

一种加入时间维的船舶轨迹高效离线压缩算法研究

徐 凯¹ 邱家瑜¹ 李 燕²

(上海海事大学上海国际航运研究中心港航大数据实验室 上海 200082)¹

(同济大学电子与信息工程学院计算机系 上海 201804)²

摘 要 全球船舶轨迹大数据加工的一个重要步骤是对船舶轨迹进行压缩。以经典船舶轨迹离线压缩算法思路为基础,探讨了传统算法在压缩效果和压缩效率方面存在的问题。例如,经典 Douglas-Peucker 压缩算法没有考虑时间维,导致压缩后的数据对船舶的速度与航行状态信息的保留效果不佳。但是,动态 Douglas-Peucker 算法虽然考虑了时间维,却因此增加了计算量,导致压缩效率不高。在上述算法的基础上,将考虑时间维的轨迹压缩算法对应到向量空间中,利用向量的内积、外积的几何性质,对算法压缩效率和效果进行了改善,并提出了快速 Douglas-Peucker 算法。经过实验验证,该算法较经典算法效率提高了约 30% 以上,压缩效果比已有算法更优。

关键词 船舶轨迹,船舶自动识别系统,轨迹压缩,时空序列,道格拉斯-普克算法

中图法分类号 TP391 **文献标识码** A

Offline Efficient Compression Algorithm for AIS Data Retains Time Elapsing Dimension

XU Kai¹ QIU Jia-yu¹ LI Yan²

(Shipping & Port Big Data Laboratory, Shanghai Maritime University, Shanghai 200082, China)¹

(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)²

Abstract Ship track compressing is an important step in global ship trace big data processing. This paper firstly expounded some existing offline compression algorithms for ship trajectory and also exhibited their problems. For example, classic Douglas-Peucker compression algorithm just compresses the ship trajectory in two dimensions, losing time element, which makes that the compression algorithm cannot keep the information of speed and status of ships. On the other hand, although dynamic Douglas-Peucker compression algorithm needs to divide ship trajectory into different groups before compression, it will yet result in some efficiency problems. The compression algorithms which takes into account time dimensions was token to spatial vector. In order to improve the effect and efficiency, we used inner product and outer product. Through experiment, we found that the efficiency of the algorithm had a 30% increase and a better effect in most cases. The innovation is putting the inner product and outer product into compression algorithms.

Keywords Ship trajectory, Automatic identification system (AIS), Trajectory compression, Space-Time sequence, Douglas-Peucker algorithm

1 引言

随着船舶自动识别系统(Automatic Identification System, AIS)的强制使用,海事系统及船舶公司为了监控船舶,将接收大量的船舶 AIS 数据,因此航运业需要大数据计算与分析技术对这些海量数据进行分析。为了在存储这些船舶 AIS 数据的同时确保数据的质量并降低存储成本,对 AIS 数据的压缩处理产生了需求。因此涌现出以 Douglas-Peucker 算法^[1]为代表的一系列压缩算法。

船舶 AIS 数据在海事管理、船舶搜救、船舶交通流调查、船舶监控查询^[2]以及电子海图动态船舶场景回放等方面^[3]得到了广泛应用。AIS 数据包括船舶的静态信息和船舶的动态信息,本文主要研究动态信息,包括船位(实时)、航迹向、对地航速、船首向、航行状态、转向率等。动态信息的发布频率高,一般 2~360s 发布一条^[4],数据体量巨大。

全球船舶每天大约发布上亿条 AIS 数据,目前的计算机性能无法满足在单台计算机上有效、快速地完成对这些数据的分析与利用。在对海量大数据进行可视化分析的研究中,压缩算法已经体现出重要的价值^[5]。如何将这海量数据压缩至单台计算机,使得能够在较短时间内分析利用数据量成为重要研究方向。本文根据经典的压缩算法提出了一种更高效、失真率更低的轨迹离线压缩算法,创新点如下。

(1)利用代数空间中向量外积与内积的几何性质,在首尾两点之间找出所有从远离首尾两点连线而靠近首尾两点连线的点(本文称为“状态驻点”),并从状态驻点集合中找到距离极值点。由于计算外积与内积仅需使用加减法与乘法,减少了三维距离计算的次数,因此从计算性能上提高了压缩效率。若以距离求取次数来计算,算法的时间复杂度由 $T(n) = O(n \log n)$ 减少为 $T(n) = O(\log n)$ 。

