

动态阈值结合全局优化的船舶 AIS 轨迹在线压缩算法

宋 鑫 朱宗良 高银萍 裴道方

(上海海事大学物流科学与工程学院 上海 201306)

摘 要 随着船舶定位技术的进一步发展,大量船舶安装了船舶定位识别系统,该系统生成了海量的船舶轨迹数据。船舶轨迹数据经过压缩处理后能有效提高处理、应用数据的工作效率。针对现有轨迹在线压缩算法处理压缩率高、耗时长等问题,提出了一种动态阈值结合全局优化的两阶段在线压缩算法(DTGO)。该算法在第一阶段对原始轨迹进行分段处理,动态更新各项阈值,从而获得简化轨迹;在第二阶段使用改进的 SPM 算法对简化轨迹进行全局优化。通过对原始轨迹进行两阶段的处理,将原始轨迹分段成若干个子轨迹段,对于轨迹段进行局部处理,最后使用全局处理算法对所有子轨迹段进行全局优化。实验结果表明,该算法在提高压缩效率的同时取得了良好的压缩效果。

关键词 AIS 船舶轨迹,在线压缩,动态阈值,全局优化

中图分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.07.051

Vessel AIS Trajectory Online Compression Algorithm Combining Dynamic Thresholding and Global Optimization

SONG Xin ZHU Zong-liang GAO Yin-ping CHANG Dao-fang

(Institute of Logistics Science & Engineering, Shanghai Maritime University, Shanghai 201306, China)

Abstract With the further development of vessel location technology, a large amount of vessels trajectory data have been generated with the vessel positioning identification system installed on vessels. These compressed data can improve the efficiency of data processing and applying to a large extent. However, compressing the vessel trajectory data online may have some problems such as high compression ratio and long time consuming. Therefore, this paper proposed a two-stage online compression algorithm (DTGO) which combines dynamic threshold value with global optimization. At the first stage, the original trajectory is processed in segments, and the threshold values are dynamically updated, thus a simplified trajectory can be obtained. At the second stage, the simplified trajectory is globally optimized by a modified SPM algorithm. Through the two-stage processing, the original trajectory is segmented into several sub-trajectory segments which are processed locally. Finally, the proposed global processing algorithm is applied to optimize all sub-trajectory segments globally. The experimental results show that the algorithm not only obtains higher compression efficiency, but also achieves better compression results.

Keywords AIS vessel trajectory, Online compression, Dynamic thresholding, Global optimization

1 引言

近年来,全球贸易联系日益紧密,航运作为低成本的运输方式,承担了全世界大部分的货物流通。在一些热点航道上,密集的来往船舶给船舶交通管理系统(Vessel Traffic System, VTS)的正常工作带来了巨大的挑战。为了增强海上交通的安全性、提高航行效率、保护海洋环境等,国际海事组织要求 300 吨以上的船舶必须安装船舶自动识别系统(Automatic Identification System, AIS)^[1]。然而,若要通过 AIS 对船舶进行动态监控则需要实时掌握船舶的航行状态,这就要求船舶按时上传 AIS 数据。全球岸基设备每天接收到的 AIS

数据量十分庞大,这给数据的存储带来了极大的负担,因此对统计、分析、利用 AIS 数据提出了较高的技术要求^[2]。

船舶轨迹数据的处理主要是指使用压缩简化算法将轨迹中不具备行驶特征的轨迹点去除,将保留的点组成新的轨迹。压缩后的轨迹与原始轨迹之间的误差必须是在可接受的范围之内,因此获取一条能够精确描述原始轨迹特征而占用存储空间小的船舶压缩轨迹就显得尤为重要^[3]。

2 研究现状

轨迹压缩领域经典的算法是由 Douglas 等于 1973 年提出的 Douglas-Peucker(DP)算法^[4]。DP 算法是一种离线的批

到稿日期:2018-06-29 返修日期:2018-11-22 本文受国家自然科学基金项目(71602114),上海市科委科研项目(16040501500, 17595810300)资助。

宋 鑫(1993-),男,硕士生,主要研究方向为 AIS 数据分析与处理;朱宗良(1986-),男,硕士,工程师,主要研究方向为人工智能及其在港航和物流中的应用、数据挖掘与知识发现;高银萍(1994-),女,博士生,主要研究方向为深度学习在港口物流中的应用;裴道方(1978-),男,博士,教授,博士生导师,主要研究方向为供应链设计与运营、物流系统运作与优化,E-mail:dfchang@shmtu.edu.cn(通信作者)。

