



(12) 发明专利

(10) 授权公告号 CN 111189459 B

(45) 授权公告日 2023. 12. 22

(21) 申请号 202010026328.6

(22) 申请日 2020.01.10

(65) 同一申请的已公布的文献号
申请公布号 CN 111189459 A

(43) 申请公布日 2020.05.22

(73) 专利权人 成都信息工程大学
地址 610000 四川省成都市西南航空港经
济开发区学府路1段24号
专利权人 四川省金科成地理信息技术有限
公司
成都探码科技有限公司

(72) 发明人 乔少杰 黄振锋 甘戈 韩楠
宋学江 魏军林 张小辉 温敏
肖月强 程维杰 陈权亮 李斌勇
张永清 张吉烈 何林波 元昌安
彭京 周凯 余华 范勇强
冉先进

(74) 专利代理机构 成都玖和知识产权代理事务
所(普通合伙) 51238
专利代理师 胡琳梅

(51) Int.Cl.
G01C 21/30 (2006.01)

(56) 对比文件
CN 104318766 A, 2015.01.28
CN 106441316 A, 2017.02.22
CN 106595680 A, 2017.04.26
KR 20100067578 A, 2010.06.21
US 2011208426 A1, 2011.08.25
CN 109405839 A, 2019.03.01
CN 105371857 A, 2016.03.02
CN 109215347 A, 2019.01.15
CN 108106620 A, 2018.06.01
CN 109029472 A, 2018.12.18 (续)

审查员 田颖

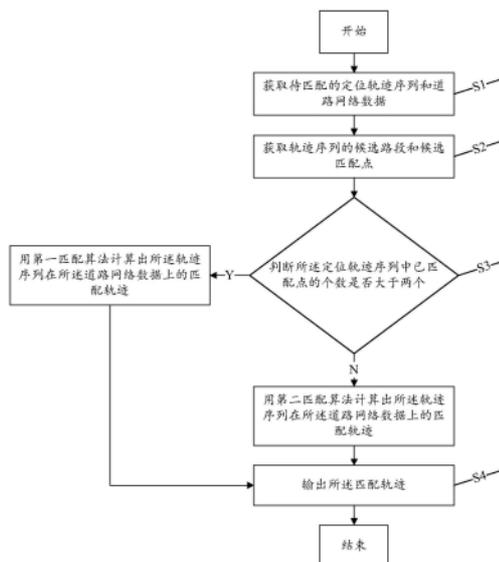
权利要求书3页 说明书10页 附图13页

(54) 发明名称

一种定位信息与道路匹配的方法和装置

(57) 摘要

本申请实施例提供了一种定位信息与道路匹配的方法和装置,该方法包括:S1.获取待匹配的定位轨迹序列和道路网络数据;步骤S2.获取所述轨迹序列的候选路段和候选匹配点;步骤S3.判断所述定位轨迹序列中已匹配点的个数是否大于两个,若是,则用第一匹配算法计算出所述轨迹序列在所述道路网络数据上的匹配轨迹;若否,则用第二匹配算法计算出所述轨迹序列在所述道路网络数据上的匹配轨迹;步骤S4.输出所述匹配轨迹。本申请提供的定位信息与道路匹配的方法和装置,其通过获取浮动车数据和城市道路网络数据,搜索和筛选出合理的候选路段和候选待匹配点,结合所设计两种不同的匹配算法来进行浮动车地图的匹配,实现了大规模浮动车地图匹配的准确性和效率性。



CN 111189459 B

[接上页]

(56) 对比文件

孙庆辉等. 路径规划和导航.《空间位置信息服务系统原理和方法》. 西安地图出版社, 2009,

陈林等. 情景感知系统.《‘互联网+’智慧校园技术与工程实施》. 电子科技大学出版社, 2017,

Shaojie Qiao. A Self-Adaptive

Parameter Selection Trajectory Prediction Approach via Hidden Markov Models.《IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS》. 2015, 第16卷 (第1期),

乔少杰. 一种基于空间编码技术的轨迹特征提取方法.《中国科学: 信息科学》. 2017, 第47卷 (第11期),

1. 一种定位信息与道路匹配的方法,其特征在于,包括:
- 步骤S1. 获取待匹配的定位轨迹序列和道路网络数据;
- 获取待匹配的定位轨迹序列包括以下步骤:
- 步骤S101. 获取原始浮动车数据;
- 步骤S102. 计算原始浮动车数据中的前一个待匹配点和后一个待匹配点的欧式距离;
- 步骤S103. 若计算结果小于两米,则剔除上述后一个待匹配点;
- 步骤S104. 检测定位轨迹序列中后一时刻待匹配点与前一时刻待匹配点的时间间隔是否大于采样时间;若是,则使用前一时刻待匹配点的瞬时速度和方向角进行插值补全定位轨迹序列,并获取补全后的轨迹序列的Geohash特征值;若否,则保持原有的定位轨迹序列,并获取原有的轨迹序列的Geohash特征值;
- 获取道路网络数据包括以下步骤:
- 步骤S111. 获取原始道路数据;
- 步骤S112. 对原始道路数据构建结点-路段的结构,得到道路网络数据;
- 步骤S113. 获取道路网络中路段折点的Geohash特征值;
- 步骤S2. 获取所述定位轨迹序列的候选路段,具体包括以下步骤:
- 步骤S201. 获取待匹配点当前的Geohash特征值;
- 步骤S202. 使用Base32编码对当前Geohash特征值反向计算得到数字串;
- 步骤S203. 对所获得的数字串取偶数位为经度码,奇数位为纬度码;
- 步骤S204. 以当前Geohash区域为中心,上下移动对应纬度码加或减一操作,左右移动对应经度码加或减一操作,进行加或减一操作后重新使用Base32编码生成Geohash特征值,获取当前Geohash区域近邻的8个Geohash区域;
- 步骤S205. 获取当前Geohash区域及近邻的8个Geohash区域在内的所有路段作为候选路段;
- 步骤S206. 计算待匹配点与候选路段的距离,选取距离最短的点作为候选路段的候选匹配点;
- 步骤S207. 计算各待匹配点与根据该待匹配点搜索到的所有候选路段的欧式距离;若待匹配点与候选路段的距离大于五十米,则剔除该候选路段;
- 步骤S208. 计算当前待匹配点与下一时刻待匹配点所构成的第一向量;计算沿着各候选路段行驶方向所构成的第二向量;判断所述第一向量和所述第二向量的夹角角度,若大于 30° ,则剔除该候选路段;
- 步骤S3. 判断所述定位轨迹序列中已匹配点的个数是否大于两个,若是,则用第一匹配算法计算出所述轨迹序列在所述道路网络数据上的匹配轨迹;若否,则用第二匹配算法计算出所述轨迹序列在所述道路网络数据上的匹配轨迹;

所述步骤S3中所述第一匹配算法为:

$$z = \alpha \frac{\arctan(k_g) - \arctan(k_r)}{Agl} + \beta \frac{d_{c_r^m \rightarrow P_i}}{R} + \gamma \left| \frac{Speed_c}{Speed_r} - 1 \right|;$$

所述第二匹配算法为:

$$z = \alpha \frac{|k_{back} - k_r|}{Agl} + \beta \frac{d_{c_i^m \rightarrow P_i}}{R} + \gamma \left| \frac{Speed_c}{Speed_r} - 1 \right|;$$

其中, k_g 是前3个待匹配点的拟合曲线的斜率; k_r 是当前候选道路的斜率; k_{back} 是当前待匹配点的后向向量的斜率; $d_{c_i^m \rightarrow P_i}$ 是候选点 c_i^m 与待匹配点 P_i 的欧式距离; $Speed_c$ 是车辆当前的瞬时速度; $Speed_r$ 是道路限制的速度; Agl 是GPS拟合曲线和候选道路夹角的阈值范围; R 是道路搜索范围; α 、 β 、 γ 为权重参数;

步骤S4. 输出所述匹配轨迹。

2. 一种定位信息与道路匹配的装置, 用于实现如权利要求1所述的一种定位信息与道路匹配的方法, 其特征在于, 包括:

数据获取模块, 用于获取待匹配的定位轨迹序列和道路网络数据;

数据搜索和筛选模块, 用于获取轨迹序列的候选路段和候选匹配点;

数据判断模块, 判断所述定位轨迹序列中已匹配点的个数是否大于两个;

第一匹配模块, 若判断结果为是, 则用第一匹配算法计算出所述轨迹序列在所述道路网络数据上的匹配轨迹;

第二匹配模块, 若判断结果为否, 则用第二匹配算法计算出所述轨迹序列在所述道路网络数据上的匹配轨迹;

结果输出模块, 输出所述匹配轨迹;

所述第一匹配算法为:

$$z = \alpha \frac{\arctan(k_g) - \arctan(k_r)}{Agl} + \beta \frac{d_{c_i^m \rightarrow P_i}}{R} + \gamma \left| \frac{Speed_c}{Speed_r} - 1 \right|;$$

所述第二匹配算法为:

$$z = \alpha \frac{|k_{back} - k_r|}{Agl} + \beta \frac{d_{c_i^m \rightarrow P_i}}{R} + \gamma \left| \frac{Speed_c}{Speed_r} - 1 \right|;$$

其中, k_g 是前3个待匹配点的拟合曲线的斜率; k_r 是当前候选道路的斜率; k_{back} 是当前待匹配点的后向向量的斜率; $d_{c_i^m \rightarrow P_i}$ 是候选点 c_i^m 与待匹配点 P_i 的欧式距离; $Speed_c$ 是车辆当前的瞬时速度; $Speed_r$ 是道路限制的速度; Agl 是GPS拟合曲线和候选道路夹角的阈值范围; R 是道路搜索范围; α 、 β 、 γ 为权重参数。

