



北京大學

本科生毕业论文

题目: 基于粗粒度手机位置信息的交
通行为识别

Traffic Behavior Recognition Based on
Coarse-Grained Cellphone Location
Information

姓名: 张闻涛

学号: 1300012758

院系: 信息科学技术学院

本科专业: 计算机科学与技术

指导教师: 邓习峰

二〇一七年五月

北京大学本科毕业论文导师评阅表

| | | | | | |
|--|--|----------------|--|--------|--|
| 学生姓名 | | 学生学号 | | 论文成绩 | |
| 学院(系) | | | | 学生所在专业 | |
| 导师姓名 | | 导师单位/ 所在研究所 | | 导师职称 | |
| 论文题目 (中、英文) | | | | | |
| 导师评语 (包含对论文的性质、难度、分量、综合训练等是否符合培养目标的目的等评价) | | | | | |
| 导师签名: | | | | | |
| 年 月 日 | | | | | |

版权声明

任何收存和保管本论文各种版本的单位和个人，未经本论文作者同意，不得将本论文转借他人，亦不得随意复制、抄录、拍照或以任何方式传播。否则，引起有碍作者著作权之问题，将可能承担法律责任。

摘要

随着智能移动设备的普及，关于人们位置信息的数据日益增加。在交通行为识别中，这些位置数据能起到十分重要的作用，特别是用于交通路线识别。

本文针对粗粒度的手机位置信息是否能有效地用于交通路线识别进行了研究。研究主要通过模拟实验。实验构建了网格状、环状两种交通道路网模型，对于采样获取的粗粒度位置信息，用最近点、最近边、多层规划三种不同的算法进行交通路线拟合，分别测试了与实际路线的匹配程度，并分析了三种算法的优劣。

关键词：粗粒度 位置信息 路线拟合 匹配度

Abstract

With the popularization of smart mobile devices, the source of people's location data also increases. The location data plays a very important role in traffic behavior recognition, especially for traffic route recognition.

We studies whether we can effectively use the coarse-grained location information in traffic route recognition. The study was mainly through simulation experiments. In the experiment, a grid-like and a ring-shaped road system models are constructed. To retrieve the traffic route with the coarse-grained location information, we propose three different algorithms, including nearest point algorithm, nearest edge algorithm, and multi-layer planning algorithm. We calculate the matching degree between the actual route and the route retrieved by algorithms, to evaluate three algorithms.

Key Words: Coarse-grained, Location Information, Traffic Route Retrieving, Matching Degree

全文目录

| | |
|----------------------------|-----------|
| 摘要..... | 1 |
| Abstract..... | 2 |
| 第一章 绪论..... | 5 |
| 1. 课题研究背景..... | 5 |
| 2. 本课题研究的意义..... | 5 |
| 3. 相关工作..... | 5 |
| 4. 本论文研究思路和所做的主要工作..... | 5 |
| 第二章 交通行为识别实验设计..... | 7 |
| 1. 仿真实验设计说明..... | 7 |
| 2. 交通道路网设计..... | 7 |
| 2.1 道路网布局介绍..... | 7 |
| 2.2 道路网模型构建..... | 8 |
| 3. 交通路线设计..... | 9 |
| 4. 采集粗粒度定位信息..... | 9 |
| 4.1 基站定位介绍..... | 9 |
| 4.2 定位数据采集方式..... | 10 |
| 5. 拟合算法设计..... | 10 |
| 5.1 最近路口（最近点）拟合..... | 10 |
| 5.2 最近道路（最近边）拟合..... | 10 |
| 5.3 多层规划拟合..... | 11 |
| 6. 拟合结果与真实路线的匹配度..... | 12 |
| 第三章 交通行为识别实验实现..... | 14 |
| 1. 实验的实现说明..... | 14 |
| 2. 道路网构建的实现..... | 14 |
| 2.1 存储数据结构选择..... | 14 |
| 2.2 环状道路网构造..... | 15 |
| 2.3 网格状道路网构造..... | 15 |
| 3. 最短路径模块的实现..... | 15 |

| | | |
|------------|------------------------|-----------|
| 4. | 构造交通路线的实现..... | 16 |
| 5. | 粗粒度采集的实现..... | 16 |
| 6. | 拟合算法的实现..... | 17 |
| 6.1 | 最近点拟合..... | 17 |
| 6.2 | 最近边拟合..... | 17 |
| 6.3 | 多层规划拟合..... | 18 |
| 7. | 匹配度计算的实现..... | 19 |
| 第四章 | 交通行为识别实验结果..... | 20 |
| 1. | 测试结果说明..... | 20 |
| 2. | 网格状道路实验结果..... | 21 |
| 3. | 环状道路实验结果..... | 23 |
| 4. | 实验结果分析..... | 25 |
| | 结束语..... | 26 |
| | 参考文献..... | 27 |
| | 致谢..... | 28 |

第一章 绪论

1. 课题研究背景

在交通发达地区，如北京、上海、深圳，道路车流量巨大，堵车是常见的现象。因此，在城市规划中，交通规划占据很重要的一部分。了解人们的交通行为有助于在交通规划中做出更好的决策。交通行为可以通过不同的方式进行分析，例如基于交通监控视频的跟踪分析。而在互联网时代，每个人都离不开智能手机等移动设备，每个移动设备都会产生粗粒度的定位信息，可用于交通轨迹的分析。本课题着重于研究基于粗粒度定位信息的交通行为识别方法。

2. 本课题研究的意义

交通行为对城市环境、能源消耗以及拥挤程度都有影响，研究交通行为有助于城市交通规划。另外，利用粗粒度手机定位信息不同于传统的监控视频分析的方式，本课题关心的交通行为是指交通路线、交通方式，忽略细小的交通行为，如违停、变道、超车等。定位信息主要来源于通信基站，往往具有较大的误差，但有一定的分布规律。通过数学模型进行拟合，结合道路匹配与数据挖掘的方法，分析出交通工具、交通路线，并且可以批量的分析，以获得整体的交通情况。

