

# IIVMM: 针对低频 GPS 轨迹的改进交互式投票匹配算法

严盛隆 于娟 周后盘

(杭州电子科技大学智慧城市研究中心 杭州 310018)

**摘要** 地图匹配是根据离散采样的定位数据(GPS 坐标)识别移动对象(车辆、行人等)在道路网络中的运动路径的过程。它是许多基于 GPS 轨迹数据分析和位置分析等相关应用的必要处理步骤。针对现有的算法在应用于低采样轨迹数据时存在的匹配准确率和效率较低的问题,文中提出一种基于交互式投票的改进地图匹配算法。该算法不仅考虑了距离特征、道路的拓扑结构以及路段的限速,还考虑了每个 GPS 点的实时移动方向和速度,以提高算法的匹配准确率。其次,该算法还加入了基于方向和限速的滤波器,通过约束条件过滤候选噪声路段,以提高算法的匹配效率。为了验证算法的性能,使用了两组真实数据集对所提算法与现有的 IVMM 算法和 AIVMM 算法进行比较。实验结果表明,所提算法在匹配性能上优于现有的两种算法。

**关键词** 地图匹配,低采样,方向,约束

**中图分类号** TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.09.050

## IIVMM: An Improved Interactive Voting-based Map Matching Algorithm for Low-sampling-rate GPS Trajectories

YAN Sheng-long YU Juan ZHOU Hou-pan

(Smart City Research Center, Hangzhou Dianzi University, Hangzhou 310018, China)

**Abstract** Map matching is the process of recognizing the movement track of moving objects (mainly vehicles, pedestrians) in the road network according to the discrete sampling location data (GPS coordinates). It is a necessary processing step for many relevant applications such as GPS trajectory data analysis and position analysis. This paper proposed an improved map matching algorithm based on interactive voting to solve the problems that the existing map matching algorithms have low accuracy and efficiency for low sampling trajectory data. The main contributions of the proposed algorithm are as follows. Firstly, the proposed algorithm considers not only the spatial distances between sampling points, road topology and road segment speed limits, but also the real-time moving direction and speed of each GPS point to improve the matching accuracy. Secondly, a filter based on driving direction and speed limits is introduced to filter out noisy candidates, thus improving the efficiency of the algorithm. To evaluate the performance of the proposed algorithm, two real-world datasets were used to compare the proposed algorithm with the existing IVMM algorithm and the AIVMM algorithm. Experimental results show that the proposed algorithm outperforms the existing two algorithms in terms of matching accuracy and efficiency.

**Keywords** Map matching, Low sampling rate, Direction, Constraints

### 1 引言

随着内置 GPS 传感器的电子移动设备的普及,每天会产生大量的行驶轨迹。然而,由于卫星定位系统的局限性,尤其是在卫星能见度有限、高楼林立的城市化地区,卫星信号容易被遮挡和折射,导致通过 GPS 传感器获取到的定位数据通常不准确,与实际位置存在误差,甚至发生数据丢失。此外,由于设备电量、存储容量和网络传输带宽的局限,我们实际收集到的轨迹数据都是离散的,且大多数是低采样率的,例如,两

个 GPS 位置采样点之间的采样间隔为 1 min 或以上。现实中,对于运动速度较快的车辆而言,它们的位置在两个采样时间间隔内会发生较大的变化,1 min 可以行驶相当长的距离。在路网密集的情况下,车辆在两个 GPS 采样点之间通过的路段会有多种可能性。这些因素导致我们无法仅根据 GPS 轨迹来确定相应移动对象的真实行驶路径。因此,在利用 GPS 轨迹数据进行与路网相关的分析和应用(车辆导航<sup>[1]</sup>、道路规划<sup>[2]</sup>、交通流量预测<sup>[3]</sup>等)之前,必须对其进行地图匹配的预处理,GPS 轨迹的处理过程如图 1 所示。

投稿日期:2018-08-24 返修日期:2018-11-16 本文受国家自然科学基金(61702148)资助。

严盛隆(1994—),男,硕士生,主要研究方向为数据分析和智能交通;于娟(1983—),女,博士,讲师,主要研究方向为数据挖掘和数据隐私保护,E-mail:yujuan@hdu.edu.cn(通信作者);周后盘(1967—),男,博士,教授,主要研究方向为智慧城市和人工智能应用。

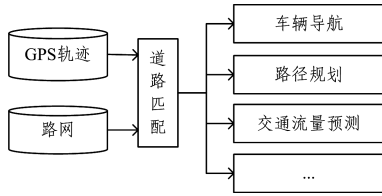


图1 GPS轨迹处理图

Fig.1 GPS track processing

在过去的近20年里,各种地图匹配算法被相继提出。根据采用方法的不同,这些地图匹配算法可分为简单的几何<sup>[4-5]</sup>算法、拓扑<sup>[6-7]</sup>算法及高级算法,其中高级算法是指采用模糊逻辑<sup>[8]</sup>、卡尔曼滤波<sup>[9]</sup>、隐马尔科夫<sup>[10-11]</sup>、条件随机场<sup>[12-13]</sup>、神经网络<sup>[14]</sup>、遗传算法<sup>[15]</sup>、蚁群优化<sup>[16]</sup>和机器学习<sup>[17]</sup>的算法。现有的这些地图匹配算法大多数专注于处理高采样率的GPS轨迹。相对而言,低采样率的GPS轨迹的地图匹配比高采样率轨迹的匹配更难,尤其是在城市复杂的路网中。

近几年,研究人员开始关注低频采样轨迹的处理。最具有代表性的算法是由郑宇等提出的ST算法<sup>[18]</sup>和IVMM算法<sup>[19]</sup>,后者在匹配准确率上优于前者,尤其是在处理低频数据时。ST算法使用轨迹中所有候选点权重的简单求和来评估候选路径与GPS数据的相似性,但当轨迹较长且车辆经过多车道道路时,ST匹配算法给出的匹配结果不可靠。而IVMM算法在采样率较低的情况下,充分考虑了道路网络的拓扑结构,以及全局GPS轨迹点之间的相互影响关系,且根据GPS轨迹点之间的距离对它们之间的影响程度进行加权处理,从而提高了匹配的正确率。然而,IVMM算法还可以在以下几个方面进一步优化。

1)IVMM算法考虑了每个GPS点之间的相互影响关系,当多个连续GPS点偏离正确的候选路径而接近错误的候选路径时,IVMM算法很有可能匹配到错误的候选路径上。

2)在计算局部最优路径时,IVMM算法沿用了ST算法中的最大权重法来求解最优路径,即权重的简单求和。如果一个点与一个错误的路段匹配,那么下面的点都会基于这个错误的匹配点来计算,这个过程的误差将被累积,最后错误路径的权重很可能大于正确路径的权重。

