经典轨迹的鲁棒相似度量算法

王前东*

(中国电子科技集团公司第十研究所 成都 610036)

摘 要：针对经典轨迹与实时轨迹之间的大差异性，该文利用最长公共子序列理论，提出一种鲁棒的轨迹相似度量方法。该方法首先利用点到线段之间的距离判断经典轨迹的点与实时轨迹的线段是否一致；然后利用改进的多对1最长公共子序列算法，计算经典轨迹与实时轨迹之间的最长公共子序列长度；最后将最长公共子序列长度与经典轨迹的点数的比值作为经典轨迹与实时轨迹之间的相似度。实验说明该算法的鲁棒性，该算法能够有效解决经典轨迹与实时轨迹之间的大差异轨迹相似度量问题。

关键词：轨迹相似度量；大差异轨迹；多对1最长公共子序列；鲁棒计算；经典轨迹

中图分类号：TP301 **文献标识码**：A **文章编号**：1009-5896(2020)08-1999-07

DOI: 10.11999/JEIT190550

A Robust Trajectory Similarity Measure Method for Classical Trajectory

WANG Qiandong

(*No.*10 *Research Institute of China Electronics Technology Group Corporation*, *Chengdu* 610036, *China*)

Abstract: In view of the great difference between classical trajectory and real-time trajectory, a robust trajectory similarity measurement method is proposed based on the longest common subsequence theory. Firstly, the distance between point and line segment is used to judge whether the point of classical trajectory is consistent with the line segment of real-time trajectory. Secondly, the longest common subsequence length between classical trajectory and real-time trajectory is calculated by using the improved multi-to-one longest common subsequence algorithm. Finally, the ratio of the longest common subsequence length to the number of points of the classical trajectory is taken as the similarity between the classical trajectory and the real-time trajectory. Experiments show that the algorithm is robust and can effectively solve the problem of similarity measurement between the classical trajectories and real-time trajectories.

Key words: Trajectory similarity measurement; Great difference trajectory; Multi-to-one Longest Common Subsequence(m-1LCS); Robust computation; Classical trajectory

1 引言

经典轨迹是对目标活动规律的总结生成的综合轨迹[1]。将在线获取的实时轨迹与总结的经典轨迹进行比较，判断两轨迹是否相似，从而分析目标的行为意图或异常活动情况[2]，是经典轨迹的重要任务之一。

轨迹间的比较主要通过轨迹之间的距离来表示。文献[3]利用相同时间的欧式距离，判断轨迹是否相关，该方法准确度较高，但该方法需要利用轨迹时间。文献[4]丢弃轨迹时间，增加轨迹位置顺序特征，提高轨迹识别率，但该方法需要大量样本进行训练。文献[5]不用轨迹时间，直接计算轨迹之间的欧氏距离。欧氏距离计算简单，但要求两轨迹必须有相同的长度。文献[6–8]分别采用动态时间弯曲距离、Frechet距离、Hausdorff距离，能够适应不同长度轨迹，但这些算法都要求轨迹不能有强噪声。文献[9]利用编辑距离来适应强噪声的处理，但编辑距离计算的轨迹相似区间是离散的，不易被直观理解。文献[10]利用最长公共子序列(Longest Common Subsequence, LCS)距离，用实验说明了LCS对强噪声的鲁棒性，但该文未考虑轨迹断裂、轨迹非等周期等情况。文献[11]引入线段与线段之间的欧氏距离，用LCS算法来计算轨迹相似度，用实验说明了该算法能够处理非等周期轨迹，但该文未考虑经典轨迹与实时轨迹的大差异性问题，即经典轨迹的多个点可能与实时轨迹的同一条线段匹配的“多对1”问题。本文针对该问题，将传统的“1对1”LCS算法改为“多对1”的LCS算法，将点与点之间的欧氏距离改为点与线段之间的欧氏距离，提出了一种鲁棒的相似度计算方法。

收稿日期：2019-07-22；改回日期：2020-04-08；网络出版：2020-04-16
*通信作者： 王前东 wangqiandong@sohu.com

2 最长公共子序列长度基础算法

LCS算法最早由Wagner等人[12]提出，具有很好的抗干扰属性，在轨迹聚类、轨迹比较中得到了广泛的应用[13]，其定义如下所示。

定义1：1对1最长公共子序列(one-to-one Longest Common Subsequence, 1-1LCS)。

设两序列$\boldsymbol{C}_m=(c_1,c_2,\cdots,c_m)$与$\boldsymbol{Q}_n=(q_1,q_2,\cdots,q_n)$的公共子序列为$\boldsymbol{Z}_r=(z_1,z_2,\cdots,z_r)$，则$\boldsymbol{Z}_r$必须满足条件

$$\left.\begin{array}{l} z_s=c_{u_s}=q_{v_s},1\le s\le r \\ 1\le u_1<u_2<\cdots<u_r\le m \\ 1\le v_1<v_2<\cdots<v_r\le n \end{array}\right\} \tag{1}$$