处理算法,它需要收集完整轨迹后再对数据进行压缩处理。而在船舶轨迹压缩领域,大部分学者在 Douglas-Pecuker 算法的基础上进行改进。李名等^[5]于 2010 年提出了一种动态的 Douglas-Pecuker 算法,通过引入时间维构造三维空间来对航向速度等动态数据进行压缩。Singh 等^[6]于 2016 年提出了一种 SPM(Scan-Pick-Move)算法,SPM 算法借鉴 DP 算法的思想调整误差度量的基线选取方式,对轨迹点依次进行误差比较。Ying 等^[7]提出了一种速度保持的轨迹简化算法,在保留轨迹点的速度信息的基础上构建伪图,利用动态规划的方法求解伪图中的最短路径,从而对轨迹进行简化。Zhang 等^[8]提出了一种确定最大值的方法来选择合适阈值的压缩算法,确保简化后的点在阈值范围内。徐凯等^[9]于 2017 年提出了一种基于向量代数方法的快速 Douglas-Pecuker 算法,降低了算法的时间复杂度,加快了算法的运行速度。Zhao 等^[10]提出了一种结合船舶轨迹和 DP 算法的轨迹简化方法,在高压缩强度下显著缩短了压缩时间。

在在线轨迹压缩领域,也有不少学者提出了各种经典的算法。Keogh 等^[11]提出了 Sliding Window 和 Open Window 两种窗口算法,这两种算法的基本思想是通过计算窗口内轨迹点的累计误差来进行轨迹点的选取。Potamias 等^[12]通过关注轨迹点的速度和方向信息来构造安全区域,进行轨迹点的筛选,从而提出了 Threshold-Guided Sampling 算法和 STTrac 算法。Trajcevski 等^[13]提出了 Dead Reckoning 在线压缩算法,该算法通过当前点和它的速度信息来估计后继点的信息。Muckell 等^[14]提出了 SQUISH 算法和 SQUISH-E 算法,通过对轨迹点的优先级赋值权重的方式选择轨迹点,SQUISH-E 算法是对 SQUISH 的优化,对压缩比和压缩误差进行调整,得到了更好的压缩效果。Cao 等^[15]提出了一种使用 LISSED 误差度量的 DOTS 算法,通过构建有向无环图(Directed Acyclic Graph, DAG)优化求解轨迹压缩问题。

以上各种在线压缩算法取得了良好的压缩效果,然而将该类算法应用于船舶轨迹压缩领域时需要考虑海面结构与路网结构的差别。海面航行的船舶虽然有固定的航道,但是由于海面的环境因素,船舶的航行具有更高的自由度,航行产生的轨迹点在航道范围内更具有随机性;并且,在船舶航行过程中非正常偏离航道和航向、突然加速和减速、避让船舶和人为操纵等也会影响船舶的轨迹^[16]。同时,AIS 数据的采集也会受到船舶运动状态的影响,不同运动状态的船舶上传 AIS 数据的时间间隔也不同^[17],如表 1 所列。不同类型的船舶在海面可能会进行各种作业活动,如渔船进行捕鱼操作、客船的长时间停留等。因此,对船舶停留位置点的检测和识别也是一个不应忽视的问题。

表 1 AIS 数据的上传时间间隔

Table 1 Uploaded time interval of AIS data

船舶运动状态	时间间隔/s
锚泊或靠泊且移动速度不大于 3 kn	180
锚泊或靠泊且移动速度不大于 3 kn	10
0~14 kn	10
0~14 kn 且改变方向	31/3
14~23 kn	6
14~23 kn 且改变方向	2
大于 23 kn	2
大于 23 kn 且改变方向	2

本文针对船舶领域的轨迹在线压缩算法展开研究。在线压缩算法大多为局部处理算法,该类算法较为简单,处理速度较快,但因为是局部算法,没有考虑部分特征点对整条轨迹曲线形态的重要性,删除一些弯曲极值点后易造成轨迹失真^[18],当轨迹较为平缓时,相邻两个轨迹点之间的特征差异小,容易造成压缩后曲线轨迹点聚集的现象。为了解决上述问题,本文提出了一种动态阈值结合全局优化的轨迹压缩(Dynamic Threshold with Global Optimized, DTGO)算法。DTGO 算法的核心是对轨迹点进行停留点的检测和特征点的检测,停留点的检测是为了将轨迹分成若干个子轨迹段,其目的是对轨迹进行全局优化处理,而特征点的检测是找出能够突出反映船舶运动状态的点。将原始轨迹分段以后,再利用改进的 SPM 算法——SSPM(SED Scan-Pick-Move)算法对轨迹进行全局优化处理。

3 相关算法

3.1 SPM 算法

SPM 算法是一种扫描式算法,其基本原理是首尾两点连接以后形成一条基线,从起点开始的第二个轨迹点依次向基线做垂线,得到垂直欧氏距离 d ,如果垂直欧氏距离小于预先设定的阈值 δ ,则该点被舍弃;直到出现轨迹点到基线的距离大于阈值,才将该点重新设置为起点。重复上述步骤,直到所有的轨迹点都被扫描、选取,算法结束。算法的原理如图 1 所示。

