

文章编号:1672-3961(2011)05-0069-07

基于经验修正策略的延时地图匹配算法

许靖, 蔡文学*, 黄晓宇

(华南理工大学物流工程系, 广东 广州 510006)

摘要:基于浮动车的交通流分析要求地图匹配快速而准确地处理 GPS 数据, 现有的地图匹配算法无法满足交通流分析的实时处理要求。本算法采用延时策略, 利用历史数据和最新数据及低速识别提取的静止点来指导延时点匹配, 同时利用延时点来修正最新点的匹配经验。使用广州市路网及广东省交通厅浮动车的数据进行实验, 结果证明在不同样本密度下, 算法匹配正确率都可以达到 80% 以上, 在精度、实时性和容错能力上均高于隐马尔科夫链匹配算法、时空匹配算法。

关键词:浮动车; 地图匹配; 交通流分析

中图分类号: TP391 **文献标志码:** A

An empirical correction strategy based delays map-matching algorithm

XU Jing, CAI Wen-xue*, HUANG Xiao-yu

(Department of Logistics Engineering, South China University of Technology, Guangdong 510006, China)

Abstract: A map-matching algorithm is an algorithm that reflects the location based on Global Positioning Data(GPS) to the digital road network. Also, the floating vehicle based traffic flow analysis requires a map-matching algorithm to processing GPS data rapidly and accurately, but the existing map matching algorithms cannot meet the real-time GPS data processing requirement of traffic flow analysis. The empirical correction strategy based delays map-matching algorithm uses the delay strategy to match GPS data, which means that the algorithm employs historical data and latest data and the static point extracted by low-speed identification to guide the delay GPS data to the match, while taking advantage of the latest delay point to fix the match empirically of the latest point. Experiments were performed on the road network of Guangzhou and the flow cars data of the Guangdong Provincial Transportation Department. The results showed that the correct match rate of the algorithm could reach more than 80% at different sample densities and the accuracy and real time and fault tolerance of the algorithm were higher than the hidden Markov matching algorithm and spatial and temporal matching algorithm.

Key words: floating vehicle; map-matching; traffic flow analysis

0 绪论

基于浮动车数据的交通流分析是近年来交通流分析的主流技术。而浮动车数据通常为 GPS 的定位数据, 而 GPS 数据是车辆或行人轨迹的离散化抽样, 往往是存在位置和抽样错误^[1]。因此, 在做交通流分析前, 必须先要对 GPS 数据进行预处理, 即利用地图匹配技术将基于经纬坐标系的 GPS 数据匹

配到其真实行驶道路上。所以地图匹配的准确度和实时性直接影响到交通流分析的准确性和实时性。

1 相关工作

1.1 地图匹配问题描述

定义物体的真实轨迹 $P = [p_1, p_2, \dots, p_t]$, 其中, 对于每一个点 $p_i = \{x, y, w, v, t\}$, 其中, x, y 为位 p_i 的坐标; v 为 p_i 点的瞬时速度; w 为 p_i 的速度方向; t

收稿日期: 2011-02-14

作者简介: 许靖(1988 -), 男, 广东河源人, 硕士研究生, 主要研究方向为智能交通与物流。 E-mail: jingxu8885@gmail.com

* 通讯作者: 蔡文学(1968 -), 男, 湖北仙桃人, 教授, 主要研究方向为智能交通与物流。 E-mail: ctwxcai@scut.edu.cn

为该点的时间。

则 GPS 轨迹 $\bar{P} = [\bar{p}_1, \bar{p}_2, \dots, \bar{p}_t]$ 为真实轨迹 P 的估计值,一个 GPS 点有 $\bar{p}_i = \{\bar{x}, \bar{y}, \bar{w}, \bar{v}, t\}$ 。

定义道路 $R = \{r_1, r_2, \dots, r_i\}$, r_i 为道路的分段,为道路的最小单元; $r_i = [n_1, n_2, \dots, n_j]$; $n = \{x, y, w\}$ 为道路的转弯点; x, y 为 n 的参考坐标系坐标, w 为道路转弯点 n 的方向。

定义 $r'(\cdot)$ 为道路分段到等距点的映射,若 $r'_i = [n'_1, n'_2, \dots, n'_j]$, 则 $r'_i(j) = n'_j$ 。

定义轨迹-道路映射 $G = [g_1, g_2, \dots, g_t]$, 其中, $i = 1, \dots, t$, $g_i = r_i(j)$ 为轨迹点 \bar{p}_j 与道路等距点 n'_j 的映射, 则地图匹配的工作可用下式描述:

已知观测值 \bar{P} , 求轨迹-道路映射函数 $f(\cdot)$, $\tilde{G} = f(\bar{P})$, 使得 $f(\cdot) = \operatorname{argmax} \|\tilde{G} - G\|_2$ 。定义运算 $|g_i - \tilde{g}_i|$, 若 $g_i = \tilde{g}_i$, 则为 1, 反之则为 0。

1.2 研究现状

在实际情况下,无法通过解数学模型得到问题的最优解,所以研究通常采用启发式算法对地图匹配进行求解。而目前根据使用的地图数据结构,地图匹配的研究可分为基于几何特征的单点地图匹配、全局拓扑地图匹配、局部拓扑地图匹配 3 大类^[6]。

基于几何特征的单点地图匹配只考虑单一 GPS 点与道路的几何关系:如距离、角度等,但不考虑道路的连接关系。在这方面的研究中,最早的研究就是 GPS 点-道路点的点对点匹配^[6],即对于 GPS 点,其最优匹配道路点为欧氏距离最小的点。为了精确计算点到道路的距离,点到线距离的匹配方法被提出,即对于 GPS 点,最优匹配道路为欧式距离最小的道路边。但无论是点到点的匹配方法还是点到线的匹配方法,都无法解决平行路匹配问题,见图 1。

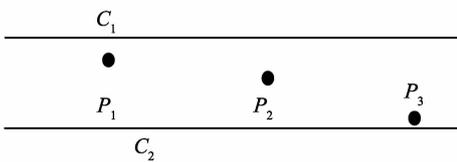


图1 点到线匹配问题

Fig. 1 Point to curve match problem

p_1, p_2, p_3 为一部车的 GPS 轨迹, c_1, c_2 , 显然,在以上的情况中,单点匹配是无法完全为 p_1, p_2, p_3 找到正确的匹配道路的。

所以有研究提出了全局地图匹配算法^[6],即将整条 GPS 轨迹或较大时间窗的 GPS 轨迹匹配拓扑路网。通过这种方法使得匹配结果保持拓扑。而当 GPS 数据的密度高,则可采用文献[3,5]中的算法,其中文献[5]通过判断在数据的高低速模式,和

在转弯模式下的轨迹变化来判断车辆行驶模式,通过识别车辆的转向信息来进行匹配,在数据密度很大的时候表现效果非常好;文献[3]采用转弯点识别及轨迹缓冲区来进行地图匹配。随着样本密度的下降,这类算法就无法识别轨迹的细节,匹配的效果就下降了。而针对低样本密度的算法有 ST 算法、HMM 算法等,大多基于 Fréchet 距离^[14], Fréchet 距离的基本模式定义两条直线的线距离或线距离函数,并对 GPS 轨迹的所有可能路线进行计算,使用权值和最小者为最优匹配路线。其中基于隐马尔科夫链的地图匹配算法^[4]认为 GPS 点的瞬时经验与移动经验是有关系的,隐马尔科夫地图匹配算法(hidden markov map-matching, HMM)^[4]证明 GPS 点间的距离位移差成指数分布,并将之作为评价指标,形成隐马尔科夫链模型;时空匹配(spatial and temporal map-matching, ST)^[7]在隐马尔科夫链模型的基础上,考虑了道路的速度变化与轨迹速度变化的关系,使得匹配更加接近真实道路的速度;文献[8]算法在 ST 算法的基础上,利用投票机制强化了点间距离的影响。

这类算法有以下缺点:(1)实时性差,即这类算法要求较大的时间窗,需要较大的时间跨度的数据进行匹配;(2)容错能力差,即如果将该类算法应用在实时性强的交通流分析系统中,必定需要缩短时间窗,这时一旦出现一点的误差过大,则可能使得时间窗内的 GPS 点全局匹配错误。

相对的,局部地图匹配就可以很好地实现实时匹配。其中文献[10,11]研究局部地图匹配算法,这类匹配的基本模式为当接收到一个 GPS 点的时候,立即处理该 GPS 点的匹配。文献[10]算法考虑 GPS 点与道路的距离和角度,同时使用历史经验计算出匹配总权重,作为匹配的依据。但是由于仅仅使用历史经验,这类算法无法解决“Y”字路匹配问题,见图 2。