(2)在加入时间维度的三维轨迹压缩算法中去除了轨迹

本文受国家社会科学基金项目(15BJY069),交通部应用基础研究主干学科项目(2014329810120)资助。

徐 凯(1983—),男,博士,工程师,CCF 会员,主要研究方向为大数据、物联网、机器学习及其在港口、航运领域的应用, E-mail: kernxu@gmail.com(通信作者)。

分段步骤,在直接压缩全局轨迹时,既保留船舶轨迹中速度和加速度的变化信息,避免了二维算法中船舶航行状态信息的丢失,又提高了算法的压缩效果和执行效率。

2 轨迹压缩算法的研究意义与现状

2.1 轨迹压缩算法的研究意义

随着物联网和移动互联网的发展,移动目标的轨迹数据变得丰富多样,轨迹数据处理技术的应用价值与日俱增。轨迹压缩算法不仅可以应用于船舶轨迹,还可以应用于各种移动物体轨迹上,例如车辆、飞机甚至是行人。此外,轨迹压缩算法还可以应用于矢量化图像压缩等领域,与位图类压缩方法^[6]互补,具有广阔的应用前景。

本文研究的压缩算法考虑了时间特征,将二维曲线压缩算法拓展至三维空间的同时有效地控制了算法的时间复杂度,不仅对船舶轨迹大数据压缩更为有效,对三维曲线压缩也具有借鉴意义。

2.2 轨迹压缩算法的研究现状

如图 1 所示,船舶 AIS 数据中的船舶轨迹数据是由点集构成的,集合中的每个点都是一条船舶 AIS 数据,反映了船舶的实时状态^[7]。轨迹压缩的目的是为了将重复、冗余数据从原始数据中剔除,尽可能保留可以体现船舶轨迹特征的子集。

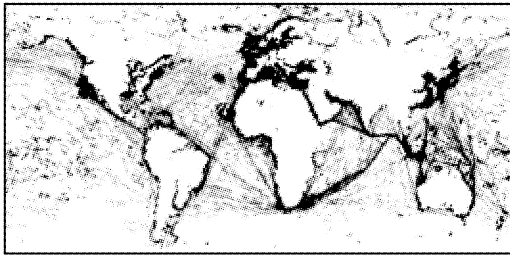


图 1 全球 AIS 轨迹点连线效果图

Douglas 等于 1973 年提出了经典的轨迹压缩算法,即 Douglas-Peucker 算法^[1]。针对船舶轨迹压缩,一些学者对 D-P 算法进行了改进。胡勤友等于 2010 年提出一种动态 Douglas-Peucker 算法^[8],在经典 Douglas-Peucker 算法的基础上引入时间维,将压缩算法推广至三维空间,保留了更丰富的轨迹细节特征,但大大增加了算法复杂性。王笑天等^[9]于 2016 年提出基于第一特征点的 Douglas-Peucker 算法,通过牺牲压缩率和误差率,减少了算法的时间复杂度。目前,缺乏既能保留时间维度特征,又具有较好时间复杂度的算法,使得轨迹大数据分析研究受到约束。

3 离线轨迹压缩算法

轨迹压缩方法分为实时压缩和离线压缩两类^[10]。离线压缩是从从全局考虑船舶的航行轨迹的最优压缩方案,相比局限于一个时间窗口内的实时数据压缩方法,它的失真率更低、压缩比更高。但由于每一个点的取舍判断都涉及全局情况,导致了多重嵌套和遍历,算法的时间复杂度难以控制。

3.1 离线轨迹压缩问题的数学表达

离线轨迹压缩即从组成的有序集合 P 中选出能够反映出原曲线主要形态特征的点,这些点组成子集合 P' ,称 P' 为集合 P 的压缩集合。

假设描述曲线的有序点集合 P 为:

$$P = \{P_1, P_2, P_3, \dots, P_n\}, P_n = (\phi_n, \lambda_n)$$

其中, ϕ_n, λ_n 分别代表第 n 条 AIS 数据的经度、纬度数据。

通常曲线在可以接受的采样粒度下可以认为是由集合 P 中相邻点之间的连线组成的折线。压缩算法负责从 P 中选择最能体现原曲线特征的点,并用集合 $K = \{k_1, k_2, k_3, \dots, k_m\}$ 表示这些点的下标,其中 $m \leq n$ 。则:

$$P' = \{P_{k_1}, P_{k_2}, P_{k_3}, \dots, P_{k_m}\}$$

即 P 代表的曲线可以由 P' 代表的曲线近似代替。

3.2 二维 Douglas-Peucker 算法

经典 Douglas-Peucker 算法^[11](经典 D-P 算法)的思路如图 2 所示。将首点和尾点分别作为初始锚点和初始漂浮点,并将其连成的直线称为基线。依次计算中间各点到基线的垂直距离,并找出最大距离对应的点。若最大距离小于设定的阈值,则用基线代替原始曲线;若最大距离大于设定的阈值,则将最大距离对应的点作为分裂点,并将该分裂点作为前向点集的漂浮点和后向点集的锚点。依次递归选取分裂点和分段,直到每一个子集中都不再出现新的分裂点。算法的时间复杂度 $T(n) = O(n \log n)$ 。

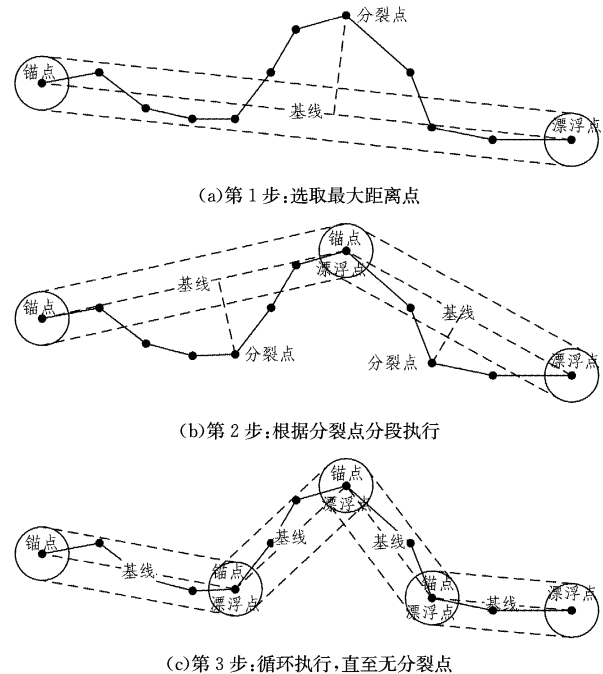


图 2 经典 D-P 算法原理图

基于第一特征点的 Douglas-Peucker 算法^[9](简称第一特征 DP 算法)为了弥补经典 D-P 算法压缩效率上的不足,从原理上减少了计算次数。将第一个距离首尾两点连线距离大于阈值的点作为新节点记录并分段,这样的点在第一特征 D-P 算法中被称为第一特征点。通过这样的计算方法,第一特征 D-P 算法的算法复杂度降低为 $T(n) = O(n)$ 。

该方法由于每次都使用第一特征点对曲线进行重新分段,未对所有特征点进行比较,因此不能保证保留所有特征点的最优结果,相比经典 D-P 算法压缩失真率增大。

以经典 D-P 算法为代表的二维 D-P 方法虽有效地保留了船舶行驶轨迹形态特征信息,但是无法保留船舶的速度变化特征,原因是未考虑时间特征。

3.3 三维 Douglas-Peucker 算法

为弥补经典 D-P 算法在时间特征方面的不足,动态 Douglas-Peucker 算法^[8](动态 D-P 算法)加入了时间维,将船舶轨迹分段后在三维空间进行压缩。对每段轨迹仍然采用经

典 D-P 算法的核心思路。

动态 D-P 算法引入时间维后对压缩效果的影响如图 3 所示。当船舶在一条直线上做变速运动时,经典 D-P 算法只会保留直线的端点,误删了包含速度特征的关键信息的位置点。当船舶由行驶转为停泊时,停泊这一事件采用经典 D-P 算法只能在二维平面保留一个点,而误删能够反映停泊时间长短的轨迹点。动态 D-P 算法通过增加时间维克服了无法保留船舶速度变化特征的缺点。

