

# HMI 车辆类别障碍物补帧优化

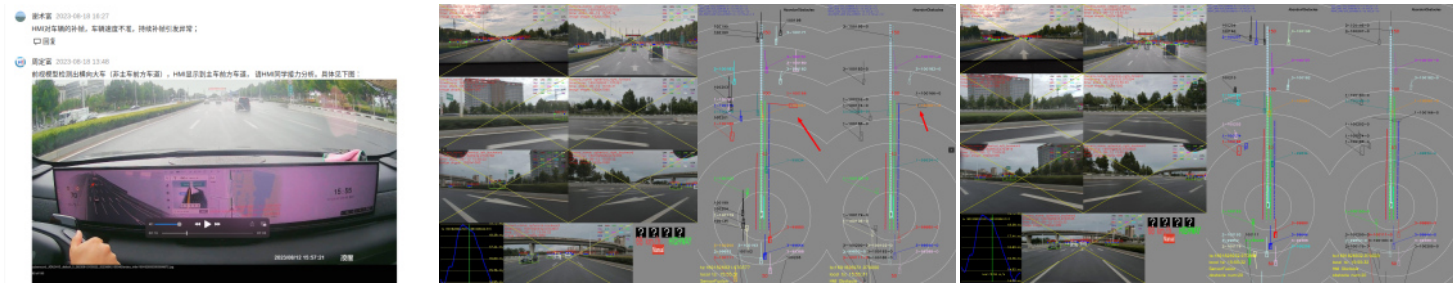
## 目录

- 背景
- 方案
- 代码
- 补帧策略
  - 1.速度可信度评估
  - 2.补帧数目计算
- case回归
  - 视频结果

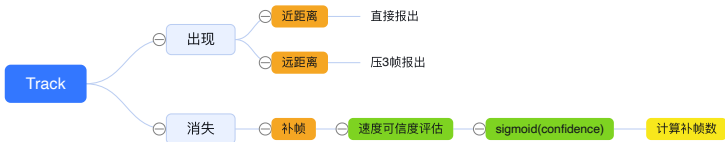
## 背景

远端部分不稳定车辆障碍物状态收敛难，而补帧策略依赖消失前一刻速度，使用错误速度补帧引起异常。

- case链接
  - 📍[JUJBS-4334] 【郑州路试】 【地图测试】 【行车】 沿途行驶时，3D图区中有横向大车闪烁显示
  - 📍[JUJBS-4264] 【地图测试】 【行车】 无交叉道路前方误识别横穿车辆切倒着走



## 方案



对HMI“补10帧”策略进一步优化，由于补帧强依赖消失前的速度。因此对于消失Track，根据其速度可信度进行打分(依据位移/速度自治)从而指定补帧次数。

## 代码

评审: [urban][perception][baidu-adu-lab-andes-113]HMI obs-compensate control

## 补帧策略

### 1.速度可信度评估

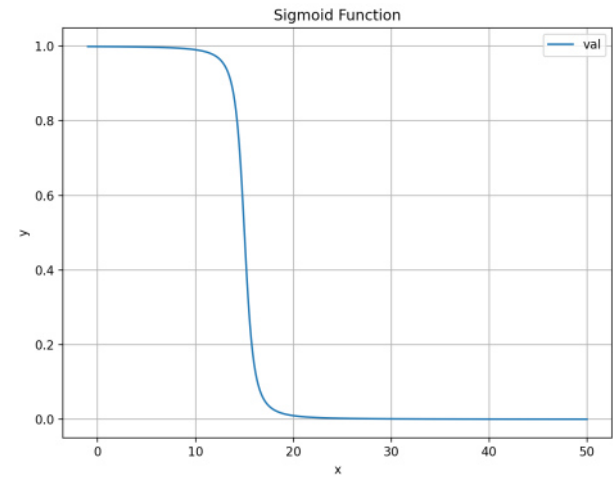
```
float pose_velocity_drift = NAN;
Eigen::Vector3f current_pose = measurement->center.cast<float>();
Eigen::Vector3f last_pose = last_measurement->center.cast<float>();
float time_diff = (hmi_object->timestamp - last_measurement->timestamp) * 1e-9;
if(time_diff > FLT_EPSILON){
    // update track pose velocity drift
    Eigen::Vector2f predict_velocity = (current_pose - last_pose).head(2) / time_diff;
    Eigen::Vector2f mean_velocity = (measurement->velocity + last_measurement->velocity).head(2) * 0.5;
    pose_velocity_drift = (predict_velocity - mean_velocity).norm();
}
```

计算帧间位移差分的速度向量与测量速度向量之差delta\_v，计算5帧窗口内，以delta\_v向量膜值来评定位置/速度的可信度confidence。

2.补帧数目计算

根据速度可信度，计算Track消失后的可补帧数目

补帧数目  $cnt = \text{sigmoid}(\text{confidence})$



上图横坐标代表confidence，纵坐标表示补帧系数k，最终补帧数目  $cnt = k * 10$

case回归

- 👉[JUJBS-4334] 【郑州路试】 【地图测试】 【行车】 沿途行驶时，3D图区中有横向大车闪烁显示

|   |        |  |  |  |  |  |  |  |  |
|---|--------|--|--|--|--|--|--|--|--|
| 1 | before |  |  |  |  |  |  |  |  |
|   | after  |  |  |  |  |  |  |  |  |

- 👉[JUJBS-4264] 【地图测试】 【行车】 无交叉道路前方误识别横穿车辆切倒着走

|   |        |  |  |  |  |  |  |  |  |
|---|--------|--|--|--|--|--|--|--|--|
| 1 | before |  |  |  |  |  |  |  |  |
|   | after  |  |  |  |  |  |  |  |  |

视频结果

- 黄圈：confidence<15
- 红圈：confidence>15 (补帧抑制)

output.mp4 (19MB)