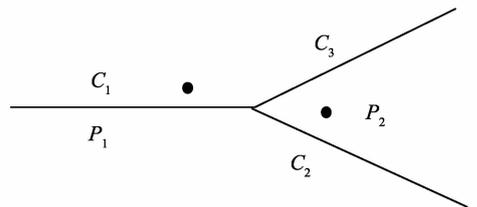


图2 实时匹配问题

Fig. 2 Real-time match problem

2 基于经验修正的延时匹配算法

2.1 经验修正的延时匹配算法概述

延时匹配算法的基本思想是,对任意一个 GPS 点,其匹配的道路必定与上一时刻和下一时刻的 GPS 点的匹配结果相关,而且它们之间的间隔时间越短,这两点的匹配结果的相互影响越大。

基于这个思想,定义最新到达匹配系统的 GPS 点为当前点;当前点上一时刻的点为延时匹配点,简称延时点;而延时点以前的点都为历史点,其中延时点上一时刻的点为最新历史点,见图3。

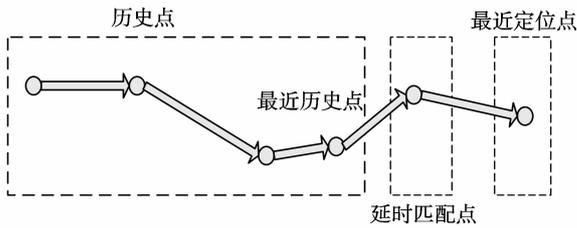


图3 历史点、延时点、当前点

Fig.3 History point, delays point and current point

对每一个延时点,算法使用最新历史点作为历史经验,当前点作为其“未来”经验,再根据匹配点与前后点的位移函数值作为经验影响因子来进行匹配。因为对每一个需要匹配的点,都必须等待它下一时刻的点到达才能进行匹配,所以称这种匹配策略为延时匹配策略。延时点匹配策略实质上属于动态时间窗的经验计算,即每到达一个 GPS 点,时间窗就往前移一位,同时匹配上一个 GPS 点。

但是,车辆往往到达十字路口处就会处于低速状态,在低速下,当前点的角度误差将被放大^[5],而且位置误差也会放大。这时利用这个当前点来匹配延时点显然是不适合的,见图4。

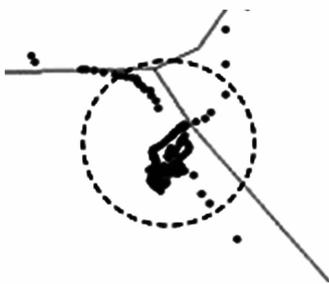


图4 低速 GPS 轨迹

Fig.4 Low-speed GPS trajectory

通过判断速度,可找出连续低速的 GPS 数据,用它们的期望位置作为静止点,并使用静止点进行匹配。称该策略为低速点检测策略。

低速点检测策略实质上属于可扩张的时间窗匹配,时间窗长度随着 GPS 流的低速状态而变化。

同时如果一个点完成了匹配,那它的匹配结果必定会影响到下一点的匹配,所以假设 GPS 点已经匹配到一条道路,则提高下一个 GPS 的候选路中

之相连接的道路,以修正下一时刻的 GPS 点的匹配,称之为匹配经验修正策略。

2.2 算法流程

基于延时匹配、低速检测和经验修正的策略,算法分成以下5个部分:选择候选集;速度模式识别;经验计算;延时点匹配及经验修正。

选择候选集的主要功能是为 GPS 数据提供匹配的候选道路。当收到 GPS 信息时,根据它的坐标信息提取缓冲区路网;然后利用角度信息对提取的路网进行过滤,生成候选路集。

算法采用低速模式识别收集低速的点,求出车辆估计位置。计算低速点将低速点聚类成一个整体进行匹配,消除低速点的匹配错误。

在当前点经验分析部分,采用 GPS 点与路网的距离、角度作为 GPS 点本身对路网的经验。同时采用当前点与延时点间的两点的位移和距离的比作为点间经验,使用这两个经验作为匹配的标准。

同时由于一个点的匹配是与该点的上一个 GPS 和下一个 GPS 点是相关的,所以在延时点匹配阶段,算法以最新历史点、延时点和当前点的自身经验和点间经验作为启发式经验来选择延时点的最优匹配道路。

最后阶段为修正当前点经验。由于延时点的匹配必定影响到当前点,所以为了体现这种影响,算法使用延时点的匹配结果对当前点的自身经验进行修正,使得在后续的匹配中,延时点的匹配更加趋向在最佳的道路中。

算法的流程如下

输入: 道路网络 R , GPS 轨迹流 \bar{P}

输出: GPS 点 \bar{p}_{i-1} 对应的最优的匹配道路 r_{i-1}^* ;