3. 一种存储介质, 其特征在于, 所述存储介质包括计算机程序, 所述计算机程序运行时, 执行所述权利要求1所述的定位信息与道路匹配的方法。

4. 一种电子设备, 其特征在于, 包括:

处理器;

存储器, 包括计算机程序;

所述处理器用于运行所述存储器中的计算机程序, 当所述计算机程序运行时, 执行所

述权利要求1所述的定位信息与道路匹配的方法。

一种定位信息与道路匹配的方法和装置

技术领域

[0001] 本发明属于大数据处理技术领域,具体涉及一种定位信息与道路匹配的方法和装置。

技术背景

[0002] 随着基于位置的服务(LBS,Location-Based Service)技术的兴起和GPS数据采集技术广泛使用。例如,在交通轨迹数据中,地图匹配技术是浮动车技术中重要的一个部分,它是将附带经纬度、时间、角度等一系列信息作为判断依据,把车辆的GPS点匹配到正确的电子地图道路上以便确认车辆正确的行驶轨迹。

[0003] 在实际技术应用中,由于北斗卫星和车载位置存在定位误差的原因,车辆GPS的位置和电子地图中道路位置存在偏差,因此需要把车辆定位位置纠正到对应道路上。基于海量的浮动车数据分析,能够提取出许多对交通有利的数据,例如出租车载客热点的发现、城市道路的更新、交通路段拥堵情况等。然而,把GPS点匹配到正确道路上是使用浮动车数据的前提条件。

[0004] 不过在大数据时代,浮动车数据增长迅速,想要把海量的浮动车数据中进行地图匹配操作,会消耗大量的时间成本。因此,如何考虑正确率和时间效率两个因素的情况下,针对海量浮动车数据进行正确、快速、可行的地图匹配是目前本技术领域的一个难题。

发明内容

[0005] 针对现有技术存在的问题,本申请提供了一种定位信息与道路匹配的方法、装置、存储介质和电子设备。

[0006] 为完成发明目的,本发明采用的技术方案是:

[0007] 一种定位信息与道路匹配的方法,包括:

[0008] 步骤S1.获取待匹配的定位轨迹序列和道路网络数据;

[0009] 步骤S2.获取所述定位轨迹序列的候选路段;

[0010] 步骤S3.判断所述定位轨迹序列中已匹配点的个数是否大于两个,若是,则用第一匹配算法计算出所述轨迹序列在所述道路网络数据上的匹配轨迹;若否,则用第二匹配算法计算出所述轨迹序列在所述道路网络数据上的匹配轨迹;

[0011] 步骤S4.输出所述匹配轨迹。

[0012] 进一步的,所述步骤S1中获取待匹配的定位轨迹序列包括以下步骤:

[0013] 步骤S101.获取原始浮动车数据;

[0014] 步骤S102.计算原始浮动车数据中的前一个待匹配点和后一个待匹配点的欧式距离;

[0015] 步骤S103.若计算结果小于两米,则剔除上述后一个待匹配点。

[0016] 步骤S104.检测定位轨迹序列中后一时刻待匹配点与前一时刻待匹配点的时间间隔是否大于采样时间;若是,则使用前一时刻待匹配点的瞬时速度和方向角进行插值补全

定位轨迹序列,并获取补全后的轨迹序列的Geohash特征值;若否,则保持原有的定位轨迹序列,并获取原有的轨迹序列的Geohash特征值;

[0017] 进一步的,所述步骤S1中获取道路网络数据包括以下步骤:

[0018] 步骤S111.获取原始道路数据;

[0019] 步骤S112.对原始道路数据构建结点-路段的结构,得到道路网络数据;

[0020] 步骤S113.获取道路网络中路段折点的Geohash特征值。

[0021] 进一步的,所述步骤S2还包括以下步骤:

[0022] 步骤S201.获取待匹配点当前的Geohash特征值;

[0023] 步骤S202.使用Base32编码对当前Geohash特征值反向计算得到数字串;

[0024] 步骤S203.对所获得的数字串取偶数位为经度码,奇数位为纬度码;

[0025] 步骤S204.以当前Geohash区域为中心,上下移动对应纬度码加或减一操作,左右移动对应经度码加或减一操作,进行加或减一操作后重新使用Base32编码生成Geohash特征值,获取当前Geohash区域近邻的8个Geohash区域;

[0026] 步骤S205.获取当前Geohash区域及近邻的8个Geohash区域在内的所有路段作为候选路段;

[0027] 步骤S206.计算待匹配点与候选路段的距离,选取距离最短的点作为候选路段的候选匹配点;

[0028] 步骤S207.计算各待匹配点与根据该待匹配点搜索到的所有候选路段的欧式距离;若待匹配点与候选路段的距离大于五十米,则剔除该候选路段;

[0029] 步骤S208.计算当前待匹配点与下一时刻待匹配点所构成的第一向量;计算沿着各候选路段行驶方向所构成的第二向量;判断所述第一向量和所述第二向量的夹角角度,若大于 30° ,则剔除该候选路段。

[0030] 进一步的,所述步骤S3中所述第一匹配算法为:

$$[0031] \quad z = \alpha \frac{\arctan(k_g) - \arctan(k_r)}{Ag1} + \beta \frac{d_{c_i^m \rightarrow P_i}}{R} + \gamma \left| \frac{Speed_c}{Speed_r} - 1 \right|;$$

[0032] 所述第二匹配算法为:

$$[0033] \quad z = \alpha \frac{|k_{back} - k_r|}{Ag1} + \beta \frac{d_{c_i^m \rightarrow P_i}}{R} + \gamma \left| \frac{Speed_c}{Speed_r} - 1 \right|;$$

[0034] 其中, k_g 是前3个待匹配点的拟合曲线的斜率; k_r 是当前候选道路的斜率; k_{back} 是当前待匹配点的后向向量的斜率; $d_{c_i^m \rightarrow P_i}$ 是候选点 c_i^m 与待匹配点 P_i 的欧式距离; $Speed_c$ 是车辆当前的瞬时速度; $Speed_r$ 是道路限制的速度; $Ag1$ 是GPS拟合曲线和候选道路夹角的阈值范围; R 是道路搜索范围; α 、 β 、 γ 为权重参数。

[0035] 本发明还提供了一种存储介质一种定位信息与道路匹配的装置,包括:

[0036] 数据获取模块,用于获取待匹配的定位轨迹序列和道路网络数据;

[0037] 数据搜索和筛选模块,用于获取轨迹序列的候选路段和候选匹配点;

[0038] 数据判断模块,判断所述定位轨迹序列中已匹配点的个数是否大于两个;

[0039] 第一匹配模块,若判断结果为是,则用第一匹配算法计算出所述轨迹序列在所述道路网络数据上的匹配轨迹;

[0040] 第二匹配模块,若判断结果为否,则用第二匹配算法计算出所述轨迹序列在所述道路网络数据上的匹配轨迹;

[0041] 结果输出模块,输出所述匹配轨迹。

[0042] 本发明还提供了一种存储介质,包括计算机程序,所述计算机程序运行时,执行所述定位信息与道路匹配的方法。

[0043] 本发明还提供了一种电子设备,包括:

[0044] 处理器;

[0045] 存储器,包括计算机程序;

[0046] 所述处理器用于运行所述存储器中的计算机程序,当所述计算机程序运行时,执行所述定位信息与道路匹配的方法。

[0047] 本申请提供的定位信息与道路匹配的方法和装置,其通过获取浮动车数据和城市道路网络数据,搜索和筛选出合理的候选路段和候选待匹配点,结合所设计两种不同的匹配算法来进行浮动车地图的匹配,实现了大规模浮动车地图匹配的准确性和效率性。

附图说明

[0048] 为了更清楚地说明本申请实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍。显而易见地,下面描述中的附图仅仅是本申请的一些示例或实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据提供的附图获得其他的附图,而且还可以根据提供的附图将本申请应用于其它类似情景。除非从语言环境中显而易见或另做说明,图中相同标号代表相同结构或操作。

[0049] 图1是本申请中道路网络结构定义的道路网络拓扑图;

[0050] 图2是本申请在一些实施例中的一种定位信息与道路匹配的方法流程图;

[0051] 图3是本申请在一些实施例中的获取待匹配的定位轨迹序列的流程图;

[0052] 图4是本申请在一些实施例中的获取道路网络数据的流程图;

[0053] 图5是本申请在一些实施例中的对候选路段进行筛选的流程图;

[0054] 图6是本申请在一些实施例中基于Geohash位置图的搜索原理图;

[0055] 图7是本申请在一些实施例中候选路段分析的原理图;

[0056] 图8a是本申请效果实验中在一条道路时的模拟GPS数据准确率对比图;

[0057] 图8b是本申请效果实验中在两条道路时的模拟GPS数据准确率对比图;

[0058] 图8c是本申请效果实验中在三条道路时的模拟GPS数据准确率对比图;

[0059] 图9a是本申请效果实验中在一条道路时的浮动车数据准确率对比图;

[0060] 图9b是本申请效果实验中在两条道路时的浮动车数据准确率对比图;

[0061] 图9c是本申请效果实验中在三条道路时的浮动车数据准确率对比图;

[0062] 图10是本申请效果实验中模拟GPS数据和真实浮动车数据使用不同算法之间时间效率对比的实验结果图。

[0063] 图11a是本申请效果实验中在一条道路时的地图匹配效果图;

[0064] 图11b是本申请效果实验中在两条道路时的地图匹配效果图;

[0065] 图11c是本申请效果实验中在三条道路时的地图匹配效果图;

[0066] 图12是本申请在一些实施例中使用的Base32编码对照图。

具体实施方式

[0067] 以下实施例将以城市出租车的定位点采样信息进行地图匹配为例,来对本申请的技术方案进行详细的说明。显然,本申请的技术方案包括但并不仅限于城市出租车的定位点采样信息的地图匹配,例如,还可以是其他移动交通工具或者任何具备定位功能的移动电子终端的定位点采样信息的地图匹配等。

