**北京航空航天大学计算机学院**

**硕士学位论文开题报告**

**论文题目**：基于轨迹模式的时空数据管理系统的研究与实现

**专 业**：计算机科学与技术

**研究方向**：计算机软件与理论

**研 究 生**：王海轮

**学 号**：SY1406318

**指导教师**：徐洁 教授

**北京航空航天大学计算机学院**

2015年12月10日

目 录

1. 课题来源 1

2. 论文选题的背景与意义 1

3. 国内外研究现状及发展动态 6

3.1时空数据系统 6

3.1.1时空数据库 6

3.1.2时空处理系统 7

3.2轨迹处理技术 7

3.2.1轨迹压缩 7

3.2.2地图匹配 9

3.3轨迹模式挖掘 9

4. 论文的研究内容及拟采取的技术方案 9

4.1论文研究目标 10

4.2论文主要研究内容 10

4.2.1研究道路网络约束下的轨迹模式挖掘方法 10

4.2.2研究基于轨迹模式的新轨迹表示方法 11

4.2.3实现基于轨迹模式的查询方法 11

4.3拟采用的技术方案 11

4.3.1系统结构 11

4.3.2拟采用的技术方案 12

5. 关键技术或技术路线 12

5.1道路网络划分 12

5.2基于轨迹模式表示新轨迹的组合优化问题 13

5.3基于轨迹模式的查询 13

6. 论文研究计划 14

7. 主要参考文献 14

**基于轨迹模式的时空数据管理系统的研究与实现**

# 课题来源

本课题来源于国家自然科学基金委员会“可信软件基础研究”重大研究计划集成项目“面向车联网的可信网络应用软件系统试验环境与示范应用”（项目编号为91118008）。

车联网项目立足国家重大科技与工程任务对可信软件技术的迫切需求，结合车联网软件的规模化、开放性和移动性等关键特性，以网络应用软件系统的客户端、网络通信构件和服务端可信软件技术研究为切入点，突破和掌握可信网络应用软件和可信环境的需求分析、构造、验证、监控和演化等理论、方法和技术，通过集成创新形成一体化的可信软件技术体系，研发相应的支撑工具与平台，建立面向车联网的可信网络应用软件系统试验环境，并通过开发车辆远程诊断、安防服务、远程控制、车队服务以及车辆监控等可信车载服务典型应用案例示范的形式进行验证。

本课题作为车联网存储系统的一部分，通过研究大规模轨迹数据的特征，挖掘轨迹模式，并使用轨迹模式进行轨迹存储和查询。最终，减少轨迹数据存储的冗余，加快轨迹查询应用的速度。

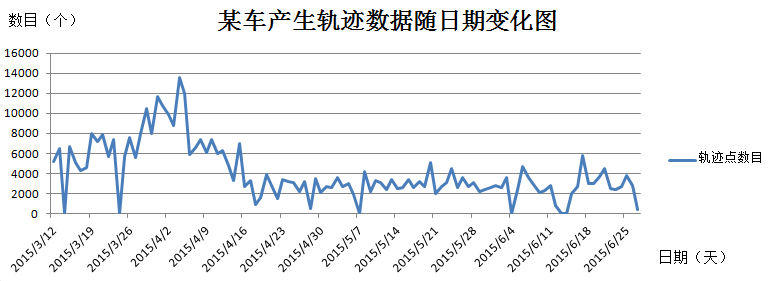
# 论文选题的背景与意义

随着GPS传感器和移动设备的广泛使用，以位置数据为核心的时空数据大量产生。车联网[1]作为物联网的重要应用，近几年取得了突飞猛进的发展。根据易观智库《中国车联网产业链研究专题报告2015》[2]，截止2014年，我国机动车保有量达2.64亿辆，其中汽车1.54亿辆。2015年，车联网渗透率将从2010年的4.67%增长至10%，市场规模有望突破1,500亿元，中国车联网市场正在迎来发展的大时代[3]。互联网厂商纷纷切入车联网应用与服务，2015年，百度推出全新车联网解决方案百度CarLife[4]，阿里巴巴集团成立阿里汽车事业部[5]，腾讯推出腾讯车联开放平台（Tencent Automotive Services）[6]，国内外厂商争相发力车联网。随着GPS、OBD[7]、智能手机等设备的普及，轨迹数据越来越多，查询也越来越复杂，车联网应用需要实时存储、处理大规模的轨迹数据，**如何有效存储和查询轨迹数据成为车联网的一大挑战**。