步骤 1 当接收到最新的 GPS 点 \bar{p}_i 时,计算候选道路集 R_i^* ;

步骤 2 判断 \bar{p}_i 的速度模式,如果为低速,进入步骤 3,否则进入步骤 4;

步骤 3 使用公式 1 计算静止点 s ,将 p_i 加入匹配序列 match points, $p_{i-1} = s$,进入步骤 1;

步骤 4 使用公式 2 计算 $Q(\bar{p}_{i-1})$,使用公式 6 计算 $S_h(p_{i-1})$ 和 $S_p(p_{i-1})$,使用公式 9 计算 r_m^* ,使用公式 10 更新 $Q'(\bar{p}_i)$,返回 r_m^* 和 match points,清空 match points。

下面详细介绍算法的各部分。

2.2.1 候选集选择

候选集选择的目的是缩小匹配范围,提高匹配的效率。算法采用缓冲区过滤^[3]和角度过滤两种方法来选择候选集。

考虑到缓冲区搜索效率,算法采用矩形缓冲区进行候选路。矩形缓冲区的原理是当一个 GPS 选择候选路时,取以 GPS 点为中心、半径为 r 的矩形,取得区域内的道路作为候选路集。使用矩形缓冲区运算速度快,同时粗糙的选集并不会给后续的运算带来庞大的运算量和算法精度的下降。考虑到 GPS 误差,采用半径为 20 m 的矩形缓冲区选择候选路网。

同时,由于在非低速下(即 $v > 5 \text{ km/h}$)时, GPS 角度与真实行驶方向的误差小于 15° ^[5],所以速度满足条件时,使用夹角差过滤可以进一步过滤所属候选集。计算 GPS 角度与道路夹角 Δw ,如果 $\Delta w > \theta_w$,则过滤该道路,其中 θ_w 是角度过滤阈值,通常的,对于城市道路, θ_w 设定为 $45^\circ \sim 60^\circ$ 。

2.2.2 速度模式识别

在真实 GPS 轨迹中,车辆低速模式是普遍存在的,在低速模式下, GPS 的角度误差会被放大^[5],所以这些 GPS 点的经验是无法指导延时点匹配,见图 4。但只要能检测出低速的点,通过求这些点的期望,就可以达到一个较准确的位置的点,这个点称为静止点。

速度识别的基本思想如下:通过判断 GPS 点间距离判断车辆是否进入低速状态,如果进入了低速状态,则进行静止点计算,并用静止点进行匹配,见图 5。

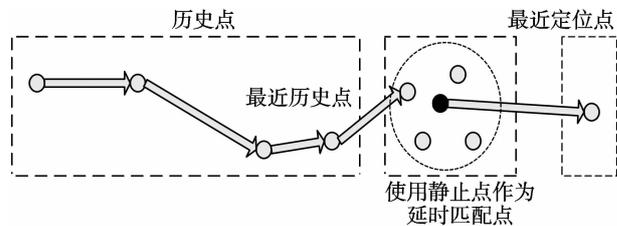


图5 将静止点当作真实 GPS 点进行匹配
Fig.5 Using a stay point as a real GPS point

根据延时匹配使用 GPS 数据流的特点,对静止点定义^[13]进行了修改:

静止点为一个具有缓冲区域的虚拟点 $s = \{x, y\}$, GPS 轨迹在这个缓冲区范围内,若停留的 GPS 轨迹为 $\bar{P} = [\bar{p}_m, \bar{p}_{m+1}, \dots, \bar{p}_n]$,必定有距离 $d(\bar{p}_k, s) \leq \theta_d$,时间差 $t(\bar{p}_k, \bar{p}_{k+1}) \leq \theta_t$ 。

$$\text{其中:} \begin{cases} x(s) \approx \sum_{i=m}^k x(\bar{p}_i) / (k-m+1), \\ y(s) \approx \sum_{i=m}^k y(\bar{p}_i) / (k-m+1). \end{cases} \quad (1)$$

$x(\cdot)$ 代表 \cdot 点的 x 坐标, $y(\cdot)$ 代表通常 \cdot 点的 y 坐标。通常根据 GPS 误差^[6], $\theta_d = 20/2 = 10 \text{ m}$, θ_t 为样本密度,定义样本密度 θ_t 为 GPS 点间时间差的期望。得到新的坐标后,修改静止点坐标。