[0068] 另外,下文实施例中,采用GPS定位来获取定位点信息。应当理解,在实际应用中,同样可以使用北斗定位或者其他定位系统来获取定位点信息,其同样应当在本申请的保护范围之内。

[0069] 如图1所示,是道路网络结构的拓扑图,道路网络结构是指:在给定的一个城市路网数据,可以抽象定义为一个有向图网络 $G=(V,E)$, $V=\{\sum v_i | i=1,2,\dots,n\}$, $E=\{\sum e_i | i=1,2,\dots,n\}$,其中 V 代表定点的集合, E 代表边的集合。在道路网络数据中,一条完整的道路是由若干条路段构成,路段由相邻两个折点连接而成。

[0070] 本申请对路段折点进行存储,通过搜索附近的路段折点得到相关的路段。在图1中可以有如下表示:道路节点集合为 $\langle 1,3,6,8,10 \rangle$;路段折点集合 $\{\langle 1,2,3 \rangle, \langle 3,4,5,6 \rangle, \langle 3,7,8 \rangle, \langle 3,9,10 \rangle\}$;路段集合 $\{\langle 12,23 \rangle, \langle 34,45,56 \rangle, \langle 37,78 \rangle, \langle 39,910 \rangle\}$;道路集合 $\langle 13,36,38,310 \rangle$ 。

[0071] 实施例1

[0072] 图2示出了本申请在一些实施例中的一种定位信息与道路匹配的方法流程图。

[0073] 该方法包括以下步骤:

[0074] 步骤S1.获取待匹配的定位轨迹序列和道路网络数据。

[0075] 上述步骤S1中,所述定位轨迹序列是一组或多组待匹配点按顺序排列而成的轨迹,其中,每组待匹配点中包括一个或多个待匹配点,每个待匹配点是按照时间戳顺序生成的定位轨迹点。例如,定位轨迹序列可以是车辆基于GPS定位来获取的浮动车数据。

[0076] 所述浮动车数据是指给定一组浮动车数据 $T=(P_1,P_2,\dots,P_n)$,其中每个点 P_i 是按照时间戳顺序生成的GPS轨迹点(待匹配点),并且每个 P_i 包括车辆的ID编号、经纬度、时间、车辆角度、车辆速度等信息。多个待匹配点按顺序排列而成的轨迹,也称为GPS轨迹序列。

[0077] 所述待匹配点,是指待匹配点 P_i 与候选路段集合 L_i 中欧式距离最近的点作为候选点,待匹配点 P_i 的候选点集合记为 $C_i=(c_i^1, c_i^2, \dots, c_i^m)$, $i=1,2,3,\dots,n$ 。

[0078] 步骤S2.获取定位轨迹序列的候选路段。

[0079] 在上述步骤S2中,需要对路段折点和车辆待匹配GPS轨迹序列进行编码,把二维地理位置信息压缩成一维的位置信息进行存储,加快候选路段的搜索速度。待匹配点对道路网络局部进行搜索后得到一条或多条候选道路,为了进一步提高匹配效率,还需要对候选道路进行筛选。

[0080] 所述候选路段,是指待匹配点 P_i 搜索附近道路,得到候选路段集合。待匹配点 P_i 可能有一条或多条候选路段,一个待匹配点的候选路段集合记为 $L_i=(l_i^1, l_i^2, \dots, l_i^m)$, $i=1,2,3,\dots,n$ 。

[0081] 步骤S3.判断所述定位轨迹序列中已匹配点的个数是否大于两个,若是,则用第一匹配算法计算出所述轨迹序列在所述道路网络数据上的匹配轨迹;若否,则用第二匹配算

法计算出所述轨迹序列在所述道路网络数据上的匹配轨迹;

[0082] 在上述步骤S3中,由于在二维坐标系下,定位轨迹序列可以使用一个m次多项式 $y = w_0 + w_1x + w_2x^2 + \dots + w_mx^m$ 进行拟合,得到一条平滑的曲线。另外,由于待匹配点的个数不同会影响拟合的斜率,故此先对待匹配点的个数进行预判,以便于采用对应的匹配算法进行拟合匹配,从而提升拟合匹配的精度。

[0083] 若是,则利用第一匹配算法计算出所述候选匹配点在所述道路网络数据上的匹配轨迹。若否,则利用第二匹配算法计算出所述候选匹配点在所述道路网络数据上的匹配轨迹。

[0084] 由于实际应用中经常需要确定多个位置点的关系。在已知变量X和Y的情况下,我们能对多个点构建一条平滑的曲线 $y = f(x)$,这就是曲线拟合,在二维坐标系下,GPS轨迹序列可以使用一个m次多项式 $y = w_0 + w_1x + w_2x^2 + \dots + w_mx^m$ 进行拟合,得到一条平滑的曲线。为了求多项式的系数w,可以将多项式转化为矩阵的形式,化简得到公式:

$$[0085] \quad \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \dots \\ w_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} \dots\dots (1),$$

[0086] 其中,GPS轨迹可看成一条m次多项式曲线方程,方程中x代表曲线的横坐标,即经度信息, $x_1 \dots x_n$ 代表多个GPS轨迹点(即待匹配点)的经度,轨迹曲线方程中y代表曲线的纵坐标,即纬度信息, $y_1 \dots y_n$ 代表多个GPS轨迹点的纬度;在m次多项式中,w代表曲线的系数,同一条轨迹中,w为固定的值,需要使用经度信息x和纬度信息y反求w, $w_0, w_1 \dots w_m$ 为多个系数值;n指轨迹曲线中轨迹点的个数;m是多项式方程的系数的个数,给定一条轨迹曲线,m的个数也随之确定,使用x和y反求w可得具体的m个数。

[0087] 进一步,由下述公式(2)得到 $XA = Y$ 的形式,可以进一步求得 $A = X^T Y$,即求得系数矩阵。

[0088] 另外,在一些实施例,考虑到一些使用低频定位采样的情况,例如30s左右获得一次定位记录,此时点与点之间的距离相差较大,为此,本实施例可以采用定位轨迹序列中3个连续的待匹配点进行曲线的拟合,拟合后曲线的斜率公式为:

$$[0089] \quad k = \frac{-\sum_{i=1}^3 x_i \sum_{i=1}^3 y_i + 3 \sum_{i=1}^3 x_i y_i}{3 \sum_{i=1}^3 x_i^2 - (\sum_{i=1}^3 x_i)^2} \dots\dots (2)。$$

[0090] 步骤S4.输出所述匹配轨迹。

[0091] 在本实施例中,上述地图匹配方法的执行主体可以为服务器或计算机终端。例如,服务器可以为位置服务器或地图服务器;计算机终端可以为车载主机、移动电子设备或控制主机等。

[0092] 本实施例提供的方法,充分考虑被定位对象(例如:车辆)和道路的速度因素,并使用两种匹配算法根据待匹配点的个数进行匹配评价得到匹配结果,相比于现有技术,本申请提供的方法具有准确率高、计算用时少等优点。

[0093] 实施例2

[0094] 在一些可能的实施例中,当车辆进入定位设备信号弱的区域,定位数据会产生漂移的情况,进而产生异常记录。另外由于车辆在拥挤路段或者等待乘客的情况下,GPS定位会出现重复的记录,应对其进行剔除处理。见图3,在上述步骤S1中,获取待匹配的定位轨迹序列的步骤可以包括:

[0095] 步骤S101.获取原始浮动车数据;

[0096] 步骤S102.计算原始浮动车数据中的前一个待匹配点和后一个待匹配点的欧式距离;

[0097] 步骤S103.若计算结果小于两米,则剔除上述后一个待匹配点。

[0098] 步骤S104.检测定位轨迹序列中后一时刻待匹配点与前一时刻待匹配点的时间间隔是否大于采样时间;若是,则使用前一时刻待匹配点的瞬时速度和方向角进行插值补全定位轨迹序列,并获取补全后的轨迹序列的Geohash特征值;若否,则保持原有的定位轨迹序列,并获取原有的轨迹序列的Geohash特征值;

[0099] 其中,插值补全轨迹数据的方法很多,本实施例并不限制具体的插值补全轨迹数据的具体方式。例如,前一时刻待匹配点的作为 (x_i, y_i) ,插值点为 (x_{i+1}, y_{i+1}) ,则插值点的计算公式为:

$$[0100] \quad x_{i+1} = x_i + \Delta t \sin \varepsilon$$

$$[0101] \quad y_{i+1} = y_i + \Delta t \cos \varepsilon \cdots \cdots (3),$$

[0102] 其中, ε 代表车辆当前定位的方向角,以正北为0度,顺时针转动,取值范围为 $[0^\circ, 360^\circ]$ 。

[0103] 在一些实施例中,由于城市道路数据使用坐标系与定位轨迹序列数据使用的坐标系不一致,因此,在对定位轨迹序列进行处理后,还可以包括:将定位轨迹序列中所有待匹配点的坐标转为以道路网络数据所在坐标系表示的坐标。例如,一般城市道路使用的是GCJ-02坐标系,GPS轨迹序列数据使用WGS-84坐标系,两者坐标系不一致,因此,需要对处理后的GPS轨迹序列数据从WGS-84转到GCJ-02坐标系。

[0104] 本实施例通过对该定位轨迹序列中时间间隔较大的待匹配点进行插值补全轨迹数据处理,可以有效修正由于定位设备在信号弱的区域出现定位数据异常记录的问题。通过本实施例对获得的定位轨迹序列中的待匹配点进行预处理,使数据更加合理,从而可以得到更加准确的候选匹配点,进而可以进一步提高候选匹配计算的效率。