对于轨迹数据，在存储方面，车联网有①轨迹数据量大、②轨迹重复的特点；在查询方面，车联网需要满足③不同查询类型和精度。

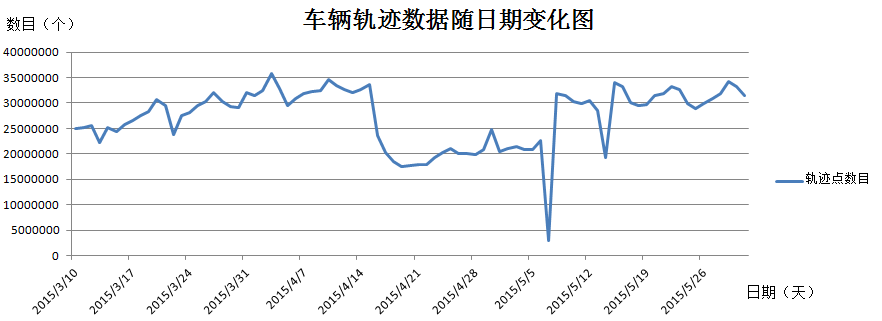
首先，车联网系统的轨迹数据量大主要表现在汽车采样频率高、运营车辆数目多、历史数据积累多等三个方面。

车联网系统中，车辆工作时间平均每5秒钟上传一次GPS位置，每辆车每天产生许多轨迹数据。以四元组point tuple表示轨迹点为例，，id为汽车标识，longitude为经度，latitude为纬度，time为时间戳。图一是某辆车每天产生的轨迹数据岁日期变化图，我们明显可以看到单辆车平均每天产生约3,000个轨迹点，系统要存储大量原始point tuple。



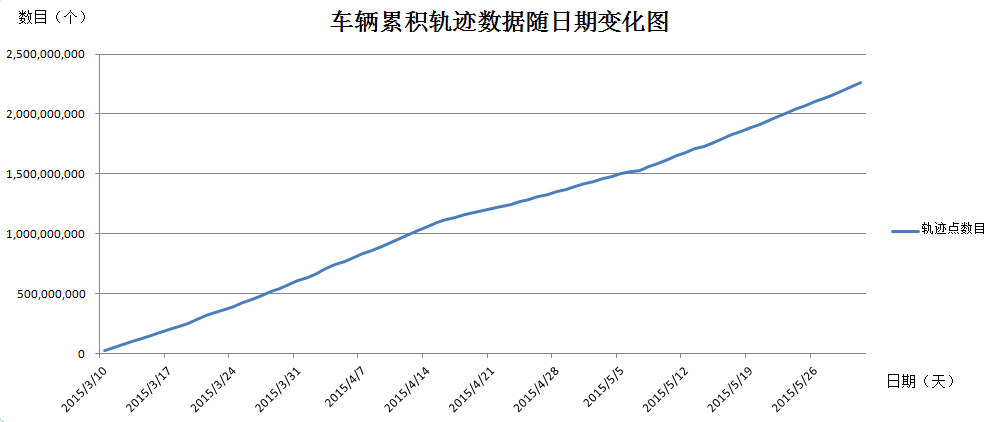
图一 某车产生轨迹数据随日期变化图

车联网系统目前约有25,000辆汽车，这些车辆每天产生大量轨迹数据。图二是全部车辆产生的轨迹数据岁日期变化图，系统平均每天产生约27,000,000个轨迹点。



图二 全部车辆轨迹数据随日期变化图

车联网系统运营一年多，积累了大量的历史数据。图三是全部车辆累积轨迹数据随日期变化图，系统仅三月份便产生了约10亿个轨迹点，随着时间的增加，系统存储的轨迹数据量基本随时间线性增加。



图三 全部车辆累积轨迹数据随日期变化图

随着系统单车上传频率的提高、车辆数目的增多、运营时间的增加，车联网系统需要存储更多的轨迹数据。文献[8]使用北京市三个月的出租车轨迹数据对驾驶方向进行了研究，出租车共有33,000辆，每辆车平均每3.1分钟上传一个轨迹点，一共790,000,000个轨迹点。文献[9]使用新加坡一个月的出租车轨迹数据对轨迹压缩进行了研究，出租车共有15,000辆，465,000条轨迹。文献[10]使用飓风数据集Best Track[11]对轨迹聚类进行了研究，数据集包括1950年到2004年的Atlantic飓风的记录，共有570条轨迹，17,736个轨迹点。车联网系统拥有25,000辆汽车，一个月产生近1,000,000,000个轨迹点，与上述数据集相比，**车联网系统明显具有轨迹数据量大的特点**。

