

一种基于交互式投票的低采样率GPS数据地图匹配算法

摘要

1. 引言

2. 问题阐述

2.1 地图匹配问题

2.2 候选点的选择

2.3 候选点之间的相互影响

2.4 查找最终匹配路径

3. 基于交互式投票的高级地图匹配算法

3.1 位置和道路分析

3.2 约束分析

3.3 相互影响建模

3.4 查找局部最优路径

3.5 候选边缘投票

4. 性能评估

4.1 数据

4.2 评估设置

4.3 评估方法

4.4 结果分析

5. 总结

摘要

车载设备提供的GPS数据极大地丰富了基于位置的系统应用，地图匹配在其中发挥着重要作用。对于低采样率的GPS数据，现有的地图匹配算法将面临许多挑战，如错误率高和复杂的城市道路网络效率低下。针对出租车GPS轨迹低采样率数据，提出了一种基于交互式投票的先进地图匹配算法。该算法利用空间分析函数、时间分析函数和带两个约束的道路分析函数来测量地图匹配中连续候选点之间的关系。算法中的约束不仅减少了大量的计算，而且显著提高了算法的精度。基于GPS采样点之间的相互作用，

提出了一种新的投票方法，即候选边缘投票，以获得最佳的地图匹配结果。我们使用一个月的真实世界轨迹数据集评估我们的算法。该算法在精度和效率上都优于该算法。

1. 引言

随着GPS设备的广泛使用，手机、车辆、无人机，甚至手表都可以收集位置数据。丰富的GPS轨迹数据池正在逐步成为智能交通系统和相关服务的数据源[1], [2]。在大多数大城市，有相当多的出租车在城市中驾驶GPS设备，这导致了大量的出租车轨迹。受电磁干扰和功耗等因素的影响，与乘客一起运行的出租车的GPS数据将始终以一分钟以上的间隔记录。由于出租车可能在一分钟内行驶了相当长的距离，因此GPS位置可能并不一直在沿路段可用。实际上，大多数数据很可能具有低采样率（采样间隔大于60秒）的偏差。所有这些数据都需要后处理，例如地图匹配。地图匹配算法的目的是将GPS点与道路网中的道路段对齐[3]。

在过去的20年里，有大量的研究人员处理GPS数据上的出租车轨迹[4]–[7]和地图匹配算法[8]、[9]。在附加信息的使用方面，地图匹配算法可分为四类：几何[10]、[11]、拓扑[12]、[13]、概率[14]和其他高级技术[15]、[16]。他们中的大多数人都集中在高采样率的GPS轨迹上。低采样率GPS轨迹的地图匹配更加困难，尤其是在城市复杂的路网中。穆罕默德·Quddus等人。[17]提出了一种利用最短路径和车辆轨迹对低采样率GPS数据进行地图匹配的方法，其中车辆航向作为关键数据。然而，实际上，由于成本的原因，高精度GPS设备并不总是在出租车上。采样的车辆航向仍然不可靠。

著名的低采样率GPS数据匹配算法是郑Y等人提出的ST-Matching算法[18]和IVMM算法[19]。这些算法考虑了GPS轨迹的空间和时间信息，并对GPS点之间的加权相互影响建模。然而，算法的过程很复杂，数据必须重复映射匹配。此外，这些算法还可以在以下方面进一步改进：

- 路径上权重的简单求和将产生这样的结果：如果错误候选路径的累积权重远大于前几个候选点中的真正匹配路径，即使错误路径在其后续候选中的权重相对较小，错误路径的最终累积权重也将大于正确路径。
- 在不考虑车辆速度的实际合理范围的情况下，有时匹配轨迹的两个采样点之间的距离太长，任何车辆都无法在规定的时间内行驶。此外，所有GPS采样点都被认为是有效的，这对于GPS采样点偏移超过100米的轨迹来说是不好的。这些样本点应在数据后处理中删除。
- 地图匹配后的一些轨迹的结果将是“曲折”的，特别是在城市高架道路上（即车辆在高架道路上频繁上下行驶）。

为了解决低采样率GPS轨迹数据地图匹配中的上述问题，提出了一种基于交互式投票的高级地图匹配算法。我们的算法起源于IVMM [19]，采用空间分析函数、时间分析函数和道路分析函数，带两个约束条件来测量地图匹配中连续候选点之间的关系。