[0105] 实施例3

[0106] 在一些可能的实施例中,本申请对原始道路数据(例如城市道路网络路段)进行压缩存储,将二维地理位置信息压缩成一维的位置信息,从而增加了路段搜索的正确性,便于后续通过扩展搜索的办法对附近相邻区域遍历搜索。

[0107] 具体的,见图4,上述所述获取道路网络数据的步骤可以包括:

[0108] 步骤S111.获取原始道路数据;

[0109] 步骤S112.对原始道路数据构建结点-路段的结构,得到道路网络数据;

[0110] 步骤S113.获取道路网络中路段折点的Geohash特征值。

[0111] 通过本实施例的处理,可以将原始道路数据抽象为道路网络数据,便于存储和后续的计算处理。

[0112] 实施例4

[0113] 在一些实施例中,在所述获取待匹配的定位轨迹序列和道路网络数据之后还可以根据待匹配点在道路网络数据进行局部搜索得到一条或多条候选路段,然后为了进一步提高匹配效率,需要对候选路段进行筛选。

[0114] 具体的,见图5,上述得到一条或多条候选路段并进行筛选的步骤可以包括:

[0115] 步骤S201.获取待匹配点当前的Geohash特征值;

[0116] 步骤S202.使用Base32编码对照图(如图12所示)对当前Geohash特征值反向计算得到数字串;

[0117] 步骤S203.对所获得的数字串取偶数位为经度码,奇数位为纬度码;

[0118] 步骤S204.以当前Geohash区域为中心,上下移动对应纬度码加或减一操作,左右移动对应经度码加或减一操作,进行加或减一操作后重新使用Base32编码生成Geohash特征值,获取当前Geohash区域近邻的8个Geohash区域;

[0119] 步骤S205.获取当前Geohash区域及近邻的8个Geohash区域在内的所有路段作为候选路段;

[0120] 步骤S206.计算待匹配点与候选路段的距离,选取距离最短的点作为候选路段的候选匹配点;

[0121] 步骤S207.计算各待匹配点与根据该待匹配点搜索到的所有候选路段的欧式距离;若待匹配点与候选路段的距离大于五十米,则剔除该候选路段;

[0122] 步骤S208.计算当前待匹配点与下一时刻待匹配点所构成的第一向量;计算沿着各候选路段行驶方向所构成的第二向量;判断所述第一向量和所述第二向量的夹角角度,若大于 30° ,则剔除该候选路段。

[0123] 由于采用了Geohash编码,如图6所示,图中大黑点为待匹配点,路段折点3、4、7和待匹配点位置在同一个区域wtw90d7中,当前能搜索路段37、34,但不能搜索到23、39路段。因此需要对搜索范围增大,以wtw90d7为中心,扩展到附近8个搜索区域。本次对待匹配点进行搜索的区域包括wtw90dd、wtw90de、wtw90ds、wtw90d6、wtw90d7、wtw90dk、wtw90d4、wtw90d5、wtw90dh。由此可见,通过上述实施例可以有效扩大搜索范围。

[0124] 这里对上述实施例中候选路段分析原理进行说明:假设当前待匹配点与下一时刻待匹配点构成向量 g ,沿着候选路段 l^m ,车辆行驶方向构成向量 r , g 与 r 的后向夹角大于 30° ,剔除该候选道路;如图7所示,50米范围内待匹配点 P_i 搜索到路段12和23,对 g 和 r 向量后向进行夹角分析,候选路段23方向向量与 $P_i P_{i+1}$ 方向向量大于 30° ,故剔除候选路段23。

[0125] 在一些实施例中,通过上述实施例的优化,可以得到候选匹配点和候选路段,通过对候选匹配点进行拟合,使其与候选路段进行匹配,即可得到待匹配点在道路网络数据上的最终匹配结果

[0126] 具体的,结合上述公式(1)和(2),在待匹配点进行道路搜索后,得到多个候选匹配点,本申请在考虑被定位对象的行进速度的因素的情况下,当候选匹配点大于两个的时候,则可以使用第一匹配算法对候选点进行计算,该第二匹配算法为如下公式(4)。

$$[0127] \quad z = \alpha \frac{\arctan(k_g) - \arctan(k_r)}{Agl} + \beta \frac{d_{c_r \rightarrow P_i}}{R} + \gamma \left| \frac{Speed_c}{Speed_r} - 1 \right| \dots (4)$$

[0128] 当候选匹配点小于两个的时候,由于不满足曲线拟合斜率的计算条件,可以使用第二匹配算法对候选点进行计算,取代价最小的候选点作为匹配结果,该第一匹配算法为如下公式(5)。

$$[0129] \quad z = \alpha \frac{|k_{back} - k_r|}{Ag1} + \beta \frac{d_{c_i^m \rightarrow P_i}}{R} + \gamma \left| \frac{Speed_c}{Speed_r} - 1 \right| \dots\dots (5)$$

[0130] 其中, k_g 是前3个待匹配点的拟合曲线的斜率, k_r 是当前候选道路的斜率; k_{back} 是当前待匹配点的后向向量的斜率, $d_{c_i^m \rightarrow P_i}$ 是候选点 c_i^m 与待匹配点 P_i 的欧式距离, $Speed_c$ 是车辆当前的瞬时速度, $Speed_r$ 是道路限制的速度, $Ag1$ 是GPS拟合曲线和候选道路夹角的阈值范围,当夹角范围超过30度时,匹配的误差率会增大,故 $Ag1$ 设置为30度。 R 是道路搜索范围, R 设置为50米, α 、 β 、 γ 分别表示权重参数,例如本实施例中对权重参数可以设置为 $\alpha=0.3$, $\beta=0.3$, $\gamma=0.4$ 。

[0131] 效果实验

[0132] 为了验证本申请提供的方法的有效性,以下将通过一组实验来进行说明。

[0133] 1、实验数据

[0134] 11、浮动车数据

[0135] 本实验GPS轨迹序列数据由两部分组成,分别为GPS模拟数据和真实浮动车数据。GPS模拟数据采用GPS误差原理,随机产生服从高斯分布20米以内的经纬度误差,共1459个待匹配GPS点。真实浮动车数据来源于香港科技大学提供的2007年2月20日上海市出租车GPS数据(<https://www.cse.ust.hk/scrg/>),行走轨迹包含整个上海市以及上海周边城市,数据采样频率大约30s至60s,数据量约600万条。从真实数据中选取697个记录作为待匹配GPS点,采用人工查找的方法标注真实道路。

[0136] 12、道路网路数据

[0137] 本实验使用上海市道路网络数据,道路类型包括国道、省道、县道、乡道、城市快速路、高速和其他类型道路。为了验证本文算法,道路网络数据缩小至上海市S20外环高速以内的区域,经过拓扑规则处理和拓扑构建后得到32916条道路和173805个路段折点,采用高德地图GCJ-02坐标系,经纬度跨度分别为:[121.347,121.646],[31.123,31.377]。

[0138] 2、实验环境

[0139] 本实验的匹配算法使用64位Windows 7旗舰版的操作系统,Intel(R)Core(TM) i7-4770HQ CPU@2.20GHz处理器和16GB的内存,编程语言使用Python3.7.2,编程工具为PyCharm2018.2.4,数据库采用PostgreSQL 10.7,并通过ArcMap10.4.1工具对待匹配点、道路网络数据和匹配结果进行可视化。

[0140] 3、评价指标

[0141] 匹配算法的指标采用准确率(CMP,全称Correct Matching Percentage)进行判定,判定方法使用ArcMap工具的图层叠加分析,与真实道路相重合的匹配点即为正确匹配,该过程由图层叠加自动完成。如公式(6):

$$[0142] \quad CMP = \frac{|Num_c|}{|Num_{all}|} \times 100\% \dots\dots (6)$$

[0143] 其中, Num_c 为正确匹配点的个数, Num_{all} 为待匹配点总数。

[0144] 4、算法复杂度分析

[0145] GPS轨迹序列包含 n 个待匹配点,道路网络数据有 m 条路段,对于每个待匹配点均需要对道路网络数据进行搜索。对 n 个待匹配点进行搜索,则时间复杂度为 $O(n)$ 。当待匹配点搜索的路段数量为0,则时间复杂度为 $O(1)$,这类搜索结果是没有意义的。当待匹配点搜索的路段数量大于0时,则时间复杂度为 $O(\log(m))$ 。所以, n 个待匹配点完成地图匹配的时间复杂度为 $O(n\log(m))$ 。

[0146] 5、算法结果分析

[0147] 为了验证本文提出的定位信息与道路匹配的方法,对比算法采用传统的垂直投影算法和曲线拟合算法。数据进行预处理后对GPS模拟数据和真实浮动车数据进行插值补全,插值补全的数据分别划分为10s、30s、50s和70s的采样频率,并且分别在一条道路、两条道路和三条道路以上的道路数据进行算法验证。本文主要对算法的准确率和时间效率两方面进行验证。首先,一条道路、两条道路和三条道路以上又分别随机选取5组不同GPS轨迹序列作为实验数据,对5组实验结果进行求平均值得到最终的CMP结果。最后,分别在GPS模拟数据和真实浮动车数据中随机抽取5数据组对算法进行时间效率验证。

[0148] 图8a、8b、8c表示了GPS模拟数据分别在一条道路、两条道路和三条道路以上使用不同算法的准确率对比实验结果。实验结果表明:在一条道路上进行实验,三种算法均由较高的准确率,定位信息与道路匹配的方法表现最好,准确率高达94.79%以上。在两条道路上进行实验,垂直投影和曲线拟合算法准确率开始下降,而定位信息与道路匹配的方法准确率仍然维持在95.14%以上。在三条道路以上进行实验,定位信息与道路匹配的方法优势显著,匹配率相对于垂直投影法平均提高了27.41%,相对于曲线拟合法提高了30.24%。相对于垂直投影算法和曲线拟合算法,定位信息与道路匹配的方法运行时间显著下降,平均单点匹配时间为8ms,时间复杂度为 $O(n\log(m))$ 。原因在于:利用Geohash扩展搜索,与全局遍历的思想不同,待匹配点只搜索本区域和附近8个区域的路网折点,搜索时间成本上从 $O(m)$ 降低至 $O(\log(m))$ 。此外,定位信息与道路匹配的方法采用Geohash技术搜索本区域内的路段和使用后向夹角对候选路段进行筛选,时间效率得到大幅度提升。

