞,贺益雄,等. 船舶 AIS-轨迹快速自适应谱聚类算法[J]. 哈尔滨工程大学学报, 2018, 39(3): 428-432.
MOU Jun-min, CHEN Peng-fei, HE Yi-xiong, et al. Fast self-tuning spectral clustering algorithm for AIS ship trajectory[J]. Journal of Harbin Engineering University, 2018, 39(3): 428-432. (in Chinese)

[4] ZHAO Liang-bin, SHI Guo-you. A trajectory clustering method based on Douglas-Peucker compression and density for marine traffic pattern recognition[J]. Ocean Engineering, 2019, 172: 456-467.

[5] ZHANG Shu-kai, SHI Guo-you, LIU Zheng-jiang, et al. Data-driven based automatic maritime routing from massive AIS trajectories in the face of disparity[J]. Ocean Engineering, 2018, 155: 240-250.

[6] LI Song, ZHOU Jiang-hua, ZHANG Yuan-qiang. Research of vessel traffic safety in ship routeing precautionary areas based on navigational traffic conflict technique [J]. The Journal of Navigation, 2015, 68(3): 589-601.

[7] PALLOTTA G, VESPE M, BRYAN K. Vessel pattern knowledge discovery from AIS data: a framework for anomaly detection and route prediction[J]. Entropy, 2013, 15(6): 2218-2245.

[8] ZHAO Liang-bin, SHI Guo-you. A method for simplifying ship trajectory based on improved Douglas-Peucker algorithm[J]. Ocean Engineering, 2018, 166: 37-46.

[9] ZHU Fei-xiang, MIAO Li-ming, LIU Wen. Research on vessel trajectory multi-dimensional compression algorithm based on Douglas-Peucker theory[J]. Applied Mechanics and Materials, 2014, 694: 59-62.

[10] 张树凯,刘正江,张显库,等. 基于 Douglas-Peucker 算法的船舶 AIS 航迹数据压缩[J]. 哈尔滨工程大学学报, 2015, 36(5): 595-599.

ZHANG Shu-kai, LIU Zheng-jiang, ZHANG Xian-ku, et al. A method for AIS track data compression based on Douglas-Peucker algorithm[J]. Journal of Harbin Engineering

- University, 2015, 36(5): 595-599. (in Chinese)
- [11] ZHANG Shu-kai, LIU Zheng-jiang, CAI Yao, et al. AIS trajectories simplification and threshold determination [J]. The Journal of Navigation, 2016, 69(4): 729-744.
- [12] VRIES G K D D, SOMEREN M V. Machine learning for vessel trajectories using compression, alignments and domain knowledge[J]. Expert Systems with Applications, 2012, 39(18): 13426-13439.
- [13] SÁNCHEZ-HERES L F. Simplification and event identification for AIS trajectories: the equivalent passage plan method[J]. The Journal of Navigation, 2019, 72: 307-320.
- [14] SINGH A K, AGGARWAL V, SAXENA P, et al. Performance analysis of trajectory compression algorithms on marine surveillance data[C]//IEEE. 2017 International Conference on Advances in Computing, Communications and Informatics. New York: IEEE, 2017: 1074-1079.
- [15] ZHENG Yu. Trajectory data mining: an overview[J]. ACM Transactions on Intelligent Systems and Technology, 2015, 6(3): 1-41.
- [16] SUN Peng-hui, XIA Shi-xiong, YUAN Guan, et al. An overview of moving object trajectory compression algorithms [J]. Mathematical Problems in Engineering, 2016, 2016: 1-13.
- [17] 高 邈, 史国友, 李伟峰. 改进的 Sliding Window 在线船舶 AIS 轨迹数据压缩算法[J]. 交通运输工程学报, 2018, 18(3): 218-227.
- GAO Miao, SHI Guo-you, LI Wei-feng. Online compression algorithm of AIS trajectory data based on improved sliding window[J]. Journal of Traffic and Transportation Engineering, 2018, 18(3): 218-227. (in Chinese)
- [18] GAO Miao, SHI Guo-you. Ship spatio temporal key feature point online extraction based on AIS multi-sensor data using an improved sliding window algorithm[J]. Sensors, 2019, 19: 1-17.
- [19] KEOGH E, CHU S, HART D, et al. An online algorithm for segmenting time series [C]//IEEE. 2001 IEEE International Conference on Data Mining. New York: IEEE, 2001: 289-296.
- [20] MERATNIA N, DE BY R A. Spatio temporal compression techniques for moving point objects[C]//Springer. Advances in Database Technology—EDBT 2004. Berlin: Springer, 2004: 765-782.
- [21] MENG Qing-bin, YU Xiao-qiang, YAO Chun-long, et al. Improvement of OPW-TR algorithm for compressing GPS trajectory data[J]. Journal of Information Processing Systems, 2017, 13(3): 533-545.
- [22] MUCKELL J, OLSEN P W J, HWANG J H, et al. Compression of trajectory data: a comprehensive evaluation and new approach[J]. Geoinformatica, 2014, 18(3): 435-460.
- [23] CAO Wei-quan, LI Yun-zhao. DOTS: an online and near-optimal trajectory simplification algorithm[J]. Journal of Systems and Software, 2017, 126: 34-44.
- [24] POTAMIAS M, PATROUMPAS K, SELLIS T. Sampling trajectory streams with spatiotemporal criteria[C]//IEEE. 18th International Conference on Scientific and Statistical Database Management. New York: IEEE, 2006: 275-284.
- [25] LONG C, WONG R C W, JAGADISH H V. Direction-preserving trajectory simplification [J]. Proceedings of the VLDB Endowment, 2013, 6(10): 949-960.
- [26] DENG Ze, HAN Wei, WANG Li-zhe, et al. An efficient online direction-preserving compression approach for trajectory streaming data [J]. Future Generation Computer Systems, 2017, 68: 150-162.
- [27] CHEN Min-jie, XU Man-tao, FRÄNTI P. A fast $O(N)$ multiresolution polygonal approximation algorithm for GPS trajectory simplification [J]. IEEE Transactions on Image Processing, 2012, 21(5): 2770-2785.
- [28] WU Lin, XU Yong-jun, WANG Qi, et al. Mapping global shipping density from AIS data [J]. The Journal of Navigation, 2017, 70(1): 67-81.
- [29] ZHAO Liang-bin, SHI Guo-you, YANG Jia-xuan. Ship trajectories pre-processing based on AIS data[J]. The Journal of Navigation, 2018, 71(5): 1210-1230.
- [30] DOUGLAS D H, PEUCKER T K. Algorithm for the reduction of the number of points required to represent a digital line or its caricature[J]. Cartographer, 1973, 10: 112-122.