低速下,算法使用静止点进行匹配运算,如果静

止点匹配成功,则静止点所含的点同时匹配到该道路上。

2.2.3 经验计算

对任意的 GPS 点,它对它周围的路网是存在经验的,主要的经验有距离经验和角度经验,这两个经验形成了该点对道路的经验。

首先定义 GPS 点 \bar{p}_i 对道路 r 的经验值:

$$Q(\bar{p}_i, r) = \alpha \times D(\bar{p}_i, r) + \beta \times V(\bar{p}_i, r), \quad (2)$$

$D(\bar{p}_i, r)$ 为基于距离的经验,根据文献[11]的定义:

$$D(\bar{p}_i, r) = \frac{1}{1 + \left(\frac{d(\bar{p}_i, r)}{\sigma^2}\right)^k}, \quad (3)$$

$d(\bar{p}_i, r)$ 为 \bar{p}_i 对道路 r 的距离, σ^2 为定位误差的方差。以往的研究^[6]发现,根据 GPS 误差,通常默认为 $\sigma^2 = 20$, k 为距离敏感度,默认为 1。

距离是衡量 GPS 匹配的最佳标准,但在复杂的路网,往往会存在由于 GPS 误差导致 GPS 点更靠近错误的道路(见图 2)的情况,所以还需使用角度经验进行匹配。

基于角度的经验为

$$V(\bar{p}_i, r) = \cos[w(\bar{p}_i) - w(r)], \quad (4)$$

$w(\cdot)$ 代表 \cdot 点的方向角, $w(\bar{p}_i)$ 为 GPS 点的行驶方向, $w(r)$ 为线路的行驶方向。通常,认为距离经验高于角度经验,设定 $\alpha = 1.5, \beta = 1$,但如果该点为静止点,则设定 $\alpha = 2, \beta = 0$ 。

显然,因为存在 GPS 误差,单凭借单点对路网的经验是无法完成匹配的,所以还需要利用 GPS 轨迹的联通关系来进行匹配。而对于正确的匹配,两个相邻的 GPS 点间的位移跟两点间的距离的差为指数分布^[2]。而为了简化计算,使用距离和位移的比值来衡量两点的联通经验。在计算位移时,假设在一段很短的时间内(如 5 min),车辆都不会选择绕道行驶,而行驶最短路。

定义 $s_{ij} = s(r_i(\bar{p}_i), r_j(\bar{p}_n))$ 为联通经验, $r_i(\bar{p}_i)$ 表示 \bar{p}_i 的匹配候选集的第 i 条候选路,公式由文献[7]定义:

$$s_{ij} = \frac{d(\bar{p}_i, \bar{p}_n)}{l(r_i(\bar{p}_i), r_j(\bar{p}_n))}, \quad (5)$$

$d(\bar{p}_i, \bar{p}_n)$ 为两点的欧氏距离, $l(r_i(\bar{p}_i), r_j(\bar{p}_n))$ 为 $r_i(\bar{p}_i)$ 到 $r_j(\bar{p}_n)$ 的位移,由采用 A* 算法^[12] 求出。同时为了使得找到的最短路符合城市限速要求,计算位移时加上限速限制,使得最短路位移满足城市限速要求。

定义道路最大限速为 v_m ,可通过取匹配样本速度置信度 95% 的上界得出, $\Delta t = t(\bar{p}_j) - t(\bar{p}_i)$,其中

$t(\cdot)$ 代表 \cdot 点的时间戳,则 $l_m = v_m \times \Delta t$,表示两个GPS点间的最大行驶距离。则有: $l(r_i(\bar{p}_t), r_j(\bar{p}_t)) \leq l_m$,若无解则 $s_{ij} = 0$ 。历史联通经验 s_h^{ij} 、延时点与当前点的联通经验 s_p^{ij} 定义:

$$\begin{cases} s_p^{ij} = s[r_i(\bar{p}_t), r_j(\bar{p}_{t+1})], \\ s_h^{ij} = s[r_i(\bar{p}_{t-1}), r_j(\bar{p}_t)]. \end{cases} \quad (6)$$

2.2.4 延时点匹配及当前点经验修正

最优路匹配是匹配的核心环节,通过使用GPS的自身经验及上下点经验进行匹配。定义 $r_{ijk} = [r_j(\bar{p}_{t-2}), r_i(\bar{p}_{t-1}), r_k(\bar{p}_t)] = [r_j, r_i, r_k]$,表示候选轨迹集里面的一条轨迹,定义 \bar{p}_{t-1} 对道路 r_i 对特定的评估函数:

$$w(\bar{p}_{t-1}, r_{ijk}) = Q'(\bar{p}_{t-1}, r_i) + \gamma \times Q_S^{ijk}, \quad (7)$$