3. 相关工作

部分关于交通行为识别的工作注重于对图像处理算法的研究，主要利用庞大的道路监控视频数据。而在[3]中，作者利用一个地图服务系统的 API，设计算法应用于交通拥堵识别，取得了良好的效果。[4]中，作者对高速公路的交通路径二义性识别进行了研究，提出了多种技术方案，并且分析了其优缺点。这些文章对我有一定的启发。

4. 本论文研究思路和所做的主要工作

我们通过仿真构造，产生了城市的道路网络模型，并且模拟不同车辆的出行方式，出行路线，按一定时间间隔采集有误差的粗粒度定位信息。作为测试数据

集。

另外，对于交通路线识别算法的准确度的评判，定义了一种比较合理的匹配度标准。

我们尝试使用多种算法对交通路线进行拟合，包括最近点拟合、最近边拟合，多层次的路线规划、最大概率规划等拟合算法。我们尝试了从朴素到综合性强的算法，进行测试，得出了详细的优劣情况。

第二章 交通行为识别实验设计

1. 仿真实验设计说明

由于基站定位数据涉及到每一个用户的隐私，数据的获取需要特定的渠道和方式，我们本次实验决定使用仿真采集数据的方式来模拟真实情况。故从道路交通网络模型和车辆路线与实际情况有一定的差别。但为了接近真实情况，道路、车辆的数据尽量与现实保持一致。当然，仿真实验也有它独特的好处，方便修改道路网模型、方便测试多种情况的车辆路线，对实验来说更加灵活

2. 交通道路网设计

2.1道路网布局介绍

目前城市道路布局大致有以下几种方式：

网格状道路网（棋盘式），其特点是道路横平竖直，容易辨认方向。所划分的街坊较为规整。同时，道路密集、交叉口多，功能不易分清；对角方向的街道较少，从一个起点到目的地的道路非直线系数较大。

环形放射式道路网，其特点是市中心和郊区联系方便，道路非直线系数小；道路有曲有直，宜于适合各种地形；比较易于组织。过境车辆可通过环线过境，不增加中心城区交通压力。但这种道路容易把交通导向市中心，造成市中心交通混乱；环向道路不易辨认方向。

混合式；因地制宜，在不同的区域采取最佳路网形式。国内很多城市都是在历史上形成中心城区方格式，后期扩张采用了环状加放射的形式，实际上严格说北京也算混合式，比如中心老城一带呈方格式，整体又呈环状放射式。他基本上兼具了方格式和环状的各种优缺点。

另外，自由式布局也是一种形式。

2.2道路网模型构建

本人生活在北京，较为熟悉北京的交通网络（如图 2.1），故我们将北京的环状放射式道路网加入到道路网模型中。北京有 7 条环路，从最内环的紫禁城一带，到最七环的京津冀环线高速公路，环与环之间有道路相连。

另外，曼哈顿方方正正的棋盘式道路颇有特色（如图 2.2），道路横平竖直，规整分布，易于构造，所以我们也尝试了构建这种道路网模型。

模型中需要体现出道路、路口两个主要元素。道路具有通畅度和长度的属性。

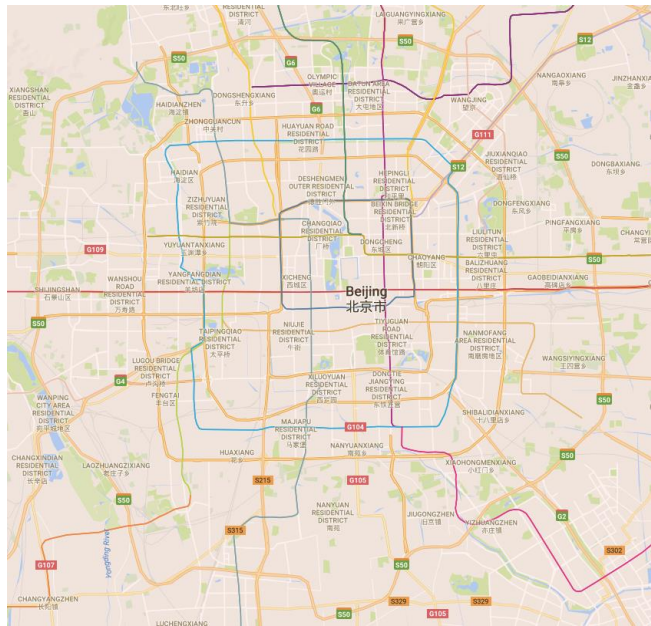


图 2.1 北京环路（来自谷歌地图）

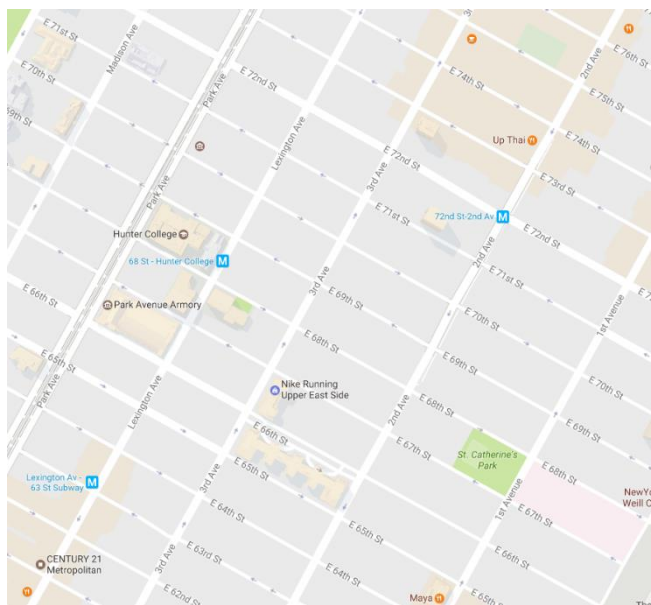


图 2.2 曼哈顿棋盘式道路网（来自谷歌地图）

3. 交通路线设计

为了测试的方便，我们把整条交通路线的起点和终点都选取在路口上。在日常生活中，人们一般会选择消耗时间最短的路线。根据中这种情况，我们的交通路线有如下设计：

最简单的路线由单一的起点和终点组成，它们之间的最短路径，所经过的路口和道路即形成一条交通路线。按我们起点与终点都在路口上的简化设定，只需要记录所有经过的路口即可记录一条交通路线。

