文章编号: 1007-5321(2015)02-0094-05 DOI: 10.13190/j.jbupt.2015.02.017

路网空间中 GPS 轨迹压缩的新方法

李 川[1,2,3], 张 彪[1,2,3], 李艳梅[1], 杨 宁[1], 王 勇[1,4]
(1. 四川大学 计算机学院, 成都 610065; 2. 武汉大学 软件工程国家重点实验室, 武汉 430072;
3. 国家空管自动化系统技术重点实验室, 成都 610065; 4. 防空兵学院 指挥控制系, 郑州 450000)

摘要: 传统轨迹压缩算法要对每个具体轨迹进行建模与存储,未利用路网对轨迹的限制,故空间性能较差. 针对该问题,首先对路网空间进行建模,继而探索个体轨迹的活动规律. 提出基于轨迹的空间信息和轨迹的时态信息相结合的轨迹间投影距离度量(SRTD);提出基于 SRTD 距离相似轨迹双层压缩算法(SDTC),实验表明,SDTC 算法相对于原始算法降低了存储空间开销;SDTC 算法精度较原始算法有较大改进.

关 键 词: 全球定位系统轨迹; 轨迹压缩; 路网空间; 轨迹距离
中图分类号: TN929.53 **文献标志码:** A

New Method for Road-Network GPS Trajectory Compression

LI Chuan[1,2,3], ZHANG Biao[1,2,3], LI Yan-mei[1], YANG Ning[1], WANG Yong[1,4]
(1. College of Computer Science, Sichuan University, Chengdu 610065, China;
2. State Key Laboratory of Software Engineering of Wuhan University, Wuhan 430072, China;
3. National Key Laboratory of Air Control Automation System Technology, Chengdu 610065, China;
4. Air Defense Forces Academy, Zhengzhou 450000, China)

Abstract: The traditional trajectories compression methods handle each trajectory individually, but it does not take into account the actual route situations, so it shows limited space performance. To solve this problem, the route network model is designed, and regulations of these trajectories are deeply explored. The main contributions include: 1) proposing the distance measure SRTD (shadow reference trajectory distance) which incorporates the space and time information of trajectories together; 2) proposing an algorithm called SDTC (SRTD distance based trajectory compression), which compresses dual-layer trajectories based on SRTD distance similarities. Experiments show that, compared with traditional methods, SDTC algorithm significantly reduces the storage consumption, and is of good precision.

Key words: Global positioning system trajectory; trajectory compression; road network; trajectory distance

近年来随着基于位置的社交网络如 Foursquare 等的蓬勃发展以及带有 GPS 定位功能的智能移动设备的广泛普及,产生了大量需要存储和传输的轨迹数据.

在 GPS 轨迹数据压缩领域,目前已有很多相关的文献和算法. 通过对原始轨迹进行线性拟合来减少轨迹点的数量是一个重要的研究方向[1-5],其中与 Douglas-Peucker[2] 相关的研究比较多. 随后,Merat-

收稿日期: 2014-05-20
基金项目: 国家自然科学基金项目(61103043, 61173099, U1233118); 国家"十二五"科技支撑计划项目(2012BAG04B02); 武汉大学软件工程国家重点实验室开放基金项目(SKLSE2012-09-26)
作者简介: 李 川(1977—),男,副教授,硕士生导师,E-mail: lcharles@scu.edu.cn.

nia[3] 指出 Douglas-Peucker 算法并不适合 GPS 轨迹数据,因为 GPS 轨迹数据同时具有空间和时间两个属性,而 Douglas-Peucker 算法在计算误差时忽略了轨迹的时态信息. Meratnia[3] 提出一种新的距离计算方法同步欧式距离(SED, synchronous euclidean distance),且基于 SED 提出了一种新的自顶向下压缩算法(TD-TR, top down time ration). 为了描述方便,统一把基于 SED 的轨迹压缩算法称为 STC(SED based trajectory compression).

上述算法在轨迹压缩时,需对每条原始轨迹分别进行计算、存储,且未考虑路网空间的现实约束,因而压缩率较差、还原精度不高. 针对上述问题,通过探索个人活动的活动规律及轨迹在路网空间的受限移动特征,提出了一种基于轨迹点投影的轨迹间距离度量,并基于该度量提出了一种双层轨迹压缩算法. 实验表明,提出的方法在降低轨迹存储空间开销的同时,还能提高轨迹还原的精度.

1　基于轨迹点投影的轨迹间距离度量

为了描述方便,先给出一些相关符号的形式化定义.

