

阅读笔记：基于车辆轨迹的动态路况挖掘（张禹）

相关工作

道路匹配算法

- 几何算法（纯几何计算，lwq道路修正工作中主要使用的算法）
- 网络拓扑算法
 - 结合网络拓扑
 - 差错域、差错参数
 - 参考信息、参数调节空间大
- 概率统计算法
- 高级算法

路况挖掘

- 道路稀疏区域（高速）：固定探测器
- 城市区域
 - 高采样率：<10s
在浮动车终端进行道路匹配，减轻服务器压力，有一定延迟
 - 低采样率：>30s

系统框架

整体

- 低采样率信息+服务器端路况挖掘系统
- 高采样率信息+智能合约地图信息：终端计算，合约汇总
- 两部分路况汇总得到最终存储的结果

服务器端路况挖掘

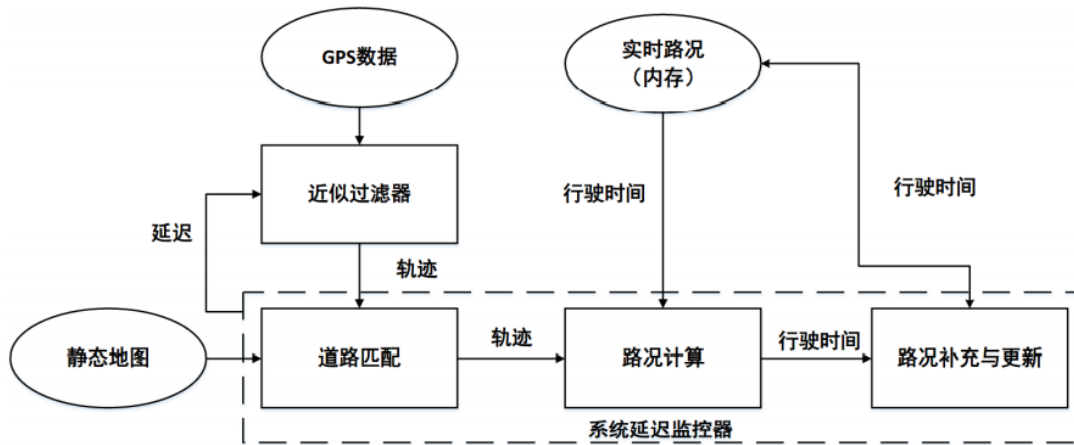


图 3.2 服务器端路况挖掘系统框架图

- 轨迹切割：GPS数据预处理，按时间排序、分片
- 轨迹过滤：近似算法，删除冗余轨迹
- 道路匹配：在线匹配算法进行修正，最短路径算法
- 路况计算：道路和虚拟路口的权重
- 更新与补充：平滑处理

终端路况挖掘

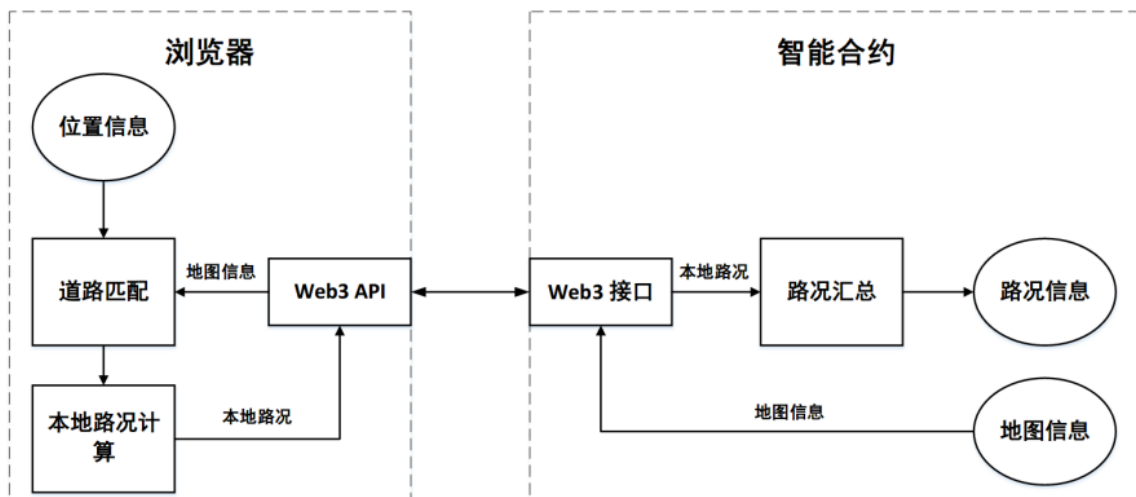


图 3.3 终端路况挖掘系统框架图

- 用手机app作为终端
- 智能合约保存同一道路、不同车辆写入状况
- 智能合约只保存最新路况，服务器端有历史记录

算法

服务器端

近似算法

- 丢弃过时轨迹
- 丢弃收敛、冗余轨迹
- 以区域为单位做近似

在线道路匹配算法

- 使用信息
 - 时间
 - 经纬度
 - 道路网络拓扑
 - 道路级别
 - 单双向行驶信息（双向道路被拆分成两个单向道路存储）
- OHMMM + BVSUW
Online Hidden Markov Map Matching + Bounded Variable Sliding Window
- 计算候选道路之间的最短路由、计算两个候选道路之间的转移概率

虚拟路口

只考虑两条道路之间的转向延迟

路况计算

加权分配，平滑处理

补充缺失路况

横向与当前时间段路况对比，纵向与历史路况对比

手机端

Geohash编码

6位编码，每个块覆盖大小1220m*1220m

道路匹配

- 9个Geohash块中的道路，计算距离得到候选者
- 计算匹配概率

路况计算

- 平均速度，取道路中间段行驶的部分
- 虚拟路口与轨迹切割：车辆减速、距离道路终点小于阈值时进行路段与路口切割

智能合约

- 道路平滑：历史路况占90%比重

想法与问题

服务器端路况计算：路由在道路上的覆盖率？

终端路况挖掘不是论文重点，属于对服务器端系统的一个补充。代码上可能主要只实现了流程，可参考的内容应该不是很多，有时间也可以简单看一下，便于思考自己的入手方向或者和其他工作的结合。一些在这篇文章基础上需要发展改进的方向：

- 在切换道路的时候匹配准确度下降。可以沿用服务器端的虚拟路口模型，然后做进一步的处理：
 - 网页端不能保存完整的道路拓扑结构，根据历史/当前位置对附近的道路、虚拟路口进行预加载（维护小规模数据）
 - 道路匹配：由于采样率高，可以尝试尽可能直接地通过实时数据确定车辆在虚拟路口的走向
 - 路况计算：历史路况在平滑时占的比重非常高，考虑到历史数据是最好缩减的无用部分，改进平滑方法，或者增加其他排除异常数据的方法（结合信誉？）
- 挖空矿现象（暂时不了解）
- 无用数据：轻节点+全节点（考虑引入RSU单元）

V Buterin. A next-generation smart contract and decentralized application platform. [EB/OL]. (2014)http://www.the-blockchain.com/docs/Ethereum_white_paper_a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf

- 区块链分支：可参考*Using Blockchain for Peer-to-Peer Proof-of-Location*
- 网络环境较差的情况（这个不一定能解决）