实际的道路会有拥堵情况，与车流量、时间、道路宽度都有关。在实验中我们将其简化成通畅度这一简单的属性。在交通路线设计中，我们让车辆在满足行经最短路径的情况下，尽量选择通畅度高的路线。

另外，车辆的一次交通行为不会是简单的一个起点和终点。例如一位驾车出行的男士可能会从公司先去便利店购物，再回到家中，那么就产生了一个路线中的中间点，且在中间点停留时间对于粗粒度定位信息来说非常短，不足以视作两次路线。因此，我们在路线中加入一个到多个中间点，先从起点出行到中间点，再前往终点，以使实验更贴近实际。

4. 采集粗粒度定位信息

4.1 基站定位介绍

蜂窝网络由蜂窝基站构成的，这些蜂窝基站把整个通信区域划分成一个个蜂窝小区。这些小区小则几十米，大则几千米。GSM 定位，就是借助这些蜂窝基站进行定位。定位的原理有如下几种：

COO 单基站定位，即根据设备当前连接的蜂窝基站的位置来确定设备的位置。

七号信令定位，以信令监测为基础，能够对移动通信网中特定的信令过程，如漫游、切换以及与电路相关的信令过程进行过滤和分析，并将监测结果提供给业务中心，以实现特定用户的个性化服务。

基于到达时间和到达时间差的三基站定位，是基于电波传播时间的定位方法。

基于到达角度计算的两基站定位。

基于场强的定位，是通过测出接收到的信号场强和已知的信道衰落模型及发射信号的场强值估计收发信短的距离，根据多个三个距离值就可以得到设备的位置。

混合定位，就是同时使用两种以上的定位方法来进行定位。通过各种定位方法之间结合使用，互补短长，以达到更高的定位精度。

定位精度在较好的情况下，可达 50 米以内，较坏情况下定位精度只能粗略到几千米。考虑到城市基站分布较为密集，实验中我们采用了较为乐观的误差数值。

4.2 定位数据采集方式

定位数据采集中，我们模拟多种交通工具，把真实情况简化，按既定的交通路线以匀速行使。对于不同规模的道路模型和不同的交通路线，我们设定了不同的采集频率，下至 10 分钟，上至 30 分钟。模拟从起点出发，按照设定的时间频率进行打点，在当前所在的道路上计算出坐标值。另外，为了构造合理的定位误差，我们采用了两种分布方式，一种是按照误差区间均匀分布，另一种是东西和南北坐标相互独立的二维正态分布。具体的分布参数将会在实现中说明。

5. 拟合算法设计

5.1 最近路口（最近点）拟合

依次访问每个采集点，选取离各个采集点欧氏距离最近的路口。此算法朴素地认为离采集点最近的路口就是交通路线中最可能经过的那些路口。按照最短路径的出行方式，依次经过每个路口，最终到达最后一个采集点，生成整条拟合的路径。

5.2 最近道路（最近边）拟合

依次访问每个采集点，选取离各个采集点最近的道路。最近道路的选取不同于最近路口，需要求出采集点到道路所在的线段的距离。此算法认为离采集点最

近的道路很有可能是交通路线中所经过的道路。

与最近路口拟合相同，按照最短路径的出行方式，依次经过每个道路的两个点，最终到达最后一条道路的其中一端生成整条拟合的路径。

但是，连接这些道路的最短路径的求法不同于最近路口拟合。此算法中，从起始点开始，我们尝试分别到达下一条需要经过的道路的两端，取两端中较近的一端，把从当前位置到这一端的路径加入拟合路线，同时舍弃另一端。然后再尝试经过下一条道路。我们可以用舍弃一端的处理方式，是因为若这条道路确实在真实路线上以最短路径的形式经过，那么在前往下一条道路时我们自然会到达另一端，否则的话舍弃它也不会使路线结果变差。

5.3 多层规划拟合

之前的算法均未考虑车辆行驶速度和采集频率的关系。在道路通畅的情况以及采样频率较低的情况下，我们可以认为车辆是匀速行驶的，反应在基站定位数据的特点上，就是两个采样点之间的路线长度是相近的。前两种算法没有利用这一点，而且对于粗粒度采样点的还原仅仅保留了最少的可能性，即认为只有最近一个路口或一条道路是实际定位点所在的区域。为了改进它，我们扩充实际定位点的可能性，选取多条道路的路口进行路线规划尝试，而尝试的依据就是对匀速行驶进行采样的特性。

文献[4]中针对高速公路二义性路径识别，提出了一种多级目标规划算法。受其启发，我们对于每隔一段时间采集的粗粒度定位信息，找到离它较近的那些道路的路口，这个路口集合作为一个层。如此划分层次，采样了几次就有几个路口层。从起点的下一层开始，对于当前层的每一个路口，尝试从上一层的每一个路口行经最短路径到达。我们要求上一层的点保存下之前的路线方案中行驶的速度，根据这个速度，可以找出如果路线到达当前这个路口，最大概率是来自上一层的哪个路口。如此就把上一层的路线结果更新到了当前层，也可以获得行驶的速度。直到最后一个定位点，可以选取与各采样点偏差最小的路线作为答案。另外也可以各段速度的偏差等其他因素进行考量来选取路线。