**其次，车联网系统中大量轨迹存在重复，存储冗余**。以北京市为例，每天约有3,600辆汽车在城市道路运营，这些车辆之间每天的运行轨迹存在大量重复，而且，大量车辆每天新产生的轨迹和历史轨迹也存在着重复，这些重复数据给存储带来了大量冗余。表一是某辆车在北京道路网络连续几天行驶轨迹，该车每天的行驶轨迹在机场、环路等路段存在明显重复。

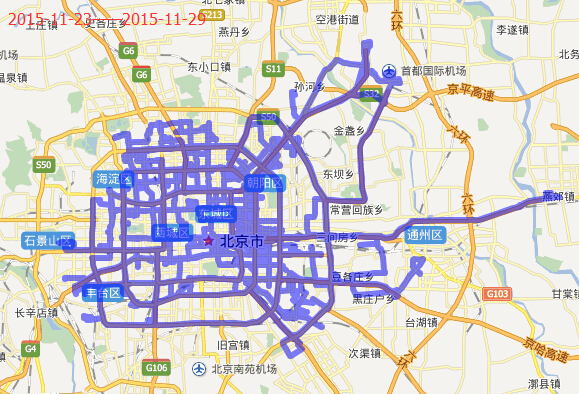
表一 某辆车连续几天行驶轨迹表

|  |  |  |
| --- | --- | --- |
| 某辆车2015-11-23至2015-11-28每天行驶轨迹 | | |
| 2015-11-23-small-copy |  | 2015-11-25-small |
| 2015-11-26-small | 2015-11-27-small | 2015-11-28-small |

表二是某两辆车每天行驶轨迹对比，两辆车行驶轨迹在机场、环路、主干路等地存在重复。图四是两辆车一周的行驶轨迹，可以发现，两辆车重复的轨迹更多。因此，随着运营时间的增加，大量车辆的行驶轨迹在道路网络中会存在更多的重复，存储更加冗余。

表二 两辆车每天行驶轨迹对比表

|  |  |  |
| --- | --- | --- |
| 第一辆车每天行驶轨迹 | | |
| 2015-11-27-small | 2015-11-28-small | 2015-11-29-small |
| 第二辆车每天行驶轨迹 | | |
|  | 2015-11-28-small-another | 2015-11-29-small-another |

图四 两辆车一周行驶轨迹对比图

**最后，车联网轨迹数据的查询主要集中在道路网络上，对原始轨迹点的查询很难满足不同查询类型、精度的要求**。汽车实时监控需要实时查看汽车的具体位置，查找最近的车辆需要返回距离乘客道路距离最近的汽车，分析某辆车的运行轨迹需要统计该车的运行路线，这些都需要对轨迹的高效存储和查询。目前，车联网轨迹查询主要分为单条轨迹查询、范围查询、最近邻查询、联合查询等类型。系统以point tuple的形式存储轨迹数据，查询也相应转化为原始轨迹点的大量计算；而且，由于GPS设备的限制、采样的测量误差[12]、实际道路网络环境等，对原始轨迹点的查询得到的结果也不一定准确；同时，不同的查询需要满足不同的精度，如查询某辆车是否经过某一区域、某一时刻某辆车的具体位置等，查询都需要对原始轨迹点进行大量计算。随着数据量的增大，对原始轨迹点的计算、查询越来越难以满足车联网的查询需求。

基于以上对车联网轨迹数据的分析，**车联网轨迹数据应用面临着存储和查询的矛盾**。一方面，大量、重复的轨迹数据，需要越来越大的存储空间，而越来越大的存储空间又会增加查询的计算，降低查询速度。另一方面，不同查询类型需要在道路网络的基础上满足不同精度，又影响了轨迹数据的存储方式。因此，如何在满足不同查询精度的要求下，利用道路网络信息，减少轨迹数据的存储，加快查询速度是车联网时空数据系统的重要研究内容。

