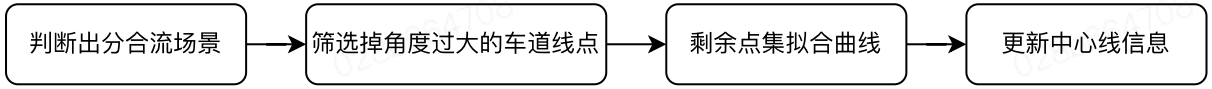
分合流中心线策略说明

目录

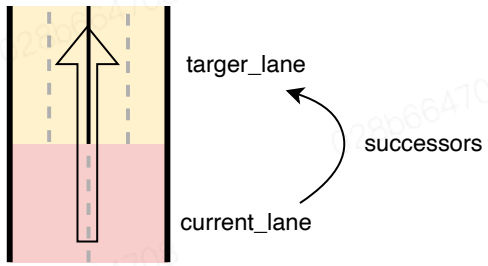
分合流中心线细化Pipeline



1. 场景判断

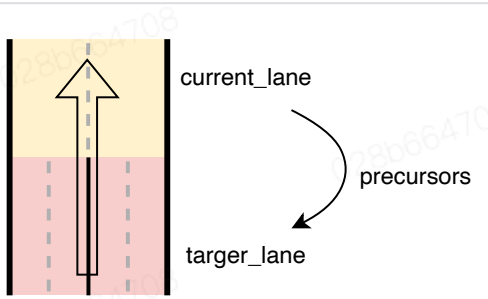
分流场景条件

- 当前车道存在多条后继
- 后继LaneTopo类型为Split、Continue only by left、Continue only by right



合流场景条件

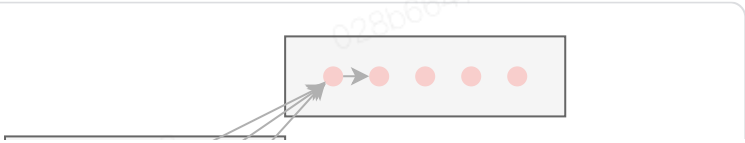
- 当前车道存在多条前继
- 后继LaneTopo类型为Merge、Continue only by left、Continue only by right

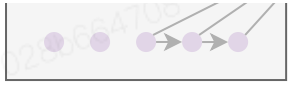


当满足分合流场景判断则对target_lane进行中心线优化

2. 点集筛选

为解决中心线挂接生硬问题，筛选掉与current_lane角度差比较大的点集。**当算法找到除首尾点外满足阈值的点时，停止筛选**

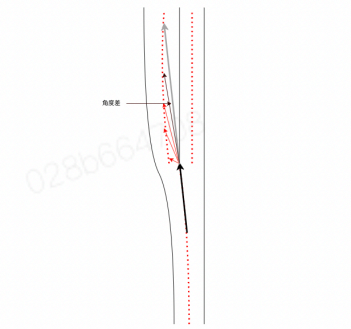




```cpp
double CenterLineEstimator::CalculateAngle(const std::vector<Point3DF>& points, const LanePtr& cur_lane, bool is_merge, int index) {
  Point2DF direction;
  if (is_merge) {
    direction = index + 1 == static_cast<int>(points.size()) ?
      cur_lane->center_line_start_direction : Point2DF(points[index + 1].x - points[index].x, points[index + 1].y - points[index].y);
  } else {
    direction = index - 1 == 0 ?
      cur_lane->center_line_end_direction : Point2DF(points[index].x - points[index - 1].x, points[index].y - points[index - 1].y);
  }
  Point2DF point_vec(points[index].x - points[0].x, points[index].y - points[0].y);
  return std::acos(direction.dot(point_vec) / (direction.norm() * point_vec.norm() + 1e-7)) * 180 / M_PI;
}
```

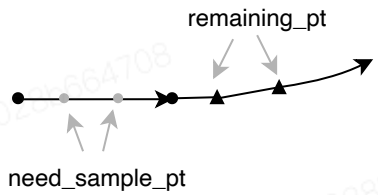
需要注意的是，分流和合流场景current lane和target lane顺序不同，因此所筛选掉顺序有所不同，分流删选掉的是起始点附近的点，合流筛选掉的是终止点附近的点，因此在算法处理上略作区分。

注：策略变更，解决弯道场景的分合流角度判断问题



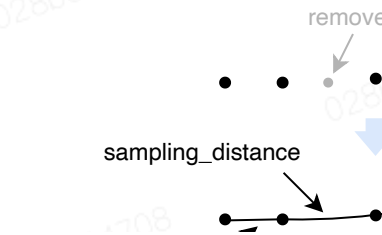
被筛选掉的点集数量，即 **need_sample_pt_size**

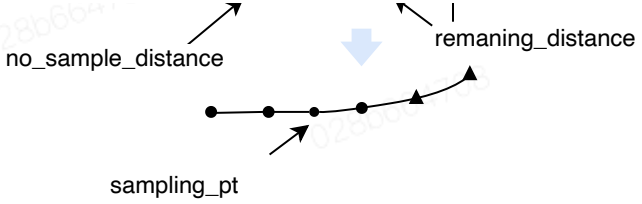
当前车道不满足4个点数时，向前继借点进行曲线拟合，即 **remaining_pt_size**



3. 拟合剩余点集曲线

1. 对剩余点进行曲线拟合
2. 在曲线上找到需要重新采样的曲线段，根据长度占比信息得到出采样步长，计算出重新采样的点





注：原有cubic_spline_2d通过三次样条插值方法在线段之间拟合存在突变情况，应该改用最小二乘方法对所有点进行拟合，再根据距离进行采样。

4. 更新中心线信息

根据上一步得到的曲线，以及**need_sample_pt_size**、**remaining_pt_size**两个数据，选取合适的点集以及曲线段进行中心线信息更新。