$\boldsymbol{Z}_r$为两序列的1-1LCS，当且仅当$\boldsymbol{Z}_r$为满足式(1)条件中长度最长的序列。

根据最长公共子序列定义，文献[14]给出了如下的求最长公共子序列长度的算法。

设$\boldsymbol{C}_i=(c_1,c_2,\cdots,c_i)$为$\boldsymbol{C}_m$的子序列，$\boldsymbol{Q}_j=(q_1,q_2,\cdots,q_j)$为$\boldsymbol{Q}_n$的子序列，令$\boldsymbol{L}1(i,j)$表示序列$C_i$与序列$Q_j$的最长公共子序列长度，则$\boldsymbol{L}1(i,j)$由式(2)所示的1对1递推计算公式求出

$$\boldsymbol{L}1(i,j)=\begin{cases}\max\{\boldsymbol{L}1(i-1,j),\boldsymbol{L}1(i,j-1)\}, & i,j>0,\ c_i\ne q_j\\ \boldsymbol{L}1(i-1,j-1)+1, & i,j>0,\ c_i=q_j\\ 0, & i=0\text{ 或 }j=0\end{cases} \tag{2}$$

式(2)为1对1最长公共子序列长度算法，$\boldsymbol{L}1(i,j)$计算的时空复杂度为$O(mn)$。

3 改进的最长公共子序列长度算法

LCS基础算法中的最长公共子序列是1对1的最长公共子序列，即1个序列中至多只能有1个元素与另一个序列中的1个元素进行匹配，为了适应多对1情况，定义如下的多对1最长公共子序列。

定义2：多对1最长公共子序列(multi-to-one Longest Common Subsequence, m-1LCS)。

设两序列$\boldsymbol{C}_m=(c_1,c_2,\cdots,c_m)$与$\boldsymbol{Q}_n=(q_1,\cdots,q_j,\cdots,q_n)$的多对1公共子序列为$\boldsymbol{Z}_r=(z_1,z_2,\cdots,z_r)$，则$\boldsymbol{Z}_r$必须满足条件

$$\left.\begin{array}{l} z_s=c_{u_s}=q_{v_s},1\le s\le r \\ 1\le u_1\le u_2\le\cdots\le u_r\le m \\ 1\le v_1<v_2<\cdots<v_r\le n \end{array}\right\} \tag{3}$$

$\boldsymbol{Z}_r$为两序列$\boldsymbol{C}_m$与$\boldsymbol{Q}_n$的m-1LCS，当且仅当$\boldsymbol{Z}_r$为满足式(3)条件中长度最长的序列。

根据多对1最长公共子序列定义，可推出如下的多对1最长公共子序列长度计算公式，其推导过程略。

设$\boldsymbol{C}_i=(c_1,c_2,\cdots,c_i)$为$\boldsymbol{C}_m$的子序列，$\boldsymbol{Q}_j=(q_1,q_2,\cdots,q_j)$为$\boldsymbol{Q}_n$的子序列，令$L2(i,j)$表示序列$\boldsymbol{C}_i$与序列$\boldsymbol{Q}_j$的多对1最长公共子序列长度，则$L2(i,j)$由如式(4)所示的递推计算公式求出

$$L2(i,j)=\begin{cases}\max\{L2(i-1,j),L2(i,j-1)\}, & i,j>0,\ c_i\ne q_j\\ L2(i-1,j)+1, & i,j>0,\ c_i=q_j\\ 0, & i=0\text{ 或 }j=0\end{cases} \tag{4}$$

式(4)为多对1最长公共子序列长度算法，$L2(i,j)$计算的时空复杂度为$O(mn)$。

4 经典轨迹的相似度量算法

为了说明将点与点之间的欧氏距离改为点与线段之间的欧氏距离，将1-1LCS改为m-1LCS的必要性，特提出点到点1对1、点到点多对1、点到线1对1、点到线多对1共4种算法。

4.1 点到点1对1的轨迹相似度量算法

根据最长公共子序列定义，有如下的点到点1对1的最长公共子轨迹定义。

定义3：点到点1对1的最长公共子轨迹(Point-to-Point one-to-one Longest Common Subtrajectory, P2P1-1LCS)。

设经典轨迹$\boldsymbol{C}_m=(c_1,\cdots,c_i,\cdots,c_m)$与实时轨迹$\boldsymbol{Q}_n=(q_1,\cdots,q_j,\cdots,q_n)$的点到点1对1公共子轨迹为$\boldsymbol{Z}_r=(z_1,z_2,\cdots,z_r)$，则$\boldsymbol{Z}_r$必须满足条件

$$\left.\begin{array}{l} z_s=c_{u_s},1\le s\le r \\ \mathrm{dis}(z_s,q_{v_s})<\varepsilon,1\le s\le r \\ 1\le u_1<u_2<\cdots<u_r\le m \\ 1\le v_1<v_2<\cdots<v_r\le n \end{array}\right\} \tag{5}$$

其中，ε为距离阈值，$\mathrm{dis}(z_s,q_{v_s})$为两点$z_s$与$q_{v_s}$之间的欧式距离。

$\boldsymbol{Z}_r$为两轨迹C_m与Q_n的P2P1-1LCS，当且仅当$\boldsymbol{Z}_r$为满足式(5)条件中点数最多的轨迹。

根据P2P1-1LCS定义，有如下的P2P1-1LCS长度计算的递推公式[15]。

令$\varepsilon>0$为距离阈值，$L1_\varepsilon(m,n)$为经典轨迹$\boldsymbol{C}_m$与实时轨迹$\boldsymbol{Q}_n$的P2P1-1LCS长度，则$L1_\varepsilon(m,n)$可由式(6)所示的递推公式求出

$$L1_\varepsilon(i,j)=\begin{cases}L1_\varepsilon(i-1,j-1,)+1, & \mathrm{dis}(c_i,q_j)<\varepsilon\\ \max\{L1_\varepsilon(i,j-1),L1_\varepsilon(i-1,j)\}, & \mathrm{dis}(c_i,q_j)\ge\varepsilon\end{cases} \tag{6}$$

