

# LETS节点和SD匹配

## 目录

- [匹配输入](#)
- [预处理](#)
- [总体流程](#)
- [匹配流具体程](#)
  - [Sd Link和Lets中心线配准](#)
    - [思路](#)
    - [方法](#)
  - [匹配LETS节点和SD Link，获取匹配结果](#)
- [过滤重叠的Node节点](#)
- [根据保留下来的Node生成中心线](#)
- [输出Link需要的中心线](#)

## 匹配输入

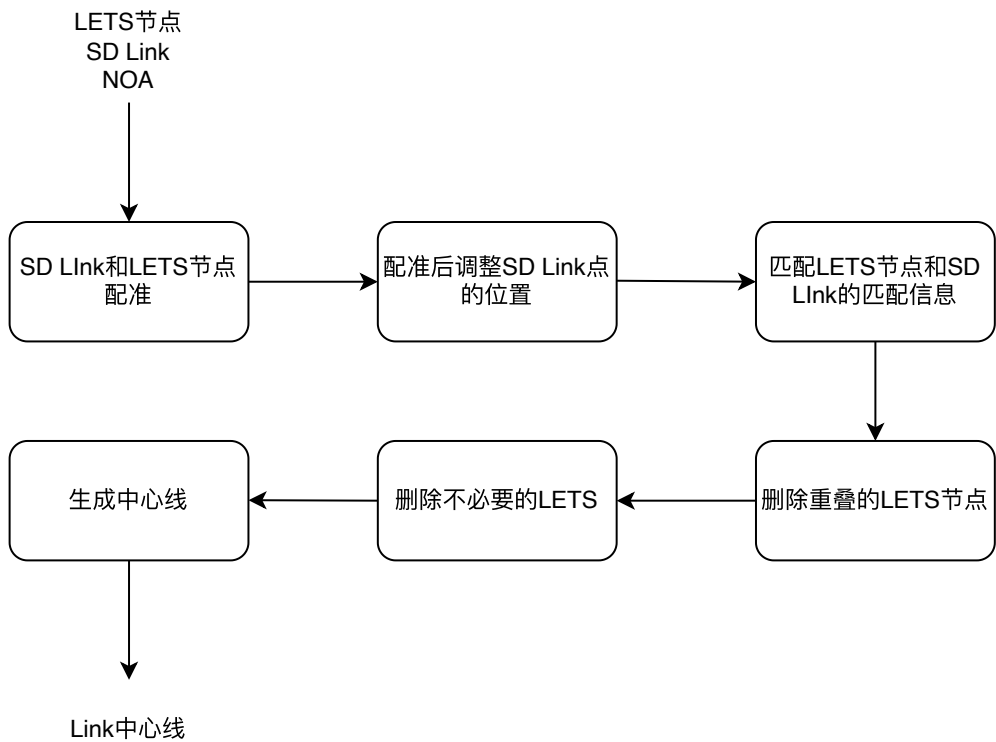
1. LETS搜索的node节点的图（不需要直接输出中心线），可以以有向无环图图的方式给出
2. 所有满足要求的SD Link（比如过滤掉7级及以上道路）

## 预处理

1. 根据有向无环图，计算分流，起点，终点等关键点
2. 根据SD Link的前后继续，生成有向无环图（待定）

## 总体流程

1. SD Link和LETS节点配准，获取旋转矩阵和平移向量
2. 调整SD Link的点的位置
3. 匹配LETS节点和SD Link，获取匹配结果
4. 删除重叠的LETS节点
5. 生成LETS中心线，删除不符合条件的LETS中心线
6. 根据LETS中心线生成感知Link