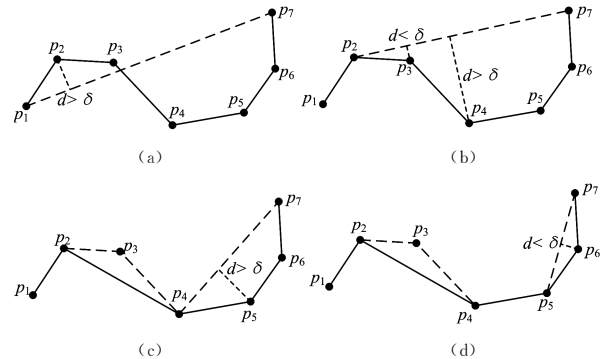


图 1 SPM 算法

Fig. 1 SPM algorithm

图 1 中的原始轨迹为 $T = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$,经过算法压缩处理后点 p_3 和 p_6 被舍去。压缩后的轨迹为 $T_{sm} = \{p_1, p_2, p_4, p_5, p_7\}$ 。SPM 算法通过将起点和终点连接形成基线,对轨迹上的各个点依次进行扫描、选取操作,剔除垂直欧氏距离小于阈值的点,而将垂直欧氏距离大于阈值的点作为新的起点。这样的做法其实是一种分段策略,改进了 DP 算法的递归运算,使得算法的时间复杂度降到了 $O(n)$ 。然而,SPM 由于是一种离线算法,需要将起点和终点连接形成第一条基线,因此不适用于在线压缩领域。此外,SPM 算法是一种二维平面上的算法,当船舶运动状态改变时,容易将包含重要特征的轨迹点剔除,而这些特征点恰恰能够反映轨迹的重要曲线样式。

3.2 VPTS 算法和 DOTS 算法

VPTS 算法首先保留轨迹点中的速度信息,根据速度对原始轨迹进行分段,并计算各个子轨迹段的平均速度,从而确

定速度误差的大小;其次,在原始轨迹的基础上构建伪图,为每个边缘附加速度权重值;最后,使用动态规划的方法求解简化轨迹。VPTS算法可以在保留原始轨迹样式特征的情况下保留轨迹点的移动行为。

DOTS算法是一种基于有向无环图(DAG)的在线轨迹简化算法,其基本原理是使用局部积分平方同步欧氏距离(Local Integral Square Synchronized Euclidean Distance, LISSSED)作为误差的度量,通过构建DAG找出在阈值范围内的所有潜在的简化轨迹。在DAG中,从起点到终点的每条路径都可以作为简化轨迹的近似解,因此对每一条近似解做边缘测试,验证该近似解是否属于边缘集合 E 。大量的边缘测试能够获得优秀的压缩效果,但同时也使得算法的运行时间大大延长。

以上两种算法把轨迹压缩问题转变为求解图中的最优路径问题,找到与原始轨迹最为相似的图中的轨迹,保证了较高的压缩率和压缩精度,能够最大程度地反应轨迹的样式特征,然而其极高的算法运行时间给在线处理轨迹数据带来了较大的负担。

4 算法设计与实现

4.1 相关定义与公式

船舶AIS轨迹是一个包含经度、纬度、时间、速度等信息的轨迹点集合。这些含有不同信息的轨迹点反映了轨迹的不同特征,组成了不同曲线样式的轨迹。

定义1(轨迹点) 船舶的位置点用 p 表示,一个位置点含有4个变量信息,分别是经度、纬度、时间、航速。

$$p = (x, y, v, t)$$

定义2(原始轨迹和简化轨迹) 原始轨迹是指船舶在航行一段时间后上传的AIS数据组成的轨迹,即原始轨迹是一个轨迹点组成的集合,用 T 表示。含有 n 个轨迹点的轨迹表示为:

$$T = \{p_1, \dots, p_i, \dots, p_n\} (i \in [1, n]), p_i = (x, y, v, t), t_1 < \dots < t_i < \dots < t_n \quad (1)$$

简化轨迹 T_{sim} 是指使用压缩算法对原始轨迹进行处理以后,剔除了冗余的轨迹点,按照轨迹点产生的时间顺序组成的新轨迹。

$$T_{sim} = \{p_{s1}, \dots, p_{s2}, \dots, p_{sn}\} (s_i \in [s1, sn]), t_{s1} < \dots < t_{s2} < \dots < t_{sn} \quad (2)$$