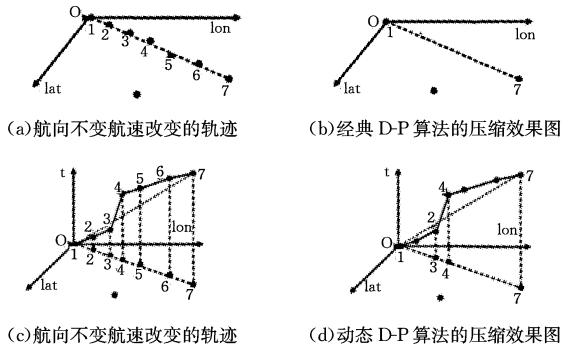


图 3 三维 D-P 算法与二维 D-P 算法的对比

但是,动态 D-P 算法也存在缺陷:虽然动态 D-P 算法在压缩前通过数据遍历找出虚拟停泊点对轨迹进行了分段,将轨迹压缩问题的计算规模大大缩减,但是在每一段轨迹的压缩过程中仍然使用经典 D-P 算法过程,导致每一段轨迹压缩的时间复杂度均为 $T(n) = O(n \log n)$,效率并未产生量级上的改变;同时,由于需要在三维空间中多次计算首尾两点连线与所有点的偏离距离,反复运用了海伦公式,涉及大量乘方、开方运算,耗费的 CPU 时钟周期远大于基于加减法、乘法的运算。因此,动态 D-P 算法耗费的计算时间远大于经典 D-P 算法。

4 快速 Douglas-Peucker 算法研究

针对已有算法的缺陷,本文将三维轨迹压缩问题放入代数空间中进行研究,通过向量外积与内积的几何特性,寻找与以往通过扇形区域和几何包络等方法来降低 D-P 算法时间复杂度的研究思路截然不同的新算法。本文将这一新算法称为快速 Douglas-Peucker 算法(快速 D-P 算法)。

4.1 基于向量代数方法的压缩思路

本文发现三维空间中的轨迹可以被分解为与其首尾连线的平行方向和垂直方向的运动合成,而距离变化只与垂直方向的运动有关,因此可以利用代数外积运算将轨迹运动分解到垂直平面上,从而替代海伦公式计算距离。

轨迹被分解到垂直于连线的平面后,可以通过向量内积运算,找出轨迹由远离转为靠近的点(本文称为“状态驻点”)。显然,距离的极值一定出现在状态驻点中,这样就减少了对计算机而言计算周期较大的海伦公式的距离求解次数,避免了除法、三角函数与开方运算。因此,快速 D-P 算法的主要计算时间是在利用海伦公式求距离上,算法的时间复杂度为 $T(n) = O(\log n)$ 。

4.2 快速 D-P 算法的流程和计算公式

图 4 给出快速 D-P 算法的流程图,其主要过程为:首先,对首点与尾点之间的所有点利用向量外积的性质,找出所有的状态驻点。其次,依次计算所有状态驻点到首尾两点连线的距离并从中找出距离最大值 d 。最后,比较 d 与距离阈值

DST 的大小,若 d 小于距离阈值 DST ,输出首尾两点,子程序结束;若 d 大于距离阈值 DST ,记录该状态驻点并以该点为新的节点将曲线分为两段。以相同方法递归运算,直至没有新的分裂点出现,子程序结束。

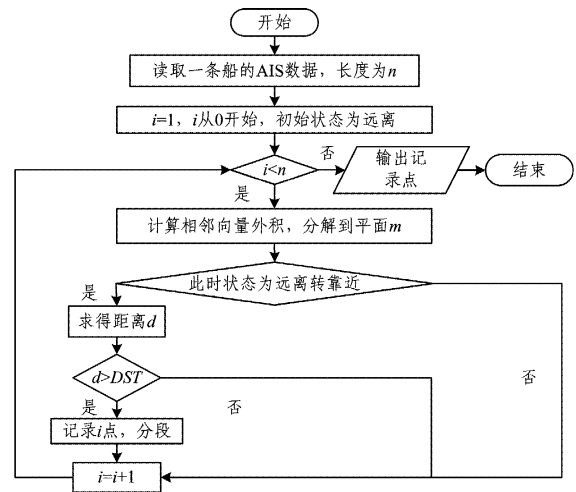


图 4 快速 D-P 算法的流程图

图 5 所示为船舶轨迹压缩中的随机轨迹。曲线 SE 中 S 为起点, E 为终点。 $\overrightarrow{P_{k-1}P_k}$ 与 $\overrightarrow{P_kP_{k+1}}$ 为轨迹中任意相邻三点间的连线,虚线连接的 \overrightarrow{SE} 为起点与终点的连线,平面 M 为垂直于 \overrightarrow{SE} 的平面。根据算法的核心思路,首先用 $\overrightarrow{P_{k-1}P_k}$ 与 $\overrightarrow{P_kP_{k+1}}$ 对 \overrightarrow{SE} 计算得到外积 $\overrightarrow{P_{k-1}P_k}'$ 与 $\overrightarrow{P_kP_{k+1}}'$, 即:

$$\overrightarrow{P_{k-1}P_k}' = \overrightarrow{P_{k-1}P_k} \times \overrightarrow{SE}$$

$$\overrightarrow{P_kP_{k+1}}' = \overrightarrow{P_kP_{k+1}} \times \overrightarrow{SE}$$

利用向量的平移性质,将 $\overrightarrow{P_{k-1}P_k}'$ 与 $\overrightarrow{P_kP_{k+1}}'$ 沿着平行于 \overrightarrow{SE} 的方向平移至平面 M 后,继续进行下一步压缩。

利用向量的平移性质,将与沿着平行于方向平移至平面 M 后,继续进行下一步压缩步骤。

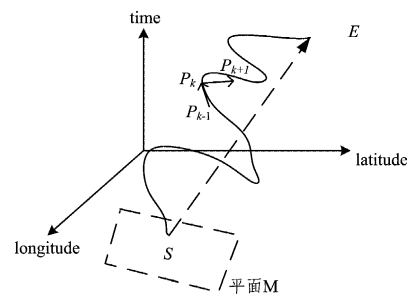


图 5 快速 D-P 算法思路示意图

如图 6 所示, $\overrightarrow{P_{k-1}P_k}'$ 与 $\overrightarrow{P_kP_{k+1}}'$ 为外积 $\overrightarrow{P_{k-1}P_k}'$ 与 $\overrightarrow{P_kP_{k+1}}'$ 平移到平面 M 后的向量,这两个向量包含 $\overrightarrow{P_{k-1}P_k}$ 与 $\overrightarrow{P_kP_{k+1}}$ 相对 \overrightarrow{SE} 的靠离状态。其中, $\angle \alpha$ 是 $\overrightarrow{SP_{k-1}}'$ 与 $\overrightarrow{R_{k-1}P_k}'$ 所形成的夹角, $\angle \beta$ 是 $\overrightarrow{SP_k}'$ 与 $\overrightarrow{P_kP_{k+1}}'$ 所形成的夹角。

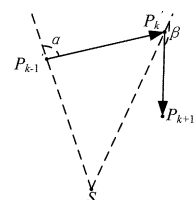


图 6 垂直平面 M 上的分向量

假设点 P_{k-1}' 的坐标为 $(x'_{k-1}, y'_{k-1}, z'_{k-1})$, P_k' 的坐标为 (x'_k, y'_k, z'_k) , P_{k+1}' 的坐标为 $(x'_{k+1}, y'_{k+1}, z'_{k+1})$, S 的坐标为 (x_0, y_0, z_0) 。利用向量的点积,可以判断 $\angle\alpha$ 与 $\angle\beta$ 是锐角还是钝角。

其中:

$$\overrightarrow{SP_{k-1}'} \cdot \overrightarrow{P_k'P_{k-1}'} = \begin{cases} \overrightarrow{SP_{k-1}'} \cdot \overrightarrow{P_k'P_{k-1}'} < 0, & \angle\alpha \text{ 为钝角} \\ \overrightarrow{SP_{k-1}'} \cdot \overrightarrow{P_k'P_{k-1}'} > 0, & \angle\alpha \text{ 为锐角} \\ \overrightarrow{SP_{k-1}'} \cdot \overrightarrow{P_k'P_{k-1}'} = 0, & \angle\alpha \text{ 为直角} \end{cases}$$

当 $\angle\alpha$ 为锐角且 $\angle\beta$ 为钝角时,可以得到点 P_k 是由远离 S 点转变为靠近 S 点的点,记录点 P_k 为状态驻点。根据上述步骤依次找出所有状态驻点。

找出状态驻点后,计算所有状态驻点到首尾两点连线的距离,找出距离最大值与距离阈值进行判断。假设图 6 中的点 P_k' 为距离最大值所在状态驻点。

$$SP_k' = \begin{cases} SP_k' \geq DST, & \text{令 } P_k \text{ 为新节点} \\ SP_k' < DST, & \text{忽略 } P_k \end{cases}$$