3)IVMM算法的过程比较复杂,且数据必须是重复匹配的映射,该算法无法兼顾匹配准确率与效率。

针对低采样率GPS轨迹数据的地图匹配存在的上述问题,本文提出了一种基于交互投票的改进地图匹配算法IIVMM(Improved Interactive Voting-based Map Matching)。该算法的创新之处主要在于:1)与IVMM算法相比,IIVMM算法加入了基于GPS实时方向的分析函数,考虑了方向因素,旨在找到一条与实时运动方向相似的路段,以提高匹配准确率;2)IIVMM算法中加入了新的滤波器,用于去除候选路段中的部分噪声,以提高算法的匹配准确率和匹配效率。

## 2 基本概念和问题描述

为了方便描述,首先定义一些基本概念和符号,然后对地图匹配问题进行形式化描述。

**定义1(GPS轨迹)** 一条GPS轨迹  $T: p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_n$  是由一系列GPS坐标点  $p_i (1 \leq i \leq n)$  构成的一个有序序列,其中每个GPS点  $p_i = (t, lat, lon, heading, speed)$ ,包含采样时间  $t$ 、经纬度坐标  $lat$  和  $lon$ 、GPS实时方向  $heading$  和速度  $speed$  等信息。

**定义2(路网)** 路网由一系列路口和连接路口的路段构成,可以表示为一个有向图  $G(V, E)$ ,其中  $V$  是顶点集,包含了路网中的所有路口或道路终点,  $E$  是有向边集(单向或双向),表示路网中的路段,其中每条路段  $e = (length, speed\ limit)$ ,  $e \in E$  包含了路段长度、限速等信息。

**定义3(路径)** 给定路网  $G(V, E)$ ,以及其中两个顶点  $r_i$  和  $r_j$ ,这两个顶点之间的一条路径  $P: e_1 \rightarrow e_2 \rightarrow \dots \rightarrow e_n$  是始于  $r_i$  终于  $r_j$  的一组首尾连接路段序列,其中  $e_1.start = r_i, e_n.end = r_j, e_k.end = e_{k+1}.start, 1 < k < n$ 。

地图匹配的问题可以描述为:给定某移动对象在路网  $G(V, E)$  环境中的一条GPS轨迹  $T$ ,从中推断出该移动对象的轨迹  $T$  在路网  $G$  中对应的真实运动路径  $P$ 。例如,图2(a)中有一条GPS轨迹(红点连成的折线)及对应的路网(绿色线条),地图匹配的目的是根据轨迹数据推断轨迹在路网中对应的路径,即图2(b)中的红色路段。

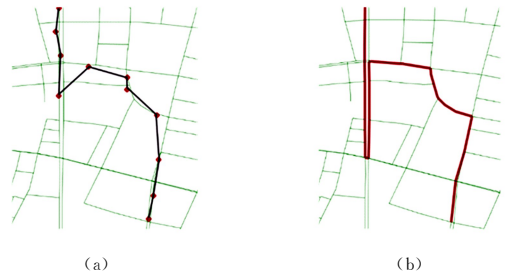


图2 地图匹配示例(电子版为彩色)

Fig.2 Toy example of map matching

## 3 IIVMM 算法

本文提出的地图匹配算法框架如图3所示,其大致分为4个步骤:候选集准备、位置环境分析、加权分析和交互式投票。本节最后对IIVMM算法进行了算法复杂度分析。

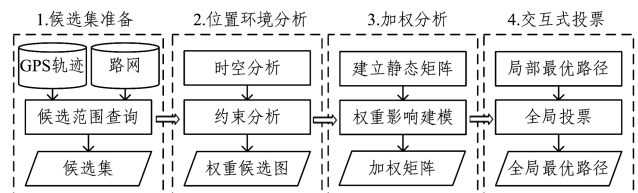


图3 IIVMM算法的框架

Fig.3 Framework of proposed IIVMM algorithm

### 3.1 候选集准备

候选集准备过程分为两个子步骤:1)建立路网数据的R-tree索引<sup>[20]</sup>,主要针对路段建立索引,以方便快速查找候选路段;2)基于路段的R-tree索引,快速查询轨迹  $T$  上每个采样点在路网中的所有可能候选路段,进而计算对应的候选点。

具体地,针对GPS轨迹  $T: p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_n$  中的每个采

样点  $p_i (1 \leq i \leq n)$ , 以 GPS 点  $(p_i, lon, p_i, lat)$  为圆心, 在路网中搜索半径  $r$  范围内的所有路段作为候选路段, 记为  $R_i = \{r_i^k | k=1, 2, \dots\}$ ,  $r_i^k$  表示 GPS 点  $p_i$  的第  $k$  个候选路段。对每个候选路段  $r_i^k$ , 通过计算轨迹点  $p_i$  到该路段上的投影得到对应的候选点, 记为  $c_i^k$ 。如果 GPS 点  $p_i$  在路段  $r_i^k$  的范围内存在垂点, 则选取该垂点作为轨迹点  $p_i$  的候选点; 否则, 选取该路段离轨迹点最近的起点或者终点作为候选点。如图 4 所示, GPS 点  $p_1$  拥有一条候选路段  $r_3$ , 对应有一个候选点  $c_1^1$ ; GPS 点  $p_2$  拥有 4 条候选路段  $r_1, r_2, r_3$  和  $r_4$ , 路段  $r_2$  和  $r_3$  对应的候选点  $c_2^2$  和  $c_2^3$  落在路段范围内, 而路段  $r_1$  和  $r_4$  对应的候选点  $c_2^1$  和  $c_2^4$  落在路段对应的起始节点。

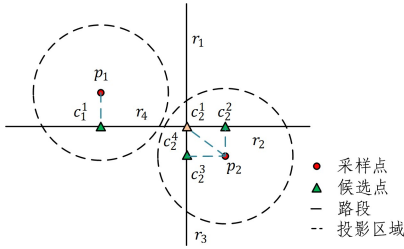


图 4 候选点选择示意图

Fig. 4 Schematic illustration of candidate points selection

### 3.2 位置环境分析

位置环境分析包括空间分析、时间分析和约束分析 3 个部分。

(1) 空间分析不仅要考虑单个 GPS 点与该点候选路段之间的距离和路网的拓扑信息, 还要考虑移动对象的实时方向 (即 GPS 采样点的方向) 和候选路段间方向的误差信息。