如图2所示,原始轨迹为:

$$T = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9\}$$

假设点 p_3, p_6, p_8 是冗余点,将其剔除,则简化轨迹为:

$$T_{sim} = \{p_{s1}, p_{s2}, p_{s4}, p_{s5}, p_{s7}, p_{s9}\}$$

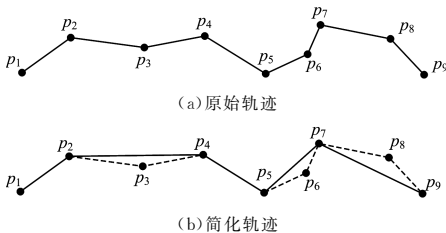


图2 原始轨迹与简化轨迹

Fig. 2 Original trajectory and simplified trajectory

定义3(轨迹段) 在原始轨迹 T 或者简化轨迹 T_{sim} 中,在轨迹点 p_i 和 p_j ($1 \leq i < j \leq n$)之间存在一个点或多个点,将 p_i 和 p_j 相连组成的线段作为原始轨迹 T 或简化轨迹 T_{sim} 的轨迹段,用 $\overline{p_i p_j}$ 表示。

定义4(航速差值) 航速差值VSD(Vessel Speed Difference)反映的是船舶运动状态的改变,VSD为相邻位置点速度差值的绝对值:

$$VSD = |v_j - v_i| (1 \leq i < j \leq n) \quad (3)$$

定义5(航向角差值) 航向角差值AD(Angle Difference)的航向角是指相邻轨迹段的方向差值,如图3所示。

$$AD = \alpha (\alpha \in [0^\circ, 180^\circ]) \quad (4)$$

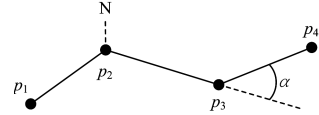


图3 航向角差值

Fig. 3 Heading angle difference

定义6(特征点) 若某点处的距离差值、航速差值、航向角差值中的一个值大于对应的距离阈值、航速阈值、航向角阈值,则称该点为特征点。

定义7(停留点) 原始轨迹 T 或简化轨迹 T_{sim} 中,若船舶在某个位置进行海上作业或者长时间停留,则其在该位置处上传的轨迹点被称为停留点,用 p_i' ($1 < i < n$)表示。图4中, p_2, p_3 两点的上传时间间隔远远大于正常情况下AIS数据上传的时间间隔。因此, p_2 点被称为停留点, p_3 点为子轨迹段的起点。

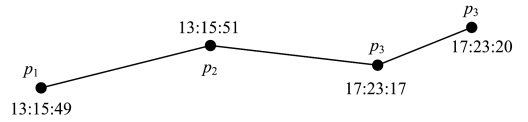


图4 停留点

Fig. 4 Stay point

定义8(停留间隔值) 停留间隔值STI(Stay Time Interval)是指船舶在停留点处停留的时间,即下一个轨迹点和停留点之间的时间差值。假设 p_i' 为停留点,则:

$$STI = t_{i+1} - t_i' \quad (5)$$

式(1)为墨卡托坐标转换公式。由AIS得到的轨迹点的数据为经纬度数据,为了提高压缩精度,需要将经纬度坐标转换成墨卡托坐标,假设某点的经纬度坐标为 (ϕ, ω) ,平面坐标为 (x, y) ,则转换公式如下:

$$r_0 = a / \sqrt{1 - e^2 \sin^2 \phi} \times \cos \phi \quad (6)$$

$$q = \ln \tan \left(\frac{\pi}{2} + \frac{\phi}{4} \right) - \frac{e}{2} \ln \frac{1 + e \sin \phi}{1 - e \sin \phi} \quad (7)$$

$$x = r_0 \times \omega \quad (8)$$

$$y = r_0 \times q \quad (9)$$

其中, r_0 为基准纬度圈的半径, a 为地球椭球长轴半径, q 为等量纬度, e 为地球椭球第一偏心率。

式(2)为同步欧氏距离(Synchronous Euclidean Distance, SED)。同步欧氏距离为原始轨迹中的点在简化轨迹中按照时间比例对应的欧氏距离。假设原始轨迹中的点为 $p(x_i, y_i, t_i)$,简化轨迹中的点为 $p'(x_{si}, y_{si}, t_{si})$,则 p' 的坐标计算公式为:

$$x_{s_i} = x_{i-1} + \frac{t_i - t_{i-1}}{t_{i+1} - t_{i-1}} (x_{i+1} - x_{i-1}) \quad (10)$$

$$y_{s_i} = y_{i-1} + \frac{t_i - t_{i-1}}{t_{i+1} - t_{i-1}} (y_{i+1} - y_{i-1}) \quad (11)$$

由 p 和 p' 计算同步欧氏距离:

$$SED = \sqrt{(x_i - x_{s_i})^2 + (y_i - y_{s_i})^2} \quad (12)$$

4.2 DTGO 算法

4.2.1 停留点检测

船舶在海面行驶至作业地点时会停留在海面进行各种作业,停留点反映了原始轨迹 T 的重要特征信息。如果不对停留点进行检测、识别,容易造成轨迹重要信息的缺失。因此,在对每一个轨迹点进行特征点的判断之前,首先要对其进行停留点的检测,而停留点检测也是为了在轨迹优化时能够分段处理。通过设置 STI 的值,判断轨迹点是否能够成为停留点,如果成为第一个停留点,则该停留点成为子轨迹段 $\overline{p_1 p_i'}$ 的终点,利用这样的原理将原始轨迹划分为多个子轨迹段,每一个子轨迹段都有自己的起点和终点。