本文的其余部分如下。第二节阐述了地图匹配算法的问题，并澄清了初步的思想。所提出算法的细节见第三节。实验结果在第四节中进行了论证和分析。结论见第五节。

2. 问题阐述

2.1 地图匹配问题

首先，我们给出了与地图匹配算法相关的一些定义。

- （路段）作为道路网中的基本单元的道路段 r 是有向（单向或双向）边缘。每个路段都有长度 $r.length$ 和速度限制 $r.v$ 。每个路段都有其起点 $r.s$ 和终点 $r.e$ 。道路交叉口 c 是两个或多个路段的连接点。
- （GPS轨迹）GPS轨迹 T 是GPS采样点 $p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_n$ 的序列。每个GPS点都有经度、纬度、时间戳、方向和速度的关键信息。任何连续GPS点之间的时间间隔超过60秒被定义为低采样率。
- （道路网）有向图 $G(V, E)$ ，其中 V 是代表道路交叉口或道路段末端的一组顶点， E 是代表道路段的一组边。
- （路径）路径 P 是路段序列 $r_1 \rightarrow r_2 \rightarrow \dots \rightarrow r_n$ ，其中 $r_{i-1}.e = r_i.s$ 。 $r.s$ 和 $r.e$ 分别表示路段 r 的起点和终点。
- （局部最优路径）对于GPS轨迹 T 中采样点 p_i 的每个候选点 c_i^j ，假设通过候选点 c_i^j 的最佳路径是 T 的真实路径。

地图匹配：给定GPS轨迹 T 和道路网 G ，在 G 中找到 T 的真实路径 P 。

2.2 候选点的选择

到采样点的候选点对于地图匹配至关重要。对于每个GPS点 $p_i, i = 1, 2, \dots, n$ ，在轨迹中，有几个候选匹配点 $c_i^j, j = 1, 2, \dots, m$ 。我们使用[20]中提出的KNN算法来获得候选点。即使路网复杂或GPS数据量很大，该算法仍然可以快速找到候选点。在图1中，采样点 p_2 垂直投影在投影区域中，得到候选点 c_2^1 和 c_2^2 。如果投影区域有十字路口，则道路的交叉口也应视为候选点，即获得候选点 c_2^3 。将交点视为候选点是至关重要的。如果采样点 p_1 和 p_3 分别只有一个候选点 c_1^1 和 c_3^1 ，则点 p_2 通过垂直投影获得候选点 c_1^1 和 c_2^2 ，并且不将交点 c_2^3 视为候选点。在地图匹配中，路径 $c_1^1 \rightarrow c_2^1 \rightarrow c_3^1$ 和 $c_1^1 \rightarrow c_2^2 \rightarrow c_3^1$ 都不能选择。在这种情况下，大多数车辆不会选择掉头。

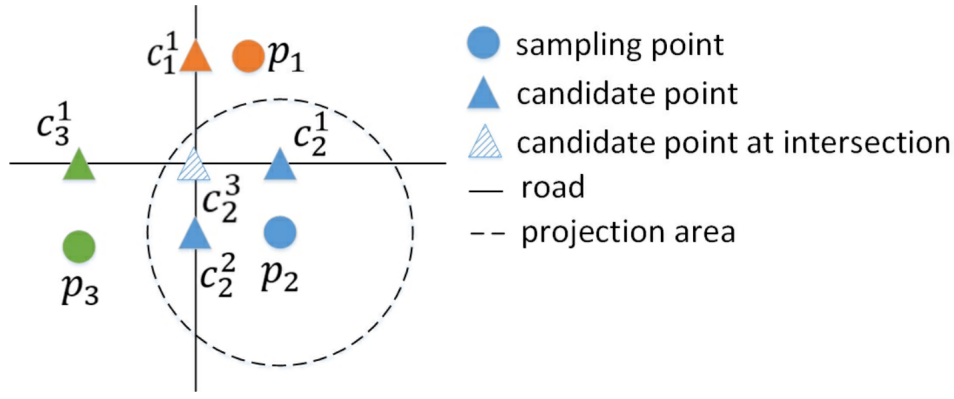


Fig. 1. Sampling point and candidate points selection

2.3 候选点之间的相互影响

GPS采样点相互影响，尤其是相邻采样点。在图1中，我们有信心从 p_1 到 p_3 的最终路径经过 c_2^3 ，而不是 c_2^1 或 c_2^2 。因为采样点 p_2 受到 p_1 和 p_3 的影响。显然，越接近点，影响就越大。基于采样点和投票过程的相互影响[19]，我们提出了一种基于交互式投票的高级地图匹配算法(AIVMM)。对于采样点的每个候选点，通过第三节讨论的位置上下文分析过程，找到通过候选点的局部最优路径。

2.4 查找最终匹配路径

我们工作背后的一个基本假设是，局部最优路径可以反映最终匹配路径所经过的路径的可能性。如果路径 $c_{i-1}^t \rightarrow c_i^s$ （用 P 表示）没有被任何局部最优路径通过，则最终匹配路径将不包括路径 P 。在此假设下，找到从 p_{i-1} 的候选者到 p_i 的候选者的路径，该路径被所有局部最优路径遍历最多，并将该路径定义为最终子路径。此外，我们还提出了一个称为候选边缘投票的过程，以计算有多少局部最优路径将通过候选路径。对于每个局部最优路径，如果两个连续的候选点在路径上，则从两个候选点中的一个到另一个的路径的投票率增加1。最后，所有最终子路径都可以通过候选边缘投票过程获得。此外，将它们拼接在一起，将形成最终的匹配路径。