若线段 SP_k 小于距离阈值 DST ,忽略点 P_k ,直接输出点 S 与点 E 两点,压缩算法结束;若线段 SP_k 大于距离阈值 DST ,输出点 P_k 并将该点设为新的节点且将轨迹分为两段。分段后递归调用压缩算法,直至不产生新的分裂点,程序结束。

5 实验结果与对比

实验基于 Ubuntu16.04 系统、Python2.7.12 的开发环境,算法运行的硬件环境为 I7 处理器,主频 3.4GHz,内存 16GB。为了验证本文方法的有效性,分别进行单船轨迹细节对比与多船轨迹航程误差对比。

5.1 单船压缩结果对比

距离阈值在本文研究的算法中起着重要作用。经实验发现,使用不同的距离阈值进行压缩将产生不同的结果。

本文随机抽取了某船为时一个月的船舶 AIS 数据,并进行压缩。并本文实验采用的 AIS 数据由亿海蓝(北京)数据技术股份公司提供、港航大数据实验室整理。船舶的唯一识别号(mmsi)为 412148000,时间为 2014 年 9 月,原始数据为 12043 条。

如图 7 所示,当取压缩阈值为 20m 时,一条船发生了两段停泊状态,带星号连线为原始数据连线,不带星号连线为压缩后的数据连线。图 7(a)为经典 D-P 算法压缩后的轨迹,图 7(b)为快速 D-P 算法压缩后的轨迹。显然,经典 D-P 算法将船舶的停泊时间分成了两段低速航行状态,忽略了一段停泊状态,漏判了这两段停泊状态,压缩产生了严重失真,而快速 D-P 算法则避免了这种情况。

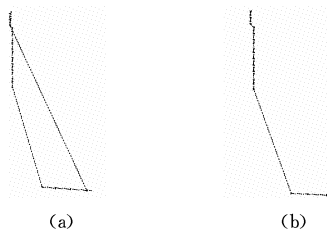


图 7 压缩后轨迹在三维坐标系中的对比

图 8 为选取不同压缩阈值时,3 种不同的 D-P 压缩算法之间航程差百分比与压缩率的对比直方图。其中压缩比为柱状,表示压缩后数据条数与原始数据条数的比值(主坐标轴为

柱状数值);航程差百分比为折线,表示压缩后轨迹点连线长度的减少值与压缩前轨迹点连线长度的比值(次坐标轴为折线数值)。从图 8 可知,当距离阈值取得越大时,压缩比越高,航程差越大。压缩比、航程差与距离阈值正相关;可以看出快速 D-P 算法在压缩比与航程差上都优于经典 D-P 算法与动态 D-P 算法。

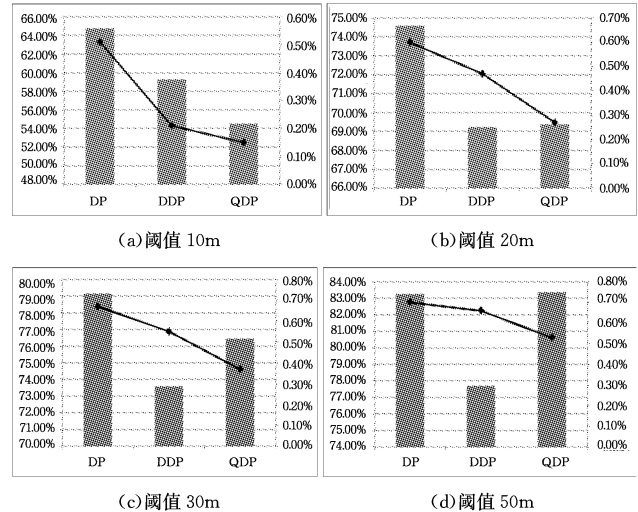


图 8 不同阈值下压缩效果对比

5.2 多船轨迹压缩效果对比

本节使用多个 AIS 数据集来对比快速 D-P 算法与动态 D-P 算法的压缩效果。实验共选取了 3 个数据集:1)中国沿海干散货船 2016 年 9 月轨迹数据集;2)全球 1000-4000TEU 集装箱船舶 2016 年 9 月轨迹数据集;3)全球 CAPE 型干散货船 2016 年 9 月轨迹数据集,数据集说明如表 1 所列。