(2) 时间分析测量任何相邻点之间的实际平均速度, 然后将平均速度与每个候选路径上的速度约束进行比较, 得到一条拥有与 GPS 移动速度相似的限速路段。

(3) 约束分析通过考虑移动对象的运动方向与候选路段方向的差值, 以及移动对象的平均运动速度与路段限速的关系, 设置 3 个约束条件, 提前过滤错误的候选路径, 以减少额外的匹配开销, 提高匹配效率。

通过位置环境分析, 可构造出一个带权重的候选图, 图的节点是每个 GPS 候选点的集合, 边是相邻两个候选点之间最短路径的集合。节点和边都带有权重, 权重值的分配由位置环境分析结果确定。位置环境分析过程详见算法 1。

#### 算法 1 位置环境分析

输入: 路网  $G(V, E)$ , 轨迹  $T: p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_n$

输出: 权重候选图  $G''$

1. 初始化一个列表 candidate\_list 用于存放候选点
2. for  $i$  in  $(1, n)$
3. candidate\_list.append(Get\_candidate( $G, p_i, r$ ))
4.  $G' = \text{Constraint\_analysis}(\text{candidate\_list})$
5.  $G'' = \text{Spatial\_Temporal\_analysis}(G')$
6. return  $G''$

#### 3.2.1 空间分析

在空间分析中, 本文同时考虑路网的几何信息和拓扑信息来计算每个候选点的匹配概率, 即每个候选节点为对应时间移动对象的真实位置的可能性。

通常, 人们假设 GPS 采样点 (即观测位置) 与移动对象的实际位置的距离误差服从高斯分布  $N(\mu_1, \sigma_1^2)^{[10]}$ 。由此定义候选点的  $c_i^j$  的距离分析函数为  $N(c_i^j)$ :

$$N(c_i^j) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(X_i^j - \mu_1)^2}{2\sigma_1^2}} \quad (1)$$

其中,  $c_i^j$  是 GPS 点  $p_i$  的候选点,  $X_i^j$  是  $c_i^j$  与  $p_i$  的欧几里得距离, 即地面直线距离。

随着采样时间间隔的增加, 两个相邻 GPS 点连线的方向通常不能反映移动对象在对应的时间区间的真正移动方向。因此, 我们利用 GPS 的实时方向来评估移动对象的运动方向与路段方向的相似性。如图 5 所示, 如果只考虑 GPS 点和对应候选点的距离因素, 轨迹中的 GPS 点  $p_2$  和  $p_3$  将会匹配到路段  $r_2$ 。但是, 在考虑 GPS 实时方向和对候选路段之间的角度差  $\Delta\theta(c_i^j)$  后,  $p_2$  和  $p_3$  将会匹配到正确的路段  $r_3$ 。

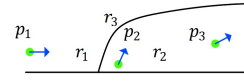


图 5 方向分析示意图

Fig. 5 Diagram of directional analysis

假设移动对象的实时方向与候选路段方向的夹角也服从高斯分布, 进而定义候选点的方向分析函数  $D(c_i^j)$  为:

$$D(c_i^j) = \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(\Delta\theta(c_i^j) - \mu_2)^2}{2\sigma_2^2}} \quad (2)$$

结合距离分析函数式 (1) 和方向分析函数式 (2), 定义观测概率  $O(c_i^j)$  为:

$$O(c_i^j) = N(c_i^j) \times D(c_i^j) \quad (3)$$

通过分析发现, 在有些情况下仅考虑距离与方向信息无法帮助我们选择正确的候选点。如图 6 所示, 路段  $r_1$  和  $r_2$  是平行的两个路段, 如果只考虑观测概率, GPS 点  $p_2$  会被匹配到错误的路段。但是, 综合考虑其前后 GPS 点的转移可能性时, 可以断定  $p_2$  应该在路段  $r_1$  上, 因为其对应的前后采样点  $p_1$  和  $p_3$  在路段  $r_1$  上。由此可见, 在匹配过程中还须考虑路网的拓扑信息, 以提高匹配结果的准确性。

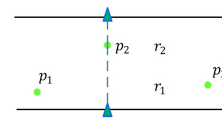


图 6 路网拓扑分析

Fig. 6 Topological analysis

为了在匹配过程中结合考虑路网拓扑信息, 定义转移概率, 其根据两个连续候选点之间的最短路径和直线路径的相似性来评估候选点为真实位置的概率。具体地, 两个候选点  $c_{i-1}^s$  和  $c_i^t$  之间的转移概率<sup>[13]</sup>  $V(c_{i-1}^s \rightarrow c_i^t)$  为:

$$V(c_{i-1}^s \rightarrow c_i^t) = \frac{\min(d(i-1, i), w(i-1, s), (i, t))}{\max(d(i-1, i), w(i-1, s), (i, t))} \quad (4)$$

其中,  $d(i-1, i)$  是两个相邻 GPS 采样点  $p_{i-1}$  和  $p_i$  之间的欧几里得距离,  $w(i-1, s), (i, t)$  是两个 GPS 点对应的两个候选点  $c_{i-1}^s$  和  $c_i^t$  之间的最短路径距离。本文使用 Dijkstra 算法<sup>[21]</sup> 计算最短路径。

结合观测概率式(3)和转移概率式(4),定义一个空间分析函数: $F_s(c_{i-1}^s \rightarrow c_i^s)$ 。

$$F_s(c_{i-1}^s \rightarrow c_i^s) = O(c_i^s) \times V(c_{i-1}^s \rightarrow c_i^s) \quad (5)$$

### 3.2.2 时间分析

在路网分布复杂的情况下,例如高速路段与普通路段平行且距离较近时,仅通过空间分析的权重大小通常难以区分车辆是行驶在高速路上还是普通路段上。因为当某个GPS点 $p_i$ 到一条普通路段和一条高速路段的距离近似时,空间分析函数会产生接近相同的权重(或概率)。现实中,普通路段和高速路段的限速是不同的,并且车辆的平均行驶速度不能超过道路限速。根据这一事实,我们可以通过车辆的行驶速度很好地区分其所在的行驶路段。例如,如果GPS采样点中测得的车辆时速约90 km/h,那么就将其与高速路段匹配。

根据上述分析,定义一个时间分析函数,结合采样速度与路段的限速来计算权重,它反映了车辆的平均行驶速度与路段限速的相似性,倾向于帮助我们选择行驶速度与限速最为相似的路段。具体地,已知候选点到 $c_{i-1}^s$ 到 $c_i^s$ 的最短路径为 $[r_1, r_2, r_3, \dots, r_k]$ ,时间分析函数 $^{[22]} F_t(c_{i-1}^s \rightarrow c_i^s)$ 定义为:

$$F_t(c_{i-1}^s \rightarrow c_i^s) = \frac{\hat{v}_{(i-1,s) \rightarrow (i,t)}}{|\hat{v}_{(i-1,s) \rightarrow (i,t)} - \bar{v}_{(i-1,s) \rightarrow (i,t)}| + \hat{v}_{(i-1,s) \rightarrow (i,t)}} \quad (6)$$

$$\bar{v}_{(i-1,s) \rightarrow (i,t)} = \frac{\sum_{i=1}^k l_{r_i}}{\Delta t} \quad (7)$$

$$\hat{v}_{(i-1,s) \rightarrow (i,t)} = \sum_{i=1}^k \frac{l_{r_i}}{l} \times v_{r_i} \quad (8)$$