3. 基于交互式投票的高级地图匹配算法

GPS采样点具有其经度、纬度和时间戳的信息。还有车辆的即时方向和即时速度。信息的准确性由GPS设备的精度决定。信息不准确可能会导致高干扰。对于采样的GPS数据中的瞬时方向和速度不够可靠，

我们不会简单地直接使用它们的值，而是从不同数据点之间的关系中提取有用的信息，如两点之间的平均速度、两点之间的权重。

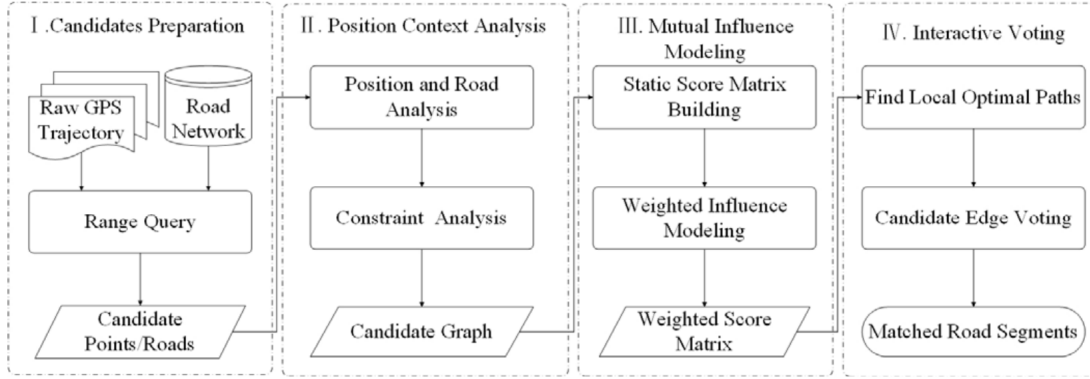


Fig. 2. Overview of the AIVMM algorithm

提出的AIVMM算法包括四个阶段：候选点准备、位置上下文分析、相互影响建模和交互式投票。图2示出了AIVMM算法的框架。我们使用基于KD树[20]的KNN算法的范围查询，获得第一阶段每个采样点的候选点，如第II-B节所述。然后在第二阶段，我们利用位置和道路分析和约束分析构建候选图。在第三阶段，通过加权影响建模的静态评分矩阵构建加权评分矩阵后，我们可以找到局部最优路径。根据路径，所有候选边都将在最后阶段投票为最佳匹配路径。

3.1 位置和道路分析

我们使用空间时间分析来测量两个连续候选点之间的影响。GPS点的测量误差满足高斯分布 $N(\mu, \sigma^2)$

[21]，因此观测概率为： $N(c_i^j) = e^{-\frac{(x_i^j - \mu)^2}{2\sigma^2}}$ 。其中 c_i^j 是采样点 p_i 的候选点， x_i^j 是从候选 c_i^j 到采样点的欧几里德距离。从候选点 c_{i-1}^t 到候选点 c_i^s 的空间分析函数[18]定义为：

$$F_s(c_{i-1}^t \rightarrow c_i^s) = N(c_i^s) \times V(c_{i-1}^t \rightarrow c_i^s), 2 \leq i \leq n$$

其中 $V(c_{i-1}^t \rightarrow c_i^s)$ 是过渡概率。它的目的是测量两个连续候选点之间的最短路径和直路径的相似性。它的定义为：

$$V(c_{i-1}^t \rightarrow c_i^s) = \frac{d_{i-1 \rightarrow i}}{\omega_{i-1, t \rightarrow (i, s)}}$$

其中 $d_{i-1 \rightarrow i}$ 是采样点 p_{i-1} 到采样点 p_i 的欧几里得距离， $\omega_{i-1, t \rightarrow (i, s)}$ 是从候选 c_{i-1}^t 到 c_i^s 的最短路径的长度。时间分析函数定义为：

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

其中 $\hat{v}_{(i-1, t) \rightarrow (i, s)}$ 是从候选点 c_{i-1}^t 到 c_i^s 的最短路径的加权速度限制， $\bar{v}_{(i-1, t) \rightarrow (i, s)}$ 是沿着候选点 c_{i-1}^t 至 c_i^s 之间最短路径行驶的车辆的平均速度。在没有交通事故的情况下，车速一般接近道路限速。因此，适当的时间分析函数应在道路限速附近获得最大值。