表 1 实验数据集说明

编号	船舶类型	轨迹覆盖区域	船舶数量	轨迹点数量
1	中国沿海干散货船	中国沿海	843	2949904
2	全球 1000-4000TEU 集装箱船	全球	2095	7368816
3	全球 CAPE 型散货船	全球,主要集中在中-澳、中-南美航线	1608	3683957

从表 2—表 4 所列的多个数据集的压缩实验结果中可以看出:快速 D-P 算法在选用相同阈值的情况下,压缩率与动态 D-P 算法相当,但算法执行时间加快了 30%~50%,给航程计算带来的误差更小,具有明显的优势。

表 2 中国沿海干散货船轨迹(实验数据集 1)数据压缩效果对比

阈值/m	动态 D-P 压缩率/%	快速 D-P 压缩率/%	动态 D-P 误差/%	快速 D-P 误差/%	动态 D-P 压缩时间/s	快速 D-P 压缩时间/s
50	81.35	79.83	0.001401	0.001454	188	101
100	85.70	83.01	0.001868	0.001553	170	97
150	87.87	85.11	0.002357	0.001724	166	94
200	89.29	86.40	0.002870	0.001938	157	89

表 3 全球 1000-4000TEU 集装箱船舶轨迹(实验数据集 2)数据压缩效果对比

阈值/m	动态 D-P 压缩率/%	快速 D-P 压缩率/%	动态 D-P 误差/%	快速 D-P 误差/%	动态 D-P 压缩时间/s	快速 D-P 压缩时间/s
50	81.54	81.16	0.003204	0.002922	452	241
100	85.54	83.82	0.003427	0.002957	438	244
150	87.60	85.43	0.004779	0.003010	417	238
200	88.94	86.53	0.004956	0.003053	392	233

表4 全球CAPE型散货船舶轨迹(实验数据集3)数据压缩效果对比

阈值/ m	动态 D-P 压缩率/%	快速 D-P 压缩率/%	动态 D-P 误差/%	快速 D-P 误差/%	动态 D-P 压缩 时间/s	快速 D-P 压缩 时间/s
50	78.16	80.64	0.002411	0.002154	277	152
100	83.48	83.67	0.002521	0.002174	273	147
150	86.26	85.44	0.002632	0.002200	263	145
200	88.10	86.70	0.002735	0.002233	258	139

5.3 实验结论

综上所述,快速 D-P 算法相比于经典 D-P 算法和动态 D-P 算法在失真率更低的前提下,不仅提高了压缩率,而且在压缩速度上也最优。其可以有效地从船舶 AIS 数据找出可以体现原始轨迹特征的子集,舍弃了冗余、重复数据,从根本上解决了船舶 AIS 数据的存储难题。

结束语 本文提出的快速 Douglas-Peucker 算法具有更全面保留轨迹信息、压缩率更高、压缩速度更快的特点,对于解决全球船舶 AIS 轨迹的海量数据压缩具有重要的意义,有助于提升 AIS 大数据的加工和分析能力。该算法也可以推广应用到车辆、行人轨迹的压缩处理中,并对三维曲线和矢量图像压缩算法具有借鉴意义。同时,本文提出的压缩算法仍有进一步改善的空间:一方面,可以在本文算法的基础上提出对应的实时压缩算法;另一方面,可以考虑在本文算法中加入更多维的特征信息。AIS 数据中包含很多信息,本文压缩过程中仅仅利用了时间、经度、纬度信息,而船首向、船速等有效字段数据被忽略了。若可以将数据中的所有有效字段加以利用,将可以更清楚地体现船舶的航行特征,以达到进一步降低失真率的目的。

参考文献

[1] 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]. The Canadian Cartographer, 1973, 10(2): 112-122.

