

# 道路中心线时序跟踪

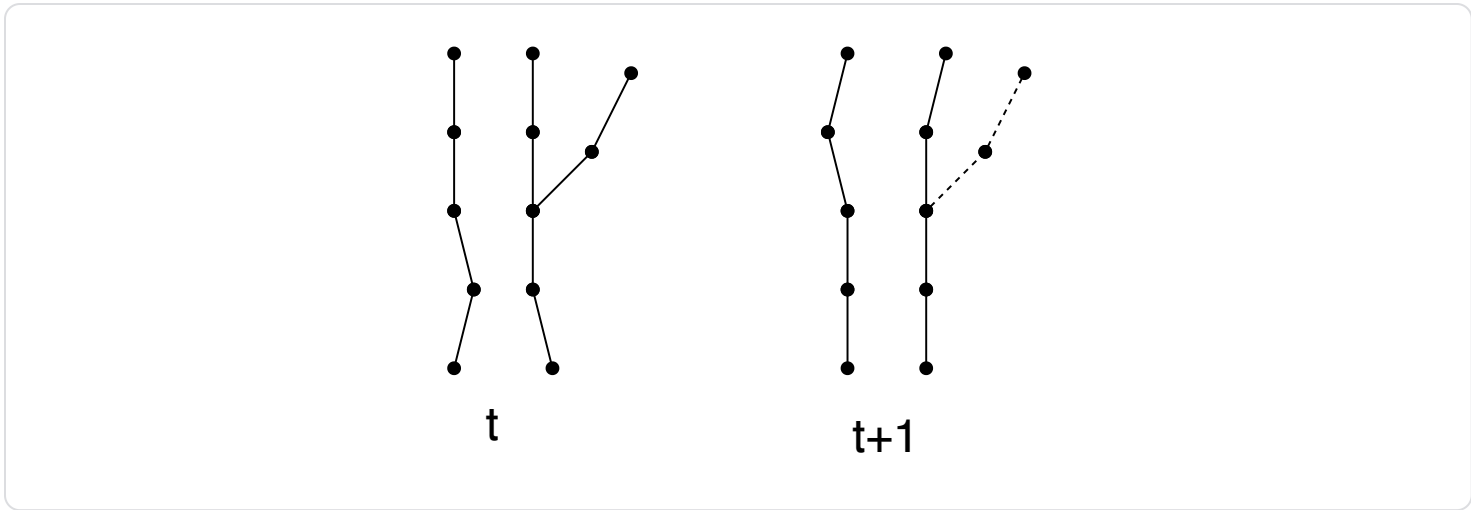
## 目录

- 背景
- 道路中心线时序保持方案
- 基本流程
- 与LETS方案对比
- 后续

## 背景

在主辅路等复杂场景，道路中心线有可能搜索不到辅路，一方面通过正向研发增加搜索成功率，另一方面通过时序跟踪来提高道路中心线的稳定性。

如下图，t时刻有两个搜索起点，右侧起点又有两条搜索路径。而在t+1时刻，虚线部分没有搜索到，希望通过时序跟踪将虚线部分找回。



## 道路中心线时序保持方案

在一帧数据中可能有多个搜索起点，每个搜索起点又可能有多条道路中心线。

因此，道路中心线时序跟踪分为两大部分：

- 搜索起点跟踪：（原则：相信起点生成结果，跟踪起点能匹配上就使用，否则不用）
  - 匹配上的起点，后续使用跟踪起点中的路径再次进行路径匹配
  - 未匹配上的搜索起点，直接使用其搜索路径
    - 起点数量跳变时，只对最优匹配结果使用跟踪起点及其中的路径

- 未匹配上的跟踪起点，不使用
  - 起点位置跳变时，如果无法匹配上，则不使用跟踪起点及其中的路径
- 搜索路径跟踪，在匹配上的搜索起点中再进行搜索路径跟踪
  - 对于匹配上的路径，直接使用当前搜索路径
  - 对于未匹配上的跟踪路径，根据其跟踪时长决定是否使用
  - 对于未匹配上的搜索路径，直接使用

## 基本流程

1. 坐标系转换，跟踪结果以local坐标系保存，使用时转换到flu坐标系
2. 起点匹配
  - a. 匹配标准：距离+道路边界
  - b. 匹配方法：贪心
  - c. 融合逻辑：如上方案所述
3. 路径匹配
  - a. 匹配标准，先使用路径边界id判断，再使用LIP距离，参考[轨迹相似度](#)
  - b. 匹配方法，贪心
  - c. 融合逻辑，如上方案所述
4. 路径合并，将跟踪路径与起点中的其他路径进行局部合并

方案优化点：

- 1.单帧尽量合并，不重复
- 2.匹配时考虑结构一致性
- 3.时序保持细化

## 与LETS方案对比

|   |                |  |                               |
|---|----------------|--|-------------------------------|
| 1 |                | 车道中心线（LETS）  | 道路中心线                         |
| 2 | 历史轨迹在lets搜索中作用 | 历史轨迹只用来算reward，与历史轨迹的接近程度是reward计算中的一个影响因素             | 当前未使用，依赖起点匹配和路径匹配，等时序跟踪完成后再考虑 |
| 3 | 起点匹配           | 维护三个实例<br>EGO/LEFT/RIGHT_LANE，主车为EGO，左右起点为主车相邻车道主车平行位置 | 起点根据车道线、路沿生成，需要进行起点匹配         |
|   | 路径匹配           | 可能通过终点进行匹配   | 1. 路径两侧边界id<br>2. LIP距离       |

|                 |           |  |  |
|-----------------|-----------|--|--|
| 2025/1/13 10:36 | 道路中心线时序跟踪 |  |  |
| 4               |           |  | 先使用1，无法判断时使用2  |
|                 | 路径匹配成功处理  | 使用最新搜索结果   | 使用最新搜索结果   |
| 5               | 未匹配的搜索路径  | 使用最新搜索结果   | 使用最新搜索结果   |
|                 | 未匹配的历史路径  | 维护三个实例<br>EGO/LEFT/RIGHT_LANE，然后每个实例记录自己的探索起点，和历史结果，如果没探索出来就fallback几帧 | 维护起点跟踪目标，每个起点目标中包含由该起点探索的道路中心线，进行起点匹配后，如果发现道路中心线出现丢失，使用跟踪目标中的道路中心线 |
| 6               |           |  |  |
| 7               |           |  |  |
| 8               |           |  |  |

## 后续

1. 起点丢失情况下，考虑使用跟踪起点及其路径