[0149] 图9a、9b、9c表示了真实浮动车数据分别在一条道路、两条道路和三条道路以上使用不同算法的准确率对比实验结果。实验结果表明:在一条道路上进行实验,定位信息与道路匹配的方法准确率维持在89.38%以上,垂直投影算法和曲线拟合算法准确率较低,曲线拟合法准确率最高只有82.26%。在两条道路上进行实验,曲线拟合算法表现稳定,平均准确率只有69.31%。垂直投影算法表现较差,准确率保持在61.25%至67.13%范围内。定位信息与道路匹配的方法表现最佳,仍然具有较高的准确率,准确率保持在93.24%以上。在三条道路以上进行实验,垂直投影算法和曲线拟合算法表现极差,平均准确率分别只有13.29%和13.3%。定位信息与道路匹配的方法较好,相比垂直投影算法提高了70.04%,相比曲线拟合法平均提高了70.03%。原因在于定位信息与道路匹配的方法筛选过滤掉不相关路段,充分考虑车辆和道路的速度信息,对路段的匹配结果更精确。相对于垂直投影算法利用空间距离思想和曲线拟合算法利用曲线斜率算法,定位信息与道路匹配的方法有较高的准确率。

[0150] 图10表示了GPS模拟数据和真实浮动车数据使用不同算法之间时间效率对比的实验结果。实验结果表明:垂直投影算法和曲线拟合算法时间效率相差不大,平均单点匹配时

间约为0.9s,时间复杂度均为 $O(nm)$ (n 是待匹配GPS点个数, m 是道路网络路段数量)。相对于垂直投影算法和曲线拟合算法,定位信息与道路匹配的方法运行时间显著下降,平均单点匹配时间为8ms,时间复杂度为 $O(n \log(m))$ 。原因在于定位信息与道路匹配的方法采用Geohash技术搜索本区域内的路段和使用后向夹角对候选路段进行筛选,时间效率得到大幅度的提升。

[0151] 图11a、11b、11c表示了分别在一条道路、两条道路和三条道路以上地图匹配效果对比图。实验结果表明,定位信息与道路匹配的方法在一条道路和两条道路上匹配准确率较高。相比垂直投影算法和曲线拟合算法,定位信息与道路匹配的方法在三条道路以上匹配效果更为突出。由图11c可以看出来,在环形路口等复杂道路仍具有良好的匹配效果。