## 匹配流程具体程

### Sd Link和Lets中心线配准

由于SD Link和LETS中心线的位置有角度偏差和位置偏差，所以需要采用配准的方法对Sd Link位置进行校正

### 思路

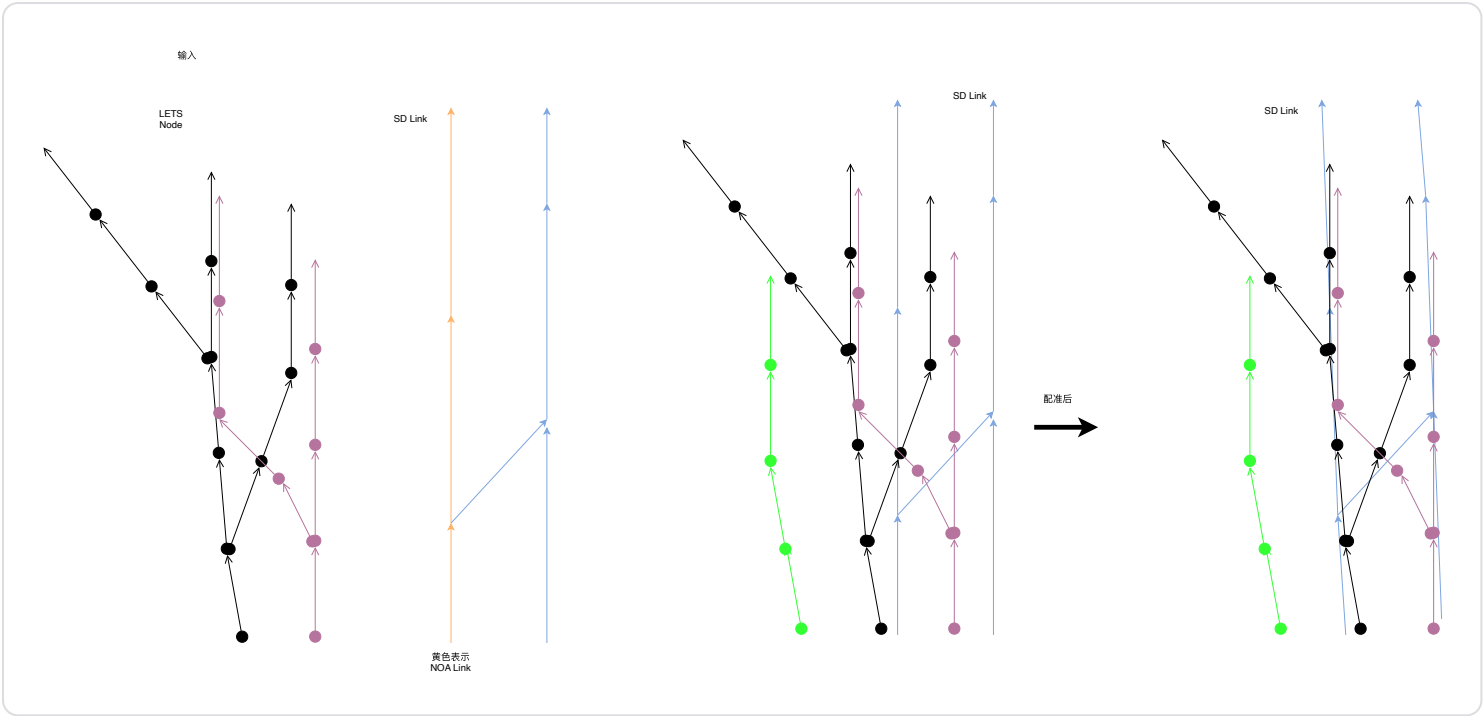
提取lets的路径点和关键点作为原始点云，以Sd Link的线段集合和关键点作为目标，采用迭代的方式计算得到旋转矩阵和平移向量

### 方法

使用点到平面（point-plane）误差度量的迭代最近点（类似ICP的一种方法）

1. Sd Link生成线段的kd tree
2. 遍历LETS节点，判断是否有最近的SD线段
3. 如果符合条件则添加到列表中
4. 根据SD Link线段和其法向量，SD Node的节点坐标，构造矩阵A和向量b
5. 根据带约束的最小二乘法计算出绕z轴旋转角度delta gamma和平移delta x,delta y值
6. 多次迭代（当达到最大迭代次数或者误差小于一定值）获取最终的旋转矩阵R和平移向量t