定义 1　轨迹 T_i. $T_i=\{(x_1,y_1,t_1),(x_2,y_2,t_2),\cdots,(x_3,y_3,t_3)\}$ 表示一条轨迹,t_1 表示开始时间,t_n 表示终止时间,且满足 $t_1<t_2<\cdots<t_n$. 设 $|T_i|=n$ 表示轨迹点的数目,且满足 $n\geqslant 1$,$T_{i,j}$ 表示第 j 个轨迹点.

定义 2　轨迹 T_i 的相邻轨迹点的距离集合 SegDis(T_i) $=\{s_1,s_2,\cdots,s_{n-1}\}$,其中 s_j 为轨迹 T_i 中的轨迹点 $T_{i,j}$ 与 $T_{i,j+1}$ 之间的空间距离,计算公式为式(1). 规定 SegDis(T_i,k) 表示 T_i 第 k 段轨迹的长度.

$$s_j=\sqrt{((T_{i,j+1})x-(T_{i,j})x)^2+((T_{i,j+1})y-(T_{i,j})y)^2} \tag{1}$$

定义 3　轨迹 T_i 中由第 1 个轨迹点到第 j 个轨迹点组成的子轨迹的长度 Length(T_i,j),其中 j 满足 $1\leqslant j\leqslant|T_i|$ 其计算公式为式(2). 设 Length(T_i) 表示轨迹 T_i 的轨迹总长度.

$$\text{Length}(T_i,j)=\sum_{k=1}^{j-1}\text{SegDis}(T_i,k) \tag{2}$$

相应的轨迹 T_i 中从第 0 个轨迹点开始的长度为 d 的子轨迹的最后一个轨迹点 $j=\text{Index}(T_i,d)$,j 必须满足 $\text{Length}(T_i,j)\leqslant d\leqslant\text{Length}(T_i,j+1)$.

定义 4　轨迹 T_i 中第 j 个轨迹点与第 $j+1$ 个轨迹点所组成的子段中距离第 j 个点的空间距离为 r 的点 JPoint(T_i,j,r),其中 r 必须满足如下条件: $0\leqslant r\leqslant\text{SegDis}(T_i,j)$.

定义 5　轨迹 T_i 中距离第 0 个轨迹点距离为 e 的空间点 DisPoint(T_i,e),计算公式为式(3),其中 e 必须满足如下条件: $0\leqslant e\leqslant\text{Length}(T_i)$.

$$\text{DisPoint}(T_i,e)=\text{JPoint}(T_i,\text{Index}(T_i,e),e-\text{Length}(T_i,\text{Index}(T_i,e))) \tag{3}$$

定义 6　轨迹 T_i 相对于轨迹 T_j 的基于轨迹点投影的轨迹间距离 SRTD(T_j,T_i). SRTD(shadow reference trajectory distance) 其意义是根据轨迹 T_i 的各个轨迹点与起始轨迹点间的距离,按照相应的距离在 T_j 中把按照定义 5 进行投影所得的点作为 T_i 中的相应点的替代点,所带来的最大误差. 其中 T_j 称为参考轨迹,由于必须在 T_j 中找到与 T_i 所对应的轨迹点,所以两条轨迹的长度必须相同.

现实的 GPS 轨迹,相同路网段的轨迹数据的长度不一定是等长的. 当被投影轨迹的轨迹距离大于参考轨迹的轨迹距离时,被投影轨迹的尾部的轨迹点不能在参考轨迹上找到相应的轨迹点与之对应,为了解决这个问题,引入轨迹间的长度因子

$$\text{DisFactor}(T_i,T_j)=\frac{\text{Length}(T_i)}{\text{Length}(T_j)} \tag{4}$$

引入轨迹间的长度因子之后,在计算两条轨迹的 SRTD 距离时,相应的轨迹间的 SRTD 距离计算公式为式(5),Eulers 表示两点间的欧式距离.

$$\text{SRTD}(T_j,T_i)=\max_{k,1\leqslant k\leqslant|T_i|-1}\text{Eulers}(\text{DisPoint}(T_j,\text{Length}(T_i,k)*\text{DisFactor}(T_j,T_i)),T_{i,k}) \tag{5}$$

定理 1　引入距离因子后,计算轨迹 T_i 相对于轨迹 T_j 的 SRTD 值时能够在 T_j 中找到轨迹 T_i 全部轨迹点的投影点. 从公式的定义可以看出原始轨迹 T_i 被投影后的轨迹点所组成的轨迹长度恰为 T_j 的轨迹长度,故证明从略.