在对原始轨迹 T 中的轨迹点进行停留点检测和特征点检测时,如果一个点是停留点,那么即使该点没有通过特征点检测,也要将该点保留下来。若该点既是停留点又是特征点,则需要在该点处对轨迹进行分段,同时更新阈值。因此,轨迹中的轨迹点若是停留点,则它一定是特征点;它若不是停留点,则有可能是特征点。而停留点的存在反映了船舶的真实运动情况,需要将其保留。

4.2.2 动态阈值和全局优化

在使用阈值对轨迹进行压缩时,合理的阈值能够使轨迹压缩得更加理想。然而,初始设定的固定阈值随着船舶航行时间的推移而增加,船舶运动状态的多次改变导致初始阈值不能更好地判断轨迹点的特征。因此,阈值的选取应该是动态的,这样才能更加符合船舶的实际运动。动态阈值的基本思想是,设定距离阈值 SED_{mit1} 、速度阈值 VSD 、角度阈值 AD 、停留点间隔值 STI 等初始阈值,随着后续轨迹点的到来,依次判断到来的轨迹点是否保留或删除,将确定为停留点或特征点的轨迹点保留下来,剔除其他冗余点。由各个子轨迹段组成第一阶段的初始简化轨迹。

然而,这是一种局部处理方法,难以兼顾全局的走势。因此,在局部处理完成之后,对轨迹再进行全局优化处理,以得到更加平滑的轨迹。在对轨迹进行停留点检测之后,会将原始轨迹 T 分成若干个子轨迹段,对每个轨迹段使用 SPM 算法进行处理,最后得到简化轨迹 T_{sim} 。全局优化使用时间复杂度较低的 SPM 算法能够降低处理的时间,同时得到较为理想的处理结果。

4.2.3 DTGO 算法的基本步骤

(1) 输入距离阈值 SED_{mit1} 和第二阶段的优化阈值 SED_{mit2} , 根据经验设置速度阈值 VSD 、角度阈值 AD , 在收集到第一个轨迹点 p_1 的信息后,将其作为起始特征点。

(2) 等待点 p_2, p_3 的到来,先对点 p_2 进行停留点检测,如果 p_2 通过检测成为停留点 p_2' ,则轨迹在此处分段,组成子轨迹段 $\overline{p_1 p_2'}$,同时在停留点处对阈值进行初始化。如果不是停留点则轨迹不分段。对停留点进行特征点检测时,如果停留

点不是特征点,则该停留点不会被剔除;若该停留点为特征点,则仍需进行阈值更新操作。

(3) 接着对点 p_2 进行特征点检测,计算点 p_2 处的 SED, VSD, AD , 将各个差值与初始阈值进行比较,如果有一个差值大于阈值,则点 p_2 为特征点,将点 p_2 保留;如果差值均小于阈值,则点 p_2 不是特征点,将点 p_2 舍去。

(4) 等待点 p_4 的到来,对点 p_3 进行停留点检测和特征点检测。依次对所有到来的轨迹点分别进行停留点检测和特征点检测。处理完所有轨迹点以后,运用改进的 SPM 算法对每一段子轨迹进行全局优化处理,最后导出简化轨迹 T_{sim} 。

算法 1 DTGO 算法

Input: $p_1, p_2, p_3, SED_{mit1}, SED_{mit2}$

Output: T_{sim}

1. $i = 1$;
2. for each p_i in T
3. $i + 1$; $\Delta t = t_i - t_{i-1}$;
4. if $\Delta t > STI$
5. $p_i' = p_i$;
6. trajectory $_i = \overline{p_1 p_i'}$;
7. end if
8. if $SED_i > SED_{mit} \& \& VSD_i > VSD_{mit} \& \& AD_i > AD_{mit}$
9. $SED_{mit} = SED_i$; $VSD_{mit} = VSD_i$; $AD_{mit} = AD_i$;
10. end if
11. $SSPM(\text{trajectory}_1, \text{trajectory}_2, \dots, \text{trajectory}_n)$
12. end for

DTGO 算法因其两阶段的特性,在第一阶段对所有原始轨迹中的点进行了循环比较,在第二阶段使用改进的 SPM 算法对轨迹进行全局优化处理,因此 DTGO 算法的最坏时间复杂度为 $O(n^2)$ 。

5 实例验证与算法对比

为了验证 DTGO 算法的有效性,进行单条船舶轨迹和多条船舶轨迹的压缩比较。算法采用 Python 语言编写,版本为 3.6.5, IDE 使用 PyCharm。实验设备的具体配置为: Windows 10, Intel(R) Core(TM) i5-3337U 1.8GHz, 8GB 内存。