- [2] 周岳明. 船舶自动识别系统的应用及其关键技术的研究[D]. 大连:大连海事大学,2002.
- [3] 张宇. 通用船舶自动识别系统(AIS)及其关键技术研究[D]. 武汉:武汉理工大学,2004.
- [4] 毕月琨. AIS 基站系统中的数据解析与压缩[D]. 杭州:浙江海洋学院,2014.
- [5] 时学凯,王文珂,黄辉,等. 基于压缩域的脑成像大数据体可视化方法[J]. 计算机科学,2017,44(3):27-31.
- [6] 冯飞,刘培学,李晓燕,等. 离散余弦变换在图像压缩算法中的研究[J]. 计算机科学,2016,43(11):240-241.
- [7] 徐凯,李燕. 基于 Arduino 的 AIS 数据分布式采集研究[J]. 计算机测量与控制,2017,25(1):188-191.
- [8] 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.
- [9] 王笑天,吕海洋. 基于第一特征点的道格拉斯-普克压缩算法[J]. 软件导刊,2016,15(11):68-70.
- [10] 黄伟明,杨建宇,陈彦清,等. 基于扇形筛选法的矢量数据压缩方法[J]. 武汉大学学报(信息科学版),2016,41(4):487-491.
- [11] 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(5):595-599.
- [12] 王平利. 船舶自动识别系统应用关键技术研究[D]. 武汉:武汉理工大学,2007.
- [13] ROSEN I. Real-time GPS track simplification algorithm for outdoor navigation of visually impaired[J]. Journal of Network and Computer Applications, 2012, 35(5):1559-1567.
- [14] SHI S Z, CHARLTON M. A new approach and procedure for generalizing vector-based maps of real-world features[J]. GIScience & Remote Sensing, 2013, 50(4):473-482.
- [15] XU K, ZHEN H, LI Y, et al. Big Data Acquisition and Analysis Platform for Intermodal Transport[J]. International Journal of Database Theory and Application, 2016, 9(12):67-78.

(上接第 459 页)

参考文献

- [1] 李孟杰,谢强,丁秋林. 基于正交非负矩阵分解的 K-means 聚类算法研究[J]. 计算机科学,2016,43(5):204-208.
- [2] 胡海涛,朱建民. 基于 K-means 聚类的大学教学管理利益相关者分析[J]. 湘潭大学自然科学学报,2015,37(3):107-114.
- [3] LIN K, LI X, ZHANG Z, et al. A K-means clustering with optimized initial center based on Hadoop platform[C]//International Conference on Computer Science & Education. IEEE, 2014: 263-266.
- [4] YUNOH M F M, ABDULLAH S, NOPIAH Z M, et al. Fatigue Feature Extraction Analysis based on a K-Means Clustering Approach[J]. Journal of Mechanical Engineering and Sciences, 2015, 8(3):1275-1282.
- [5] CHEN Z, XUAN L, FAN Y. Constrained K-means with external information[C]//International Conference on Computer Science & Education. IEEE, 2013:490-493.
- [6] 何云斌,张晓瑞,万静,等. 基于改进遗传模拟退火 K-means 的心电波形的分类研究[J]. 计算机应用研究,2014,31(11):3328-3332.
- [7] AMORIM R C D, MIRKIN B. Selecting the Minkowski Exponent for Intelligent K-Means with Feature Weighting [J]. Springer Optimization & Its Applications, 2014, 9(2):103-117.
- [8] COVOES T F, HRUSCHKA E R, GHOSH J. A study of K-Means-based algorithms for constrained clustering[J]. Intelligent Data Analysis, 2013, 17(3):485-505.
- [9] 王越,王泉,吕奇峰,等. 基于初始聚类中心优化和维间加权的改进 K-means 算法[J]. 重庆理工大学学报,2013,27(4):77-80.
- [10] 王慧,申石磊. 一种改进的特征加权 K-means 聚类算法[J]. 微电子学与计算机,2010,27(7):161-163.
- [11] AI H, LI W. K-means initial clustering center optimal algorithm based on estimating density and refining initial[C]//Information Science and Service Science and Data Mining. IEEE, 2012:603-606.
- [12] ASHOK P, NAWAZ G M K. Outlier Detection Method on UCI Repository Dataset by Entropy Based Rough K-means[J]. Defence Science Journal, 2016, 66(2):113.
- [13] 李翠霞,史苇杭,李占波. 一种基于密度的加权模糊均值聚类算法[J]. 计算机科学,2012,39(5):180-182.
- [14] FEYEREISL J, AICKELIN U. Privileged information for data clustering[J]. Information Sciences, 2012, 194(5):4-23.
- [15] 侯勇,郑雪峰. 基于数据集特点的增强聚类集成算法[J]. 计算机应用,2013,33(8):2204-2207.