

臻识API定义问题汇总

1. 接口函数输出参数错误

EventInfo定义为单个结构体，用以表示单个车辆事件结果，单帧处理函数int ProcessFrame(aih::ImageInfo frame[2], aih::EventInfo& result)，该函数一帧图像只能输出单一事件，是否需要替换ProcessFrame参数为 `typedef std::vector<EventInfo> EventInfos;`

确认需要修改

2. EventInfo结构体定义缺少注释：



复制代码

```
struct EventInfo {
    EventType event_type_ = EVENT_TYPE_NO_EVENT; //事件类型（细分）
    U64 track_id_ = 0; //控件ID，ID在AddControl时获取
    S32 control_id_ = -1; //控件ID，ID在AddControl时获取
    S32 event_id_ = -1; //event_type_对应的id
    TimeInfo time_info_; //时间信息
    MoveDirectionThroughLine move_line_dir_ = MOVE_DIRECTION_ANY;
//卡口方向类型，数据是否有误
    BerthEventInfo berth_event_info_; //如果event_type_与泊位相关，该数据有效
    std::list<U64> event_related_ids_; //不确认具体作用
};}
```

该结构中：

move_line_dir_：该字段的具体作用是什么，类型是否定义错误？ **卡口事件触发方向**

event_related_ids_：在字段的具体含义以及作用是什么？ **未使用，忽略**

3. BerthEventInfo字段定义问题



复制代码

```
struct BerthEventInfo
{
    U32 result_cnt_ = 0; //结果个数，最大为MAX_RESULT_CNT
```

信息	<code>AlarmInfo</code>	<code>alarm_info_</code>	<code>;</code>	//报警
式	<code>U32</code>	<code>report_mode_</code>	<code>= 0;</code>	//上报模
属性	<code>CarProperty</code>	<code>car_property_</code>	<code>;</code>	//车辆
晰度	<code>FLOAT</code>	<code>sharpness_[MAX_RESULT_CNT]</code>	<code>= { 0 };</code>	//车牌清
素均值	<code>FLOAT</code>	<code>plate_mean_[MAX_RESULT_CNT]</code>	<code>= { 0 };</code>	//车牌像
丢弃	<code>U32</code>	<code>can_drop_[MAX_RESULT_CNT]</code>	<code>= { 0 };</code>	//是否可
1车头, 2车尾	<code>LoopStatus</code>	<code>parking_state_[MAX_RESULT_CNT] = {</code> <code>LOOP_STATUS_UNKNOWN };</code>	<code>//停车状态</code>	
数	<code>BerthCarInfo</code>	<code>car_info_[MAX_RESULT_CNT]</code>	<code>;</code>	
	<code>CachedFrameInfo</code>	<code>frame_info_[MAX_RESULT_CNT]</code>	<code>;</code>	
	<code>U32</code>	<code>car_gesture_</code>	<code>= 0;</code>	//0无效,
件, 0无遮挡, 1遮挡	<code>U32</code>	<code>order_score_</code>	<code>= 0;</code>	//订单分
位开始时间戳	<code>U32</code>	<code>covered_event</code>	<code>= 0;</code>	//遮挡事
	<code>U64</code>	<code>enter_start_timestamp_</code>	<code>= 0;</code>	//进入泊
开标志	<code>U32</code>	<code>force_leave_flag_</code>	<code>= 0;</code>	//强制离
ID	<code>U32</code>	<code>group_id_</code>	<code>= -1;</code>	//泊位组
车类型	<code>U32</code>	<code>non_motor_car_type_</code>	<code>= 0;</code>	//非机动
车id	<code>U32</code>	<code>non_motor_id_</code>	<code>= -1;</code>	//非机动
制视频的标志, 0:不处理, 1:出场, 2 : 入场	<code>U32</code>	<code>out_video_start_</code>	<code>= 0;</code>	//开始录
否移动过	<code>U32</code>	<code>recent_plate_</code>	<code>= 0;</code>	//车牌是
否被遮挡	<code>U32</code>	<code>plate_covered_</code>	<code>= 0;</code>	//车牌是
稳定的帧计数	<code>U32</code>	<code>car_stable_count_</code>	<code>= 0;</code>	//车辆停

该结构中：

`sharpness_`, `plate_mean_`相关字段为什么是使用数组表示,具体有什么作用

`sharpness_`网页配置只输出一张或两张的情况下, 或者输出合成图的情况下, 这张图可以丢弃, ,用来选图, 合成图的车身图根据这个清晰度来选一张; (0-10000)

plate_mean_ 车牌像素均值,暂未使用.先可以不管

can_drop_: 该字段的具体作用是什么?

can_drop_ 网页配置只输出一张或两张的情况下, 或者输出合成图的情况下, 这张图可以丢弃, 和sharpness类似, 先排除can_drop_的图片, 然后再比较sharpness_

4. 缺少默认城市字段配置, 是否可以新增该字段

5. 证据链更新问题:



复制代码

```
struct TrackObjUpdateInfo {
    S32 control_id_ = -1;
    S32 event_id_ = -1;
    U64 track_id_ = 0;

    ObjMainCls obj_cls_ = OBJ_MAIN_CLS_UNKNOWN; //属性类别
    bool obj_update_ = false;
};
```