实验数据集来自于中国港口网¹⁾。单条船舶轨迹为在数据集中随机抽取一条船舶一个月的运行轨迹,船舶海上移动通信业务标识 MMSI 为 413501160, 轨迹点的数量为 3310。多条船舶验证数据则从数据集中抽取 100 条船舶连续 5 个月的轨迹运行数据, 轨迹点数量为 1520720。

5.1 实验对比算法

本文选用了经典的 DP 算法、扫描式的 SPM 算法、基于图论的 VPTS 算法和 DOTS 算法。其中 DP 算法作为经典的轨迹压缩算法,其良好的压缩效果可以作为其他算法的参照。扫描式的 SPM 算法,优秀的压缩效率和压缩时间可以作为本文算法的参照。而基于图的基本理论的 VPTS 算法和 DOTS 算法可以作为本文算法的对比算法,以证明本文算法相比使用其他理论形成的算法也具有一定的优势。对比算法的相关信息如表 2 所列。

¹⁾ <http://ship.chinaports.com/>

表 2 对比算法的相关信息

Table 2 Information of compare algorithms

算法	类别	误差度量	时间复杂度
DP	离线	垂直欧氏距离	$O(n \log n)$
SPM	在线	时间同步欧氏距离	$O(n)$
VPTS	离线	几何误差、速度误差	$O(n^2)$
DOTS	在线	局部积分平方同步欧氏距离	$O(n^2/m)$
DTGO	在线	多阈值误差	$O(n^2)$

其中,DTGO 算法的多阈值误差指的是速度误差、角度误差、时间同步欧氏距离。时间复杂度中的 n 代表原始轨迹中点的数目,DOTS 算法的时间复杂度中的 m 代表简化轨迹中点的数目。

5.2 单条船舶压缩结果

单条船舶轨迹的压缩结果如图 5 所示。

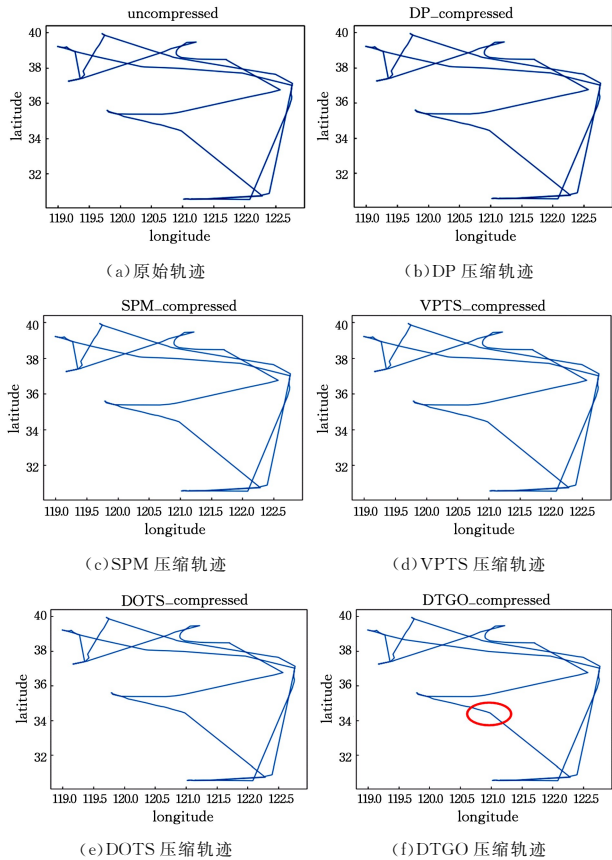


图 5 单条船舶轨迹的压缩结果

Fig. 5 Compression results of single vessel trajectory

由图 5 可知,在对单条轨迹压缩后,DTGO 算法能够获得较好的压缩精度。而原始轨迹和各种压缩算法的轨迹点数目分别如下:原始轨迹点的数目为 3310,DP 算法压缩后轨迹点的数目为 875,SPM 算法压缩后轨迹点的数目为 1288,VPTS 算法压缩后轨迹点的数目为 1389,DOTS 算法压缩后轨迹点的数目为 1156,DTGO 算法压缩后轨迹点的数目为 384。

5 种算法的压缩时间分别如下:1316 ms(DP 算法),833 ms(SPM 算法),889 ms(DTGO 算法),7486 ms(VPTS 算法),4862 ms(DOTS 算法)。SPM 算法压缩后点的数目大于 DP 算法,但是 SPM 算法的压缩时间大大短于 DP 算法。VPTS 算法和 DOTS 算法的精确度最高,较好地反映了原始轨迹的形状特征,但两者的压缩时间比其他算法长很多。而 DTGO 算法牺牲了部分的压缩时间却获得了较好的压缩精