其中联通经验:

$$Q_S^{ijk} = s_h^{ij} \times Q'(\bar{p}_{t-2}, r_j) + s_p^{ik} \times Q(\bar{p}_t, r_k), \quad (8)$$

Q' 为修正后的路网经验, γ 为经验权重,根据对经验的重视程度调节,通常为1。

得到 $w(\bar{p}_{t-1}, r_{ijk})$ 后,最佳匹配轨迹为

$$r_m^* = \arg \max_{r^* \in R^*} w(\bar{p}_{t-1}, r^*). \quad (9)$$

$r_m^* = [r_{t-2}^*, r_{t-1}^*, r_t^*]$,则延时点 \bar{p}_{t-1} 的最佳匹配道路为 r_{t-1}^* 。根据延时点 \bar{p}_{t-1} 的最佳匹配道路为 r_{t-1}^* ,对 \bar{p}_t 点的经验进行修正,公式为

$$Q'(\bar{p}_t, r_k) = s_p^{*k} \times Q(\bar{p}_t, r_k) \times \delta + (1 - \delta) \times Q(\bar{p}_t, r_k), \quad (10)$$

s_p^{*k} 为 r_{t-1}^* 与 $r_k(\bar{p}_t)$ 的联通因子, δ 为修正因子,实验证明,当 $0.2 \leq \delta \leq 0.4$ 时匹配效果较好。

3 实验

3.1 地图数据

本次实验使用广州市的地图作为基础道路匹配数据,共有114935条联通边,55423个节点,地图长为48.081 km,宽为35.3367 km。

地图数据和GPS数据分布见图6。



图6 广州的地图数据及GPS分布

Fig.6 The road network and GPS trajectories of Guangzhou

3.2 GPS数据

本次实验数据来自广东省公安厅数据中心浮动车数据,在海量的数据中挑选出12部浮动车的数据,这些数据都经过人工匹配,共14208条数据。数量分布和速度分布见图7。

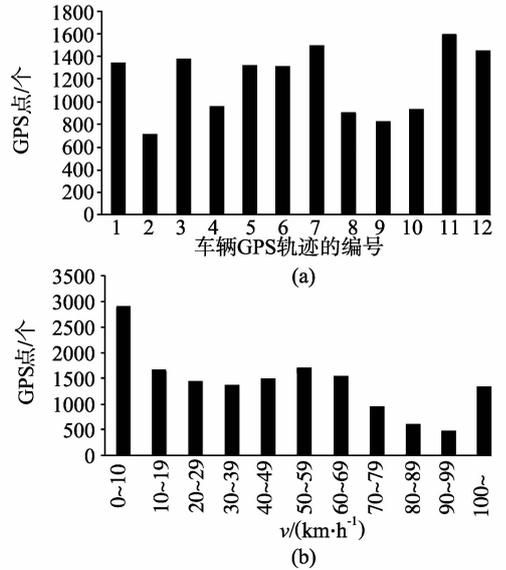


图7 GPS样本数据量和速度分布

Fig.7 GPS number of trajectories and velocity distribution

3.3 参数选择与运行环境

实验使用延时匹配算法(empirical correction strategy based deays map-matching, ECD)、时空匹配(spatial and temporal map-matching, ST)^[7]、隐马尔科夫地图匹配算法(hidden markov map-matching, HMM)^[4]作比较,ECD算法经验权重 γ 取值为0.3,定位误差的方差 σ^2 为20 m,距离敏感度 k 为1。ST算法的参数与文献[7]一致,HMM算法的参数与文献[2]一致。因为延时匹配算法是近似实时的匹配,为了算法公平,将HMM和ST算法的匹配窗限制为5。