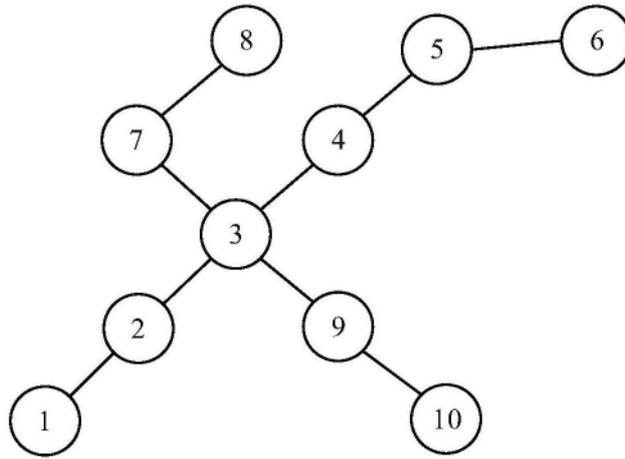


图1

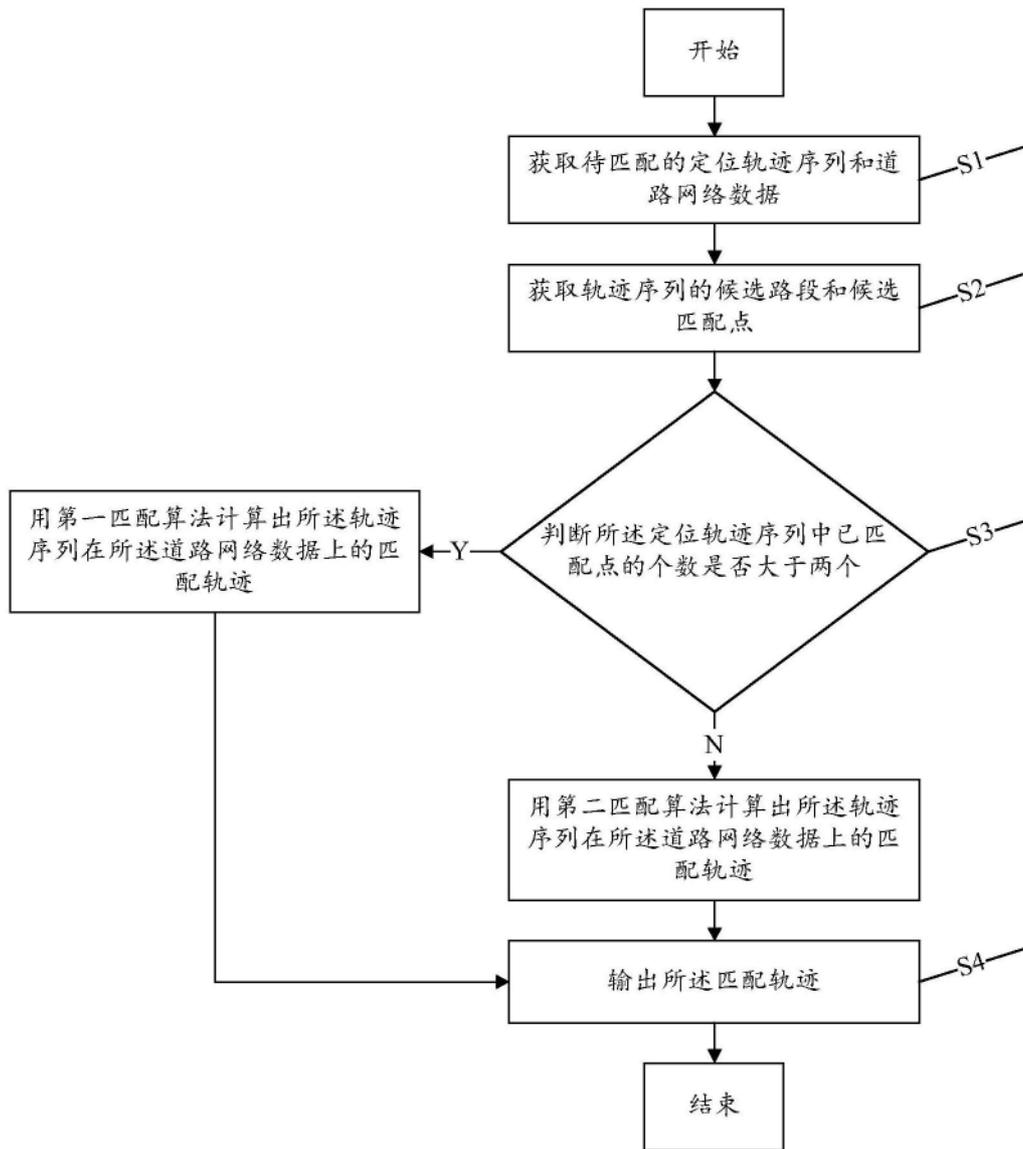


图2

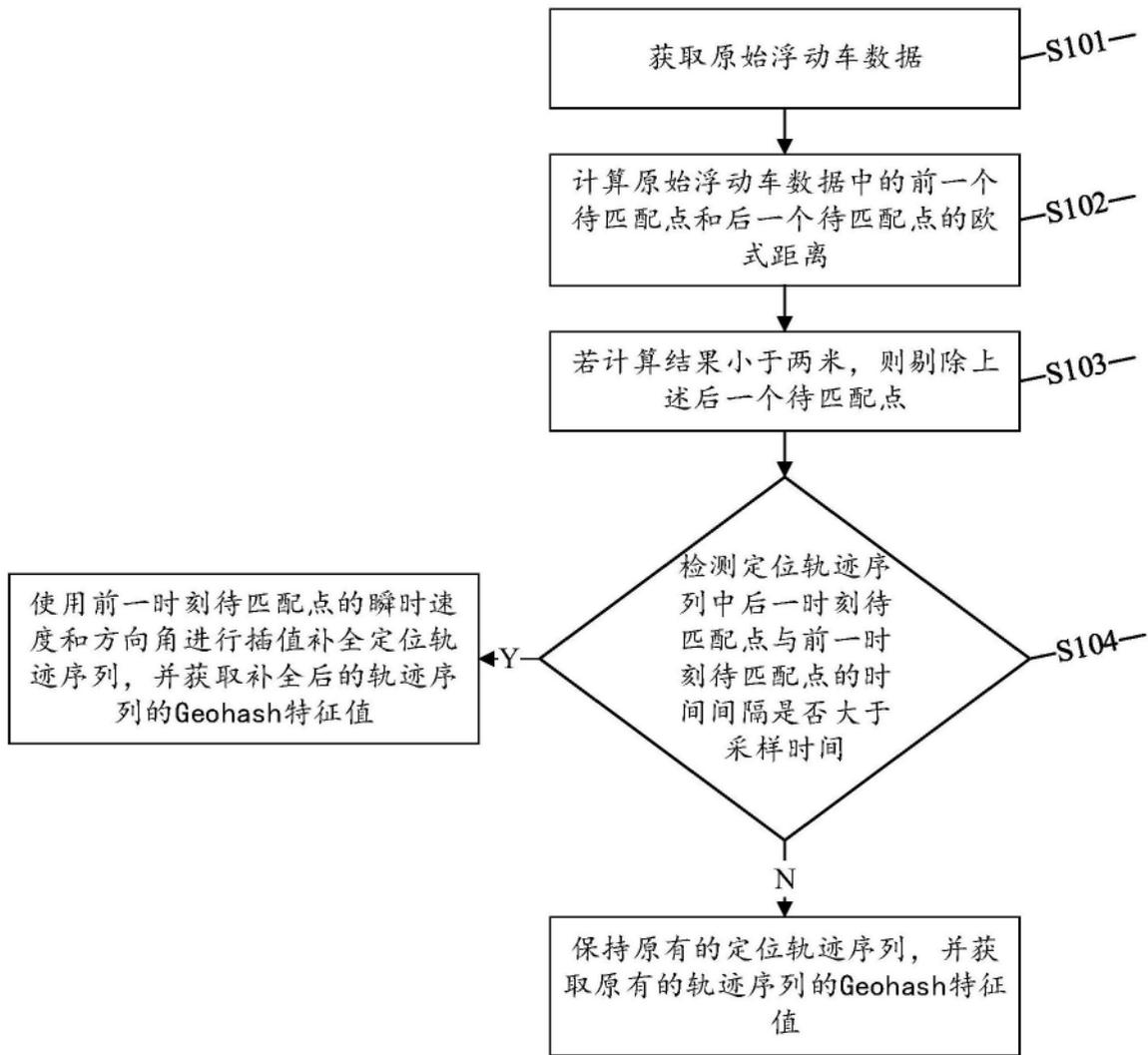


图3

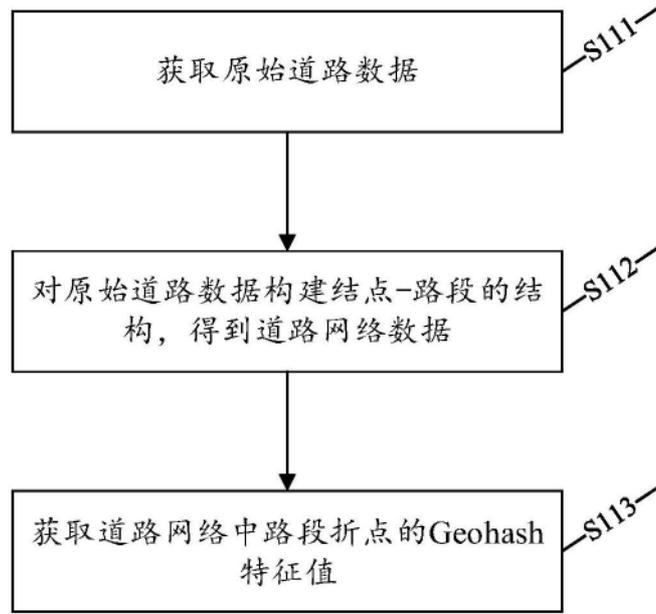


图4

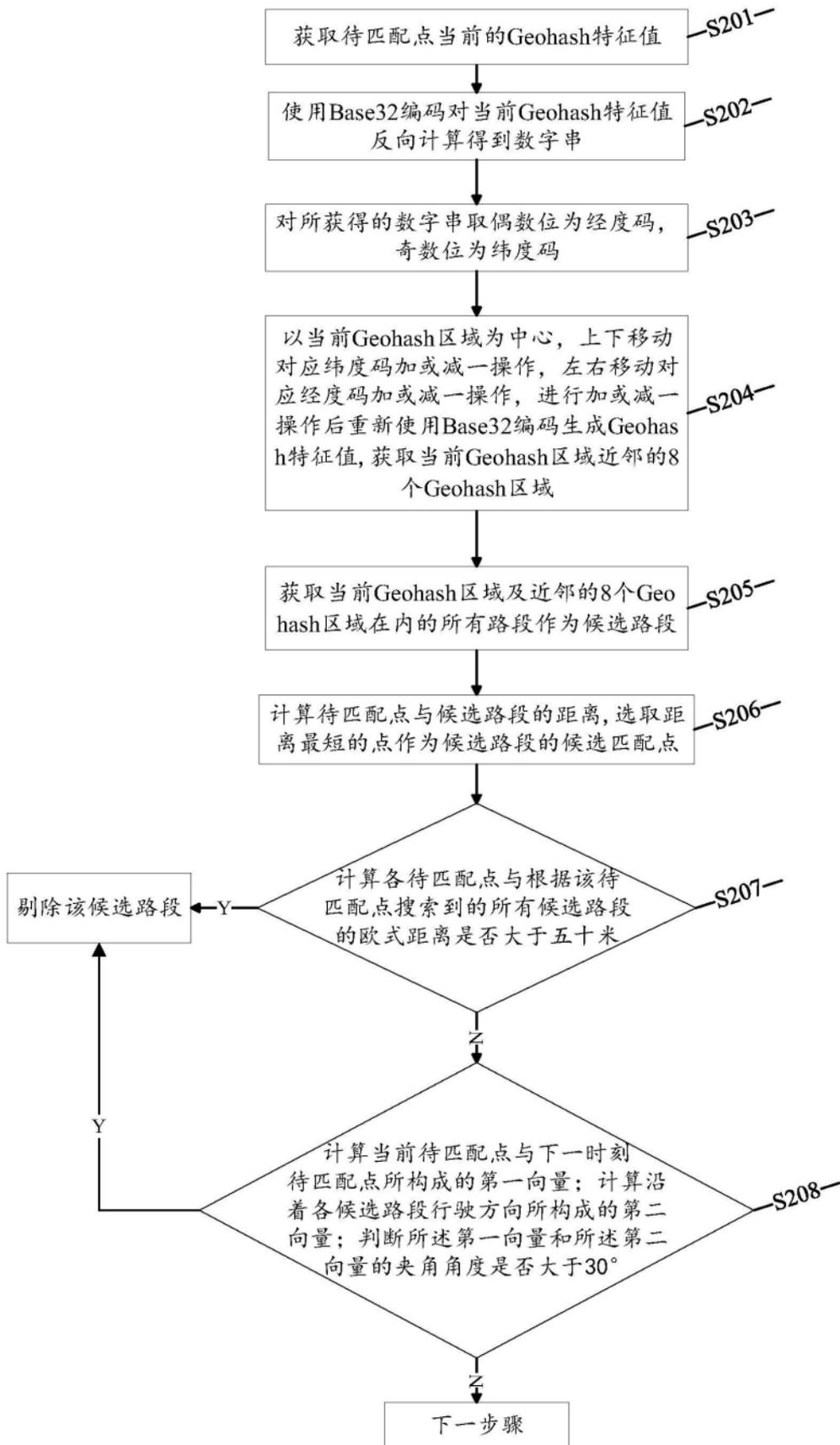


图5

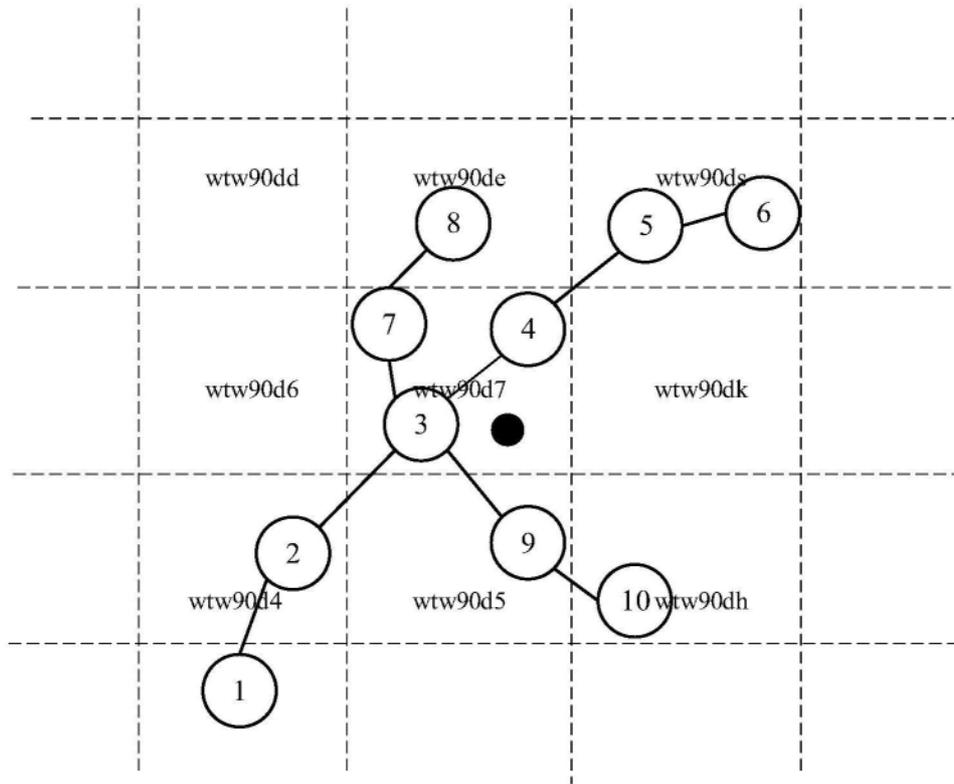


图6

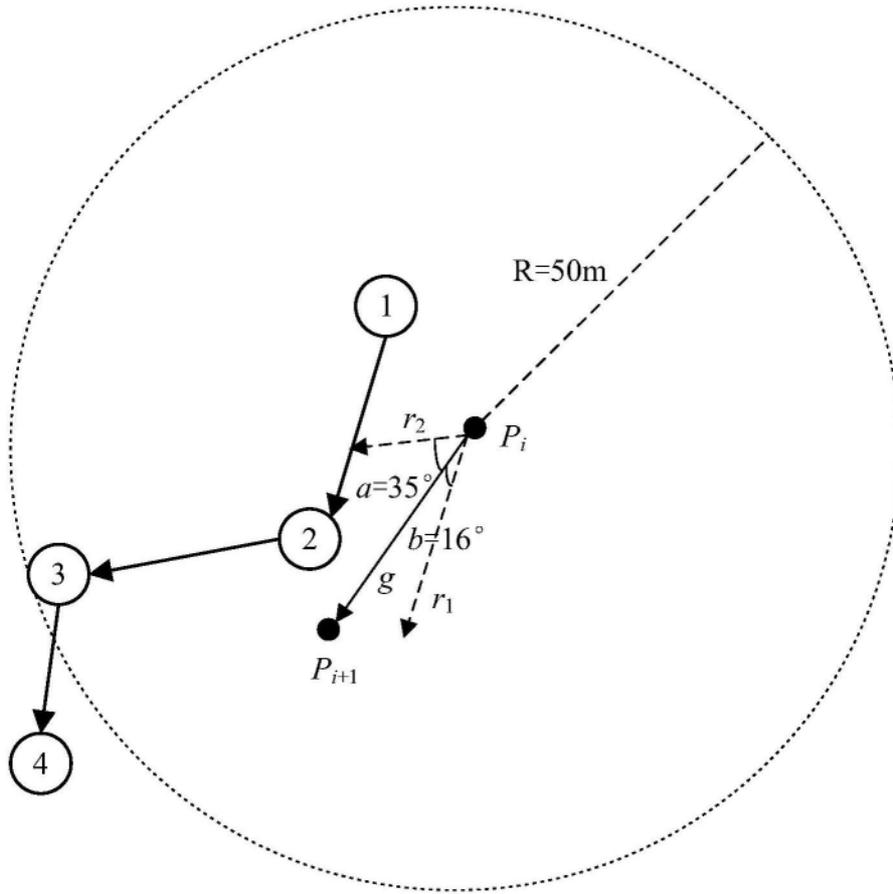