在许多地图匹配算法中，我们经常看到在复杂的城市路网中与高架道路匹配，匹配的结果是车辆在高架道路上上下行驶。如图3所示， $\hat{a}\hat{b}$ 是高架路， $\hat{c}\hat{d}$ 是地面道路。 $\hat{e}\hat{f}$ 和 $\hat{g}\hat{h}$ 分别是出口匝道和入口匝道。长虚线是IVMM算法的匹配结果，我们可以看到车辆从高架路行驶下来，然后行驶到高架路上。在实际驾驶中，车辆往往沿着当前道路行驶，而不是在高架路上上下行驶。

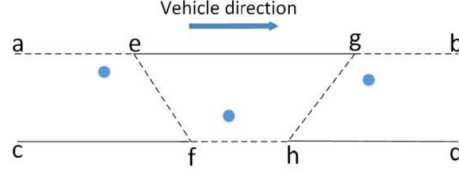


Fig. 3. Vehicle travels up and down the elevated road

在地图中，道路通常被分为几个级别，每个级别都有不同的道路限速。高架路的水平高于地面道路，即高架路的限速大于地面道路。因此，我们使用道路水平因子来模拟车辆在高限速道路上停留的趋势。

$c_{i-1}^t \rightarrow c_i^s$ 的道路水平系数（RLF）定义为：

$$RLF(c_{i-1}^t \rightarrow c_i^s) = \frac{v_s}{(v_d - v_s) + v_s}$$

其中 v_s 和 v_d 表示 c_{i-1}^t 和 c_i^s 所在道路的速度限制。因此，路径 $c_{i-1}^t \rightarrow c_i^s$ 的权重函数可以扩展为：

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

3.2 约束分析

根据我们对广泛实验收集的权重函数值的统计分析，错误匹配路径 $c_{i-1}^t \rightarrow c_i^s$ 的权重 $F(c_{i-1}^t \rightarrow c_i^s)$ 总是很低（通常小于 1.0×10^{-5} ），而且没有任何权重正确匹配路径的小于 1.0×10^{-5} 。

因此，我们将两个候选点之间的权重设置一个阈值 θ 作为约束。它的定义为：

约束1：当 $F(c_{i-1}^t \rightarrow c_i^s) < \theta$ 时，其中 θ 是用户指定的阈值，路径 $c_{i-1}^t \rightarrow c_i^s$ 被视为错误匹配。

约束1有两个优点。一个是消除第一节提到的简单权重求和的缺点。基于位置和道路分析和约束1，我们可以消除大多数错误路径。因此，即使错误候选路径的累积权重远大于前几个候选点中真正匹配的路

径，我们也不会在其后续候选中选择错误路径。最大的最终累积权重将是真正匹配的路径。另一个优点是，它消除了那些权重太小的路径，减少了寻找局部最优路径的计算工作量。

仅使用时间分析功能测量车速是不够的，因为它没有考虑车辆在沿匹配路径行驶时的速度是否在合理范围内。道路限速是关键信息。从地图中，我们可以得到每条道路的限速，并计算 $c_{i-1}^t \rightarrow c_i^s$ 经过的所有路径的加权平均限速 $\bar{v}_\omega(c_{i-1}^t \rightarrow c_i^s)$ 。考虑到车辆平均速度与道路加权平均限速之间的关系，匹配路径上的车辆速度不会超过合理的范围。

约束 2： $\bar{v}(c_{i-1}^t \rightarrow c_i^s)$ 是车辆沿着候选点 c_{i-1}^t 至 c_i^s 之间最短路径行驶的平均速度。
 $\bar{v}_\omega(c_{i-1}^t \rightarrow c_i^s)$ 是 $c_{i-1}^t \rightarrow c_i^s$ 经过的所有路径的加权平均速度限制。当 $\bar{v}(c_{i-1}^t \rightarrow c_i^s) > \alpha \bar{v}_\omega(c_{i-1}^t \rightarrow c_i^s)$ 时， $1 \leq \alpha \leq 2$ 时，路径 $c_{i-1}^t \rightarrow c_i^s$ 被视为误差匹配。

出租车中支持GPS的设备通常精度在20米以内。由于车辆靠近建筑物、桥梁和树木，它们的准确性会恶化。然而，大多数GPS点的偏移量不超过100米。然而，仍有一些GPS采样点不在此范围内。在这种情况下，应删除这些点，以防止它们的干扰。Lee和克鲁姆[23]使用卡尔曼滤波器和粒子滤波器来固定嘈杂的轨迹点。但该方法不能应用于低采样率的GPS轨迹。由于卡尔曼滤波器在运动目标长时间被阻塞时将失去跟踪目标。

我们提出了一种基于约束1和约束2的噪声轨迹点滤波方法。当采样点 p_{i-1} 和 p_i 之间的所有候选路径不满足约束1和约束2时，认为采样点 p_i 是噪声点。删除采样点 p_i ，并将 p_{i-1} 的后继点更改为 p_{i+1} 。然后恢复地图匹配过程。