其中，计算$L1$的初始值为：对任意$i\ge0$, $L1_\varepsilon(i,0)=0$；对任意$j\ge0$, $L1_\varepsilon(0,j)=0$。

令$f1$为经典轨迹$\boldsymbol{C}_m$与实时轨迹$\boldsymbol{Q}_n$的点到点1对1的轨迹相似度，则$f1$可由式(7)求出

$$f1_\varepsilon(C_m,Q_n)=\frac{L1_\varepsilon(m,n)}{m} \tag{7}$$

4.2 点到点多对1的轨迹相似度量算法

定义4：点到点多对1的最长公共子轨迹(Point-to-Point multi-to-one Longest Common Subtrajectory, P2Pm-1LCS)。

设经典轨迹$\boldsymbol{C}_m=(c_1,\cdots,c_i,\cdots,c_m)$与实时轨迹$\boldsymbol{Q}_n=(q_1,\cdots,q_j,\cdots,q_n)$的点到点多对1公共子轨迹为$\boldsymbol{Z}_r=(z_1,z_2,\cdots,z_r)$，则$\boldsymbol{Z}_r$必须满足条件

$$\left.\begin{array}{l} z_s=c_{u_s},1\le s\le r \\ \mathrm{dis}(z_s,q_{v_s})<\varepsilon,1\le s\le r \\ 1\le u_1\le u_2\le\cdots\le u_r\le m \\ 1\le v_1<v_2<\cdots<v_r\le n \end{array}\right\} \tag{8}$$

其中，ε为距离阈值。

$\boldsymbol{Z}_r$为两轨迹$\boldsymbol{C}_m$与$\boldsymbol{Q}_n$的P2Pm-1LCS，当且仅当$\boldsymbol{Z}_r$为满足式(8)条件中点数最多的轨迹。

根据P2Pm-1LCS定义，有如下的P2Pm-1LCS长度计算的递推公式，其推导过程略。

令$\varepsilon>0$为距离阈值，令$L2_\varepsilon(m,n)$为经典轨迹$\boldsymbol{C}_m$与实时轨迹$\boldsymbol{Q}_n$的P2Pm-1LCS长度，则$L2_\varepsilon(m,n)$可由式(9)所示的递推公式求出

$$L2_\varepsilon(i,j)=\begin{cases} L2_\varepsilon(i-1,j)+1, \\ \quad \mathrm{dis}(c_i,q_j)<\varepsilon \\ \max\{L2_\varepsilon(i,j-1),L2_\varepsilon(i-1,j)\}, \\ \quad \mathrm{dis}(c_i,q_j)\ge\varepsilon \end{cases} \tag{9}$$

其中，计算$L2$的初始值为：$L3_\varepsilon(i,0)=0$，$i\ge0$；$L3_\varepsilon(0,j)=0$，$j\ge0$。

$$L3_\varepsilon(i,j)=\begin{cases} L3_\varepsilon(i-1,j-1)+1, & \mathrm{dis}(c_i,\overrightarrow{q_jq_{j+1}})<\varepsilon,j=1 \\ \max\{L3_\varepsilon(i,j-1),L3_\varepsilon(i-1,j)\}, & \mathrm{dis}(c_i,\overrightarrow{q_jq_{j+1}})\ge\varepsilon,j=1 \\ L3_\varepsilon(i-1,j-1)+1, & \min\{\mathrm{dis}(c_i,\overrightarrow{q_{j-1}q_j}),\mathrm{dis}(c_i,\overrightarrow{q_jq_{j+1}})\}<\varepsilon,1<j<n \\ \max\{L3_\varepsilon(i,j-1),L3_\varepsilon(i-1,j)\}, & \min\{\mathrm{dis}(c_i,\overrightarrow{q_{j-1}q_j}),\mathrm{dis}(c_i,\overrightarrow{q_jq_{j+1}})\}\ge\varepsilon,1<j<n \\ L3_\varepsilon(i-1,j-1)+1, & \mathrm{dis}(c_i,\overrightarrow{q_{j-1}q_j})<\varepsilon,j=n \\ \max\{L3_\varepsilon(i,j-1),L3_\varepsilon(i-1,j)\}, & \mathrm{dis}(c_i,\overrightarrow{q_{j-1}q_j})\ge\varepsilon,j=n \end{cases} \tag{12}$$

其中，计算$L3$的初始值为：$L3_\varepsilon(i,0)=0$，$i\ge0$；$L3_\varepsilon(0,j)=0$，$j\ge0$。

令$f3$为经典轨迹C_m与实时轨迹Q_n的点到线1对1的轨迹相似度，则$f3$可由式(13)所示的公式求出

$$f3_\varepsilon(C_m,Q_n)=\frac{L3_\varepsilon(m,n)}{m} \tag{13}$$

4.4 点到线多对1的轨迹相似度量算法

定义6：点到线多对1的最长公共子轨迹(Point-to-Segment multi-to-one Longest Common Subtrajectory, P2Sm-1LCS)。

设经典轨迹$\boldsymbol{C}_m=(c_1,\cdots,c_i,\cdots,c_m)$与实时轨迹$\boldsymbol{Q}_n=(q_1,\cdots,q_j,\cdots,q_n)$的点到线多对1公共子轨迹为$\boldsymbol{Z}_r=(z_1,z_2,\cdots,z_r)$，则$\boldsymbol{Z}_r$必须满足条件