证据链跟track id走, 如果trackid 变更如何操作?

event_id_: 具体代表含义

是否可以把单独保存某张帧作为一个事件传出

该结构体未使用

6. 车牌颜色, 车辆颜色问题



复制代码

```
struct SinglePropDefine {
    int type_id_ = PROP_UNKNOWN;
    std::string type_str_ = "";
    float type_score_ = 0.f;
};

typedef SinglePropDefine CarTypeProp; //车型
```

```
typedef SinglePropDefine CarColorProp;      //车色
typedef SinglePropDefine CarBrandProp;        //车标
```

CarColorProp::type_id_：车身颜色是否是定义在aih_prop.h中，或者是目前未定义

车牌信息定义：



复制代码

```
struct PlateInfo {
    float plate_score_ = 0.f;           //车牌号置信度
    float plate_type_prob_ = 0.f;        //车牌类型置信度
    std::string plate_str_ = "";         //车牌号

    int plate_type_id_ = 0;             //车牌类型ID,见PLATE_TYPE // TODO预留
    int plate_color_id_ = 0;             //车牌颜色ID,见PLATE_COLOR //TODO 具体颜色没有定义，需要我们这边自己定义结构
    int plate_layer_id_ = 0;             //车牌单双层ID

    std::string plate_type_str = "";    //车牌类型
    std::string plate_color_str_ = "";   //车牌颜色
    std::string plate_layer_str_ = "";   //车牌单、双层

    std::vector<float> plate_scores_;   //车牌字符置信度
    std::vector<std::string> plate_chars_; //车牌每一个字符
};
```



复制代码

```
enum PLATE_COLOR {
    COLOR_OTHER = 0,    //其他颜色
    COLOR_BROWN = 1,    //棕色
    COLOR_WHITE = 2,    //白色
    COLOR_RED = 3,      //红色
    COLOR_BLUE = 4,     //蓝色
    COLOR_SILVER = 5,   //银色
    COLOR_BLACK = 6     //黑色
};
```

PLATE_COLOR：车牌颜色定义不准确

命名错误，需要修改，算法内部只支持这些颜色，给到松立协议的颜色是经过映射的

7. 假驶入/驶出之后，冗余的上报信息怎么处理

比如车辆驶入泊位，但没有发生停车行为，之前上报的预驶入event怎么通知删除

不删除；

8. 拌线配置：



复制代码

```
struct ControlInfo {
    S32 id_ = -1;      //控件control ID >=0 有效
    ControlType control_type_ = CONTROL_TYPE_UNKNOWN;
    std::vector<Point2i> point_infos_; //输入点连续，且0-1-2...n为线段
方向
};
```

拌线条数可根据点的数量来判断？比如画面中会有设置双拌线的情况，point_infos_是否存在四个坐标点，以及坐标点顺序。

每条拌线是一个空间，只支持线段

9. 事件间隔触发的含义

泊位/区域配置中，字段事件间隔触发的适用场景

10. 区域业务问题



复制代码

```
typedef struct MutiStayAreaEvent
{
    bool interval_trigger_ = false;           //事件是否间隔触发
    U32 stay_count_ = 5;
    U32 stay_time_ = 5000;                   //滞留时间阈值，单位毫秒
    FLOAT in_area_percent_ = 0.5;            //滞留在区域内的面积比
}MutiStayAreaEvent , SingleLaneCongestionEvent;
```

结构体的具体含义是什么，区域业务场景对应的功能是什么？

默认清空下，事件只触发一次，interval_trigger_ = true表示会重复触发，重复次数由stay_count_决定

11. SingleShieldAreaEvent业务场景



复制代码

```
typedef struct _area_event {
    FLOAT in_area_percent_ = 0.5f; //面积比阈值，大于阈值，才认为进入区域
} SingleEnterAreaEvent, SingleShieldAreaEvent;
```

结构定义的具体含义是什么？

IOU阈值, 当空间区域和目标区域iou> in_area_percent_的时候才触发

12. 泊位事件的report_mode字段

证据链模式与事件模式的区别是什么？完整行为的所有事件一起上报和分别上报？

证据链模式：会补齐完整的证据链上报

事件模式：只上报实际发生的事件

比如，一个车在泊位停好，被遮挡后离开，证据链模式会补齐 出场空闲事件，事件模式则不会补齐

13. 违停配置



复制代码

```
//控件类型
enum ControlType {
    CONTROL_TYPE_UNKNOWN = -1,
    CONTROL_TYPE_AREA = 0, //区域
    CONTROL_TYPE_LINE, //线
    CONTROL_TYPE_BERTH, //泊位
    CONTROL_TYPE_VIOLATION, //违停
    CONTROL_TYPE_LANE, //车道

    CONTROL_TYPE_NUM
};
```

不规范停车是否以泊位为单位进行配置，也即配置类型是CONTROL_TYPE_BERTH时，不规范停车开关有效

禁停区域配置是否以CONTROL_TYPE_VIOLATION为字段作为开关

其他修改项:

车牌颜色定义

确认目标信息在后续处理中是否有用.