行车轨迹线的绘制程序

使用说明书

上海寅家电子科技股份有限公司

2024.3.20

[**第一部分 设计说明** 4](#_Toc162368344)

[**1 系统概况** 4](#_Toc162368345)

[**2 软件架构** 5](#_Toc162368346)

[**3 使用环境** 7](#_Toc162368347)

[**4 设计依据** 8](#_Toc162368348)

[**第二部分 系统功能介绍** 9](#_Toc162368349)

[**1** 进入及退出 9](#_Toc162368350)

[**1.1 启动系统** 9](#_Toc162368351)

[**1.2 退出系统** 11](#_Toc162368352)

[**1.3 与车身系统联立** 12](#_Toc162368353)

[**第三部分 故障处理** 14](#_Toc162368354)

[**1 显示故障处理** 14](#_Toc162368355)

[**1.1** **线段显示异常** 14](#_Toc162368356)

[**1.2** **整体无法出现** 14](#_Toc162368357)

[**2** **初始化故障处理** 15](#_Toc162368358)

[**2.1** **GPU异常** 15](#_Toc162368359)

[**2.3** **Memory异常** 15](#_Toc162368360)

[**3** **运行过程中的故障处理** 16](#_Toc162368361)

# **第一部分 设计说明**

## **1 系统概况**

在360的顶视，前后视图，倒车影像中可以见到这些行车轨迹线的身影。其主要目的，是模拟车辆运行时车轮的运动轨迹，辅助车辆停靠，防止车辆在行驶的过程中撞到障碍物。其借助方向盘转角信息将汽车可能的行驶路线叠加到摄像头画面上输出，以直观形象化的行驶协助驾驶人员调整选择行驶路线，减少驾驶人员特别时新手的误判断，对使用者是一个不错的实用功能。行车轨迹线在智能驾驶领域的实用性和辅助性是一个有效的补充。

## **2 软件架构**

软件架构分为软件流程和方向盘角度处理流程。

软件处理流程如下：

输入起始坐标点

计算圆心

计算转弯半径

将待绘制的轨迹线分段

传入对应的半径，每段计算一个半径

坐标获取结束

将计算的三维坐标点保存

映射到摄像头画面

对应屏幕顶点填充OpenGL坐标

方向盘角度处理流程如下：

启动程序

初始化已知数据

获取方向盘角度

转换为车轮角度

根据公式计算车辆行驶的实际点坐标

转向角度变化

车轮轨迹计算结束

是

否

## **3 使用环境**

开发工具：Visual Studio Code

　Ubuntu18.04

运行环境：类Unix系统，如Linux，QNX

Android系统

OpenGL3.0版本

## **4 设计依据**

(1)《计算机软件开发规范》GB 8566；

(2)《软件开发规范》

(3)《计算机软件可靠性和维护性管理》

(4)《计算机软件质量保证计划规范》

(5)《中华人民共和国公共安全行业标准》

(6)《安全技术防范规范工程程序技术规范》

**第二部分 系统功能介绍**

**1** 进入及退出

**1.1 启动系统**

调用AVM\_DrawCamviewGuideLine\_Rear(stVTDScrnWin win)API即可查看轨迹线，如下图：

图片包含 游戏机, 灯光, 播放器

描述已自动生成

图1 轨迹线实际效果确认图

图片包含 游戏机, 灯光

描述已自动生成程序可以根据获得的方向盘角度实时转变方向盘角度，如下图：

图2a 轨迹线实际效果确认图—左转向

图2b 轨迹线实际效果确认图—左转向形状

描述已自动生成

程序可以输出计算出的每个点位，并以车辆所在空间建立一个实际的线条，在三维空间中的仿真效果如下图：

图3 三维空间中的仿真效果

在三维空间中的仿真结果中可以看到，在以摄像头为原点建立空间坐标系的情况下，轨迹线为左右两条平行的线段。2500mm为后保险杠的投影位置，从这个距离开始进行计算初始值，设置轨迹线的有效值为3000mm，因此轨迹线的终点值为5500mm。

