**车道线检测（传统法）**

**技术说明文档**

学 校： 成都信息工程大学

开发人员： 钟弋辰

**摘要**

该项目使用opencv和ros进行视频的发布和车道线的检测。

关键字：opencv，ros，图像处理

**目录**

[摘要 - 2 -](#_Toc3738)

[（一）实现步骤 - 4 -](#_Toc19991)

[1.透视变换，将路面转换到识别区域内 - 4 -](#_Toc21514)

[1.1原理 - 4 -](#_Toc27486)

[1.2实现 - 6 -](#_Toc1668)

[2.筛选车道线，并二值化 - 7 -](#_Toc15262)

[2.1 原理 - 7 -](#_Toc19168)

[2.1实现 - 8 -](#_Toc2778)

[- 8 -](#_Toc21848)

[3.寻找色块，并提取感兴趣的车道线坐标点 - 8 -](#_Toc7050)

[3.1原理 - 8 -](#_Toc18724)

[3.2实现 - 8 -](#_Toc11318)

[4.对色块进行遍历 - 9 -](#_Toc14867)

[5.二次函数拟合 - 10 -](#_Toc15152)

[5.1原理 - 10 -](#_Toc2022)

[5.2实现 - 10 -](#_Toc269)

[6.对拟合后的曲线计算曲率 - 11 -](#_Toc18799)

[6.1原理 - 11 -](#_Toc8119)

[6.2实现 - 12 -](#_Toc28101)

[7.将拟合后的曲线绘制到原图 - 12 -](#_Toc13145)

[7.1原理 - 12 -](#_Toc11282)

[7.2实现 - 13 -](#_Toc7033)

[（二）测试视频效果 - 13 -](#_Toc24253)

[（三）测试视频出现的问题 - 14 -](#_Toc17906)

[（四）额外 - 15 -](#_Toc3333)

[1.ros部分视频读取并发布 - 15 -](#_Toc21164)

[2。视频的订阅并调用回调处理 - 18 -](#_Toc15937)

[3.关于原测试视频左右车道线区分实现的讨论 - 19 -](#_Toc24145)

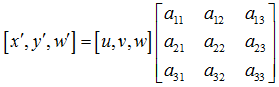
[3.1原测试视频最后的区分效果 - 21 -](#_Toc19559)

# （一）实现步骤

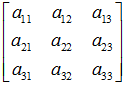
## 1.透视变换，将路面转换到识别区域内

#### 1.1原理

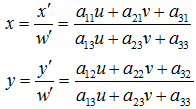
getPerspectiveTransform 通过该函数可以将图像变换到正交视图下，可以排除一部分路面外的干扰，并且易于分析曲率。



u,v是原始图片左边，对应得到变换后的图片坐标x,y,其中。

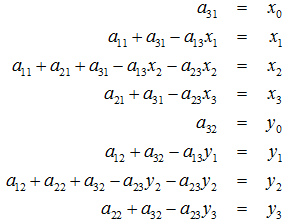
变换矩阵可以分作四部分来理解，IMG_258表示线性变换，IMG_259表示平移，IMG_260产生透视，

所以可以理解成仿射等是透视变换的特殊形式。经过透视变换之后的图片通常不是平行四边形（除非映射视平面和原来平面平行的情况）。

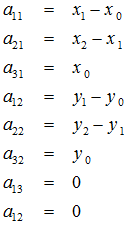
重写之前的变换公式可以得到：

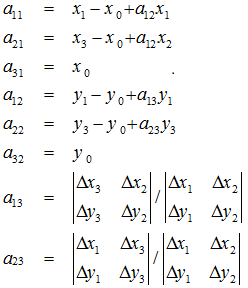
所以，已知变换对应的几个点就可以求取变换公式。反之，特定的变换公式也能新的变换后的图片。简单的看一个正方形到四边形的变换：

IMG_262

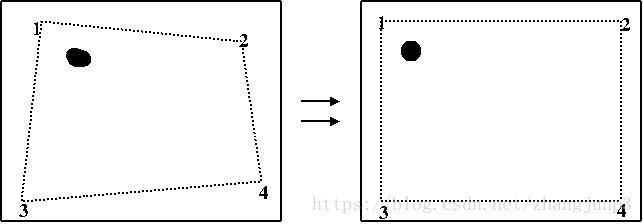
根据变换公式得到：

定义几个辅助变量：IMG_264

IMG_265都为0时变换平面与原来是平行的，可以得到：

IMG_267不为0时，得到

求解出的变换矩阵就可以将一个正方形变换到四边形。反之，四边形变换到正方形也是一样的。于是，我们通过两次变换：四边形变换到正方形+正方形变换到四边形就可以将任意一个四边形变换到另一个四边形。