其中, $\bar{v}_{(i-1,s) \rightarrow (i,t)}$ 为车辆从 $c_{i-1}^s$ 到 $c_i^s$ 的平均速度; $\hat{v}_{(i-1,s) \rightarrow (i,t)}$ 为最短路径的权重限速; $\Delta t$ 表示从候选点 $c_{i-1}^s$ 到候选点 $c_i^s$ 的时间差; $l_{r_i}$ 表示最短路径中第 $i$ 条路段的长度; $l$ 表示最短路径的总长度; $v_{r_i}$ 表示最短路径中第 $i$ 条路段的限速。

结合空间分析函数式(5)和时间分析函数式(6),定义车辆从候选节点 $c_{i-1}^s$ 迁移到候选节点 $c_i^s$ 的最终权重函数 $F(c_{i-1}^s \rightarrow c_i^s)$ 为:

$$F(c_{i-1}^s \rightarrow c_i^s) = F_s(c_{i-1}^s \rightarrow c_i^s) \times F_t(c_{i-1}^s \rightarrow c_i^s) \quad (9)$$

### 3.2.3 约束分析

实际上,在构造候选图的过程中还可以进一步滤除一些明显错误的候选节点。具体地,可以通过设置以下约束条件滤除一部分候选节点。

首先,我们认为GPS采集到的车辆实时运动方向与其所在路段的方向的差别不会太大。因此,采用实时运动方向与路段方向的差值作为一个约束条件滤除方向差别过大的候选节点,具体见约束1。

**约束1** 定义方向差阈值 $\theta$ ,如果车辆实时运动方向与对应路段方向之间的方向差值 $\Delta\theta(c_i^s) < \theta$ ,保留对应的候选点,并使用方向分析函数对候选路段分配权重;否则,当 $\Delta\theta(c_i^s) \geq \theta$ 时,直接将对应的候选节点滤除,后续的处理过程将不再考虑此候选点。阈值 $\theta$ 的设置可根据经验或统计数据进行设置,具体见实验环境参数设置。

其次,现实中车辆在路网中的运动速度都是在一个合理的范围内。根据这一事实,可以考虑通过分析车辆平均速度

与最短路径的权重限速的关系,将速度超过合理范围的对应候选路段去除,具体见约束2。

**约束2** 定义参数 $\alpha$ ,当 $\bar{v}_{(i-1,s) \rightarrow (i,t)} > \alpha \cdot \hat{v}_{(i-1,s) \rightarrow (i,t)}$ 时,将 $c_{i-1}^s \rightarrow c_i^s$ 视为错误的匹配路径,其中 $v_{(i-1,s) \rightarrow (i,t)}$ 为两个连续候选点之间的车辆平均速度, $\hat{v}_{(i-1,s) \rightarrow (i,t)}$ 为它们之间的最短路径的权重限速。如果采样点 $p_{i-1}$ 的所有候选点到 $p_i$ 的候选点 $c_i^s$ 全为错误匹配路径,将GPS点 $p_i$ 的候选点 $c_i^s$ 所对应的候选路段作为噪声路段过滤。参数 $\alpha$ 的设置见实验环境参数设置。

最后,现实中,车辆一般被允许超速15%左右,根据这一事实我们还可以设置一个超速极限值将部分不符合条件的候选路段去除。具体见约束3。

**约束3** 定义参数 $\gamma$ ,设 $v_i$ 为GPS点 $p_i$ 的实时速度, $v_j$ 是对应候选路段的限速,当 $v_i > (1 + \gamma) \cdot v_j$ 时,我们将对应的候选路段 $r_j$ 作为噪声路段滤除。参数 $\gamma$ 的具体设置见实验环境参数设置。

在构造候选图的过程中加入约束分析有以下两点优势:1)通过约束条件可以过滤一部分错误候选路段,因此匹配准确率会有一定的提高;2)过滤一部分错误的候选路段在很大程度上减少了加权分析和交互式投票过程的运行时间,从而提高了整个算法的匹配效率。

### 3.3 加权分析

位置环境分析得到的权重值可以表示成矩阵的形式,称之为静态矩阵 $^{[19]}$ 。加权分析通过考虑全局GPS轨迹点之间的相互影响关系,在位置环境分析的基础上进行权重影响建模,即对静态矩阵的加权。影响关系的强弱程度取决于GPS点之间的距离。距离越近,影响关系越强,反之越弱。加权分析产生 $n$ 个新的权重矩阵,称之为加权矩阵,其将被用于计算局部最优路径,其中 $n$ 表示一条轨迹中GPS采样点的个数。

位置环境分析产生的静态矩阵表示为 $\mathbf{M} = \text{diag}\{M^2, M^3, M^4, \dots, M^n\}$ ,其中, $M^i = (m_{ij}^i)_{a_i \times a_{i-1}} = (F(c_{i-1}^s \rightarrow c_i^s))_{a_i \times a_{i-1}}$ , $a_i$ 和 $a_{i-1}$ 分别代表第 $i$ 个和 $i-1$ 个GPS点的候选点个数。例如,给定GPS轨迹 $p_1 \rightarrow p_2 \rightarrow p_3$ ,其静态矩阵 $\mathbf{M}$ 如式(10)所示。它的值反映两个相邻候选点转移的可能性。矩阵静态矩阵只是一个中间结果,它只能反映两个相邻采样点之间的相互影响关系,但不能反映全局GPS轨迹点之间的相互影响关系。

$$\mathbf{M} = \begin{bmatrix} 0.6 & 0.7 & -\infty & -\infty \\ 0.4 & 0.5 & -\infty & -\infty \\ 0.3 & 0.1 & -\infty & -\infty \\ -\infty & -\infty & 0.8 & 0.6 \\ -\infty & -\infty & 0.2 & 0.5 \end{bmatrix} \quad (10)$$

为了建立权重影响模型,首先定义距离加权矩阵 $^{[19]} \mathbf{W}_i$ :

$$\mathbf{W}_i = \text{diag}\{\omega_i^1, \omega_i^2, \dots, \omega_i^{i-1}, \omega_i^{i+1}, \dots, \omega_i^n\} \quad (11)$$