转成屏幕的坐标，并附着摄像头内外参数之后的效果如图：

图4 映射到屏幕的坐标点

映射后的屏幕坐标点中可以看到，在OpenGL的屏幕坐标系中，坐标值的范围在(-1.0, 1.0)的区间内。考虑景深的情况，在进行映射时添加了摄像头外参中的R和T矩阵，这样仿真的到的点位能够贴合摄像头像素的变化，即近处平行，远处几近相交的情况，满足实际情况。

## **1.2 退出系统**

取消调用AVM\_DrawCamviewGuideLine\_Rear(stVTDScrnWin win)API，轨迹线将不再继续显示。

## **1.3 与车身系统联立**

想要让轨迹线的显示更贴合实际情况，还需要进行以下的修改，还需要与车身信息联立。需要输入车身信息和辅助线信息。在代码的开始有定义一些车身，和轨迹线的相关参数：

车身信息：

* 车长

CAR\_LENGTH\_CAMVIEW

* 后轮轮距

CAR\_WHEELBASE\_CAMVIEW

* 车宽

CAR\_OUTLINEWIDTH\_CAMVIEW

* 轴距

CAR\_WIDTH\_CAMVIEW

* 车辆的最大转向值

LUT\_StartAngle

LUT\_EndAngle

辅助线信息：

* 辅助线长

CAR\_LENGTH\_

* 辅助线宽

CAR\_OUTLINEWIDTH\_CAMVIEW

* 辅助线距保险杠的宽度

CAMLINE\_COORD\_START\_REAROFFSET

* 辅助线绘制宽度

CAMLINE\_OUTLINE\_WIDTH

需要进行一些实际情况的添加，以达到最终的准确效果：

1. 调整辅助线距后保险杠的距离，让其满足功能规范
2. 调整辅助线长度，让其满足功能规范
3. 保持方向盘为0度角，调整辅助线宽度，辅助线外侧贴合车辙外侧痕迹。

# **第三部分 故障处理**

## **1 显示故障处理**

## **线段显示异常**

当前的软件无法处理一些突变情况，当获得的角度信号发生突变时（从100变成0或者从-100变成0）的时候会出现辅助线断开的情况，再次发送连续且无突变的角度信号时，显示恢复正常。当前传递的角度值应当都是连续的值，因此这种情况被认为是只存在于理论情况下。

## **整体无法出现**

使用OpenGL进行绘制和填充时，因为会涉及到显示窗口的大小和位置等情况，会导致无法出现的问题，将窗口设置为合理位置即可

## **初始化故障处理**

## **GPU异常**

当出现“CamLine UvProg was not loaded”字样时，说明是OpenGL的着色器初始化出现了问题，可能是由GPU显存过小或GPU不支持OpenGL3.0版本，需要换用合适的GPU方能恢复。

## **Memory异常**

程序在保存三维坐标点和屏幕坐标点的时候均使用到了std::vector这样的容器保存，如果没有足够的内存进行保存，可能会导致程序初始化失败，建议保留10kb的空间运行程序。

## **运行过程中的故障处理**

程序使用了枚举管理控制线段的显示，如果因为某些意外访问到非定义的枚举，会导致程序越界，产生code dump文件，换用定义后的枚举即可正常运行。当前定义的枚举如下：

typedef enum{

OBJ\_OUTLINE\_L\_BOTTOM\_DYNAMIC,

OBJ\_OUTLINE\_R\_BOTTOM\_DYNAMIC,

OBJ\_OUTLINE\_L\_MIDDLE\_DYNAMIC,

OBJ\_OUTLINE\_R\_MIDDLE\_DYNAMIC,

OBJ\_OUTLINE\_L\_TOP\_DYNAMIC,

OBJ\_OUTLINE\_R\_TOP\_DYNAMIC,

OBJ\_DISLINE\_TOP\_DYNAMIC,

OBJ\_DISLINE\_MIDDLE\_DYNAMIC,

OBJ\_DISLINE\_BOTTOM\_DYNAMIC,

OBJ\_OUTLINE\_L\_STATIC,

OBJ\_OUTLINE\_R\_STATIC，

OBJ\_DISLINE\_STATIC,

OBJ\_NUM, //12

}enCamviewLineObj;