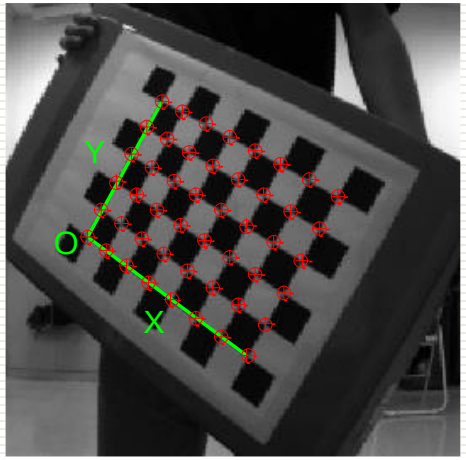
第四章

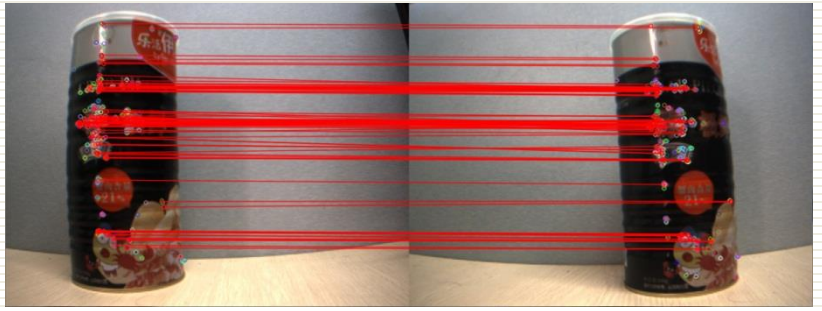
目标检测

目标检测方法

- **静态目标检测**：目标几何尺寸测量、瑕疵检测等，需要提高单帧图像的处理速度；
- **动态目标检测**：提取运动目标特征进行检测。

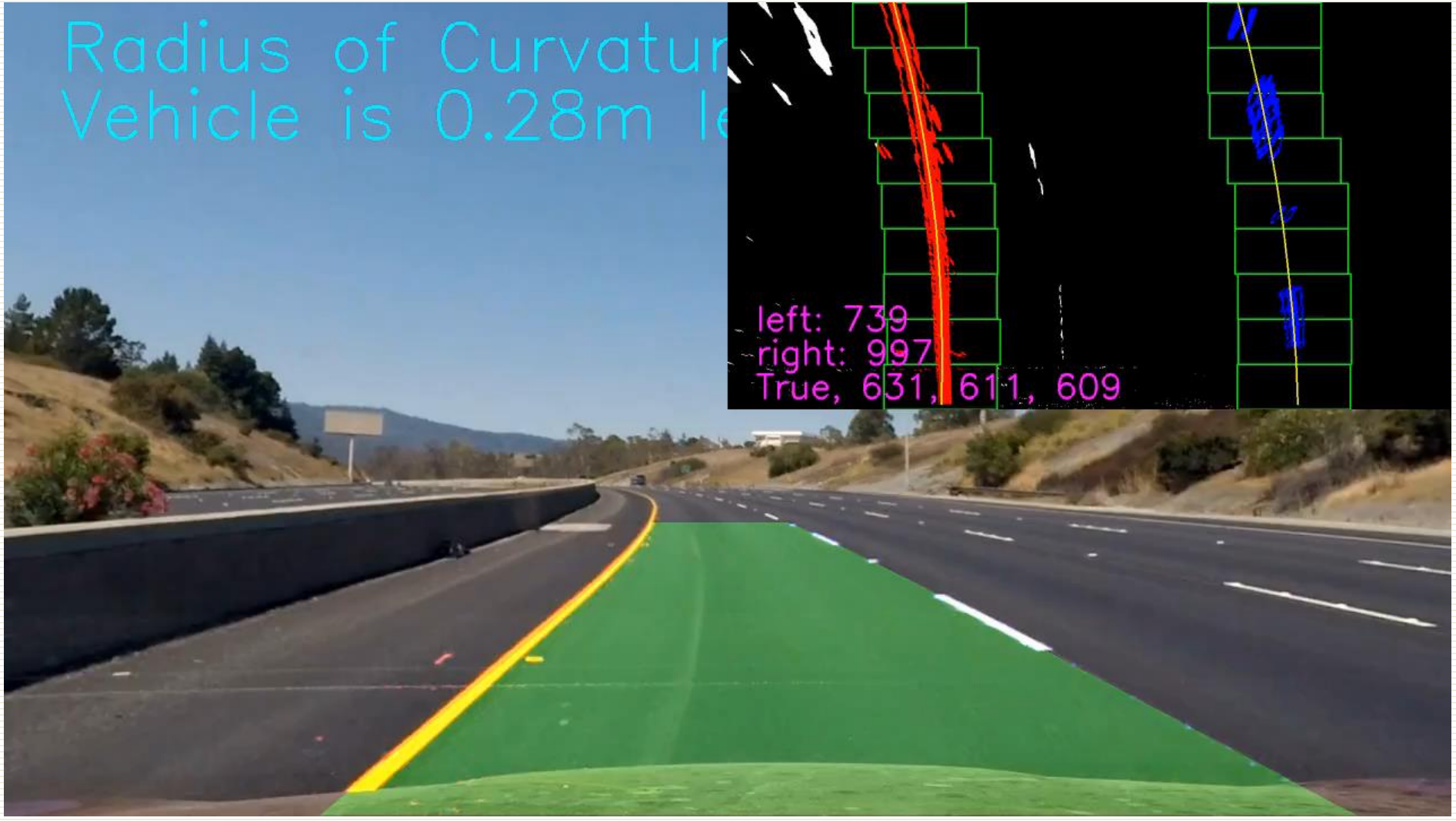


- 常用**目标特征**及表示方式：
  - 颜色：RGB、HSV空间，提取颜色直方图；
  - 纹理：灰度共生矩阵；
  - 轮廓：边缘检测、模板等；
  - 角点：图像中局部曲率突变的点，在角点处沿各个方向都存在较大的灰度值变化。
  - 特征点：SIFT、Surf、ORB等。