2　基于 SRTD 距离的轨迹压缩算法

给定某人在路网中的两条轨迹 T_i 和 T_j,若 T_i 相对于 T_j 的 SRTD(T_j,T_i) 小于给定阈值 α,则可以结合 T_i 中的轨迹点按照定义 5、定义 6 的方法在 T_j 中进行投影,得到相应的投影轨迹 T_p. 这样 T_p 与 T_i 相对应的轨迹点之间的最大误差为 α. 因此可以只保

留 T_i 中的相邻轨迹点间的距离及时态信息，利用 T_j 还原时最大误差为 α. 存储 GPS 轨迹点的传统方法需要保存所有轨迹点的经度、纬度和时间，代价较大，而这种方法只需要存储原始轨迹中相邻轨迹点的距离信息和时态信息，很好地节省了轨迹的存储空间. 而且在保证精度的情况下，距离可以存为整数，而经纬度全部为浮点数，在进行存储时，这种方法需要的空间也更小.

基于上面的思想，提出了一个朴素的基于 SRTD 距离的轨迹压缩算法(N-SDTC，naive SRTD distance based trajectory compression). 算法的基本步骤如下:

1) 计算待压缩的轨迹集合中任意两条轨迹之间的 SRTD 值，若该值小于给定阈值，则把相应的轨迹及其 SRTD 值加入候选集合;

2) 对于候选集合的轨迹按照 SRTD 值进行升序排列;

3) 选取其中 SRTD 值最低的轨迹按照图 1 所示的数据结构进行压缩存储，并从候选集合中删除对应的两条轨迹;

4) 如候选集合不为空，继续第 3) 步，否则程序结束.

在现实基于路网的 GPS 轨迹数据中，由于 GPS 存在一定的误差，且道路交通状况不同，并非所有位于相同路网中的轨迹都能够用一条基准轨迹表示. 从后面的实验也可以看出，在误差相对较低的情况下，由于起始点的不同以及某些角度的差异，原始轨迹间的距离满足给定阈值的轨迹条数很少. 因此，N-SDTC 算法只适合理想的情况，不能对找不到基准轨迹的轨迹进行压缩，也不能对基准轨迹进行压缩.

由前述分析可知，两轨迹的 SRTD 值基本上是由两条轨迹的形状决定的，STC 压缩算法能保留大部分轨迹的空间形态信息. 因此，可首先利用 STC 算法对原始轨迹进行压缩，而后分析原始轨迹与其压缩后的轨迹，计算它们之间的 SRTD 距离. 若距离满足给定条件，则用该压缩后的轨迹作为原始轨迹的基准轨迹. 这样既可解决基准轨迹不能被压缩的问题，同时也能解决找不到基准轨迹时原始轨迹的压缩问题. 这就是提出的双层压缩算法 SDTC 的核心思想，这里的双层一共有 2 层: 第 1 层是利用 STC 算法先对轨迹进行简化压缩，第 2 层是利用 NSDTC算法对简化后的轨迹进行压缩.

根据上述思想，双层轨迹压缩算法 SDTC 的基本步骤如下:

1) 采用 STC 算法对原始轨迹进行压缩;

2) 分别计算原始轨迹与被 STC 压缩后的轨迹的 SRTD 距离;

3) 对于距离小于给定阈值的轨迹按照 N-SDTC 算法压缩原始轨迹，与 N-SDTC 算法不同的是，被压缩轨迹的参考轨迹为采用 STC 算法压缩后的轨迹.

图 1 给出了 SDTC 算法压缩后的轨迹数据结构示意图. 基准轨迹的 ID 编号表示为 2 个字节的整形，轨迹的起始时间表示为 4 个字节的无符号整型，最后的部分共有 $N-1$ 个 2 个字节大小的节点，N 表示原始轨迹中轨迹点的数量. 其中，第 i 个节点的前 1 个字节表示原始轨迹中的第 i 个轨迹段的长度，后 1 个字节表示原始轨迹中的第 i 个轨迹段的持续时间.

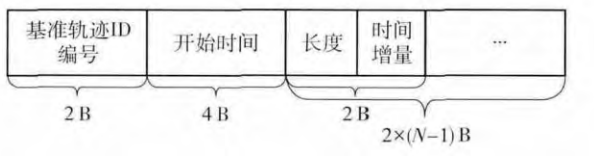

图 1 SDTC 算法压缩后轨迹的数据结构

通过实验观察发现，在真实的 GPS 轨迹数据中相邻的轨迹点之间的速度往往比较接近，因此为了进一步降低空间开销，在不增加误差的前提下，可以把具有相近速度的相邻轨迹点进行合并.