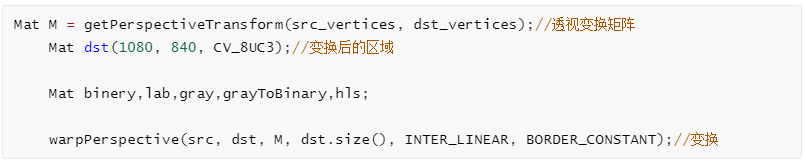


#### **1.2实现**

##### 1.2.1 创建映射点数组



##### 1.2.2 透视变换



## 2.筛选车道线，并二值化

#### **2.1 原理**

仅考虑一般路面容易出现的黄色线和白色线，黄色线使用lab色域进行筛选，白色线使用rgb或hls色域进行筛选。

inRange(hls, Scalar(0, 232, 150), Scalar(255, 255, 255), binery);

使用inrange函数可以实现对颜色在三个维度内的筛选，并且二值化。

#### **2.1实现**

##### 2.1.1.创建不同色域图片变量



##### 2.1.2.转换色域



##### 2.1.3.二值化

### 

## 3.寻找色块，并提取感兴趣的车道线坐标点

#### **3.1原理**

findContours

使用findContours函数可以寻找色块的轮廓。根据轮廓组成的点，我们可以提取出色块线条的大致点序列。以便用于后面的拟合

#### **3.2实现**



## 4.对色块进行遍历

遍历中根据相邻距离。区分色块是否为同一车道线

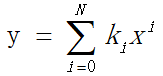
由于后面测试视频中色块过于离散，我还没有能力实现作区分的算法。

原来的视频中线比较连续。是实现了的。

## 5.二次函数拟合

#### **5.1原理**

基本原理：幂函数可逼近任意函数。



上式中，N表示多项式阶数，实际应用中一般取3或5；

假设N=5，则：

IMG_271

共有6个未知数，仅需6个点即可求解；

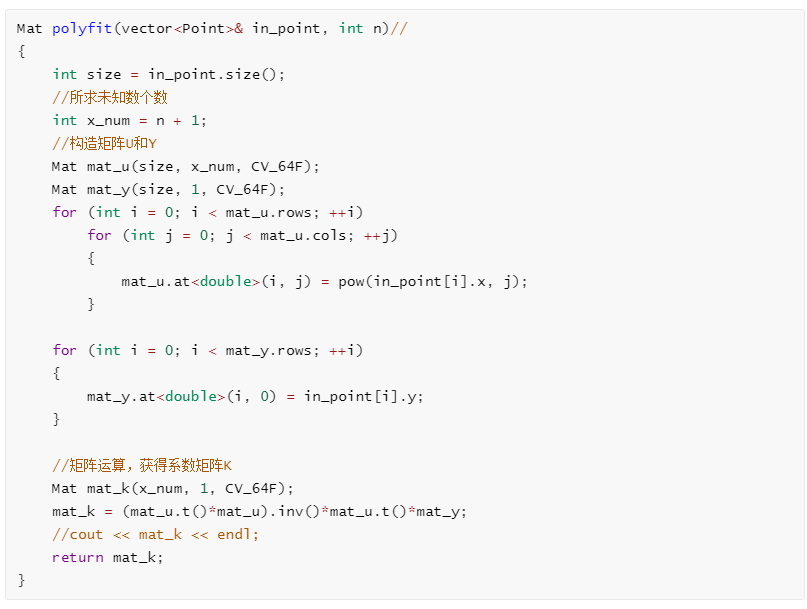
可表示为矩阵方程：IMG_272

Y的维数为[R\*1]，U的维数[R \* 6]，K的维数[6 \* 1]。

R> 6时，超定方程求解：

IMG_273

#### **5.2实现**



## 6.对拟合后的曲线计算曲率

#### **6.1原理**

IMG_274

求解一阶导数的公式：y’(i) = (y(i+1)-y(i))/h;

（ｙ(i)处y’(i) = △y(i)/△x(i)）

求解二阶导数的公式：y’’(i) = (y(i+1)+y(i-1)-2\*y(i))/h^2;