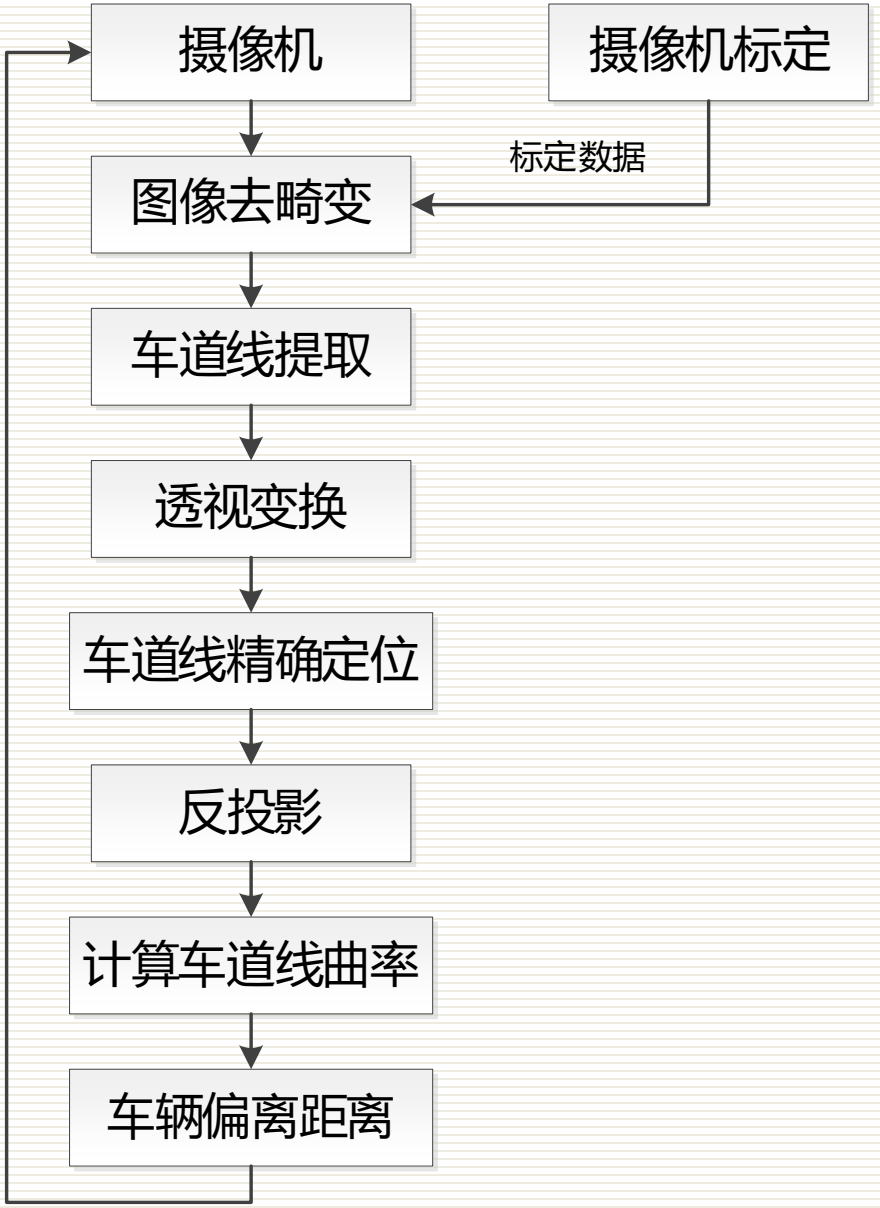
车道线检测

车道线检测


Radius of Curvatur
Vehicle is 0.28m l
left: 739
right: 997
True, 631, 611, 609

车道线检测流程


读取车道线视频/图像
图像去畸变
• 相机校正
• 图像校正
车道线提取
• 边缘检测
• 梯度阈值
• 颜色阈值
透视变换
• 转换成鸟瞰图
精确定位车道线
• 利用直方图确定左右车道线
• 最小二乘拟合车道线
• 绘制车道线多边形
透视变换
• 车道线检测结果
计算曲率及车辆偏离中心距离
结果展示


摄像机
摄像机标定
标定数据
图像去畸变
车道线提取
透视变换
车道线精确定位
反投影
计算车道线曲率
车辆偏离距离

一、阈值分割

（1）直方图阈值法

基于直方图有两个波峰，可通过阈值T把前景从背景里提取出来。

设图像以灰度形式表示为 $f(x,y)$，$x=1,\cdots,m, y=1,\cdots,n$

m和n分别为CCD所采图像的行像素数和列像素数。阈值处理过程为：

$$F(x,y)=\begin{cases} f(x,y), & f(x,y)\geq T \\ 0, & f(x,y)<T \end{cases}$$

阈值100

一、阈值分割

（2）自适应阈值法——最大类间方差法

1979年由日本学者大津于提出的，是一种自适应阈值确定方法，又叫大津法，简称OTSU。