度。图 5(f)中圆圈标出的部分为 DTGO 算法出现的压缩误差,从整体上看,其压缩效果良好。

5.3 多条船舶压缩结果

为了观察不同阈值下的各种算法的压缩效果,选取了 10m,20m,30m,40m,50m 作为压缩阈值参数,适用算法为 DP 算法、SPM 算法、DOTS 算法、DTGO 算法。其中 DTGO 算法的 VSD 和 AD 依据经验选取为 5kn 和 5° 。VPTS 的阈值为 10%,20%,30%,40%,50%。

表 3 为不同压缩阈值下不同算法压缩后的结果。从表 3 中可以看出,随着阈值的增加,DP 算法、VPTS 算法、DOTS 算法、DTGO 算法的压缩率有较大的下降趋势,而 SPM 算法的压缩率较为稳定。

表 3 不同阈值下的实验结果

Table 3 Experimental results under different thresholds

算法	阈值	压缩后点数	压缩率/%	压缩时间/ms
DP	10	852290	56	571656
DP	20	680593	45	568369
DP	30	598685	39	524677
DP	40	547199	36	518317
DP	50	509306	33	518401
SPM	10	990644	65	100142
SPM	20	908983	60	104462
SPM	30	866702	57	103642
SPM	40	837520	55	106596
SPM	50	814253	54	106781
VPTS	10	999131	66	2932530
VPTS	20	844592	55	2797293
VPTS	30	693946	46	2504543
VPTS	40	619284	41	2510581
VPTS	50	554477	36	2533740
DOTS	10	887607	58	1734295
DOTS	20	779501	51	1647946
DOTS	30	718339	47	1622502
DOTS	40	656410	43	1612960
DOTS	50	556532	37	1652854
DTGO	10	462791	30	115979
DTGO	20	338726	22	115032
DTGO	30	282149	19	117495
DTGO	40	248067	16	118118
DTGO	50	224024	15	110385

在压缩时间上,VPTS 算法和 DOTS 算法是 DP 算法的 3~5 倍,而 SPM 算法相比 DP 算法得到了 80%左右的提升。基于图论的 VPTS 算法和 DOTS 算法在牺牲了大量的压缩时间后获得了优秀的压缩效果。相比其他 4 种算法的高压缩率,本文提出的改进算法 DTGO 不仅获得了较好的压缩精度,同时在压缩时间上相比其他 4 种算法也有较大的优势。DTGO 算法的压缩时间平均比 SPM 算法高 10%左右,这是因为 DTGO 算法在动态阈值阶段对点不断进行比较操作。通过比较 5 种算法的压缩率和压缩时间可以发现,DTGO 算法继承了 SPM 算法良好的压缩时间,而轨迹的在线压缩需要在线对数据进行分析处理,因此较低的压缩时间使得算法具有更好的效果,且压缩率的降低使得数据的存储成了能够准确反映轨迹样式的特征点。

图 6 为算法误差比较图,DP 算法和 DTGO 算法的误差随着压缩率的降低而增大,SPM 算法、VPTS 算法、DOTS 算法的误差较小且相对稳定。然而,从单船压缩结果来看,DTGO 算法仍然能够保持良好的轨迹形状。

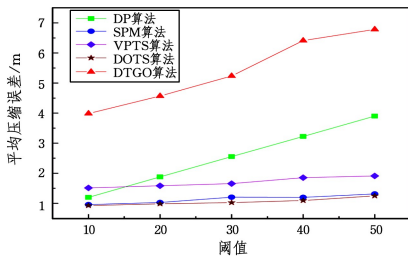


图6 平均压缩误差

Fig. 6 Average compression error

图7为5种算法的压缩时间随着原始轨迹点数目的增长而变化的曲线图,其中轨迹点的数目从300 000增长到1 500 000。轨迹点数目的增长给DP算法带来了更多次的迭代计算,使得压缩时间大幅增加。VPTS算法和DOTS算法在构建图以后,需要对所有边进行搜索,从而确定最优轨迹,导致算法运行时间大大延长,而SPM算法和DTGO算法的压缩时间的增长较为平缓。DTGO算法的压缩时间比SPM算法长约10%左右,然而DTGO算法获得了较低的压缩率,因此具有明显的优势。

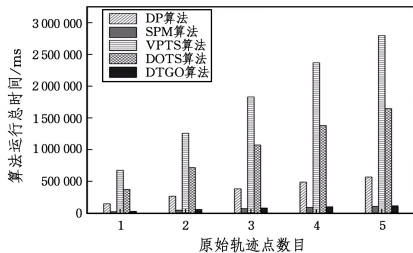


图7 轨迹点对算法运行时间的影响

Fig. 7 Effect of trajectory points on algorithm running time

结束语 针对船舶轨迹在线压缩,本文的提出的DTGO算法,通过分阶段、局部处理与全局优化相结合的方式,对船舶轨迹进行在线压缩。实验结果表明:该算法得到了较好的轨迹样式,并且提高了压缩效率,降低了压缩时间。然而,本文的研究也还存在一些不足,在设定了较多阈值的情况下,若各个阈值的选取不合理,则会对最后的压缩结果产生较大的影响,接下来将做进一步的研究。

参考文献

[1] MUCKELL J, HWANG J H, LAWSON C T, et al. Algorithms for compressing GPS trajectory data: an empirical evaluation[C]// Sigspatial International Conference on Advances in Geographic Information Systems. ACM, 2010:402-405.