其中, $dis(p_i, p_j)$ 是两个GPS点之间的欧氏距离(地面直线距离), $\beta$ 是一个道路网络参数。因为相互影响的强弱程度取决于GPS点之间的欧氏距离,且距离越近,影响越大,所以 $\omega_i^j$ 的计算可表示为:

$$\omega_i^j = e^{-\frac{(dis(p_i, p_j))^2}{\beta^2}}, 1 \leq i, j \leq n \quad (12)$$

距离加权矩阵给出了其他 GPS 点到当前 GPS 点  $p_i$  的距离影响权重, 每一个 GPS 点都有一个对应的距离加权矩阵。对于每个 GPS 点  $p_i$ , 结合位置环境分析得到的静态矩阵和基于 GPS 间距影响得到的距离加权矩阵, 可以得到一个加权矩阵  $\Phi_i^{[19]}$ :

$$\Phi_i = W_i M = \text{diag}\{\Phi_i^{(2)}, \Phi_i^{(3)}, \dots, \Phi_i^{(j)}\} \quad (13)$$

其中,  $\Phi_i^{(j)} = (\varphi_{st}^{i,j})_{a_{j-1} \times a_j}$ , 具体计算为:

$$\Phi_i^{(j)} = (\varphi_{st}^{i,j})_{a_{j-1} \times a_j} = \begin{cases} w_i^{(j-1)} M^j = (w_i^{(j-1)} m_{ts}^{(i,j)})_{a_{j-1} \times a_j}, & \text{if } 1 \leq j \leq i \\ w_i^{(j)} M^j = (w_i^{(j)} m_{ts}^{(i,j)})_{a_{j-1} \times a_j}, & \text{其他} \end{cases} \quad (14)$$

为了方便计算说明, 取  $w_i^j = 2^{-|i-j|} \neq e^{-\frac{\text{dis}(p_i, p_j)^2}{\beta^2}}$ 。例如, 在前面的例子(式(10))中, 可以得到 3 个加权矩阵  $\Phi_1, \Phi_2$  和  $\Phi_3$ 。以计算  $\Phi_3$  为例, 可得到如下加权矩阵:

$$\Phi_3 = \begin{bmatrix} w_3^{(1)} M^2 & 0 \\ 0 & w_3^{(2)} M^3 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} M^2 & 0 \\ 0 & \frac{1}{2} M^3 \end{bmatrix} = \begin{bmatrix} 0.15 & 0.175 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

矩阵  $\Phi_3$  是 GPS 点  $p_3$  的加权矩阵, 其中  $\Phi_3^{(0,1)}$  表示  $c_1^1 \rightarrow c_2^2$  的权重,  $\Phi_3^{(3,2)}$  表示  $c_2^2 \rightarrow c_3^3$  的权重。

### 3.4 交互式投票

交互式投票过程包括寻找局部最优路径和全局投票两部分。其中寻找局部最优路径就是通过最大权重法<sup>[18]</sup>查找一条必须通过指定候选点  $c_i^j$  的路径。全局投票过程是统计局部最优路径结果中每个候选点出现的次数, 将每个 GPS 点对应计数值最高的候选点连接组成最终的最优路径, 详见算法 2。

#### 算法 2 交互式投票

输入: 权重候选图  $G''$

输出: 最终地图匹配结果 Path:  $c_1^{s_1} \rightarrow \dots \rightarrow c_1^{s_n} \rightarrow \dots \rightarrow c_n^{s_n}$

1. 初始化一个数组  $\text{vote}[][]$  记录每个候选点的出现次数
2. compute  $M$
3. for  $i$  in  $(1, n)$
4. compute  $W_i$  和  $\Phi_i$
5. for  $k$  in  $(1, a_i)$
6. 初始化一个列表 LocalPath[] 记录分别经过每个候选点的局部路径  
LocalPath = Find\_LocalPath( $G'', \Phi_i, i, k$ )
7. if  $c_i^j$  in LocalPath then
8.  $\text{vote}[i][j] = \text{vote}[i][j] + 1$
9. Path.append( $\text{argmax}\{\text{vote}[i][j], i=1, 2, \dots, n; k=1, 2, \dots\}$ )
10. return Path

#### 3.4.1 寻找局部最优路径

每一个候选点  $c_i^j$  必定有一条经过它的路径, 将其称之为  $c_i^j$  的局部最优路径。通过枚举, 整个候选图中共有  $\sum_{i=1}^n a_i$  条局部最优路径, 其中  $a_i$  表示 GPS 点  $p_i$  候选点的个数。例如, 在图 7 所示的一条轨迹的候选图中, 假设  $c_2^2 \rightarrow c_3^3$  和  $c_2^2 \rightarrow c_3^3$  的权重为  $-\infty$ , 这就意味着整条路径必须经过  $c_3^3$ 。每个候选点

的累计权重计算为:

$$fValue(c_i^j) = \max\{fValue(c_{i-1}^j) + \varphi_{st}^{i,j}\} \quad (15)$$

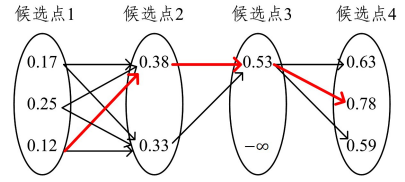


图 7 查找局部最优路径图

Fig. 7 Find local optimal path

对于每个候选点, 计算其累计权重后, 记录得到该候选点累计权重的前一个候选点  $c_{i-1}^j$ 。计算完所有采样点对应的候选点的累计权重之后, 由最后一个采样点的候选节点的累计权重得到最大累计权重值, 即:

$$f_{\max} = \text{argmax}\{fValue(c_n^j), 1 \leq i \leq a_n\} \quad (16)$$

如图 7 中, 最大累计权重值  $f_{\max} = fValue(c_4^2) = 0.78$ , 它来自于  $p_4$  的 3 个候选点。最终, 可通过全局最大权重值, 以及每个最大权重值记录的其前一采样点对应的候选点的位置, 得到权重最大的局部最优路径, 即图 7 中的加粗路径  $c_1^1 \rightarrow c_2^2 \rightarrow c_3^3 \rightarrow c_4^2$ 。

#### 3.4.2 全局投票

每找到一条局部最优路径, 对路径中出现的候选点进行计数。查找完所有的局部路径后, 得到一组拥有计数值的候选点。全局投票<sup>[19]</sup>就是将每个 GPS 点中计数值最大的候选点取出来组成全局最优路径。在上节例子, 最终可得到一组含有计数值的候选点, 如表 1 所列。  $c_1^1, c_2^2, c_3^3$  和  $c_4^2$  分别拥有其对应 GPS 点的候选点的最高计数值, 所以经过全局投票得到的全局最优路径如图 8 中加粗路径所示:  $c_1^1 \rightarrow c_2^2 \rightarrow c_3^3 \rightarrow c_4^2$ 。