6. 拟合结果与真实路线的匹配度

匹配度对于拟合结果好坏的判定十分重要，故我们需要找到一种简单有效的匹配度定义方式。

我们考虑过多种匹配度计算方式，有两种定义方式比较合理：

第一种是对于拟合路线与真实路线，构建成一个二部图，两两连边。拟合路线的每条道路与真实路线的每条道路用几何公式计算出道路间的最短距离（即线段的距离），平滑处理后取其倒数作为二部图的边权。对于二部图采用 Kuhn-Munkres 算法求最优匹配，作为匹配程度。

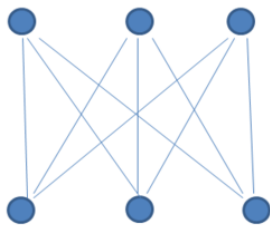


图 2.3 道路作为点的二部图

在这种二部图匹配的方法中，一条道路有可能与较远的路径相匹配，看似忽略了路线的连续性。但实际上，由于所求出的最优匹配的特性，实际路线和拟合路线中相近的路线会匹配在一起，拟合路线中连续的道路所匹配的实际道路也会是连续的，否则对于匹配度的贡献就很小。

第二种计算方式是对于拟合路线与真实路线，利用其序列的存储方式来进行匹配。路线的存储，最直接的一种方式就是记录下所经过的每一条道路或每一个路口。以道路为例，我们将道路用数字编号之后，路线就成为了一个由数字组成的线性序列。对于拟合路线和真实路线这两个线性序列，它们的匹配程度可以用与最长公共子序列相关的参数来表示。这里我们取最长公共子序列的长度与较长序列的长度比值作为匹配程度。取较长的序列长度作为分母，是为了避免因为有意地加长拟合路线而使公共子序列也能加长，从而得到较高匹配度的这种情况。

举一个计算的例子，假设真实路线经过了：道路 1 道路 3 道路 4 道路 5 道路 7。

拟合路线经过了：道路 1 道路 2 道路 8 道路 4 道路 5 道路 7。

它们的最长公共子序列为 1 4 5 7，其长度为 4，较长的拟合路线长度为 6，

则匹配度为 66.67%。

第二种匹配度计算方式简洁而且合理。由于第一种定义的距离计算过程和匹配算法实现较为复杂，并且需要通过多次尝试得出合适的参数平滑方式，计算所消耗的时间也较多，故我们最终采用了第二种匹配度的定义。

第三章 交通行为识别实验实现

1. 实验的实现说明

本次交通行为识别的实验主要基于 Python 3.5.1 实现，利用了一些数值计算库、绘图库。

根据程序的结构，分为两个部分：数据集构造部分，拟合及匹配度测试部分。数据集构造部分包括：交通道路网构建模块，获取最短路径的模块，构造交通路线的模块，采集粗粒度定位信息的模块。拟合及匹配度测试部分包括：交通道路网载入模块、拟合计算模块、匹配度计算模块。

2. 道路网构建的实现

2.1 存储数据结构选择

显而易见的，一般的交通道路网模型符合图论中无向图的结构，即路口视作无向图中的点，道路视作无向图中的边，这里我们暂时忽略单行道的情况。设置无向图的边权为道路拥堵程度，用于后续的路线选择。而且，道路网中的道路和路口是有实际地理坐标的，故对于每个路口（每个无向图中的点），需要存储该点的横纵坐标。

另外，实际的道路交通网有一个性质，即在不考虑高架桥的情况下，是一个平面图，不会出现边与边交叉的情况。所以图中的边是相对比较稀疏的，我们使用邻接链表来存储图中的边。邻接链表的存储结构如图 3.1

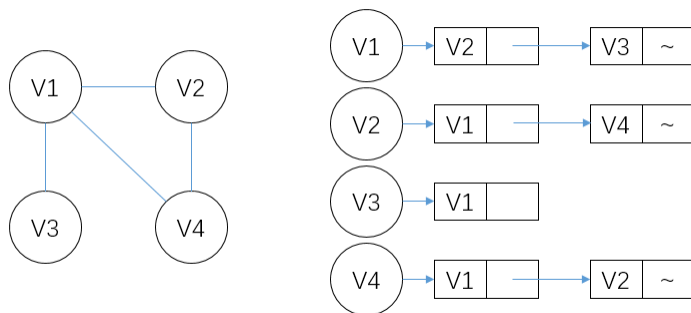


图 3.1 无向图的邻接链表存储

2.2 环状道路网构造

环形道路在我们的模型中简化为正方形，换半径即边长的一半。首先随机确定要构造几条环形道路，确定每个环形的大小，保证环形道路的环半径从小到大。确定这些之后，我们将环形道路安排到坐标系中，做法是：将最大的环形道路的左下角作为坐标原点。有了这个基准，再确定每个环上各个路口的坐标。

在相邻的两个环之间，对应的路口按照一定的概率进行道路连接，对于主要的八个路口（环的四个角以及四个边的中点），我们设定较大的道路连接的概率值，其他路口采用较小的概率。

再将路口进行编号，依次将路口坐标信息和所有道路的信息传给构造无向图的模块进行结构化存储。

2.3 网格状道路网构造

通过随机确定网格的行数和列数，然后依据实际道路的长度确定各个行与行之间的间隔，列与列之间的间隔。以此为基准确定各个路口的横纵坐标。再对各行各列进行网格状的道路连接，对于这些横向和纵向上相邻的路口，以较大的概率进行道路连接，对于斜对角相邻的路口，以较小的概率进行道路连接。