本文在道路网络的基础上，利用大量车辆的历史轨迹来挖掘轨迹模式，并使用轨迹模式来表示原始轨迹，最后，基于轨迹模式进行不同精度的轨迹查询。

# 国内外研究现状及发展动态

时空数据，特别是轨迹数据，在国内外得到了大量的研究。本文主要分析时空数据库和处理系统、轨迹压缩、轨迹模式挖掘（聚类）等研究进展和发展动态。

## 3.1时空数据系统

实时、高效、弹性的轨迹数据存储和处理是车联网应用的基础，以下简要介绍时空数据库和时空处理系统的发展。

### 3.1.1时空数据库

传统关系型数据库广泛应用于现代信息系统中，对于多维的轨迹数据，关系型数据库通过扩展地理数据库来实现地理数据的操作。一般地理数据库[13]，如PostGIS[14]，通过增加地理类型（如点、线、多边形等）、索引（R-tree[15]等）、查询函数（几何对象交叉、接触、距离等）等来实现地理查询，其研究重点主要集中在地理数据的处理、优化上。文献[17]提出了针对特定时空区域检索的动态存储系统，通过将轨迹分段，把相近的轨迹段、时间段放在临近的磁道上，并使用自适应多层Quad-tree[18]结构来索引磁盘页，查询时通过减少读取磁盘次数来减少磁盘IO延时，加快查询速度，系统特别适用于经过某一特定区域的大规模轨迹查询。以上时空数据系统难以满足车联网等大数据系统对高数据插入速率、实时查询、TB级大容量存储、可扩展等要求。

近年来，分布式数据存储系统迅速发展，例如HBase[19]，并广泛应用于大数据存储系统。这些系统在可靠性、扩展性、性能方面有着明显的优势，一般可以满足大规模数据的存储、查询需求。但是，这些系统在地理数据的查询方面明显存在低效、功能缺失等问题。文献[20]、[21]基于HBase设计了适用于时空数据的存储系统。其原理是在HBase上增加多维索引层来实现地理查询功能、规模化操作。一方面，通过多维索引（K-d tree[22]、Quad tree）实现高效的地理查询，另一方面，使用HBase原始的存储系统来满足数据的高插入速率、大容量、高可用性的需求。

### 3.1.2时空处理系统

近几年，基于Hadoop[23]提出了一些时空数据处理系统。文献[24]通过扩展Hive[39]，在Hadoop之上创建了一个满足范围、联合查询的全局网格索引的数据仓库Hadoop-GIS。文献[25]SpatialHadoop是在Hadoop内核中引入标准地理索引和MapReduce来实现几何计算和地理数据挖掘操作。作者进一步提出通过增加时间维度处理扩展Spatio-temporal Hadoop。

## 3.2轨迹处理技术

轨迹处理技术是轨迹计算的基础，本文主要分析研究广泛的轨迹压缩和地图匹配技术。

### 3.2.1轨迹压缩

经典的轨迹压缩算法是Douglas-Peucker[16]算法，其思想是通过近似的线段来代替原始轨迹，进而减少轨迹点的数目来达到压缩目的。开始时，算法连接起点S、终点E，从剩余点中找出离线段SE垂直欧拉距离最远的超过阈值的点F，然后，算法再对线段PF、FE按照上述方法递归分解，最后得到的所有线段起始点即为压缩后的轨迹点，原始的DP算法的时间复杂度是，N是一条轨迹的轨迹点数目。其改进算法[26]将时间复杂度进一步提升为，[27]使用了时间复杂度为的动态规划算法来取得最优的近似轨迹。[28]使用了增加时间维度的time synchronized Euclidean distance代替垂直欧拉距离，进一步保证精确性。同时，为了满足在线处理的需要，滑动窗口[29]和开放窗口[28]压缩算法也被提出，滑动窗口通过在error bound内动态扩展压缩的线段，开放窗口算法使用启发式的DP算法选择分解点。另外，一些在线压缩算法也考虑了行驶方向、速度等信息，例如[30]使用最后两个点和阈值构建的安全区，来决定是否保留新产生的轨迹点。[31]为解决压缩后的轨迹与原先轨迹不相似，提出了保留方向的轨迹简化方法。