实验用的电脑配置为Intel® Xeon(R) CPU E5520 2.27 GHz×2, 4 G内存。

3.4 正确性检验

为了评估匹配的精确度,本文使用正确匹配百分比(Z_{CMP})来衡量正确匹配率:

$$Z_{CMP} = \frac{R}{A} \times 100\%, \quad (11)$$

其中, R 为匹配正确的GPS点的数量, A 为所有匹配的GPS点的数量。

3.5 实验结果

3.5.1 正确匹配率

图8为EMD、ST、HMM的匹配结果的匹配正确率CMP,从图8看出,在5~20 s的样本密度中,延时匹配的匹配正确率远高于ST算法和HMM算法。

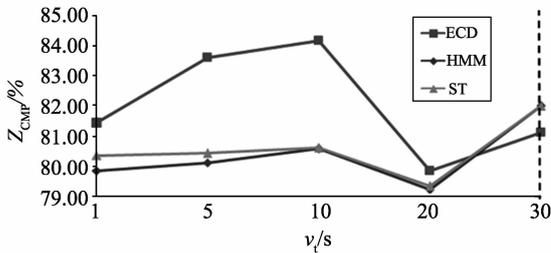


图8 不同样本密度下的CMP

Fig. 8 The precision of CMP in different sample density

而到了 20 ~ 30 s 的样本密度时,各算法的正确率持平,但因为延时算法使用的匹配时间窗短于 HMM 算法和 ST 算法,所以本算法对经验的利用率高于 HMM 算法和 ST 算法。

图9为匹配的可视结果。

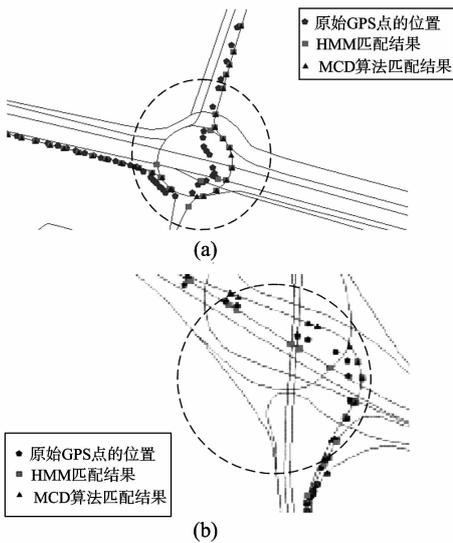


图9 匹配结果

Fig. 9 The results of matching on the map

由图9(a)可以看出,由于匹配经验得到纠正,所以延时算法匹配点的道路基本都在一条转弯上,而对于 HMM 算法,由于在一些点的 GPS 误差使得那些点接近错误的道路,导致匹配错误。

由图9(b)看出,由于本算法匹配采用动态时间窗,所以在转弯处,虽然轨迹偏向了下方的道路,但根据道路的联通, GPS 点能够正确匹配,而对于 HMM 算法,恰好时间窗就在转弯处,由于距离因素,导致 HMM 算法匹配全错。

本算法优于 ST 和 HMM 的原因有:(1) 由于使用动态时间窗,历史的经验动态传递到延时点的匹配;而对于 HMM 算法和 ST 算法,匹配时使用的匹配经验却是静态的,仅限在时间窗内。(2) 延时匹配算法不会直接匹配最新的 GPS 数据,因为最新的 GPS 数据是没有未来经验的;而 HMM 和 ST 算法每次都匹配最新的 GPS 点,导致最新 GPS 容易出现匹配错误。(3) 延时匹配算法采用动态可伸缩的

匹配时间窗,使得算法可根据实际情况调节匹配时间窗;而 HMM 和 ST 算法的时间窗却是固定的。

3.5.2 匹配效率

匹配效率的统计方法是统计每种算法运行 10 次的时间,然后取平均值,运行效率见图 10。

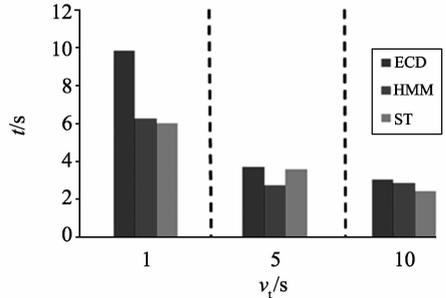


图10 匹配效率

Fig. 10 Matching efficiency

从图10看出,EMD 匹配的速度明显慢于 ST 和 HMM 匹配,主要原因在于延时匹配需要判断速度模式,这个过程的距离运算需要耗费一定的时间,但是 ST、HMM 算法没有以上的计算,所以效率高。但是随着间隔时间的增大,HMM 和 ST 计算最短路的开销使得延时算法的运行时间也慢慢接近其他两种算法。