SDTC 算法中采取一种基于动态窗口的启发式规则: 若一个轨迹点的速度与窗口中所有轨迹点的平均速度相差小于给定阈值 β，则把该轨迹加入窗口，否则对窗口中的轨迹点进行合并. 经过轨迹点合并后的轨迹再计算 SRTD 距离，假设窗口中合并后的轨迹点在参考轨迹上相应的轨迹段上的速度是匀速的.

3 实验分析

实验数据来源于微软亚洲研究院提供的 Geolife[6] 数据集. 从编号为 2 的轨迹数据中选取如图 2 所示的 44 条轨迹数据. 编号为 2 的轨迹数据集共包括 121 条轨迹数据，去掉其中轨迹点数目小于 300 的数据后，剩余 106 条. 其中与图 1 所示的行为模式相似的轨迹共有 52 条，占轨迹总数的 49%，从数据方面初步证实了提出的轨迹数据确实存在大量重复的事实.

SDTC 算法的第一层压缩算法采用的是 TD-TR[4]. 为了消除增量表示方法给空间开销带来的影响,在进行对比试验时,对 TD-TR 算法压缩的轨迹也采用增量的方式进行存储. 对图 2 所示的 1 ~ 6 段各段轨迹分别采用两种算法进行压缩. 实验共设计了 7 组不同的误差阈值,所允许的最大误差分别为 5,10,15,20,25,30,40. SDTC 对应的合并相邻的轨迹点的速度阈值 β 分别为 0.5,1,2,2.5,2.5,3.5,3.5.

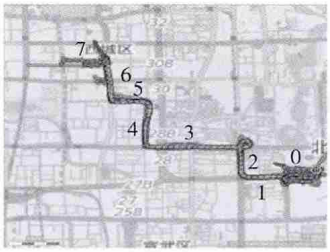

图 2　Geolife 中 ID 为 2 的个体部分行动轨迹

对能够找到基准轨迹的轨迹分别进行 SDTC 与 TD-TR 算法压缩,得到的存储空间开销比值如图 3 所示. 从图中可以看出,对于能够找到基准轨迹的轨迹,SDTC 算法能在大部分情况下降低 40% 以上的空间开销. 图 4 为 SDTC 算法与 TD-TR 算法的总空间开销比值. 从图中可以看出,SDTC 算法在 TD-TR 算法的基础上能够再降低 10% ~ 20% 的存储空间开销,并且随着最大误差阈值的增大,能够降低更多的空间开销.

表 1 给出了在不同的误差下,原始轨迹中能够被 SDTC 算法所压缩的轨迹与 TD-TR 算法压缩该

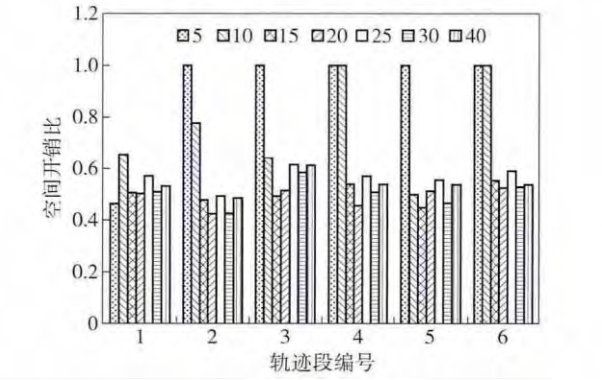

图 3　SDTC 算法中能找到基准轨迹的轨迹与 TD-TR 算法空间开销对比

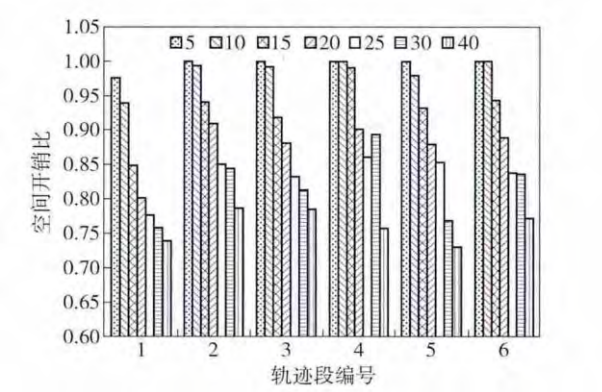

图 4　SDTC 算法与 TD-TR 算法整体的空间开销对比

轨迹所产生的误差的差值的平均情况. 从表中可以看出,SDTC 几乎都能够取得比 TD-TR 算法更小的误差,并且随着 TD-TR 算法误差的增大,SDTC 能够降低更多的误差,提高算法的精度.