表 1 投票计数表

Table 1 Vote count

候选点	$c_1^1$	$c_1^2$	$c_1^3$	$c_2^1$	$c_2^2$	$c_2^3$	$c_3^1$	$c_3^2$	$c_3^3$	$c_4^1$	$c_4^2$	$c_4^3$
计数值	2	1	7	9	1	2	8	1	8	1		

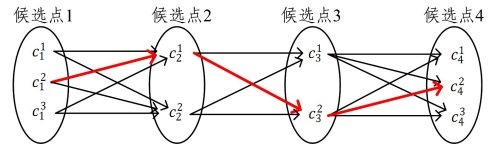


图 8 算法匹配结果

Fig. 8 Matching result of algorithm

### 3.5 算法复杂度分析

与 IVMM 算法类似, IIVMM 算法分 4 步进行地图匹配。设  $n$  为整条轨迹采样点的个数,  $m$  为路网中路段的个数,  $k$  为一个采样点最大候选点个数。算法第一步是候选路段查询过程, 此过程就是一个基于路段 R-tree 索引的  $k$ -近邻查询, 以获得每一个 GPS 点对应的  $k$  个候选路段, 这个过程一共执行  $n$  次。因此, 第一步的时间复杂度为  $O(nk)$ 。第二步, 通过位置环境分析建立权重候选图, 其中最耗时部分为转移概率的计算。整个候选图的构建过程共需计算  $(n-1)k^2$  个转移概率, 每个转移概率需要计算一次最短路径, 我们采用 Dijkstra 算法查询最短路径, 其时间复杂度为  $O(m \log m)$ , 因此第二步

的时间复杂度约为  $O(nk^2 m \log m)$ 。第三步加权分析可以并入第四步交互式投票过程中进行动态计算,因此在估计计算复杂度时可以仅考虑第四步的复杂度。交互式投票过程中最耗时的部分为加权权重矩阵的计算,其时间复杂度为  $O(nk^2)^{[19]}$ 。综上, IIVMM 算法总的的时间复杂度为  $O(nk^2 m \log m)$ ,其中  $n \ll m, k \ll m$ 。理论上该计算复杂度与 IVMM 类似。但是,在构建候选图的过程中本文引入了约束条件对候选点和路段进行过滤,其构造出的候选图比 IVMM 算法更简单,所以实际上 IIVMM 算法较 IVMM 算法更快。

## 4 实验

### 4.1 实验环境

#### 4.1.1 实验数据

本实验采用从 OpenStreetMap 下载的南京市路网作为路网测试数据。GPS 轨迹采用两组真实测试集,分别是南京市出租车的活动轨迹和 OpenITS 公开数据集,两组数据集轨迹主要包括以下信息:出租车 ID 信息、时间戳、经度、纬度、GPS 实时速度和 GPS 实时方向。对于数据集 1,随机抽取 5 条 GPS 轨迹作为测试数据,轨迹信息如表 2 所列。对测试数据的采样间隔进行统计,得到的结果如图 9 所示,采样间隔大部分在 30-150 s 内。数据集 2 的轨迹如图 10 所示,其由两条实验路线构成,红色是实验路线经过的普通道路部分,蓝色是实验路线经过下层道路的部分,绿色是实验路线经过高架桥的部分。其中,实验路线 1 包括 3360 个 GPS 采样点,实验路线 2 包括 3534 个 GPS 采样点,且数据集是以 1 s 为采样间隔,因此在实验中将数据集重采样成时间间隔为 30 s, 60 s, 90 s, 120 s 和 150 s 的 10 条轨迹作为测试数据。真实的对比数据是先通过 ST 算法<sup>[18]</sup>得到一条近似真实轨迹,然后经过人工修正匹配后得到。

表 2 轨迹信息表

Table 2 Trajectory information

组别	GPS 点数目	轨迹行驶时间
track1	259	4 h 49 min 13 s
track2	311	5 h 26 min 41 s
track3	392	6 h 32 min 01 s
track4	305	4 h 28 min 54 s
track5	357	5 h 53 min 27 s

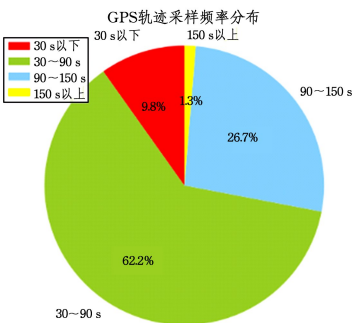


图 9 GPS 轨迹采样频率分布(电子版为彩色)

Fig. 9 GPS track sampling frequency distribution

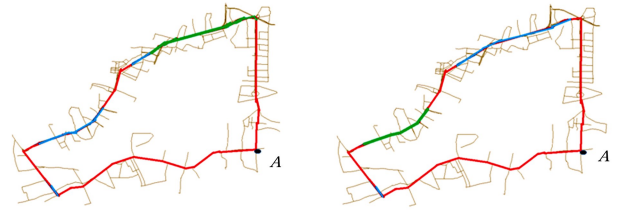


图 10 实验路线(电子版为彩色)

Fig. 10 Experimental routes

#### 4.1.2 参数设置

在候选集准备阶段,将缓冲半径  $r$  设为 120 m,且候选点个数不超过 12。在位置环境分析的时空分析过程中,在距离分析时将参数  $\mu_1$  设为 5 m,  $\sigma_1$  设为 20 m;在方向分析时将参数  $\mu_2$  设为 0,  $\sigma_2$  设为  $30^\circ$ 。在约束分析中,将  $\theta$  设为  $45^\circ$ ,  $\alpha$  设为 1.5,  $\gamma$  设为 15%。在权重影响建模中,将道路参数  $\beta$  设为 5 km。此外,本算法是在 Intel i5-7200u, CPU 2.5 GHz, 内存 8 GB, Windows 10 操作系统运行的 python 2.7 上编写并测试的。

#### 4.1.3 评估标准

IIVMM 算法是基于 IVMM 算法的改进算法,因此将 IIVMM 与 IVMM 算法在匹配准确率和效率上进行比较。此外,将本文算法与当前较优的 AIVMM 算法<sup>[22]</sup>进行比较。匹配算法的准确率(或精度)是正确匹配到真实行驶路段的 GPS 点所占的百分比,效率是比较算法在同一平台上的运行时间。本文使用匹配精度来评估算法的准确性。