（两处h为△x(i)）

#### **6.2实现**



每遍历一个点和上一个点作差，即为y',此次求得的y‘和上一次求得的y’作差，即为y‘’，每次遍历可以计算一次，求和并且计数，最后在循环外取平均即可。

## 7.将拟合后的曲线绘制到原图

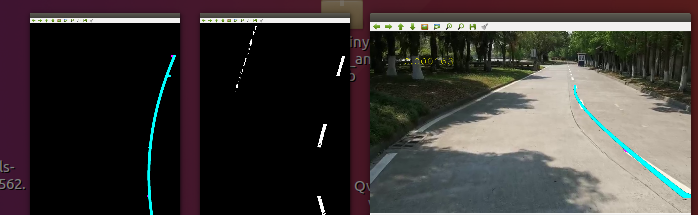
#### **7.1原理**

现将线条绘制到一张透视变换后一样大的黑色图里，将其逆透视变换回原图。然后在创建一个他的拷贝并进行二值化用于作为遮罩。然后使用二值化图对其进行剔除来绘制到原图

#### **7.2实现**

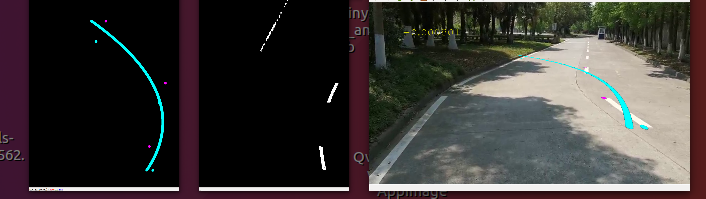