经过位置和道路分析和约束分析，构建了候选图。节点是候选点的集合，边缘是两个相邻候选点之间的最短路径集合。节点和边都是根据位置和道路分析的结果分配权重值的。

3.3 相互影响建模

在计算相互影响的权重之前，我们构建了一个静态评分矩阵[19] $M = \text{diag}\{M^{(2)}, M^{(3)}, \dots, M^{(n)}\}$ ，其中 $M_{(i)} = (m_{ts}^{(i)})_{a_{i-1} \times a_i} = (F(c_{i-1}^t \rightarrow c_i^s))_{a_{i-1} \times a_i}$ ，此静态评分矩阵中的每个项目仅考虑两个连续点的信息，表示候选点的权重。此信息并不反映互动的相互影响。因此，我们可以将其构建为静态矩阵，以减少计算相互影响的权重时的计算。

要对候选点的加权影响建模， $(n-1)$ 维距离权重矩阵[19]定义如下：

$$W_i = \text{diag}\{\omega_i^{(1)}, \omega_i^{(2)}, \dots, \omega_i^{(i-1)}, \omega_i^{(i+1)}, \dots, \omega_i^{(n)}\}$$

其中, $\omega_i^{(j)} = e^{\frac{(dist(p_i, p_j))^2}{\beta^2}}$ 和 $dist(p_i, p_j)$ 是 p_i 和 p_j 之间的欧几里德距离, β 是相对于道路网的参数。此矩阵给出了所有其他点到 p_i 的距离的效果的权重。对于 $i = 2, 3, \dots, n$ 受距离影响的权重可计算为:

$$\Phi_i^{(j)} = (\varphi_{ts}^{(i,j)})_{a_{i-1} \times a_i} = f(x) = \begin{cases} \omega_i^{j-1} M^{(j)} & \text{if } 1 \leq i \leq j \\ \omega_i^j M^{(j)} & \text{otherwise} \end{cases} \quad (1)$$

加权得分矩阵 Φ_3 如下:

$$\Phi_3 = \begin{bmatrix} 0.4 & 0.3 & -\infty & -\infty \\ 0.35 & 0.25 & -\infty & -\infty \\ -\infty & -\infty & 0.05 & 0.1 \end{bmatrix}$$

矩阵 Φ_3 是采样点 p_3 的加权得分矩阵。而 $\Phi_3^{(1,1)} = 0.4$ 表示 $c_1^1 \rightarrow c_2^1$ 的权重为0.4。

3.4 查找局部最优路径

对于每个候选点 $c_i^j = 1, 2, \dots, n$, 和 $j = 1, 2, \dots, m$, 假设 c_i^j 是最终映射匹配结果中的正确点。找到一条通过点 c_i^j 的概率最大的路径作为局部最优路径。我们使用图4(a)中的GPS轨迹来说明寻找局部最优路径的过程。在图4(b)中, 该值表示来自 p_1 的候选点的累积权重。实线和虚线表示从 c_{i-1}^t 到 c_i^s 有合法的路径 (即, 任何车辆都可以在合理的时间内通过路径 $c_{i-1}^t \rightarrow c_i^s$)。在这个例子中, 我们假设 c_3^2 是正确的点, 并设置路径 $c_2^1 \rightarrow c_3^1$ 和 $c_2^2 \rightarrow c_3^1$ 的权重为 $-\infty$ 。这意味着路径必须通过候选点 c_3^2 。累积权重 $fValue(c_i^s) = \max\{fValue(c_{i-1}^t) + \psi_{ts}^{(i,j)}\}, t = 1, 2, \dots, n$ 。 $\psi_{ts}^{(i,j)}$ 是 $c_{i-1}^t \rightarrow c_i^s$ 的权重, 受 p_i 和 p_j 的距离影响。在计算累积权重后, 我们得到了所有 p_4 候选点的 $fValues$ 。从 p_4 可以得到局部最优候选点为 c_4^2 , 因为 $fValue(c_4^2) > fValue(c_4^1)$ 。从 c_4^2 中我们可以得到上一个局部最优候选点 c_3^2 。同样, 我们得到了局部最优路径为 $c_1^1 \rightarrow c_2^3 \rightarrow c_3^2 \rightarrow c_4^2$, 如图4(b)中的虚线所示。