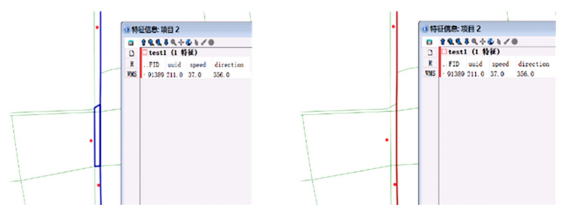
$$\text{匹配精度} = \frac{\text{匹配正确的 GPS 点}}{\text{需要匹配的 GPS 点}} \times 100\%$$

## 4.2 实验结果

### 4.2.1 可视化匹配结果

图 11-图 13 给出了 IIVMM 算法与 IVMM 算法测试同一条轨迹得到的可视化匹配结果对比。在这些图片中,橙色的点代表 GPS 采样点,绿线代表路网,红线和蓝线分别代表 IIVMM 算法和 IVMM 算法的匹配结果。下面是 3 种常见的可视化场景。

可视化场景 1 在没有考虑方向的情况下,经过 IVMM 算法匹配得到图 11(a)所示的错误匹配结果。图 11(a)中匹配错误的 GPS 点的方向是  $356^\circ$ (正北),而左边道路的方向大约在  $180^\circ$ 左右。在 IIVMM 算法中加入约束 1 的情况下,左边的平行错误候选路段被筛选出来并过滤。最后正确的匹配结果如图 11(b)所示。



(a) IVMM

(b) IIVMM

图 11 场景一

Fig. 11 Scenario one

可视化场景 2 IVMM 算法没有考虑车辆在路网中的运动速度是否在一个合理的范围内。当连续几个 GPS 轨迹远离正确的匹配路径时,IVMM 算法在全局影响下就会产生如图 12(a)所示的错误匹配结果,而实际上车辆无法在有限的时间从普通路段到达高速路段。IIVMM 算法在加入约束 2 后,图 12(a)右下的 GPS 点的所有候选点无法在一定时间内转移到高速路段上,因此可以根据约束 2 过滤高速路段,使得 GPS 点只能匹配到普通路段中的一条。最后的正确匹配结果如图 12(b)所示。

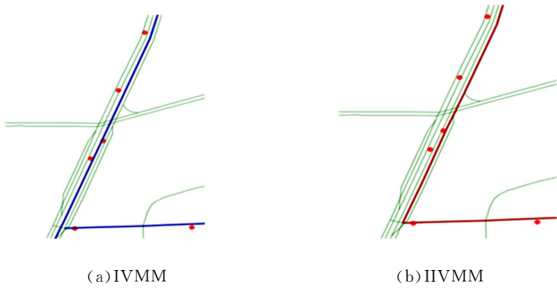


图 12 场景二  
Fig. 12 Scenario two

可视化场景 3 我们经常看到在复杂的城市路网中有高架道路,匹配的结果是车辆在高架道路上行驶。图 13(a)是 IVMM 算法的匹配结果,可以看到车辆从高架路下来,然后又马上又开到高架路,违背驾车出行常识,可推断这段匹配是错误的。而我们发现携带 GPS 设备的移动对象速度是 67 km/h,下方道路限速是 50 km/h,明显超出了约束 3 的条件,因此可以将下方道路作为错误候选路段过滤。最后的正确匹配结果如图 13(b)所示。

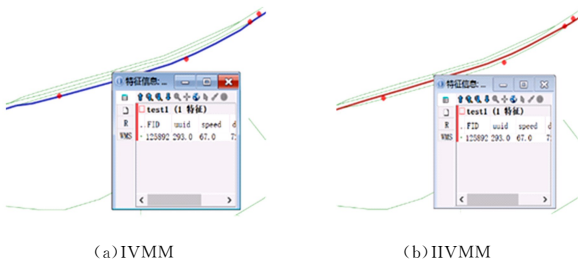


图 13 场景三  
Fig. 13 Scenario three

4.2.2 匹配性能对比

图 14 和图 15 是 IVMM 算法、AIVMM 算法和 IIVMM 算法在同一个平台用相同的测试数据得到的匹配性能对比。其中,图 14(a)、图 15(a)和图 15(b)反映了 3 种算法在匹配准确率上的差异。AIVMM 算法和 IIVMM 算法在匹配准确率上明显优于 IVMM 算法,约高出 3%~9%;且 IIVMM 算法的结果与 AIVMM 算法的近乎相同。实际上,IVMM 算法在稀疏路网中能得到较为准确的地图匹配结果,但在密集的道路网络中会产生错误的结果。IIVMM 算法在加入方向分析函数和 3 个约束条件后,有效改善了算法在复杂道路网中的匹配性能。

图 14(b)反映了 IVMM 算法、AIVMM 算法和 IIVMM 算法测试数据集一时每条轨迹的运行时间,而图 15(c)反映了 3 种算法运行同一频率下的轨迹平均运行时间。AIVMM 算法和 IIVMM 算法在匹配效率上明显优于传统的 IVMM 算法;同时,IIVMM 算法的匹配效率又高于 AIVMM 算法。IIVMM 算法和 AIVMM 算法的共同点在于通过去除噪点来提高整个算法的匹配性能;不同之处在于 AIVMM 算法通过统计权重函数中错误转移概率得到一个极限值作为约束条件来去除错误候选路径,而 IIVMM 算法在计算总权重值之前就通过方向极限值将大量的错误候选路段去除,这在很大程度上减少了计算权重概率所花费的时间。因此,IIVMM 算法的匹配效率优于 AIVMM 算法,且轨迹点个数越多,效果越明显。

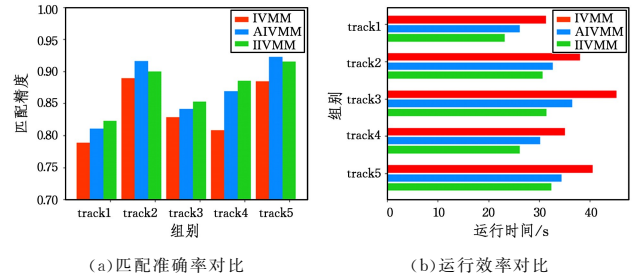


图 14 匹配性能对比(数据集一)

Fig. 14 Matching performance comparison(Dataset one)