令$f2$为经典轨迹C_m与实时轨迹Q_n的点到点多对1的轨迹相似度，则$f2$可由式(10)所示的公式求出

$$f2_\varepsilon(C_m,Q_n)=\frac{L2_\varepsilon(m,n)}{m} \tag{10}$$

4.3 点到线1对1的轨迹相似度量算法

定义5：点到线1对1的最长公共子轨迹(Point-to-Segment one-to-one Longest Common Subtrajectory, P2S1-1LCS)。

设经典轨迹$\boldsymbol{C}_m=(c_1,\cdots,c_i,\cdots s,c_m)$与实时轨迹$\boldsymbol{Q}_n=(q_1,\cdots,q_j,\cdots,q_n)$的点到线1对1公共子轨迹为$\boldsymbol{Z}_r=(z_1,z_2,\cdots,z_r)$，则$\boldsymbol{Z}_r$必须满足条件

$$\left.\begin{array}{l} z_s=c_{u_s},1\le s\le r \\ \min\{\mathrm{dis}(z_s,\overrightarrow{q_{v_s-1}q_{v_s}}),\mathrm{dis}(z_s,\overrightarrow{q_{v_s}q_{v_s+1}})\}<\varepsilon, \\ \quad 1\le s\le r \\ \quad 1\le u_1<u_2<\cdots<u_r\le m \\ \quad 1\le v_1<v_2<\cdots<v_r\le n \end{array}\right\} \tag{11}$$

其中，ε为距离阈值，$\mathrm{dis}(z_s,\overrightarrow{q_{v_s-1}q_{v_s}})$与$\mathrm{dis}(z_s,\overrightarrow{q_{v_s}q_{v_s+1}})$为点到线段之间的欧式距离。

$\boldsymbol{Z}_r$为两轨迹$\boldsymbol{C}_m$与$\boldsymbol{Q}_n$的P2S1-1LCS，当且仅当$\boldsymbol{Z}_r$为满足式(11)条件中点数最多的轨迹。

根据P2S1-1LCS定义，有如下的P2S1-1LCS长度计算的递推公式，其推导过程略。

令$\varepsilon>0$为距离阈值，$L3_\varepsilon(m,n)$为经典轨迹$\boldsymbol{C}_m$与实时轨迹$\boldsymbol{Q}_n$的P2S1-1LCS长度，则$L3_\varepsilon(m,n)$可由式(12)所示的递推公式求出

$$\left.\begin{array}{l} z_s=c_{u_s},1\le s\le r \\ \min\{\mathrm{dis}(z_s,\overrightarrow{q_{v_s-1}q_{v_s}}),\mathrm{dis}(z_s,\overrightarrow{q_{v_s}q_{v_s+1}})\}<\varepsilon, \\ \quad 1\le s\le r \\ \quad 1\le u_1\le u_2\le\cdots\le u_r\le m \\ \quad 1\le v_1<v_2<\cdots<v_r\le n \end{array}\right\} \tag{14}$$

其中，ε为距离阈值，$\mathrm{dis}(z_s,\overrightarrow{q_{v_s-1}q_{v_s}})$与$\mathrm{dis}(z_s,\overrightarrow{q_{v_s}q_{v_s+1}})$为点到线段之间的欧式距离。

$\boldsymbol{Z}_r$为两轨迹$\boldsymbol{C}_m$与$\boldsymbol{Q}_n$的P2Sm-1LCS，当且仅当$\boldsymbol{Z}_r$为满足式(14)条件中点数最多的轨迹。

根据P2Sm-1LCS定义，有如下的P2Sm-1LCS长度计算的递推公式，其推导过程略。

令$\varepsilon>0$为距离阈值，令$L4_\varepsilon(m,n)$为经典轨迹C_m与实时轨迹Q_n的P2Sm-1LCS长度，则$L4_\varepsilon(m,n)$可由式(15)所示的递推公式求出

$$L4_\varepsilon(i,j)=\begin{cases}L4_\varepsilon(i-1,j)+1, & \mathrm{dis}(c_i,\overrightarrow{q_jq_{j+1}})<\varepsilon, j=1\\ \max\{L4_\varepsilon(i,j-1),L4_\varepsilon(i-1,j)\}, & \mathrm{dis}(c_i,\overrightarrow{q_jq_{j+1}})\geq\varepsilon, j=1\\ L4_\varepsilon(i-1,j)+1, & \min\{\mathrm{dis}(c_i,\overrightarrow{q_{j-1}q_j}),\mathrm{dis}(c_i,\overrightarrow{q_jq_{j+1}})\}<\varepsilon, 1<j<n\\ \max\{L4_\varepsilon(i,j-1),L4_\varepsilon(i-1,j)\}, & \min\{\mathrm{dis}(c_i,\overrightarrow{q_{j-1}q_j}),\mathrm{dis}(c_i,\overrightarrow{q_jq_{j+1}})\}\geq\varepsilon, 1<j<n\\ L4_\varepsilon(i-1,j)+1, & \mathrm{dis}(c_i,\overrightarrow{q_{j-1}q_j})<\varepsilon, j=n\\ \max\{L4_\varepsilon(i,j-1),L4_\varepsilon(i-1,j)\}, & \mathrm{dis}(c_i,\overrightarrow{q_{j-1}q_j})\geq\varepsilon, j=n\end{cases}\tag{15}$$

其中，计算$L4$的初始值为：对任意$i\geq 0$, $L4_\varepsilon(i,0)=0$; 对任意$j\geq 0$, $L4_\varepsilon(0,j)=0$。