[2] DE VRIES G K D, VAN SOMEREN M. Machinelearning for vessel trajectories using compression, alignments and domain knowledge[J]. Expert Systems with Applications, 2012, 39(18): 13426-13439.

[3] JIANG J W, WANG X L. Review on trajectory data compression [J]. Journal of East China Normal University(Natural Science), 2015, 2015(5): 61-76. (in Chinese)
江俊文, 王晓玲. 轨迹数据压缩综述[J]. 华东师范大学学报(自然科学版), 2015, 2015(5): 61-76.

[4] DOUGLAS D H, PEUCKER T K. Algorithms for the reduction of the number of points required to represent a digitized line or

its caricature[J]. Canadian Cartographer, 2011, 10(2): 112-122.

[5] LI M, HU Q Y, MENG L. Research on Vessel Motion Trajectory Compression Based on AIS[J]. MARINE TECHNOLOGY, 2010(1): 11-13. (in Chinese)
李名, 胡勤友, 孟良. 基于 AIS 的船舶运动轨迹压缩技术研究 [J]. 航海技术, 2010(1): 11-13.

[6] SINGH A K, AGGARWAL V, SAXENA P, et al. Performance analysis of trajectory compression algorithms on marine surveillance data[C]// International Conference on Advances in Computing, Communications and Informatics. 2017: 1074-1079.

[7] YING J C, SU J H. On Velocity-Preserving Trajectory Simplification[M]// Intelligent Information and Database Systems. Springer Berlin Heidelberg, 2016: 241-250.

[8] ZHANG S K, LIU Z J, YAO C, et al. AIS Trajectories Simplification and Threshold Determination[J]. Journal of Navigation, 2016, 69(4): 729-744.

[9] XUK, QIU J Y, LI Y. Offline Efficient Compression Algorithm for AIS Data Retains Time Elapsing Dimension[J]. Computer Science, 2017, 44(S2): 498-502. (in Chinese)
徐凯, 邱家瑜, 李燕. 一种加入时间维的船舶轨迹高效离线压缩算法研究[J]. 计算机科学, 2017, 44(S2): 498-502.

[10] ZHAO L B, SHI G Y. A method for simplifying ship trajectory based on improved Douglas-Peucker algorithm [J]. Ocean Engineering, 2018, 166: 37-46.

[11] KEOGH E, CHU S, HART D, et al. An Online Algorithm for Segmenting Time Series[C]// Proceedings 2001 IEEE International Conference on Data Mining, 2001: 289-296.

[12] POTAMIAS M, PATROUMPAS K, SELIS T. Sampling Trajectory Streams with Spatiotemporal Criteria[C]// International Conference on Scientific and Statistical Database Management. IEEE Computer Society, 2006: 275-284.

[13] TRAJCEVSKI G, CAO H, SCHEUERMANN P, et al. Online data reduction and the quality of history moving objects databases[C]// ACM International Workshop on Data Engineering for Wireless and Mobile Access. ACM, 2006: 19-26.

[14] MUCKELL J, HWANG J H, PATIL V, et al. SQUISH: an online approach for GPS trajectory compression[C]// International Conference on Computing for Geospatial Research Applications. ACM, 2011: 13.

[15] CAO W, LI Y. DOTS: An online and near-optimal trajectory simplification algorithm[J]. Journal of Systems and Software, 2017, 126: 34-44.

[16] MA W Y, WU Z L, LI W F. Conformal detection algorithm of anomalous behavior of vessel[J]. Journal of Traffic and Transportation Engineering, 2017, 17(5): 149-158. (in Chinese)
马文耀, 吴兆麟, 李伟峰. 船舶异常行为的一致性检测算法[J]. 交通运输工程学报, 2017, 17(5): 149-158.

[17] QI L, ZHENG Z Y, LI G P. AIS-data-based ship domain of ships in sight of one another[J]. Journal of Dalian Maritime University, 2011, 37(1): 48-50. (in Chinese)
齐乐, 郑中义, 李国平. 互见中基于 AIS 数据的船舶领域[J]. 大连海事大学学报, 2011, 37(1): 48-50.

[18] SAALFELD A. Topologically Consistent Line Simplification with the Douglas-Peucker Algorithm[J]. American Cartographer, 1999, 26(1): 7-18.