构造完成后，将路口和道路信息进行结构化存储。

3. 最短路径模块的实现

最短路径模块主要实现了以一个点作为起点，求到各个点的最短路径的功能，使用的是[5][6]中提出的 SPFA (Shortest Path Faster Algorithm) 算法。该算法本质上是 Bell-man-Ford 算法的一个“队列”实现，其主要思想是动态优化逼近。算法中，设 D 数组是记录当前从起点到其余各点的最短路径的值，初始时除起点外都为无限大，另设立了一个先进先出队列 Queue 用来保存待优化的顶点，优化时从此队列里顺序取出一个点 w ，并且用 w 点的当前路径 $D[w]$ 去优化调整其他各点的路径值 $D[j]$ ，若有调整，即 $D[j]$ 的值改小了，就将 j 点放入 Queue 队列以待继续进一步优化。反复从 Queue 队列里取出点来对当前最短路径进行优化，直至队列为空不需要再优化为止，此时 D 数组里就保存了从起点到各点的最短

路径值。

在道路网中求最短路径，我们还考虑了道路的拥挤程度，对于来自两个不同路口的相同长度的路径方案，优先选择拥挤程度良好的那条道路。另外，还需要记录下最短路径的具体路线，这只需要在 D 数组的基础上，增加一个 F 数组，对于 i 路口，路径长度为 $D[i]$ 的情况下，上一个经过的路口记录在 $F[i]$ 中。要还原出一条最短路径，只需要从终点开始沿着 $F[i]$ 数组不断往回走，直到起点。

4. 构造交通路线的实现

随机选取起点，终点，以及中间点，利用最短路径模块的功能求出起点到各个中间点，然后再到终点的路线。该路线作为实际行驶路线，进行基站定位数据采集。

5. 粗粒度采集的实现

首先沿着给定行驶路线进行定位数据精确采样，设定车辆行驶速度为 10 千米至 50 千米每小时，采样时间为 10 至 30 分钟一次。在精确采样获得的采样点上，加入误差。

误差采用两种分布方式，第一种是平均分布，分布范围最小设定在精确采样点的 50 米内，最大在 300 米内，平均分布只需要通过随机一个点，使其与目标点距离在规定范围内即可。另一种分布方式是正态分布，生成正态分布需要使用 Numpy 库，实现中， σ 的取值下至 150，上至 300，生成若干以原点为中心，横纵坐标相互独立的二维正态分布点，再与精确采样点的横纵坐标相加得到带有误差的基站定位点。至此，完成了数据集的构造。

图 3.2 是一个模拟基站信令数据的例子，蓝色 X 型标注了各个定位点的位置，绿色的路线代表实际的行驶路线，红色的细线是后续算法拟合出来的路线。

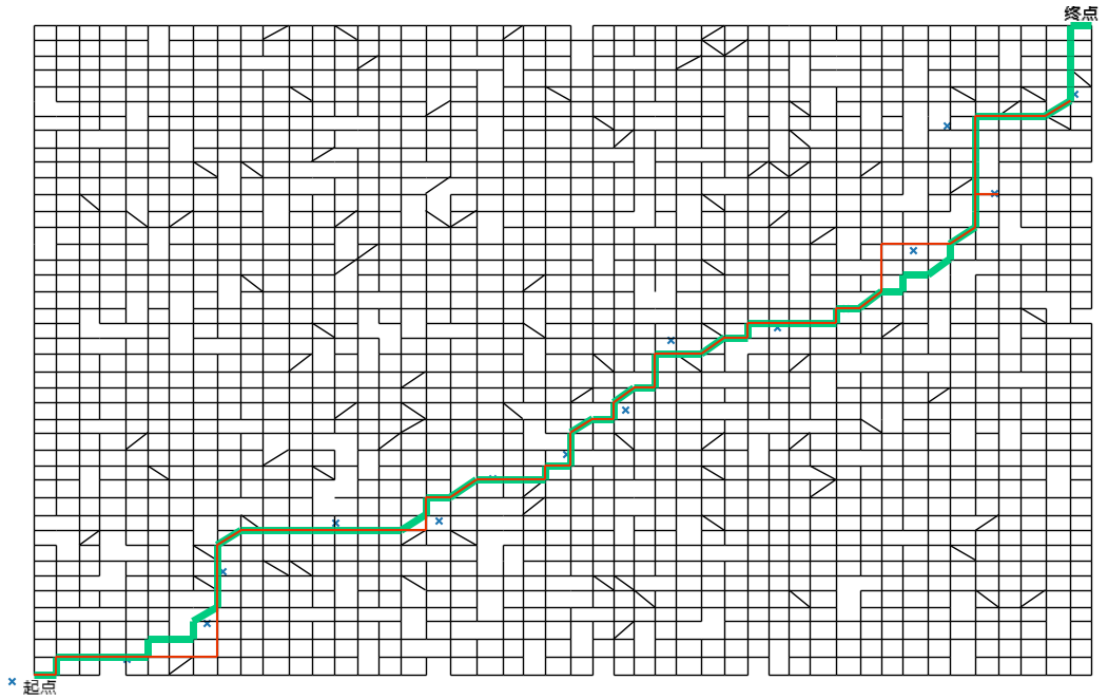


图 3.2 模拟路线识别的一个例子

6. 拟合算法的实现

6.1 最近点拟合

对于每一个带有误差的基站定位采样点，遍历道路网络中的所有路口，计算出两点之间的欧几里得距离，取距离误差点最近的那个路口作为还原出来的采样地。将各个时间的误差点还原出来的路口，按照采集顺序排列，再通过最短路径算法进行连接起来，就形成了一条拟合路径。

6.2 最近边拟合

依次访问每个采集点，选取离各个采集点最近的道路。这里需要计算点（采集点）到线段（道路）的距离，需要分情况讨论。我们将采集点沿着线段所在的直线作一条垂线，与直线的交点就是采集点在直线上的投影。如图 3.3，若投影点落在线段上，那么这条垂线的长度就是点到线段的距离，若投影点不在线段上，则点到线段的最近距离为点到线段的两端中较近的一个。