令$f4$为经典轨迹$\boldsymbol{C}_m$与实时轨迹$\boldsymbol{Q}_n$的点到线多对1的轨迹相似度，则$f4$可由式(16)所示的公式求出

$$f4_\varepsilon(C_m,Q_n)=\frac{L4_\varepsilon(m,n)}{m}\tag{16}$$

4.5 鲁棒的相似度量算法性质

设任意经典轨迹为$\boldsymbol{C}_m$，实时轨迹为$\boldsymbol{Q}_n$，令$\varepsilon>0$为距离阈值，$f1$, $f2$, $f3$, $f4$分别为经典轨迹$\boldsymbol{C}_m$与实时轨迹$\boldsymbol{Q}_n$之间的点到点1对1轨迹相似度、点到点多对1轨迹相似度、点到线1对1轨迹相似度、点到线多对1轨迹相似度，则有如下性质成立：

(1) $f4\geq\max\{f3,f2,f1\}$, $f2\geq f1$, $f3\geq f1$。

(2) 当$\varepsilon\to\infty$时，有$f4=f2=1$，$3=f1=\min\{m,n\}/m$。

(3) 当$\boldsymbol{Q}_n$=$\boldsymbol{C}_m$时，有$f4=f3=f2=f1=1$。

上述性质证明略。

5 实验

为了检验该算法的鲁棒性，通过设计不同的距离门限、对数据进行不同的轨迹抽样删除及抽样扰动等不同情况，将点到线多对1轨迹相似度与其它3种算法(点到点1对1轨迹相似度[16]、点到线1对1轨迹相似度、点到点多对1轨迹相似度)进行比较，说明算法的有效性。

5.1 试验数据

实验所用轨迹数据中包含6个经典目标的经典轨迹和实时轨迹。轨迹是按时间从小到大排列而成的有序的轨迹点集，轨迹点为2维坐标位置。6个目标的经典轨迹和实时轨迹长度分别为107和156, 114和107, 177和160, 162和169, 125和126, 145和149, 107和114。6个目标中的实时轨迹与经典轨迹之间的轨迹长度不同，轨迹起止点不同，轨迹相邻两点的间距不同，能有效检验算法的鲁棒性。

5.2 不同距离门限的轨迹相似度量试验

为了检验算法中距离门限对算法的影响，令ε从1000 m逐步递增到10000 m，统计各算法在选取不同的距离门限ε时的轨迹相似度。经过统计，6个目标的经典轨迹与实时轨迹之间的4种算法的轨迹相似度如图1所示。

从图1可以看出：距离门限ε从1000 m逐步递增到10000 m时，点到线多对1算法的轨迹相似度是4种算法中相似度最高的，点到点1对1算法的轨迹相似度都是4种算法中相似度最低的，这验证了4.5节性质(1)的正确性，同时也说明了点到线多对1算法对距离门限是鲁棒的。

5.3 实时轨迹抽样差异的试验结果分析

为了更深入地检验算法的鲁棒性，将6条实时轨迹进行等间隔抽样删除。设轨迹删除率d为轨迹中删除的轨迹点数k与轨迹删除前总点数n之比值，即轨迹删除率$d=k/n$。令距离门限ε=5000 m，分别计算删除后实时轨迹与经典轨迹的轨迹相似度。经过统计，6个目标的4种算法在不同抽样删除率下的轨迹相似度如图2所示。

从图2可以看出：在轨迹删除率由0%增加到50%时，点到线1对1与点到点1对1的航迹相似度下降了30%以上；点到线多对1算法的轨迹相似度始终是最高的，且几乎不受轨迹删除的影响，这说明了点到线多对1算法对轨迹删除是鲁棒的。

5.4 实时轨迹增加位置扰动的试验结果分析

为了检验位置扰动对算法的影响，将6条实时轨迹进行等间隔抽样扰动。设轨迹扰动率d为轨迹中扰动点数k与轨迹总点数n之比值，即轨迹扰动率。令距离门限ε=5000 m，扰动的噪声大小r=20000 m，分别计算扰动后实时轨迹与经典轨迹的轨迹相似度。经过统计，6个目标4种算法在不同扰动率下的轨迹相似度如图3所示。

从图3可以看出:在轨迹扰动率由0%增加到50%时，点到线多对1算法、点到线1对1算法的轨迹相似度几乎不变，但点到线多对1算法始终是4种算法中相似度最高的，这说明了点到线多对1算法对轨迹扰动是鲁棒的。

6 结束语

两轨迹的相似度量是很多应用中的关键技术，如轨迹相似度查询、轨迹识别、轨迹聚类、轨迹预测、轨迹行为分析、轨迹频繁模式挖掘等。目前大部分研究成果主要是用两轨迹点与点之间的欧氏距离，再用经典的最长公共子序列算法计算两轨迹的最长公共子轨迹长度或两轨迹的相似度。计算时，

如果轨迹不同步，则会带来较大的计算误差。本文利用点到线段之间的欧氏距离替代传统的点与点之间的欧氏距离，再通过改进的多对1最长公共子序列算法替代传统的1对1最长公共子序列算法，使得经典轨迹的多个点可以与实时轨迹中的一个线段进行匹配，提高了轨迹的相似度量。在仿真实验中表