文献[32]提出基于道路网络提出了非具体化轨迹模型，将空间轨迹信息通过道路网络表示，时间信息通过道路坐标或距离偏差和时间戳来表示。文献[33]提出了基于地图匹配的轨迹压缩方法。文章首次提出在轨迹压缩的同时考虑轨迹匹配的问题（MMTC），并提出了TC+MM、MM+TC+MM、MMTC-App的解决方法。第一种方法是先对轨迹进行压缩，然后将压缩的轨迹匹配到道路上。第二种方法是先将原始轨迹匹配到道路上，然后进行轨迹压缩，最后再把压缩的轨迹匹配到道路上。针对前两种方法对轨迹压缩和地图匹配的分割处理，第三种方法将轨迹压缩和与原始轨迹相似性同时考虑，并作为一个最优化问题，提出了近似解的算法。经过比较，第三种方法在压缩率、相似性等方面优于前两种方法。文献[9]提出了在道路网络中的轨迹压缩方法。文章对空间维度和时间维度分别表示，对空间信息使用了混合压缩的方法，对时间信息使用了限界的时间压缩方法，并通过解压实现了常用的地理查询。对于GPS轨迹，首先进行地图匹配，将匹配过的轨迹用道路段集合和时间集合分开表示，然后，对道路段集合先采用最短路径压缩算法，之后，通过挖掘出的常用子轨迹并使用Huffman编码进行压缩；时间集合通过类似于Before Opening Window（BOPW）[28]算法进行限界近似压缩。最后，根据查询类型，分别从轨迹压缩和时间压缩里面解压得到原始轨迹信息。由于对轨迹采用了两种压缩方法，系统的压缩率比较好。

### 3.2.2地图匹配

地图匹配主要分为基于几何匹配、基于拓扑匹配、概率算法、高级匹配算法等类型[34]。几何匹配主要考虑GPS点和道路网络的几何信息，使用搜索算法，在点对点匹配中，计算出GPS点最近的节点或道路段，在点到曲线的匹配中，找出最近的由分段线段组成的曲线，在曲线到曲线的匹配中，使用已知道路与轨迹比较。基于拓扑的匹配充分利用道路网络的拓扑属性（道路转向、弯曲、连接等）来进行轨迹匹配。概率匹配算法考虑轨迹位置的置信椭圆或矩形置信区域。高级匹配算法综合使用多方面信息来进行匹配，主要有卡尔曼滤波、模糊逻辑模型、隐式马尔科夫连模型等。

## 3.3轨迹模式挖掘

轨迹模式是频繁的轨迹行为，通过对大量的历史轨迹分析，我们可以发现用户在近似的时间间隔内行驶类似的路径。文献[35]形式化定义了轨迹模式问题，并提出了基于兴趣区域的挖掘方法。文献[10]提出了先分解后组合的轨迹聚类方法。与对整体相似轨迹方法不同，先对轨迹分段，然后对分段的轨迹使用基于线段密度的方法聚类，主要适用于自由运动的轨迹。文献[36]针对道路网络环境提出了轨迹聚类方法。轨迹在经过地图匹配后，可以使用道路段的集合来表示轨迹（类似字符串），并提出了与传统聚类方法不同的相似性判断标准（基于共同道路路段总长度）。文献[37]提出道路网络感知的轨迹聚类方法。传统的聚类方法主要基于密度或欧拉距离，作者提出在聚类时应考虑道路网络约束、优先级、流量。

# 论文的研究内容及拟采取的技术方案

本文用到的主要内容定义如下。

轨迹点TP id表示汽车标识，longitude表示经度，latitude表示纬度，time为时间戳。

GPS点GP longitude表示经度，latitude表示纬度。

道路R id表示道路标识，表示轨迹点集合，表示道路属性集合，如长度等。

轨迹模式P id表示轨迹模式标识，表示道路集合，表示轨迹模式属性集合，如距离等。

时间信息T d表示离该轨迹模式起点的距离，t表示时间戳。

## 4.1论文研究目标

本文设计和实现基于轨迹模式的时空数据管理系统，即在道路网络的基础上，利用大量车辆的历史轨迹来挖掘轨迹模式，并用轨迹模式代替原始轨迹点表示新轨迹，同时，使用轨迹模式进行不同类型的轨迹查询。通过地图匹配的方法将原始轨迹点匹配到道路上，研究基于道路网络的聚类方法，设计轨迹模式的存储、索引方法，并实现基于轨迹模式的新轨迹表示和查询。最终，通过挖掘和使用轨迹模式，减少轨迹数据的存储冗余，加快轨迹查询速度。