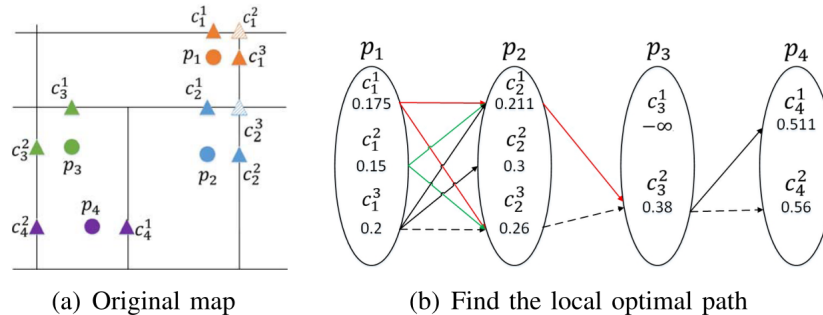


Fig. 4. Find the local optimal path

3.5 候选边缘投票

在找到每个候选点 c_i^j 的局部最优路径后，我们得到了一组局部最优路径。假设大多数局部最优路径经过道路，然后最终的地图匹配结果也经过道路，我们就可以得到最终的地图匹配结果。根据局部最优路径集，我们可以得到最终地图匹配结果概率最大的路径。与[19]中专注于候选点投票的交互式投票方法不同，我们提出了一种新的候选边缘投票方法来获得最终的地图匹配结果。更频繁地通过局部最优路径的候选点可能不是真实路径中的点。

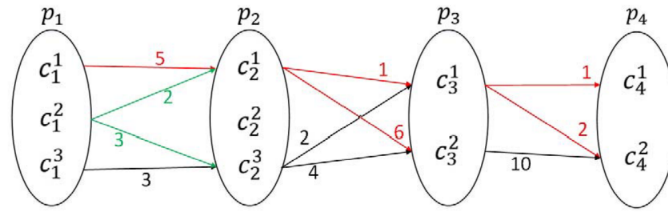


Fig. 5. Vote of Candidates

在图5中，我们继续使用图4(a)中的GPS轨迹来说明候选边缘投票过程。需要注意的是，这里的边缘是指连续采样点 p_{i-1} 和 p_i 候选的对应关系 $c_{i-1}^t \rightarrow c_i^s$ ，它们在图5中作为边缘呈现。在找到局部最优路径后，如果候选点 c_{i-1}^t 和 c_i^s 在路径 P 中，则路径 $c_{i-1}^t \rightarrow c_i^s$ 的投票率增加1。在图5中，线的值表示通过最佳路径集中道路的局部最佳路径的数量。对于 $p_1 \rightarrow p_2$ ，投票边 $c_1^1 \rightarrow c_2^1, c_1^2 \rightarrow c_2^1, c_1^2 \rightarrow c_2^2$ 和 $c_1^3 \rightarrow c_2^2$ 的数目分别为5、2、3和3。最大的投票边缘是 $c_1^1 \rightarrow c_2^1$ 。同样， $p_2 \rightarrow p_3$ 的最大投票边是 $c_2^2 \rightarrow c_3^1$ ， $p_3 \rightarrow p_4$ 的最大投票边是 $c_3^2 \rightarrow c_4^2$ 。最后，我们得到了最终的匹配结果是 $c_1^1 \rightarrow c_2^1 \rightarrow c_3^2 \rightarrow c_4^2$ 。候选边缘投票的过程在算法1中描述。

Algorithm 1 Candidate Edge Voting

Input: The set of local optimal paths LOP

Output: final map matching result P: $c_1^{s_1} \rightarrow \dots \rightarrow c_i^k \rightarrow \dots \rightarrow c_n^{s_n}$

```
1: Let  $vote[][][]$  denote the number of vote edge computed so far
2: Let  $p[]$  denote the final map matching result of candidate point
3: for  $i = 1 \rightarrow n$  do
4:   for  $j = 1 \rightarrow m$  do
5:      $c_{from} = LOP[i][j]$ 
6:      $c_{to} = LOP[i][j + 1]$ 
7:      $vote[j][c_{from}][c_{to}] = vote[j][c_{from}][c_{to}] + 1$ 
8:   end for
9: end for
10:  $p[1], p[2] = \operatorname{argmax}_{c_{from}, c_{to}} \{vote[0][c_{from}][c_{to}], c_{from} = 1, 2, \dots, a_1, c_{to} = 1, 2, \dots, a_2\}$  /*find the final map matching candidates of  $p_1$  and  $p_2$ ,  $a_i$  is the number of  $p_i$ 's candidate points*/
11: for  $s = 2 \rightarrow m$  do
12:    $p[s] = \operatorname{argmax}_{c_{to}} \{vote[s][p[s-1]][c_{to}], c_{to} = 1, 2, \dots, a_{s+1}\}$ 
13: end for
14: return  $p$ 
```