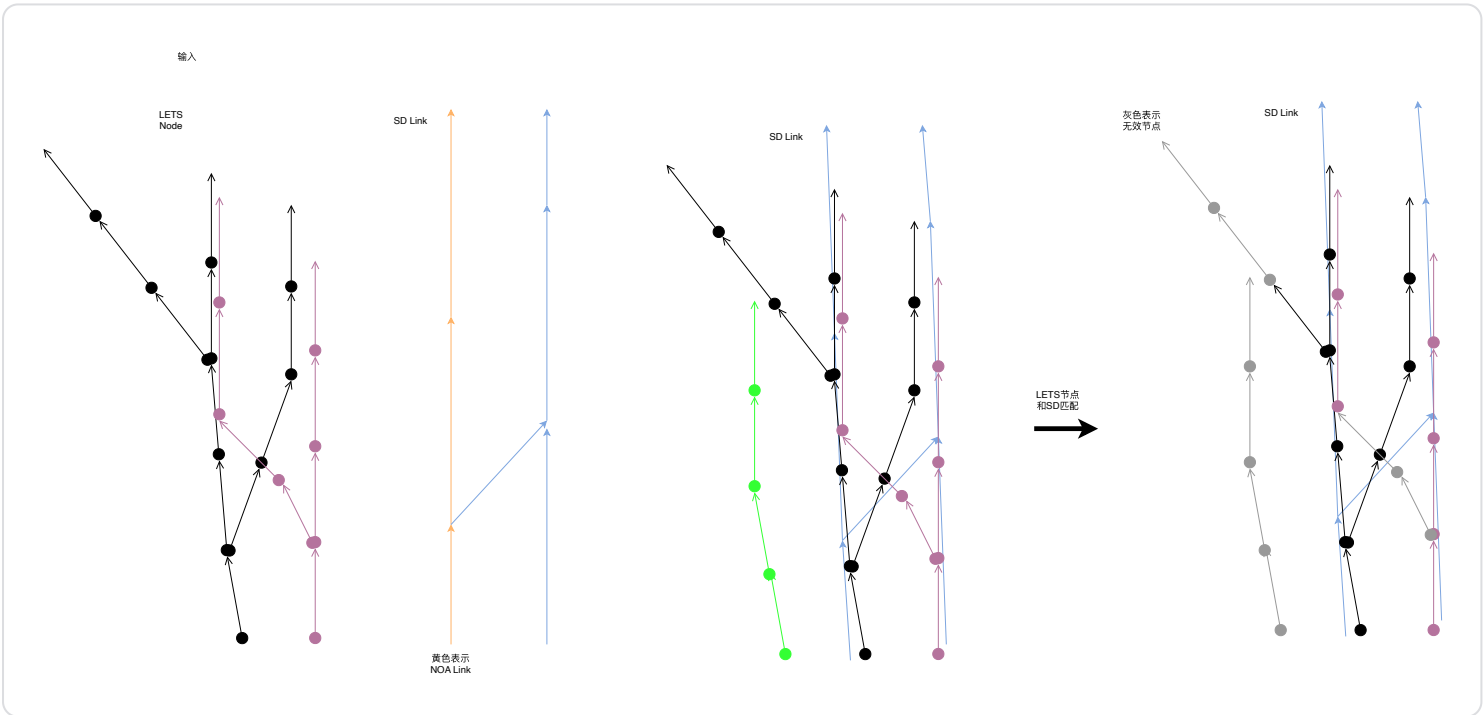
TODO：增加关联点配准



## 匹配LETS节点和SD Link，获取匹配结果

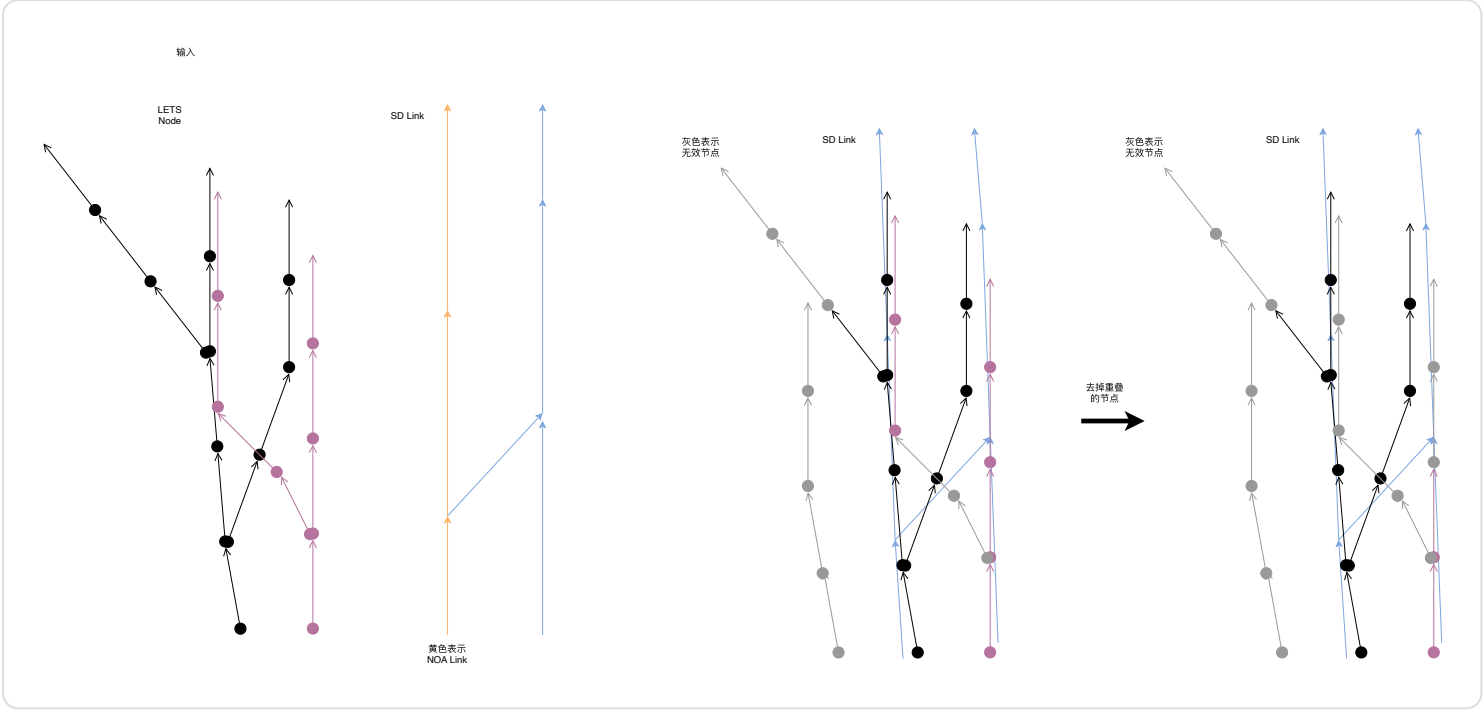
SD Link可以匹配多个LETS节点

1. 对于每一个LETS中的节点，获取和其距离最近的SD Link
2. 判断角度和位置是否符合要求，符合条件则认为SD Link和LETS节点匹配