## 4.2论文主要研究内容

本文研究内容主要分为：研究基于道路网络的轨迹模式挖掘方法、研究基于轨迹模式的新轨迹表示方法、实现基于轨迹模式的查询方法等三部分。

### 4.2.1研究道路网络约束下的轨迹模式挖掘方法

通过地图匹配方法将轨迹匹配到道路网络中，研究基于道路网络的聚类方法来挖掘轨迹模式。城市道路网络是一张复杂的网络，根据道路的几何、拓扑等属性，研究道路网络的划分方法，使用地图匹配方法来完成原始轨迹点到道路段的转化。研究如何从大量历史轨迹构成的道路网（不同权值的道路段）中挖掘轨迹模式，并依据道路覆盖率、挖掘速度等对方法进行评价。

### 4.2.2研究基于轨迹模式的新轨迹表示方法

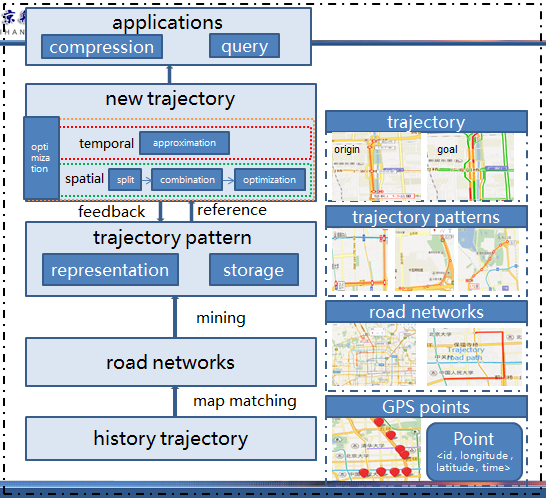
结合轨迹模式的关系，设计轨迹模式的存储系统，研究基于轨迹模式的新轨迹表示方法。基于轨迹模式的邻接、包含、交叉等关系，设计轨迹模式的多维索引结构。对于地图匹配后的原始轨迹，研究使用轨迹模式来表示新轨迹的组合优化方法。

### 4.2.3实现基于轨迹模式的查询方法

针对单条轨迹查询、范围查询、最近邻查询等常见查询类型，实现基于轨迹模式的查询方法。

## 4.3拟采用的技术方案

### 4.3.1系统结构



图五 系统框架图

系统主要由轨迹模式挖掘部分、基于轨迹模式的新轨迹生成部分、基于轨迹模式的查询部分等组成。如图一系统框架图所示，系统选用北京市运营一年多的历史数据集，采用基于道路网络的聚类方法挖掘轨迹模式，使用多维索引结构进行轨迹模式的存储。对于原始轨迹点，通过轨迹切分，使用轨迹模式来生成新轨迹，其中，原始轨迹集合set(TP)分别表示为轨迹集合set(P)、时间集合set(T)。最后，应用层面通过轨迹模式实现对原始轨迹的压缩和查询。

### 4.3.2拟采用的技术方案

* 轨迹模式挖掘

轨迹模式的挖掘本质上图聚类问题，本文拟采用在考虑道路网络约束的基础上，基于交通流密度的聚类方法进行轨迹模式挖掘。首先，结合道路优先级、几何特征等方法对北京市道路网络进行切分，然后，使用几何匹配的地图匹配算法进行轨迹匹配，最后，对轨迹模式集合联合使用Quad-tree、K-d tree、Grid等多维索引结构存储。

* 基于轨迹模式表示新轨迹的组合优化

使用轨迹模式集合表示新轨迹，本质上是类似于0-1背包的NPC[38]问题。本文拟采用近似求解算法来获取新轨迹表示的近似最优结果，根据最优解的代价函数，使用启发式的近似算法来加快新轨迹表示的处理速度，满足车联网轨迹获取、处理的实时性要求。