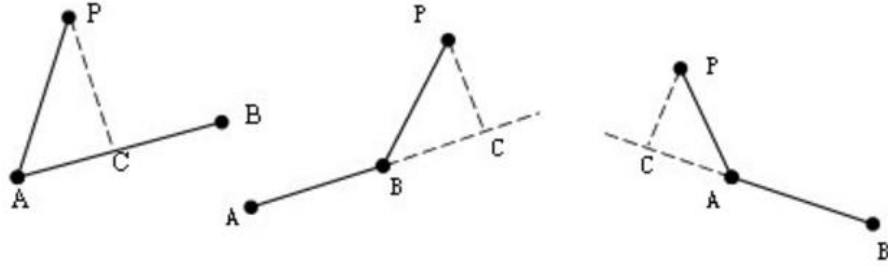


图 3.3 点到线段距离的三种情况

实际计算过程中，我们利用向量和点积的特性来简化计算。我们知道

$$(\overrightarrow{AP} \cdot \overrightarrow{AB}) = |\overrightarrow{AP}| |\overrightarrow{AB}| \cos\theta$$

所以，点乘若为负数，可以判定是图中第三种情况，则 AP 的距离即为点到线段的距离。

若点乘为正数，我们再另

$$r = (\overrightarrow{AP} \cdot \overrightarrow{AB}) / \|\overrightarrow{AB}\|^2$$

由于 $\|\overrightarrow{AP}\| \cos\theta$ 是 \overrightarrow{AP} 在 \overrightarrow{AB} 上的投影长度，所以这个 r 代表投影长度与 $\|\overrightarrow{AB}\|$ 的比值，比值大于 1 时，说明是图中第二种情况，只需求出 PB 的长度。若比值小于 1，则是第一种情况，我们先通过这个比值与 A、B 的坐标求出 C 点的坐标，再求 PC 的长度即为点到线段的最短距离。

从起始点开始，我们尝试分别到达下一条需要经过的道路的两端，通过最短路径算法模块获取路径，并且取两端中较近的一端，把从当前位置到这一端的最短路径加入拟合路线，同时舍弃另一端。然后再尝试经过下一条道路。我们可以用舍弃一端的处理方式，在之前已经论证过，舍弃它不会使路线结果变差。

6.3 多层规划拟合

路径规划的第一层只有一个点，即起始点所在的粗粒度采样点，对于此点，我们采用寻找最近点的方式来确定起始路口。从第二层开始依次到后面的所有层，对于该次采样的点，我们寻找所有与它距离 300 米范围内的街道（计算点到线段的距离的方法同最近边拟合算法）。找出与这些街道相关的所有路口，组成一个层。对于第二层，只需要简单的求出从第一层起点到第二层的各个点的最短路径，保存下路径长度，同时也作为平均每段路径长度。对于第三层以及后面的层，对

于层中的每个点 i ，枚举上一层中的各个点 j ，求出各个 j 到 i 的最短路径长度，并且取与之前平均每段路径长度的差距最小的那个 j ，作为路口 i 所来的地方，并且根据上一层的数据以及当前的层数得到这一层到达 i 路口时的的平均每段路径长度。

在最后一层，我们分别取出各个路口，沿着所记录的路线信息回溯，产生出若干条路径。对于每条路径，计算出由层所划分的各段路径长度的标准差 x ，并且对于每个粗粒度采样点，求出与拟合路径的最短距离，将其平方和开根号作为一个参考值 y 。 x 与 y 各自乘上系数求和得到路径评判值 z ，取 z 最小的那条路径作为最终的拟合路径。接下来就可以进行与实际路线的匹配度测试。

7. 匹配度计算的实现

匹配度计算中，主要是计算两条路径的最长公共子序列，再除以路径长度就得到了匹配度。以下介绍用动态规划算法来求最长公共子序列。

我们将路径 A 和路径 B 简称最长公共子序列问题中的 A 串和 B 串，用 $dp[i][j]$ 表示 A 串前 i 个符号形成的串与 B 串前 j 个的符号形成的串的最长公共子序列的长度。

则状态转移方程可写为：

$$\text{若 } A[i] = B[j], dp[i][j] \leftarrow dp[i-1][j-1] + 1$$

$$\text{否则 } dp[i][j] \leftarrow \max(dp[i-1][j], dp[i][j-1]);$$

$$\text{Answer} \leftarrow dp[N][M]$$

时间复杂度 $O(NM)$ ，其中 N 、 M 分别为两串的长度。

另外，在[7]中，作者还提出了针对该问题的一种时间复杂度更低的算法，在此不再介绍。

第四章 交通行为识别实验结果

1. 测试结果说明

由于测试涉及到大量的参数,难以完整地测试,并将完整的结果呈现在文中,故我选择了部分有代表性的数据的结果展示。并且对结果所表现出来的现象加以分析,总结。

由于采用了粗粒度定位数据,且最后一次采样至终点的数据是完全未知的,故测试中的匹配度一般是不会接近 100%的。另外,由于起点、终点、整条实际路线的随机性,不同算法所得到的结果变化较大,不稳定的结果是符合预期的。