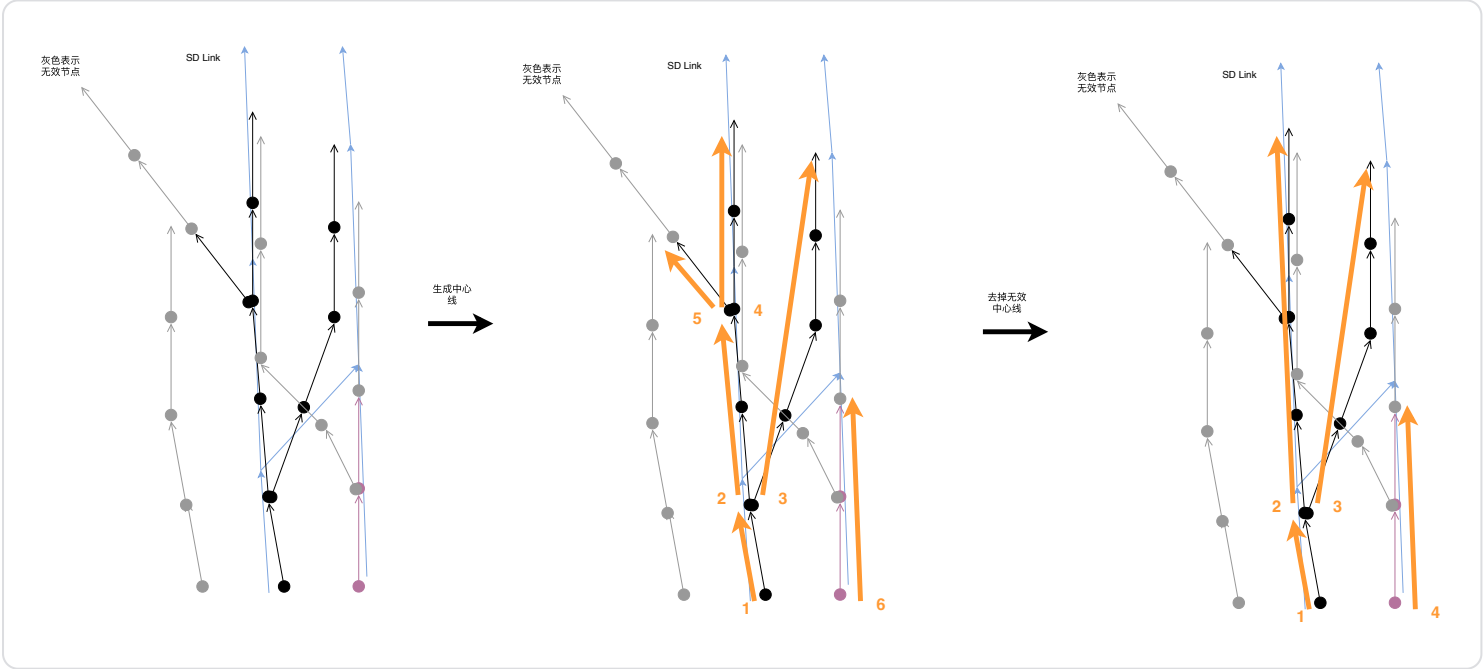
## 过滤重叠的Node节点

1. 根据和NOA 的SDLink的距离关系，删除和NOA关系比较远的重叠的LETS节点
2. 重叠节点的后继节点也一并删除



## 根据保留下来的Node生成中心线

- 1. 根据LETS节点及后继的无分叉的节点生成中心线，有分叉时打断
- 2. 删除过短的或者不符合条件的中心线
- 3. 删除后重新判断是否有中心线可以合并



## 输出Link需要的中心线

输出Link生成需要的格式