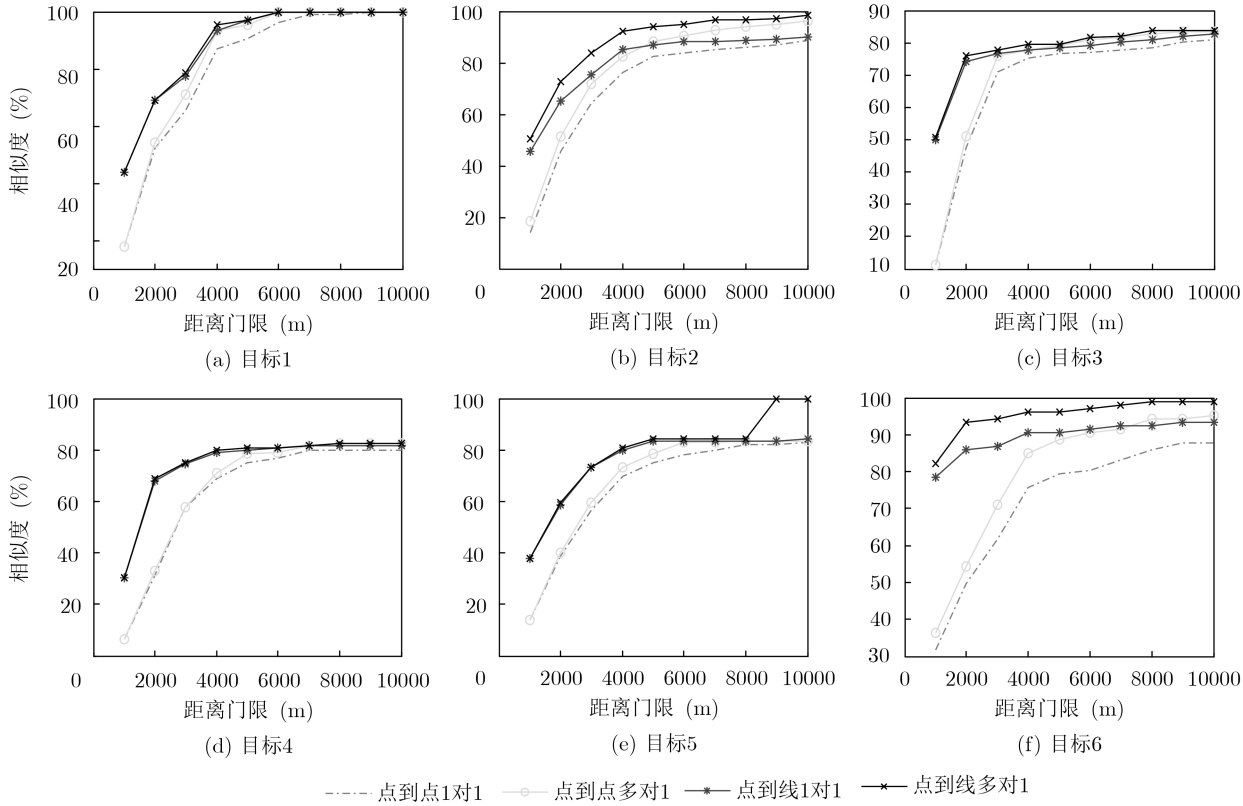

图 1 不同距离门限的轨迹相似度量

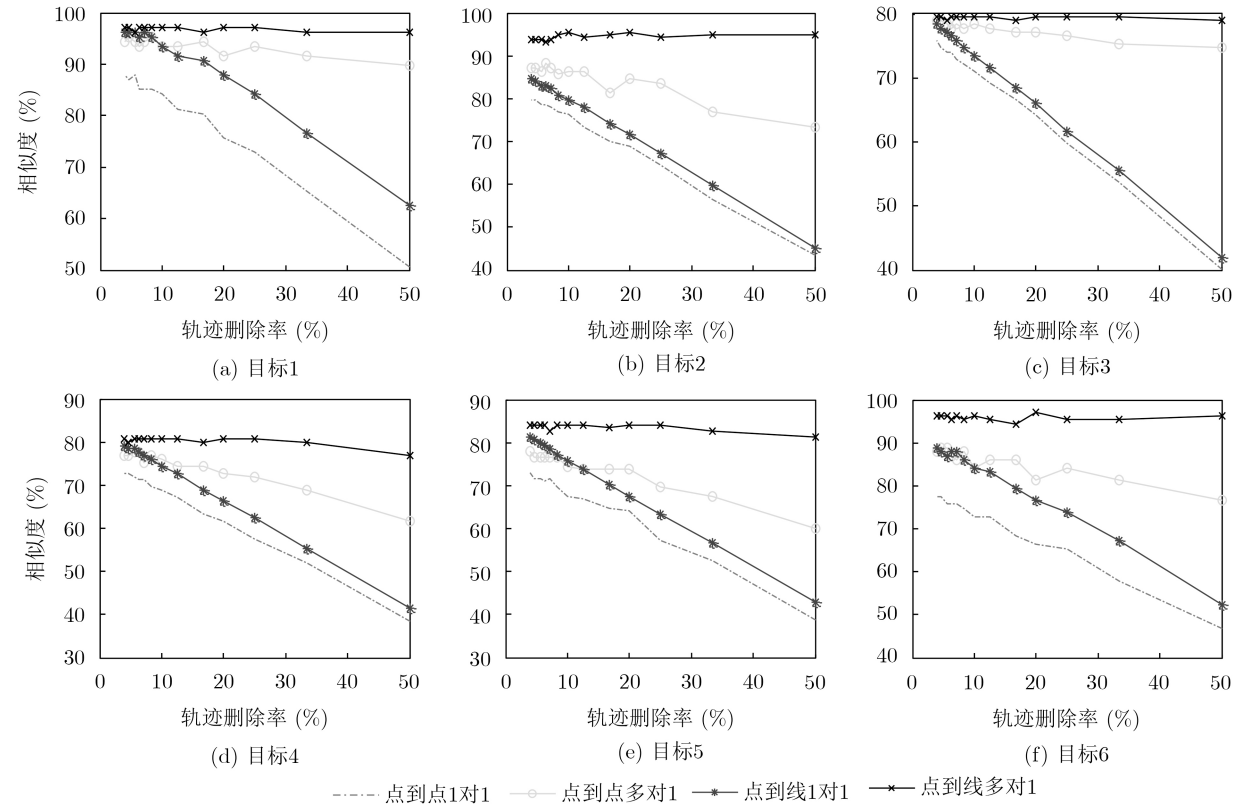

图 2 不同轨迹删除率的轨迹相似度量

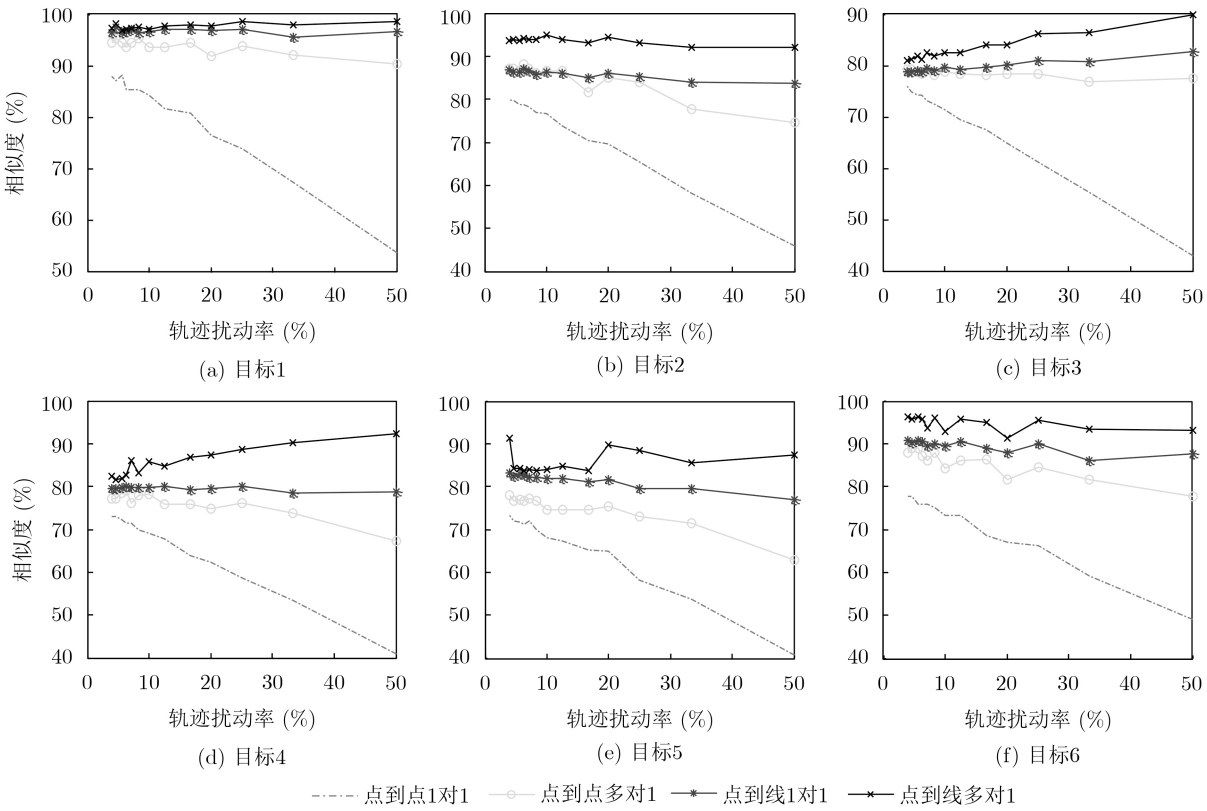

图 3 不同轨迹扰动率的轨迹相似度量

明，本文的改进方法更适用于经典轨迹的相似轨迹的度量计算，本文计算的点到线多对1的轨迹相似度比文献[16]计算的点到点1对1的轨迹相似度高，同时也说明了点到线多对1的轨迹相似度对不同的距离门限、不同的轨迹删除情况、不同的轨迹扰动情况都是鲁棒的。

本文算法主要基于经典轨迹的与实时轨迹的大差异性，提出的点到线多对1相似度算法，提高了轨迹相似度。如果待比较的两轨迹都不完整，则可以将其中的点到线段的欧氏距离替换为线段到线段的欧氏距离，但耗时更多，故本文未采用线段到线段之间的欧氏距离。如果待比较的两轨迹都比较完整，则点到点多对1相似度与点到线多对1相似度接近，但耗时更少。下一步将计划研究如何在保持精度时提高算法效率，减少算法处理时间，这也是当前工程中急需研究的课题。

参 考 文 献

[1] ANDRIENKO G, ANDRIENKO N, FUCHS G, *et al*. Clustering trajectories by relevant parts for air traffic analysis[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2018, 24(1): 34–44. doi: 10.1109/TVCG.2017.2744322.