# **（二）测试视频效果**



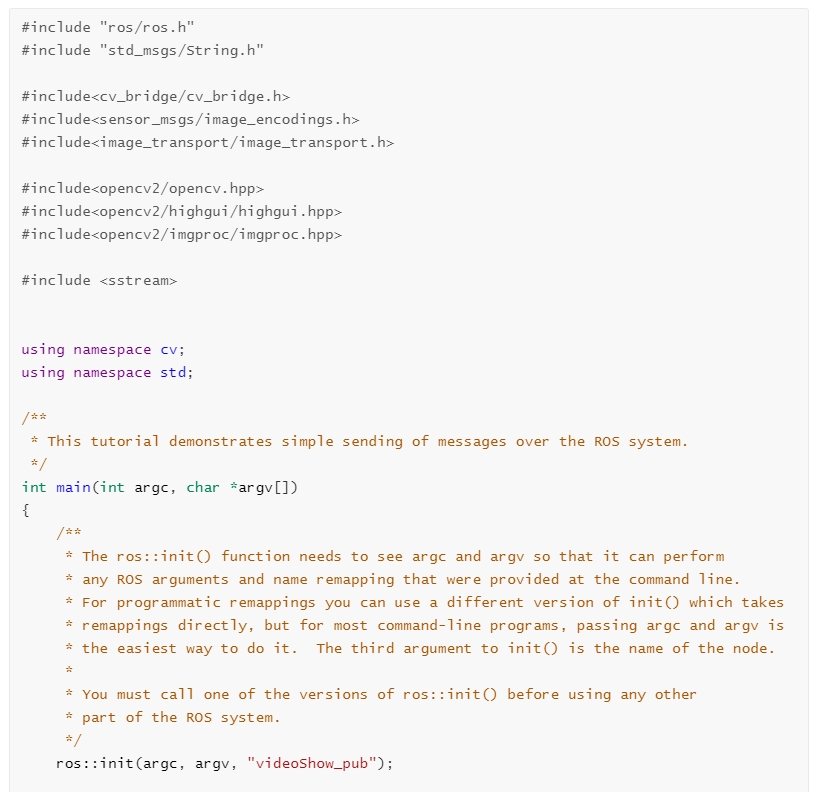
# **（三）测试视频出现的问题**

未实现区分较为离散的车道线算法。



# **（四）额外**

## **1.ros部分视频读取并发布**



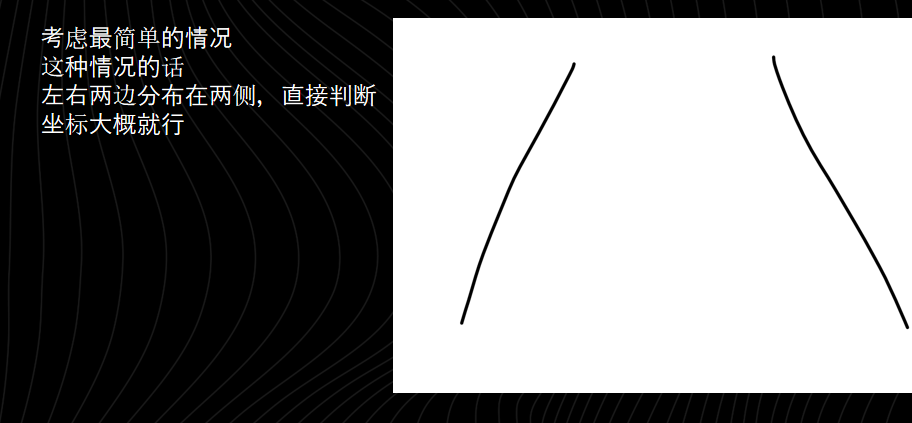


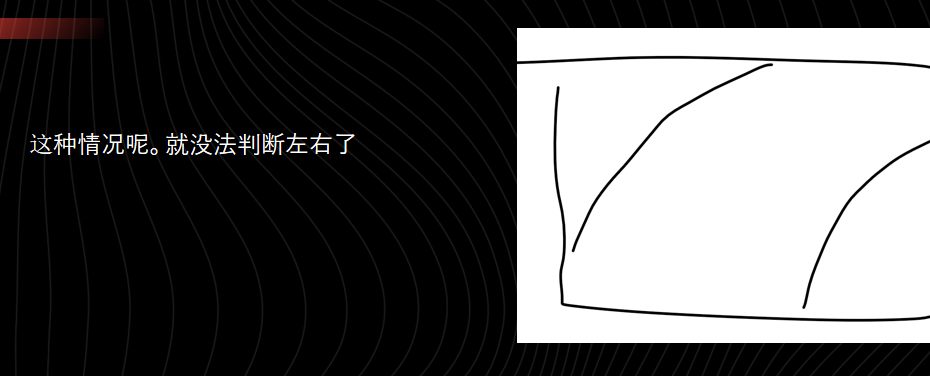


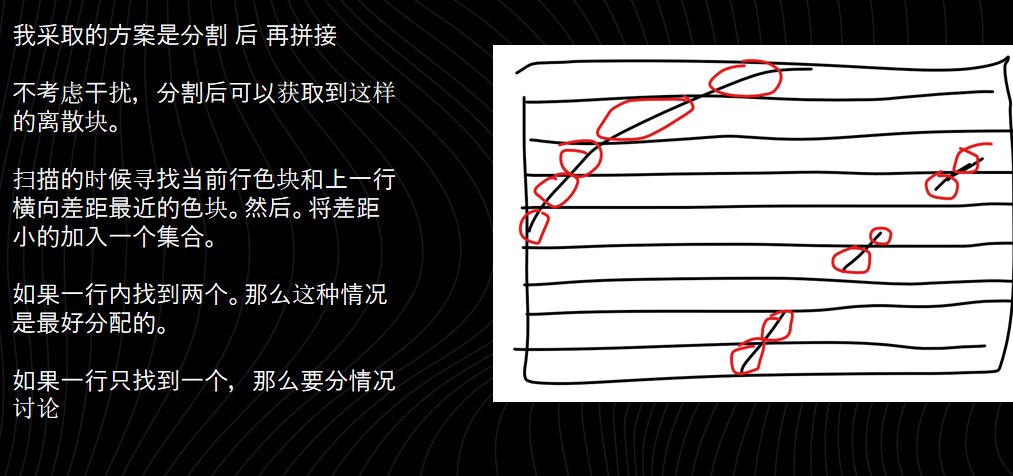
## 2。视频的订阅并调用回调处理

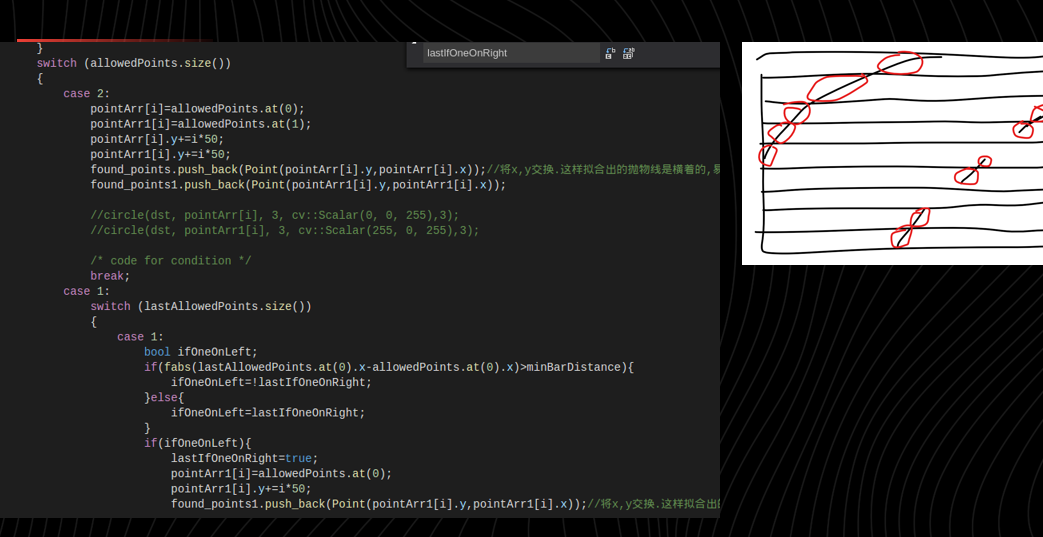


## 3.关于原测试视频左右车道线区分实现的讨论









### 3.1原测试视频最后的区分效果