# 关键技术或技术路线

## 5.1道路网络划分

城市道路网络林立交错，道路数目多，层级分明，结构比较复杂。道路网络划分是地图匹配、轨迹模式挖掘的基础。城市道路具体存在以下情况：①城市主干路、环路、机场等热门线路明显存在大量行驶轨迹，而部分支线、小街道存在车流小的情况，即道路流量不均衡；②有些道路存在车道多、交叉路口多、环形的复杂路况，而有些道路结构简单；③部分路段存在着简单固定的驾驶线路，如T型路口等，而一些道路驾驶情况复杂。针对以上道路现状，如何划分道路段的长度、粒度等，影响着轨迹模式挖掘的好坏。因此，本文需要设计合理的道路网络划分方法，一方面尽量符合道路的几何、车流信息，另一方面，又要权衡道路长度、粒度对计算复杂性、轨迹模式的数量、实际匹配效果的影响。

## 5.2基于轨迹模式表示新轨迹的组合优化问题

使用轨迹模式集合表示新轨迹，本质上是类似于0-1背包的NPC问题。如何从轨迹模式集合中快速选出匹配原始轨迹的最优方案，影响着轨迹表示、查询效果。假设轨迹模式的集合PS(Pattern Set)，待匹配的轨迹TS(Trajectory Set)，目标轨迹集合SPS(Sub-Pattern Set)，则问题定义为：求集合①，使得其满足②、③、④。

①

②

③

④

在快速找出最大匹配度的轨迹模式集合时，还需要考虑未匹配到的轨迹表示方法。由于车联网系统的实时性要求，需要设计启发式的近似算法。

## 5.3基于轨迹模式的查询

基于轨迹模式的查询需要将查询条件匹配到轨迹模式、道路上，对于点查询来说，如果点在道路上，查找包含该点的轨迹模式的轨迹，如果点不在道路上，需要查找离点近的轨迹模式；对于范围查询，需要匹配经过或包含在范围内的所有轨迹模式，特别是以某点为中心的范围查询，需要返回距离查询点道路距离近的轨迹模式；距离某点的最近邻查询同上。本文先求出到目标点距离范围内的道路，然后根据道路求出对应的轨迹模式和对应距离，查找符合条件的时间段，最后返回目标车辆或轨迹。对于时间查询，系统根据时间查找对应的轨迹模式、道路、距离等。

Trajectory为原始轨迹

各种查询转化为轨迹模式、道路，并在集合上进行。

# 论文研究计划

详细进度安排请见表三。

表三 进度安排表

|  |  |
| --- | --- |
| 时间 | 任务安排 |
| 2015年10月—2015年12月 | 系统调研、查阅资料。 |
| 2015年12月—2016年3月 | 系统概要设计，解决主要技术难点。 |
| 2016年3月—2016年6月 | 系统详细设计、实现。 |
| 2016年6月—2016年9月 | 撰写小论文。 |
| 2016年9月—2016年10月 | 系统测试、完善。 |
| 2016年10月—2016年12月 | 撰写毕业论文，毕业答辩。 |