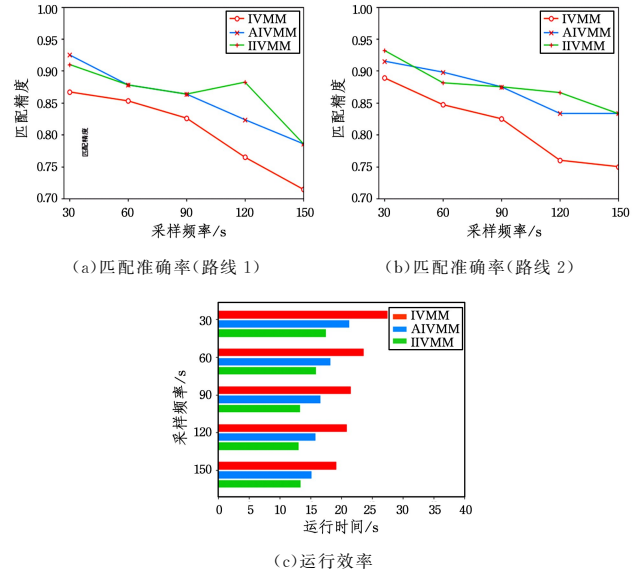


图 15 匹配性能对比(数据集二)

Fig. 15 Matching performance comparison(Dataset two)

**结束语** 本文针对出租车低频 GPS 轨迹,提出了一种改进交互式投票匹配算法。相较于传统的交互式投票匹配算法,本文算法加入了方向分析函数和 3 个用于过滤噪声路段的约束条件。实验结果表明,改进算法在匹配性能方面优于传统的算法。

本文利用了路段的最高限速、汽车的实时速度及平均速度,而每个时段的交通流量变化必然导致汽车行驶速度的变化。后续研究既可以考虑引入路段的历史速度和其他车辆的

速度信息,也可以考虑引入机器学习方法,从历史数据中学习到有模式来提升新轨迹数据的地图匹配准确率。

### 参 考 文 献

- [1] JOSHIR R. A new approach to map matching for in-vehicle navigation systems; the rotational variation metric[C]//2001 IEEE Intelligent Transportation Systems. Oakland, CA, USA, 2001: 33-38.
- [2] YUAN J, ZHENG Y, XIE X, et al. T-Drive: Enhancing Driving Directions with Taxi Drivers' Intelligence[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(1): 220-232.
- [3] PANG L X, CHAWLA S, LIU W, et al. On detection of emerging anomalous traffic patterns using GPS data[J]. Data & Knowledge Engineering, 2013, 87(9): 357-373.
- [4] WHITEC E, BERNSTEIN D, KORNHAUSERA L. Some map matching algorithms for personal navigation assistants[J]. Transportation Research Part C: Emerging Technologies, 2000, 8(1): 91-108.
- [5] BRAKATSOULAS S, PFOSE D, SALAS R, et al. On map-matching vehicle tracking data[C]//Proceedings of the 31st International Conference on Very Large Data Bases. Trondheim, Norway, 2005: 853-864.
- [6] QUDDUSM A, OCHIENGW Y, ZHAO L, et al. A general map matching algorithm for transport telematics applications[J]. GPS Solutions, 2003, 7(3): 157-167.
- [7] VELAGAN R, QUDDUSM A, BRISTOWA L. Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems[J]. Transportation Research Part C, 2009, 17(6): 672-683.
- [8] QUDDUSM A, NOLANDR B, OCHIENG W Y. A High Accuracy Fuzzy Logic Based Map Matching Algorithm for Road Transport[J]. Journal of Intelligent Transportation Systems, 2006, 10(3): 103-115.
- [9] OBRADOVIC D, LE NZ H, SCHUPFNER M. Fusion of Sensor Data in Siemens Car Navigation System[J]. IEEE Transactions on Vehicular Technology, 2007, 56(1): 43-50.
- [10] NEWSON P, KRUMM J. Hidden Markov Map Matching Through Noise and Sparseness[C]//Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. Seattle, Washington, USA, 2009: 336-343.
- [11] HSUEHY L, CHENH C, HUANG W J. A Hidden Markov Model-Based Map-Matching Approach for Low-Sampling-Rate GPS Trajectories[C]//IEEE 7th International Symposium on Cloud and Service Computing (SC2). Kanazawa, Japan, 2017: 271-274.
- [12] HUNTER T, ABBEEL P, BAYENA M. The Path Inference Filter: Model-Based Low-Latency Map Matching of Probe Vehicle Data[J]. IEEE Transactions on Intelligent Transportation Systems, 2013, 15(2): 507-529.
- [13] LIU X, LIU K, LI M, et al. A ST-CRF Map-Matching Method for Low-Frequency Floating Car Data[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(5): 1241-1254.
- [14] SU H B, WANG G Z, WANG J D. Map matching algorithm based on fuzzy neural network[J]. Journal of Beijing University of Science and Technology, 2012, 34(1): 43-47. (in Chinese)  
苏海滨, 王光政, 王继东. 基于模糊神经网络的地图匹配算法[J]. 北京科技大学学报, 2012, 34(1): 43-47.
- [15] NIKOLIĆ M, JOVIĆ J. Implementation of generic algorithm in map-matching model[J]. Expert Systems with Applications, 2017, 72: 283-292.
- [16] GONG Y J, CHEN E, ZHANG X, et al. AntMapper: An Ant Colony-Based Map Matching Approach for Trajectory-Based Applications[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(2): 390-401.
- [17] HASHEMI M. Reusability of the Output of Map-Matching Algorithms Across Space and Time Through Machine Learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(11): 3017-3026.
- [18] LOU Y, ZHANG C, ZHENG Y, et al. Map-matching for Low-sampling-rate GPS Trajectories[C]//Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. Seattle, Washington, USA, 2009: 352-361.
- [19] YUAN J, ZHENG Y, ZHANG C, et al. An Interactive-Voting Based Map Matching Algorithm[C]//2010 Eleventh International Conference on Mobile Data Management. Kanas City, Missouri, USA, 2010: 43-52.
- [20] LI C, SHEN D R, ZHU M D, et al. KNN query processing method for spatio-temporal information[J]. Journal of Software, 2016, 27(9): 2278-2289. (in Chinese)  
李晨, 申德荣, 朱命冬, 等. 一种对时空信息的kNN查询处理方法[J]. 软件学报, 2016, 27(9): 2278-2289.
- [21] DONG J, HUANG C H. Research on improving Dijkstra algorithm in GIS navigation application shortest path search[J]. Computer Science, 2012, 39(10): 245-247, 257. (in Chinese)  
董俊, 黄传河. 改进 Dijkstra 算法在 GIS 导航应用中最短路径搜索研究[J]. 计算机科学, 2012, 39(10): 245-247, 257.
- [22] ZHANG Y, HE Y. An advanced interactive-voting based map matching algorithm for low-sampling-rate GPS data[C]//IEEE 15th International Conference on Networking, Sensing and Control (ICNSC). 2018: 1-7.