[2] 毛嘉莉, 金澈清, 章志刚, 等. 轨迹大数据异常检测: 研究进展及系统框架[J]. 软件学报, 2017, 28(1): 17–34. doi: 10.13328/j.cnki.jos.005151.
MAO Jiali, JIN Cheqing, ZHANG Zhigang, *et al*. Anomaly detection for trajectory big data: Advancements and framework[J]. *Journal of Software*, 2017, 28(1): 17–34. doi: 10.13328/j.cnki.jos.005151.

[3] 李保珠, 张林, 董云龙, 等. 基于航迹矢量分级聚类的雷达与电子支援措施抗差关联算法[J]. 电子与信息学报, 2019, 41(6): 1310–1316. doi: 10.11999/JEIT180714.
LI Baozhu, ZHANG Lin, DONG Yunlong, *et al*. Anti-bias track association algorithm of radar and electronic support measurements based on track vectors hierarchical clustering[J]. *Journal of Electronics & Information Technology*, 2019, 41(6): 1310–1316. doi: 10.11999/JEIT180714.

[4] 陈鸿昶, 徐乾, 黄瑞阳, 等. 一种基于用户轨迹的跨社交网络用户身份识别算法[J]. 电子与信息学报, 2018, 40(11): 2758–2764. doi: 10.11999/JEIT180130.
CHEN Hongchang, XU Qian, HUANG Ruiyang, *et al*. User identification across social networks based on user trajectory[J]. *Journal of Electronics & Information Technology*, 2018, 40(11): 2758–2764. doi: 10.11999/JEIT180130.

[5] AGRAWAL R, FALOUTSOS C, and SWAMI A. Efficient similarity search in sequence databases[C]. The 4th International Conference on Foundations of Data Organization and Algorithms, Chicago, USA, 1993: 69–84.

[6] KEOGH E and RATANAMAHATANA C A. Exact

indexing of dynamic time warping[J]. *Knowledge and Information Systems*, 2005, 7(3): 358–386. doi: 10.1007/s10115-004-0154-9.

[7] GUO Ning, MA Mengyu, XIONG Wei, *et al*. An efficient query algorithm for trajectory similarity based on Fréchet distance threshold[J]. *ISPRS International Journal of Geo-Information*, 2017, 6(11): 326. doi: 10.3390/ijgi6110326.

[8] 魏龙翔, 何小海, 滕奇志, 等. 结合Hausdorff距离和最长公共子序列的轨迹分类[J]. 电子与信息学报, 2013, 35(4): 784–790. doi: 10.3724/SP.J.1146.2012.01078.

WEI Longxiang, HE Xiaohai, TENG Qizhi, *et al*. Trajectory classification based on Hausdorff distance and longest common subsequence[J]. *Journal of Electronics & Information Technology*, 2013, 35(4): 784–790. doi: 10.3724/SP.J.1146.2012.01078.

[9] 朱进, 胡斌, 邵华. 基于多重运动特征的轨迹相似性度量模型[J]. 武汉大学学报: 信息科学版, 2017, 42(12): 1703–1710. doi: 10.13203/j.whugis20150594.

ZHU Jin, HU Bin, and SHAO Hua. Trajectory similarity measure based on multiple movement features[J]. *Geomatics and Information Science of Wuhan University*, 2017, 42(12): 1703–1710. doi: 10.13203/j.whugis20150594.

[10] VLACHOS M, KOLLIOS G, and GUNOPULOS D. Discovering similar multidimensional trajectories[C]. The 18th International Conference on Data Engineering, San Jose, USA, 2002: 673–684. doi: 10.1109/ICDE.2002.994784.

[11] 刘宇, 王前东. 基于最长公共子序列的非同步相似轨迹判断[J]. 电讯技术, 2017, 57(10): 1165–1170. doi: 10.3969/j.issn.1001-893x.2017.10.011.

LIU Yu and WANG Qiandong. Computing similar measure between two asynchronous trajectories based on longest common subsequence method[J]. *Telecommunication Engineering*, 2017, 57(10): 1165–1170. doi: 10.3969/j.issn.1001-893x.2017.10.011.

[12] WAGNER R A and FISCHER M J. The string-to-string correction problem[J]. *Journal of the ACM*, 1974, 21(1): 168–173. doi: 10.1145/321796.321811.

[13] CHOONG M Y, ANGELINE L, CHIN R K Y, *et al*. Modeling of vehicle trajectory clustering based on LCSS for traffic pattern extraction[C]. The 2nd IEEE International Conference on Automatic Control and Intelligent Systems, Kota Kinabalu, Malaysia, 2017: 74–79. doi: 10.1109/I2CACIS.2017.8239036.

[14] 王前东. 一种带匹配路径约束的最长公共子序列长度算法[J]. 电子与信息学报, 2017, 39(11): 2615–2619. doi: 10.11999/JEIT170092.

WANG Qiandong. A matching path constrained longest common subsequence length algorithm[J]. *Journal of Electronics & Information Technology*, 2017, 39(11): 2615–2619. doi: 10.11999/JEIT170092.

[15] WANG Haoxin, ZHONG Jingdong, and ZHANG Defu. A duplicate code checking algorithm for the programming experiment[C]. The 2nd International Conference on Mathematics and Computers in Sciences and in Industry, Sliema, Malta, 2015: 39–42. doi: 10.1109/MCSI.2015.12.

[16] YUAN Guan, SUN Penghui, ZHAO Jie, *et al*. A review of moving object trajectory clustering algorithms[J]. *Artificial Intelligence Review*, 2017, 47(1): 123–144. doi: 10.1007/s10462-016-9477-7.

王前东：男，1977年生，高级工程师，研究方向为数据信息处理及应用.

责任编辑：余　蓉