# 主要参考文献

1. White Paper of Internet of Vehicles(IoV). <http://mddb.apec.org/Documents/2014/TEL/TEL50-PLEN/14_tel50_plen_020.pdf>
2. 2015年中国车联网发展现状与产业链分析. <http://www.analysys.cn/cxfx/9793.shtml>
3. 2015：车联网的大时代. <http://paper.people.com.cn/gjjrb/html/2015-01/19/content_1521834.htm>
4. 百度车联网. <http://che.baidu.com/>
5. 阿里宣布成立汽车事业部 BAT争相发力车联网. <http://finance.cnr.cn/txcj/20150409/t20150409_518270907.shtml>
6. 腾讯提速车联网推出开放平台. <http://money.163.com/15/0915/01/B3H2LBVA00253B0H.html>
7. wikipedia: On-board diagnostics(OBD). <https://en.wikipedia.org/wiki/On-board_diagnostics>
8. Yuan J, Zheng Y, Zhang C, et al. T-drive: driving directions based on taxi trajectories[C]//Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems. ACM, 2010: 99-108.
9. Song R, Sun W, Zheng B, et al. PRESS: A novel framework of trajectory compression in road networks[J]. Proceedings of the VLDB Endowment, 2014, 7(9): 661-672.
10. Lee J G, Han J, Whang K Y. Trajectory clustering: a partition-and-group framework[C]//Proceedings of the 2007 ACM SIGMOD international conference on Management of data. ACM, 2007: 593-604.
11. Atlantic hurricane data sets. <http://weather.unisys.com/hurricane/atlantic/>
12. Pfoser, D. and Jensen, C. S. Capturing the uncertainty of moving-object representations. In Proceedings of the 6th international Symposium on Advances in Spatial Databases, 111-132, 1999.
13. Steiniger S, Bocher E. An overview on current free and open source desktop GIS developments[J]. International Journal of Geographical Information Science, 2009, 23(10): 1345-1370.
14. PostGIS. <http://postgis.net/>
15. Guttman A. R-trees: a dynamic index structure for spatial searching[M]. ACM, 1984.
16. Douglas D H, Peucker T K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature[J]. Cartographica: The International Journal for Geographic Information and Geovisualization, 1973, 10(2): 112-122.
17. Cudre-Mauroux P, Wu E, Madden S. Trajstore: An adaptive storage system for very large trajectory data sets[C]//Data Engineering (ICDE), 2010 IEEE 26th International Conference on. IEEE, 2010: 109-120.
18. Finkel R A, Bentley J L. Quad trees a data structure for retrieval on composite keys[J]. Acta informatica, 1974, 4(1): 1-9.
19. George L. HBase: the definitive guide[M]. " O'Reilly Media, Inc.", 2011.
20. Nishimura S, Das S, Agrawal D, et al. MD-HBase: a scalable multi-dimensional data infrastructure for location aware services[C]//Mobile Data Management (MDM), 2011 12th IEEE International Conference on. IEEE, 2011, 1: 7-16.
21. Li S, Hu S, Ganti R, et al. Pyro: a spatial-temporal big-data storage system[C]//2015 USENIX Annual Technical Conference (USENIX ATC 15). 2015: 97-109.
22. Bentley J L. Multidimensional binary search trees used for associative searching[J]. Communications of the ACM, 1975, 18(9): 509-517.
23. White T. Hadoop: The definitive guide[M]. " O'Reilly Media, Inc.", 2012.
24. Aji A, Wang F, Vo H, et al. Hadoop GIS: a high performance spatial data warehousing system over mapreduce[J]. Proceedings of the VLDB Endowment, 2013, 6(11): 1009-1020.
25. Eldawy A. Spatialhadoop: towards flexible and scalable spatial processing using mapreduce[C]//Proceedings of the 2014 SIGMOD PhD symposium. ACM, 2014: 46-50.
26. Hershberger J E, Snoeyink J. Speeding up the Douglas-Peucker line-simplification algorithm[M]. University of British Columbia, Department of Computer Science, 1992.
27. Bellman R. On the approximation of curves by line segments using dynamic programming[J]. Communications of the ACM, 1961, 4(6): 284.
28. Meratnia N, Rolf A. Spatiotemporal compression techniques for moving point objects[M]//Advances in Database Technology-EDBT 2004. Springer Berlin Heidelberg, 2004: 765-782.
29. Keogh E, Chu S, Hart D, et al. An online algorithm for segmenting time series[C]//Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on. IEEE, 2001: 289-296.
30. Potamias M, Patroumpas K, Sellis T. Sampling trajectory streams with spatiotemporal criteria[C]//Scientific and Statistical Database Management, 2006. 18th International Conference on. IEEE, 2006: 275-284.
31. Long C, Wong R C W, Jagadish H V. Direction-preserving trajectory simplification[J]. Proceedings of the VLDB Endowment, 2013, 6(10): 949-960.
32. Cao H, Wolfson O. Nonmaterialized motion information in transport networks[M]//Database Theory-ICDT 2005. Springer Berlin Heidelberg, 2005: 173-188.
33. Kellaris G, Pelekis N, Theodoridis Y. Map-matched trajectory compression[J]. Journal of Systems and Software, 2013, 86(6): 1566-1579.
34. 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 Part C: Emerging Technologies, 2007, 15(5): 312-328.
35. Giannotti F, Nanni M, Pinelli F, et al. Trajectory pattern mining[C]//Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2007: 330-339.
36. Won J I, Kim S W, Baek J H, et al. Trajectory clustering in road network environment[C]//Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on. IEEE, 2009: 299-305.
37. Han B, Liu L, Omiecinski E. Neat: Road network aware trajectory clustering[C]//Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on. IEEE, 2012: 142-151.
38. Cormen T H. Introduction to algorithms[M]. MIT press, 2009.
39. Thusoo A, Sarma J S, Jain N, et al. Hive: a warehousing solution over a map-reduce framework[J]. Proceedings of the VLDB Endowment, 2009, 2(2): 1626-1629.