图7

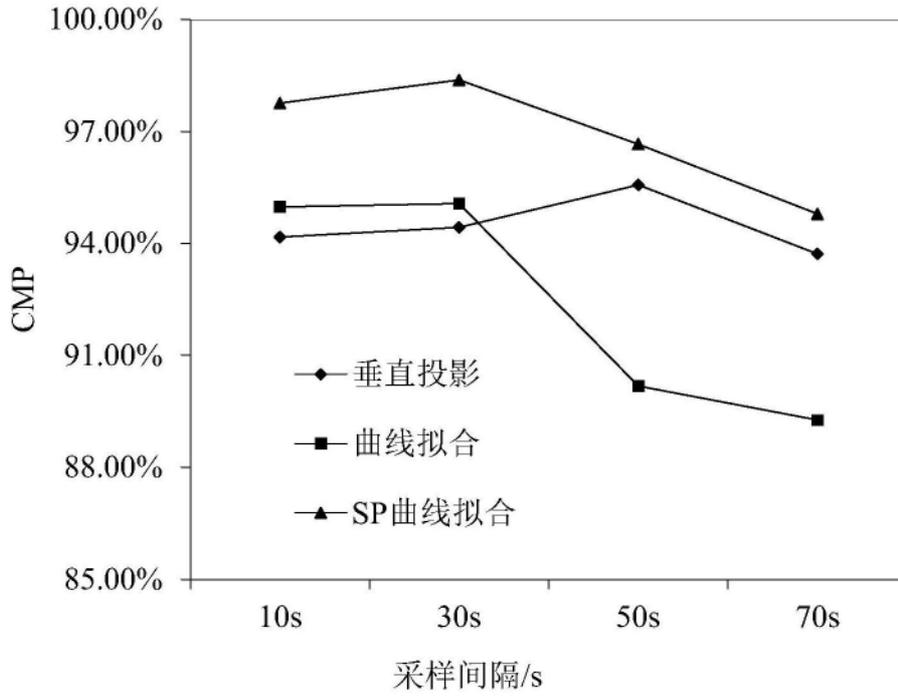


图8a

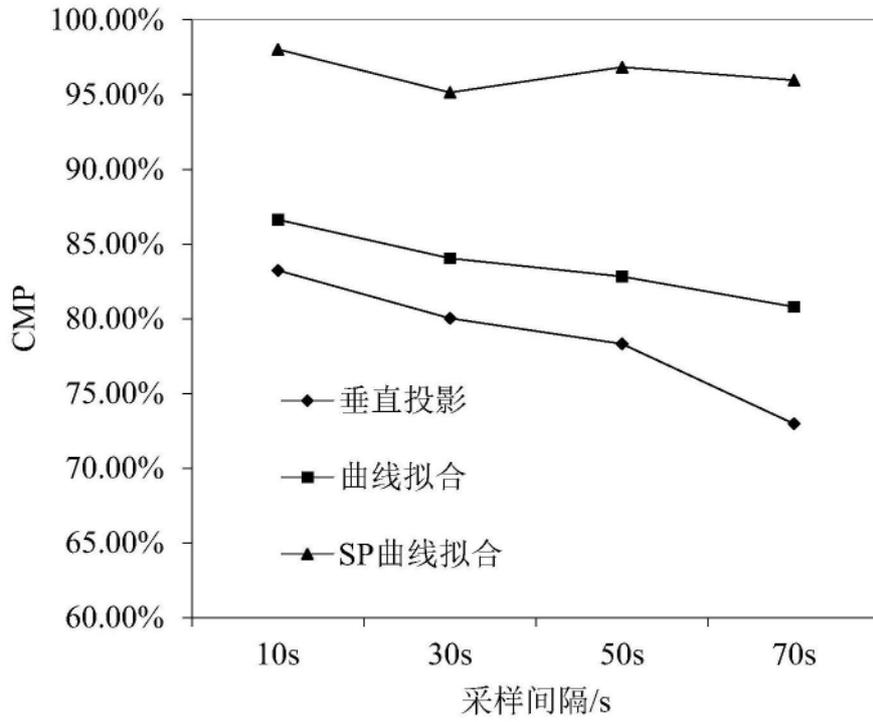


图8b

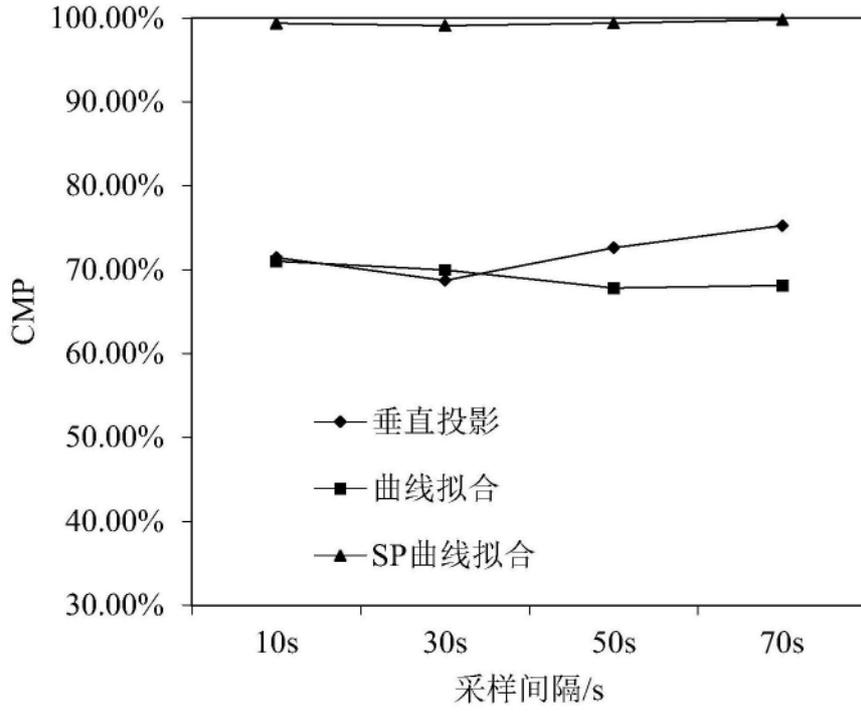


图8c

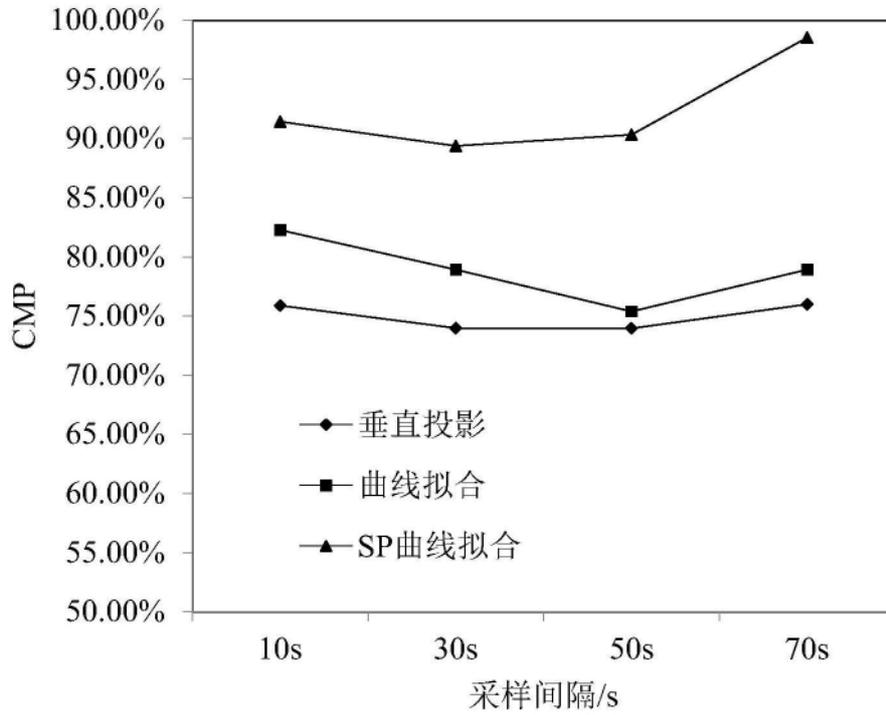


图9a

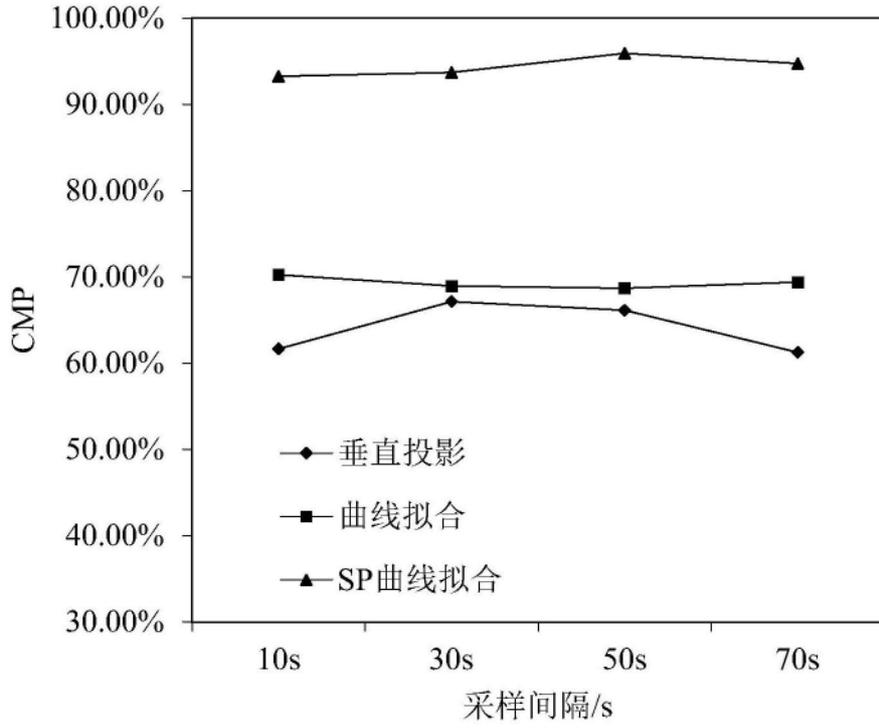


图9b

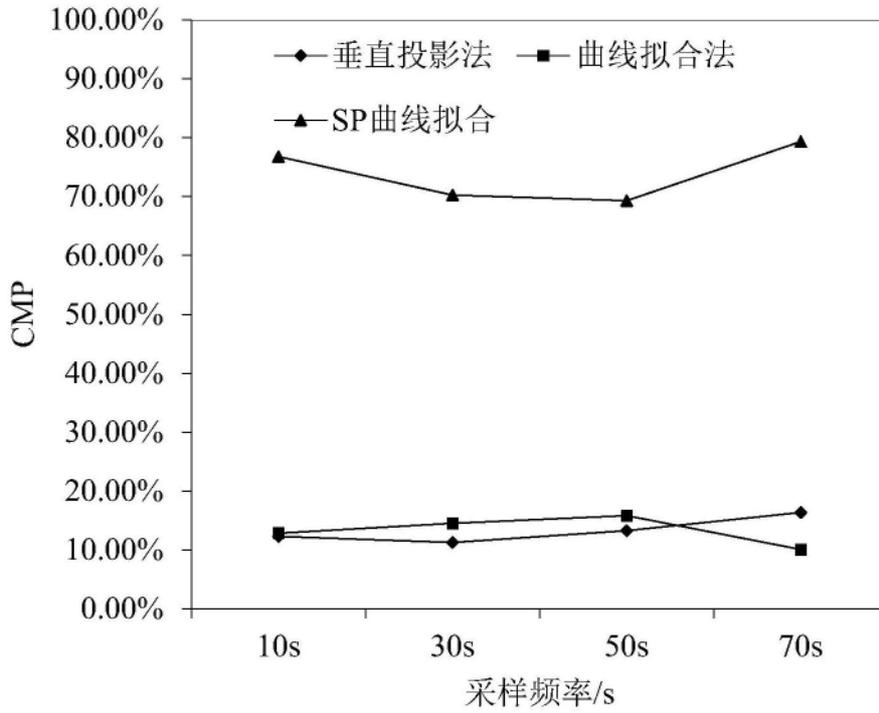


图9c

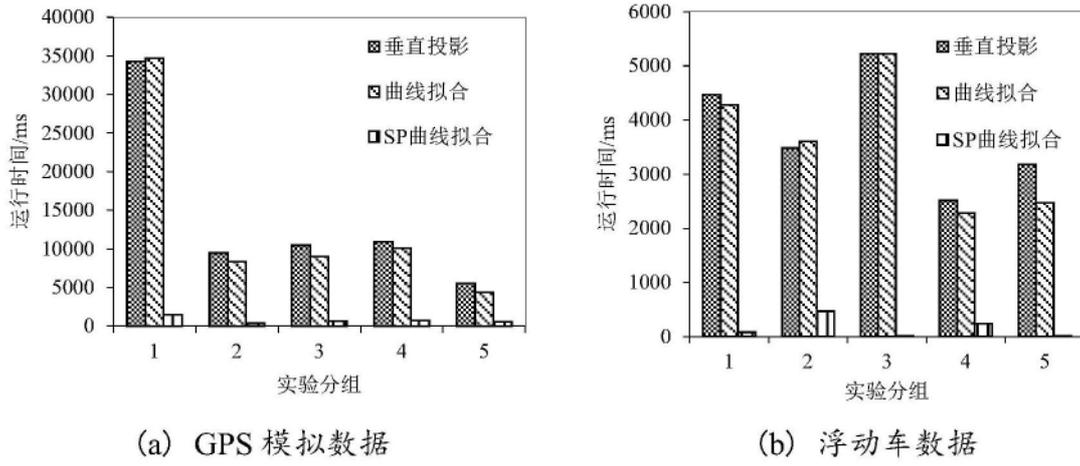