2. 网格状道路实验结果

表 1 网格状道路实验结果 采样时间间隔 10 分钟

| 算法 | 匹配度 | 其它参数 | |
|------|-------|--------|--------------------|
| | | 行驶速度 | 误差分布 |
| 最近点 | 0.727 | 10km/h | 正态分布 $\sigma=50m$ |
| 最近边 | 0.891 | 10km/h | 正态分布 $\sigma=50m$ |
| 多层规划 | 0.745 | 10km/h | 正态分布 $\sigma=50m$ |
| 最近点 | 0.736 | 10km/h | 正态分布 $\sigma=50m$ |
| 最近边 | 0.880 | 10km/h | 正态分布 $\sigma=50m$ |
| 多层规划 | 0.584 | 10km/h | 正态分布 $\sigma=50m$ |
| 最近点 | 0.511 | 10km/h | 正态分布 $\sigma=100m$ |
| 最近边 | 0.500 | 10km/h | 正态分布 $\sigma=100m$ |
| 多层规划 | 0.711 | 10km/h | 正态分布 $\sigma=100m$ |
| 最近点 | 0.541 | 10km/h | 正态分布 $\sigma=100m$ |
| 最近边 | 0.459 | 10km/h | 正态分布 $\sigma=100m$ |
| 多层规划 | 0.686 | 10km/h | 正态分布 $\sigma=100m$ |
| 最近点 | 0.906 | 10km/h | 均匀分布 范围小于 50m |
| 最近边 | 0.887 | 10km/h | 均匀分布 范围小于 50m |
| 多层规划 | 0.622 | 10km/h | 均匀分布 范围小于 50m |
| 最近点 | 0.637 | 10km/h | 均匀分布 范围小于 100m |
| 最近边 | 0.473 | 10km/h | 均匀分布 范围小于 100m |
| 多层规划 | 0.549 | 10km/h | 均匀分布 范围小于 100m |
| 最近点 | 0.671 | 10km/h | 均匀分布 范围小于 100m |
| 最近边 | 0.664 | 10km/h | 均匀分布 范围小于 100m |
| 多层规划 | 0.678 | 10km/h | 均匀分布 范围小于 100m |

表 2 网格状道路实验结果 采样时间间隔 20 分钟

| 算法 | 匹配度 | 其它参数 | |
|------|-------|--------|--------------------|
| | | 行驶速度 | 误差分布 |
| 最近点 | 0.780 | 10km/h | 正态分布 $\sigma=50m$ |
| 最近边 | 0.772 | 10km/h | 正态分布 $\sigma=50m$ |
| 多层规划 | 0.780 | 10km/h | 正态分布 $\sigma=50m$ |
| 最近点 | 0.689 | 10km/h | 正态分布 $\sigma=50m$ |
| 最近边 | 0.655 | 10km/h | 正态分布 $\sigma=50m$ |
| 多层规划 | 0.630 | 10km/h | 正态分布 $\sigma=50m$ |
| 最近点 | 0.371 | 10km/h | 正态分布 $\sigma=100m$ |
| 最近边 | 0.750 | 10km/h | 正态分布 $\sigma=100m$ |
| 多层规划 | 0.457 | 10km/h | 正态分布 $\sigma=100m$ |
| 最近点 | 0.658 | 10km/h | 正态分布 $\sigma=100m$ |
| 最近边 | 0.849 | 10km/h | 正态分布 $\sigma=100m$ |
| 多层规划 | 0.849 | 10km/h | 正态分布 $\sigma=100m$ |
| 最近点 | 0.865 | 10km/h | 均匀分布 范围小于 50m |
| 最近边 | 0.851 | 10km/h | 均匀分布 范围小于 50m |
| 多层规划 | 0.838 | 10km/h | 均匀分布 范围小于 50m |
| 最近点 | 0.677 | 10km/h | 均匀分布 范围小于 100m |
| 最近边 | 0.671 | 10km/h | 均匀分布 范围小于 100m |
| 多层规划 | 0.703 | 10km/h | 均匀分布 范围小于 100m |
| 最近点 | 0.841 | 10km/h | 均匀分布 范围小于 100m |
| 最近边 | 0.834 | 10km/h | 均匀分布 范围小于 100m |
| 多层规划 | 0.793 | 10km/h | 均匀分布 范围小于 100m |

3. 环状道路实验结果

表 3 环状道路实验结果 采样时间间隔 15 分钟

| 算法 | 匹配度 | 其它参数 | |
|------|-------|--------|--------------------|
| | | 行驶速度 | 误差分布 |
| 最近点 | 0.543 | 40km/h | 正态分布 $\sigma=50m$ |
| 最近边 | 0.962 | 40km/h | 正态分布 $\sigma=50m$ |
| 多层规划 | 0.767 | 40km/h | 正态分布 $\sigma=50m$ |
| 最近点 | 0.655 | 40km/h | 正态分布 $\sigma=50m$ |
| 最近边 | 0.724 | 40km/h | 正态分布 $\sigma=50m$ |
| 多层规划 | 0.793 | 40km/h | 正态分布 $\sigma=50m$ |
| 最近点 | 0.712 | 40km/h | 正态分布 $\sigma=200m$ |
| 最近边 | 0.709 | 40km/h | 正态分布 $\sigma=200m$ |
| 多层规划 | 0.855 | 40km/h | 正态分布 $\sigma=200m$ |
| 最近点 | 0.731 | 40km/h | 正态分布 $\sigma=200m$ |
| 最近边 | 0.827 | 40km/h | 正态分布 $\sigma=200m$ |
| 多层规划 | 0.827 | 40km/h | 正态分布 $\sigma=200m$ |
| 最近点 | 0.494 | 40km/h | 均匀分布 范围小于 50m |
| 最近边 | 0.966 | 40km/h | 均匀分布 范围小于 50m |
| 多层规划 | 0.949 | 40km/h | 均匀分布 范围小于 50m |
| 最近点 | 0.103 | 40km/h | 均匀分布 范围小于 200m |
| 最近边 | 0.8 | 40km/h | 均匀分布 范围小于 200m |
| 多层规划 | 0.814 | 40km/h | 均匀分布 范围小于 200m |
| 最近点 | 0.360 | 40km/h | 均匀分布 范围小于 200m |
| 最近边 | 0.720 | 40km/h | 均匀分布 范围小于 200m |
| 多层规划 | 0.760 | 40km/h | 均匀分布 范围小于 200m |