表 1　SDTC 算法与 TD-TR 算法压缩产生的平均误差差值

最大误差阈值	轨迹段编号					
	1	2	3	4	5	6
5	-0.268 99	0	0	0	0	0
10	1.297 344	0.728 833	0.474 506	0	0.552 669	0
15	2.308 302	0.998 171	0.717 328	-0.058 9	2.158 625	1.654 621
20	3.707 629	0.406 911	2.518 647	1.142 568	2.844 604	2.613 288
25	5.809 257	3.361 314	4.582 261	4.227 379	5.797 486	4.459 084
30	5.499 344	3.028 887	4.743 676	4.481 521	3.860 391	2.322 462
40	12.556 02	8.681 574	12.07 549	7.502 455	7.666 013	10.016 43

4　结束语

针对个人活动的规律性以及路网空间中轨迹的空间形状规律性特点,提出了用 SRTD 距离来度量轨迹间进行映射时所带来的误差,并且基于 SRTD

(下转第 103 页)

不允许被替换,最大限度地保证了高热度内容的命中率.

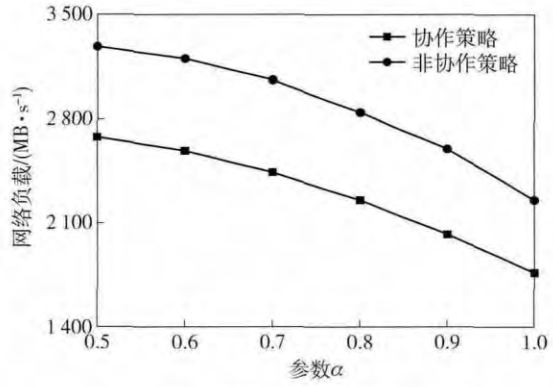

图 8　不同 α 情况下的网络负载

3　结束语

针对内容中心网络提出了一种基于分布式协作的缓存策略,建立网络中缓存节点的协作机制,有效减少网络重复流量,提高缓存利用率. 仿真结果表明,该策略在降低用户获取内容的平均时延,降低网络负载等方面取得了较好的性能.

参考文献:

[1] Jacobson V, Smetters D K, Thornton J D, et al. Networking named content [C] // Proceedings of the 5th Conference on Emerging Networking Experiments and Technologies. New York: ACM, 2009: 1-12.

[2] 林荣恒, 章晖, 邹华. 面向 SNS 用户访问行为的 Web 缓存预测替换 [J]. 北京邮电大学学报, 2012, 35(1): 111-114.
Lin Rongheng, Zhang Hui, Zou Hua. Web replacement policy based on user requests for SNS [J]. Journal of Beijing University of Posts and Telecommunications, 2012, 35(1): 111-114.

[3] Zhang Guoqiang, Li Yang, Lin Tao. Caching in information centric networking: a survey [J]. Computer Networks, 2013, 57(16): 3128-3141.

[4] Draxler M, Karl H. Efficiency of On-Path and Off-Path Caching Strategies in Information Centric Networks [C] // GreenCom 2012, Besancon: IEEE, 2012: 581-587.

[5] Hu Qian, Wu Muqing, Wang Dongyang, et al. Lifetime-based greedy caching approach for content-centric networking [C] // ICT 2014, Portugal: IEEE, 2014: 426-430.

[6] Carofiglio G, Gallo M, Muscariello L. On the performance of bandwidth and storage sharing in information-centric networks [J]. Computer Networks, 2013, 57(17): 3743-3758.

(上接第 97 页)

提出了 SDTC 双层轨迹压缩算法,实验结果表明 SDTC 能在降低空间开销的同时,降低压缩所带来的误差.

参考文献:

[1] Meyer T. Essential dynamics: a tool for efficient trajectory compression and management [J]. Journal of Chemical Theory and Computation, 2006, 2(2): 251-258.

[2] Douglas D H, Peucker T K. Algorithm for the reduction of the number of points required to represent a line or its caricature [J]. The Canadian Cartographer, 1973, 10(2): 112-122.

[3] Meratnia N, Rolf A. Advances in database technology-EDBT 2004 [M]. Berlin Heidelberg: Springer, 2004: 765-782.

[4] Cao Hu, Wolfson O, Trajcevski G. Spatio-temporal data reduction with deterministic error bounds [J]. The VLDB Journal – The International Journal on Very Large Data Bases, 2006, 15(3): 211-228.

[5] Muckell J, Hwang J H, Patil V, et al. SQUISH: an online approach for GPS trajectory compression [C] // Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications. [S. l.]: ACM, 2011: 13.

[6] Yu Zheng, Xie Xing, Ma Weiying. GeoLife: a collaborative social networking service among user, location and trajectory [J]. IEEE Data Eng Bull, 2010, 33(2): 32-39.