4. 性能评估

我们使用真实世界的的数据来评估所提出的算法。首先，我们将介绍评估的设置，然后分析评估结果。

4.1 数据

我们使用的是北京的道路网，有144151个路段和108379个十字路口。从2015年10月30日到2015年11月30日，北京的28203辆出租车收集了GPS数据，包含1821060条轨迹。在这些轨迹中，由于车辆停止，收集了大量GPS采样点。因此，我们删除这些点，仅在出租车有乘客时使用GPS采样点。我们在GPS数据中有544659447个采样点。在我们的GPS数据集中，53%的连续两个采样点的时间间隔为60–69秒。超过60%的采样点为低采样率（采样间隔大于60秒）。为了评估本文中AIVMM算法的性能，我们从上述优化数据集中随机选择了100条GPS轨迹，其中大部分数据都是低采样率的。数据连续采样点之间的时间间隔主要集中在30秒到120秒之间。根据原始GPS位置和个人经验，对100条采样轨迹进行手动地图匹配，以获得地面真实情况。

4.2 评估设置

我们设 $k = 10$ 为每个样本点的最大候选点数，查询半径为 $r = 100m$ 。在[22]中，陈光介绍了GPS偏移符合正态分布，GPS偏移在 $\pm 2\sigma$ 范围内的概率为 95.44%。正态分布函数的参数为 $\mu = 5, \sigma = 25$ 。距离权重参数为 $\beta = 5km$ 。算法在C++上实现，在英特尔I5-2400 PC上，Linux上的MINT 18操作系统上有4GB内存。

4.3 评估方法

精度是指匹配道路的正确性，效率是指在同一平台上比较算法的运行时间。我们使用正确匹配百分比(CMP)来评估算法的准确性。

$$CMP = \frac{|Corrent\ matched\ road\ segments|}{|Road\ segments\ to\ be\ matched|} \times 100\%$$

4.4 结果分析

图6-9为相同GPS轨迹下AIVMM算法与IVMM算法的可视化匹配结果。在这些图片中，点代表GPS采样点，绿线代表路网，黑粉线代表匹配结果。在IVMM算法中，简单地将权值沿路径相加会降低算法的精度。虽然在图6(a)所示的匹配路径下，车速处于合理范围内，但匹配结果并不正确。AIVMM算法可以得到如图6(b)所示的正确结果。

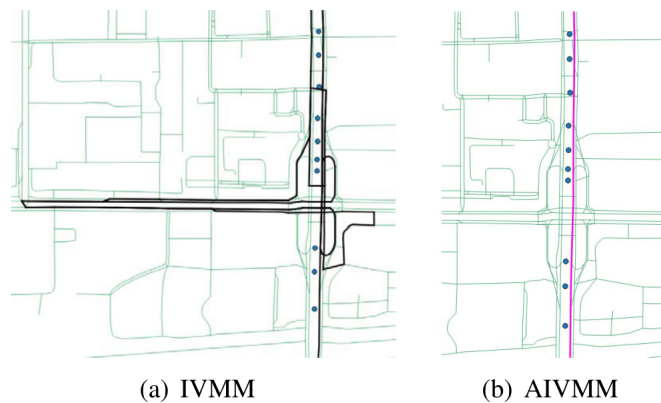


Fig. 6. A screenshot of machining result about simple weight sum

在没有车辆限速约束的情况下，经过几个连续的大偏移GPS采样点后，IVMM算法的匹配结果为图7(a)所示的扩展圆。在AIVMM算法中加入车速限制，匹配结果为一条直线，如图7(b)所示。