表 4 环状道路实验结果 采样时间间隔 30 分钟

| 算法 | 匹配度 | 其它参数 | |
|------|-------|--------|--------------------|
| | | 行驶速度 | 误差分布 |
| 最近点 | 0.875 | 40km/h | 正态分布 $\sigma=50m$ |
| 最近边 | 0.913 | 40km/h | 正态分布 $\sigma=50m$ |
| 多层规划 | 0.913 | 40km/h | 正态分布 $\sigma=50m$ |
| 最近点 | 0.268 | 40km/h | 正态分布 $\sigma=50m$ |
| 最近边 | 0.804 | 40km/h | 正态分布 $\sigma=50m$ |
| 多层规划 | 0.829 | 40km/h | 正态分布 $\sigma=50m$ |
| 最近点 | 0.250 | 40km/h | 正态分布 $\sigma=200m$ |
| 最近边 | 1.000 | 40km/h | 正态分布 $\sigma=200m$ |
| 多层规划 | 1.000 | 40km/h | 正态分布 $\sigma=200m$ |
| 最近点 | 0.672 | 40km/h | 正态分布 $\sigma=200m$ |
| 最近边 | 0.655 | 40km/h | 正态分布 $\sigma=200m$ |
| 多层规划 | 0.770 | 40km/h | 正态分布 $\sigma=200m$ |
| 最近点 | 0.746 | 40km/h | 均匀分布 范围小于 50m |
| 最近边 | 0.930 | 40km/h | 均匀分布 范围小于 50m |
| 多层规划 | 0.930 | 40km/h | 均匀分布 范围小于 50m |
| 最近点 | 0.037 | 40km/h | 均匀分布 范围小于 200m |
| 最近边 | 0.815 | 40km/h | 均匀分布 范围小于 200m |
| 多层规划 | 0.852 | 40km/h | 均匀分布 范围小于 200m |
| 最近点 | 0.541 | 40km/h | 均匀分布 范围小于 200m |
| 最近边 | 0.820 | 40km/h | 均匀分布 范围小于 200m |
| 多层规划 | 0.852 | 40km/h | 均匀分布 范围小于 200m |

4. 实验结果分析

从测试结果可以看出，最近点算法在环状道路网中表现不佳，偶尔出现匹配度特别低的结果。一种可能的解释是，在环状道路网中，一旦有一定的误差导致基站定位数据最近的路口并不是实际路线中经过的路口，那么由于最短路径不会沿着环路去绕远路，整条路线就会偏离原路线。后两种算法都考虑到了离定位点不止一个的路口，所以能避免这个问题。而在网格状道路网中，就算用错误的路口作为必经点去拟合，后续最短的路线也很容易跟实际路线重合。

定位数据误差范围较小时，最近点算法和最近边算法就能达到不错的效果，其中最近边算法的表现更加稳定。多层规划路径在定位数据误差范围较大时有优于其他两种算法的相对趋势，这也是因为加入了更大范围的路口进行整体规划。

结束语

在本次设计中，我通过构造道路网的模型，设计数据集来进行仿真实验，积累了一些经验。设计过程中，我遇到了大量的选择，模型的种类、参数的设定等等。为了尽量做出符合实际的选择，我查阅了许多文献和相关资料。在设计拟合算法的过程中，我也是从一些文章中汲取灵感，再配合自己的想法来完善。尽管测试的结果显示了我设计的算法并不是尽善尽美，但是如果将它们综合起来使用，还是能得到不错的效果的。

最重要的是，在做毕业设计的过程中，我学到了许多新的知识，巩固了大学期间在专业课程中学到的一些编程知识和算法、数据结构，达到了温故知新的效果，并将之前所学付诸实践，整个过程让我获益匪浅。

参考文献

1. 秦艳珊, 宁彬, 徐凯, 等. 蜂窝网络单基站定位技术的研究与实现[J]. 计算机时代, 2015(7):20-22.
2. 苗得文, 邱满, 许忠奇, 等. 基于 NewMap API 的地图服务系统在交通拥堵识别中的应用研究[J]. 计算机科学, 2013, 40(6a):384-386.
3. 文洁玲, 王钊, 王红燕. 二义性路径识别的技术方案及优缺点对比[J]. 公路交通科技:应用技术版, 2013(5):271-273.
4. 吴天舒, 谢昆青, 宋国杰, 等. 基于多级目标规划的高速公路二义性路径识别[J]. 中国科技论文, 2009, 4(2):115-120.
5. 段凡丁. 关于最短路径的 SPFA 快速算法[J]. 西南交通大学学报, 1994, 29(2):207-212.
6. 夏正冬, 卜天明, 张居阳. SPFA 算法的分析及改进[J]. 计算机科学, 2014, 41(6):180-184.
7. Hunt JW, Szymanski TG. A Fast Algorithm for Computing Longest Common Subsequences. Communications of the ACM. 1977 May 1;20(5):350-3.
8. Oliphant TE. A guide to NumPy. USA: Trelgol Publishing; 2006 Dec 7.
9. Hunter JD. Matplotlib: A 2D graphics environment. Computing In Science & Engineering. 2007 May;9(3):90-5.

致谢

首先，感谢我的导师邓习峰老师在毕业设计期间对我的悉心指导，不断指引我找到正确的设计思路，为我答疑解惑。另外，这次设计中的编程知识大多是我在邓老师的 Linux 程序设计课中所学到的，他在每一堂课上的细心讲解，让我在学习过程中收获良多。

感谢我的三位室友，不但体谅我在不规律的作息时间编码和写论文，忍受键盘敲击、鼠标点击带来的噪音，而且与我共同探讨学术问题，研究算法，为我毕业设计的顺利进行提供了莫大的帮助。也感谢其他大学同学，在与他们相处的四年里，我们情同手足，校园犹如家一般温馨。

感谢北京大学提供给我良好的学习氛围。在这里，学习是一件舒心的事情，而且无论想接触哪些前沿知识，北大都能使我的面前没有任何门槛。我自认学习能力不强，但是在按部就班的课程学习中，我掌握了核心的专业知识，感谢每一位教过我的老师、助教，他们传播给我的知识为我的未来打下了扎实的基础。

感谢父母对我从小到大的关心和照顾，有他们在背后默默的支持，才使我的学业生涯一帆风顺。