3.5.3 容错能力

对于 ST 算法和 HMM 算法,每一个点都是与周围的多个点相依赖的,只要有一个点出错,那么其他点就很容易跟着出错,但是,ST、HMM 算法在匹配时是很难确定最后一点的匹配位置的,所以很容易会因为最后一点匹配问题而造成全局的匹配错误。由于 MCD 采用近似的实时匹配策略,并不立即处理当前点,使得算法的容错能力提升。在图 11 中可以看到,由于 GPS 误差导致有一小段 GPS 点偏离了正确的道路,HMM 算法在偏离处匹配全部错误;而对于延时匹配算法,根据历史经验,使得匹配的误差也延时了,同时,一旦当前经验恢复正常,匹配结果就立刻恢复正常了,说明延时匹配的匹配算法具有更强的容错能力。

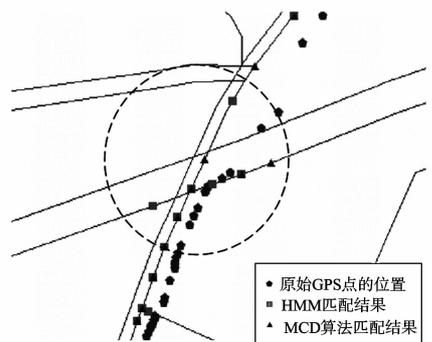


图11 匹配结果

Fig. 11 The result of map-matching on the map

4 结论

本研究提出基于延时修正的延时匹配算法,并使用角度贡献来提高匹配精度。通过固定历史经验,等到当前点到达后使用低速检验策略调节匹配时间窗,然后使用延时匹配策略匹配数据,并用匹配经验修正当前点经验。实验证明,本算法在使用比HMM和ST更小的时间窗进行匹配的情况下,匹配精度高于HMM、ST算法,而效率接近HMM和ST算法,而且具有更高的容错能力,能够满足交通流分析的实时性和准确性要求。

参考文献:

- [1] DIETER Pfoser, CHRISTIAN S Jensen. Capturing the uncertainty of moving-object representations[C]//Advances in Spatial Databases 6th International Symposium. Heidelberg: Springer-Verlag, 1999:111-131.
- [2] SOTIRIS Brakatsoulas, DIETER Pfoser, RANDALL Sallas, et al. On map-matching vehicle tracking data[C]//Proceedings of the 31st VLDB Conference. Trondheim, Norway: VLDB Press, 2005: 853-864.
- [3] MATT Weber, LING Liu, On map matching of wireless positioning data: a selective look ahead approach[C]//The 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York: ACM Press, 2010:290-299.
- [4] PAUL Newson, JOHN Krumm. Hidden markov map matching through noise and sparseness[C]//Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York: The ACM Press, 2009:336-343.
- [5] OCHIENG W Y, QUDDUS M, NOLAND R B. Map-matching in complex urban road networks[J]. Brazilian Journal of Cartography, 2003;55(2):1-16, .
- [6] QUDDUS M A, OCHIENG W Y, NOLAND R B. Current map-matching algorithms for transport applications: state-of-the art and future research directions[J]. Transportation Research C: Emerging Technologies, 2007; 15(5), 312-328.
- [7] LOU Yin, ZHANG Chengyang, ZHENG Yu, et al. Map-matching for low-sampling-rate GPS trajectories [C]//Proceedings of the 17th ACM SIGSPATIAL Conference on Geographical Information Systems. New York: ACM Press, 2010:352-361.
- [8] YUAN Jing, ZHENG Yu, XIE Xing. An interactive voting-based map matching algorithm[C]// Proceedings of mobile data management. Kansas: IEEE, 2010:43-52.
- [9] SYED S, CANNON M E. Fuzzy logic based map-matching algorithm for vehicle navigation system in urban canyon[C]// Proceedings of ION National Technical Meeting. California CA: ION Press, 2004.
- [10] KIM Sinn, KIM Jong Hwan. Adaptive fuzzy-network based c-measure map-matching algorithm for car navigation system[J]. IEEE Transactions on Industrial Electronics, 2001, 48(2):432-441.
- [11] DING Shengxi. Digital road maps based vehicle navigation system[D]. Beijing: Beijing University of Aeronautics and Astronautics, 2004.
- [12] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths [J]. IEEE Trans Systems Science and Cybernetics, 1968, 4(2):100-107.
- [13] ZHENG Yu, XIE Xing. Learning travel recommendations from user-generated GPS traces[J]. ACM Transactions on Intelligent Systems and Technology, 2011, 2(1):1-29.
- [14] ALT H, GODAU M, Computing the Fréchet distance between two polygonal curves[J]. International Journal of Computational Geometry and Applications, 1995, 5: 75-91.
- [15] QUDDUS M A, High integrity map-matching algorithms for advanced transport telemetric applications [D]. London: Imperial College London, 2006.

(编辑:陈燕)