图10

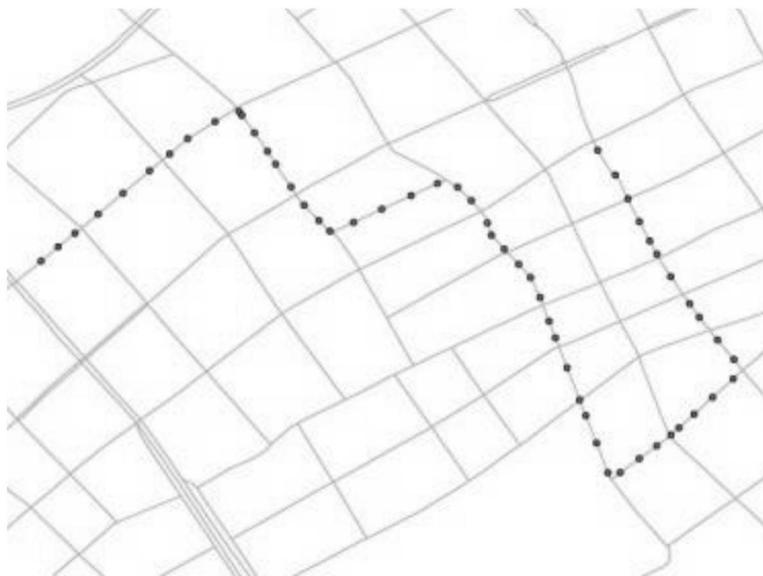


图11a



图11b

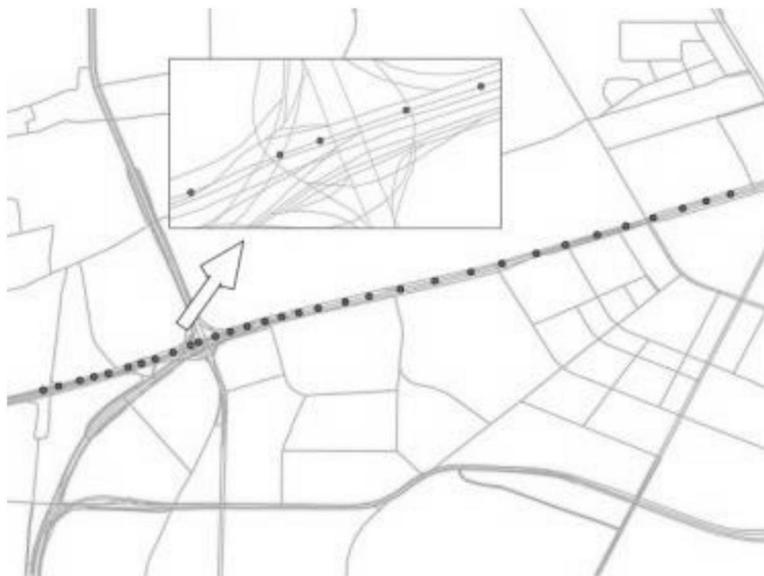


图11c

Decimal	0	1	2	3	4	5	6	7	8	9	10
Base32	0	1	2	3	4	5	6	7	8	9	b
Decimal	11	12	13	14	15	16	17	18	19	20	21
Base32	c	d	e	f	g	h	j	k	m	n	p
Decimal	22	23	24	25	26	27	28	29	30	31	
Base32	q	r	s	t	u	v	w	x	y	z	

图12