在图8中，箭头所指点的偏移量大于100米(即采样点为噪声点)。AIVMM算法能够找到噪声点并去除噪声点。因此我们可以得到如图8(b)所示的正确路径。

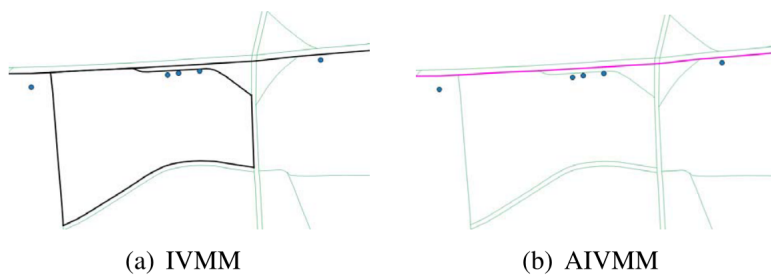


Fig. 7. A screenshot of maching result about vehicle speed

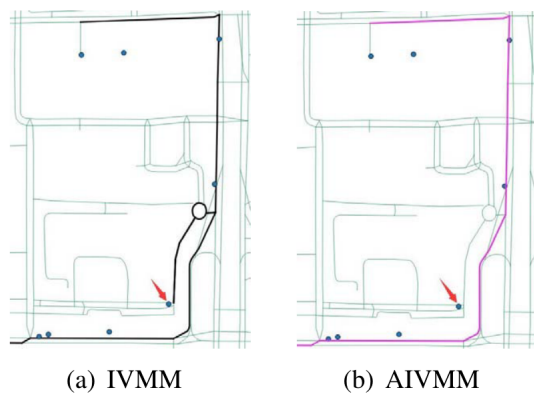


Fig. 8. Screenshot of maching result with noisy point

图9为复杂城市路网匹配结果。IVMM算法的结果如图9(a)所示。车辆在高架路上上下下行驶。AIVMM算法的匹配结果为一条直线，如图9(b)所示。

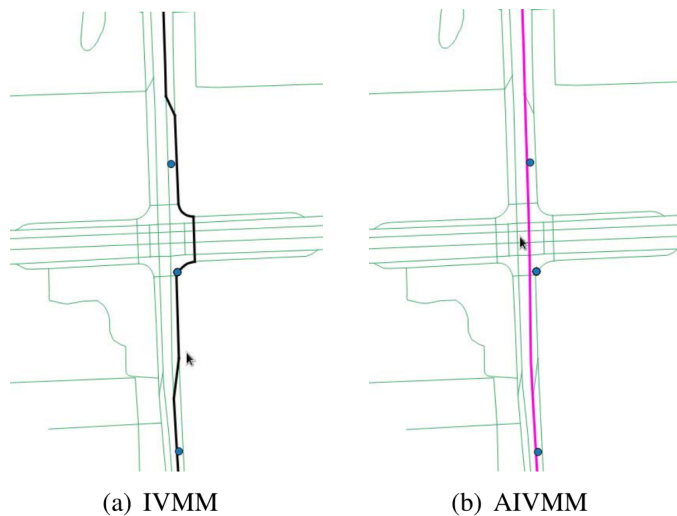


Fig. 9. Screenshot of maching result with urban elevated road

图10(a)为基于正确匹配百分比的匹配精度比较结果。AIVMM算法的准确率分别比IVMM和ST-Matching算法高近90%、15%和58%。ST-Matching算法的精度较低，因为它没有考虑采样点之间的影响关系对采样点之间的影响关系。

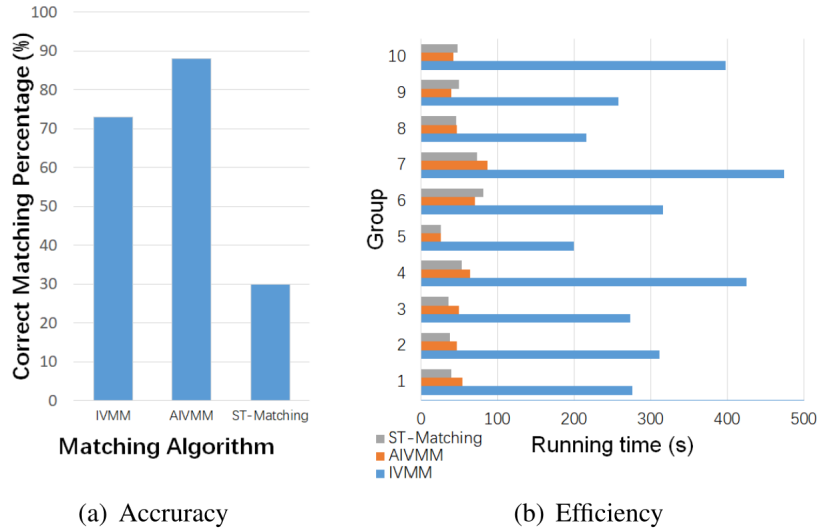


Fig. 10. Performance Analysis (IVMM V.S. AIVMM)

我们注意到，在稀疏路网中，无论车辆车速如何，IVMM算法都能得到正确的地图匹配结果。但在路网密集的情况下，将会产生错误的结果。ST-Matching算法在这两种情况下都不能得到很好的性能。AIVMM算法在这两个约束条件的基础上通过去除大量错误的候选路径来提高算法的精度。虽然AIVMM算法在很多情况下都有很好的性能，但仍有改进的空间。当车速过慢(小于15km/h)或太快时，约束2不能用于这种情况，会导致匹配结果不正确。

图10(b)为IVMM、AIVMM和ST-Matching算法的运行时间。100条用于测试的GPS轨迹被分成10组。每组有10条GPS轨迹。比较了算法在同一平台上的运行时间。我们可以看到，AIVMM算法大约比IVMM算法快5到10倍。AIVMM算法的效率与ST-Matching算法基本相同，但精度更高。

5. 总结

本文研究了低采样率GPS采样点的地图匹配算法。基于采样点之间的相互影响，我们提出了一种新的地图匹配方法，该方法使用空间分析函数、时间分析函数和道路分析函数来度量连续候选点之间的关系。该算法通过引入两种约束条件，并对噪声轨迹点进行滤波，提高了算法的精度和效率。此外，提出了一种新的候选边缘投票方法来寻找可靠的地图匹配路径。实验结果表明，AIVMM算法的正确匹配率明显高于IVMM和ST-Matching算法。即使在稠密的路网中，该方法也具有较好的匹配效果。