原理：按图像的灰度特性，将图像分成背景和目标两部分。背景和目标之间的**类间方差**越大，说明构成图像两部分的差别越大，当部分目标错分为背景或部分背景错分为目标都会导致两部分差别变小。因此，使类间方差最大的分割意味着错分概率最小。

类间方差公式：$$g=(\overline{x}_1-\overline{x})^2\cdot\frac{n_1}{n}+(\overline{x}_2-\overline{x})^2\cdot\frac{n_2}{n}$$

$\overline{x}$ n个数据平均值

$\overline{x}_1$ 第一组n_1个数据平均值，$\overline{x}_2$ 第二组n_2个数据平均值。

一、阈值分割

算法：

对于图像I(x,y)，图像的大小为M×N，

- 前景(即目标)和背景的分割阈值记作T，
- 属于前景的像素点数占整幅图像的比例记为ω0，其平均灰度μ0；
- 背景像素点数占整幅图像的比例为ω1，其平均灰度为μ1。
- 图像的总平均灰度记为μ，类间方差记为g。
- 图像中像素的灰度值小于阈值T的像素个数记作N0，像素灰度大于阈值T的像素个数记作N1，

$\omega0=N0/M\times N$　　$\omega1=N1/M\times N$　　$N0+N1=M\times N$　　$\omega0+\omega1=1$

$\mu=\omega0*\mu0+\omega1*\mu1$　　$g=\omega0(\mu0-\mu)^2+\omega1(\mu1-\mu)^2$

$g=\omega0\omega1(\mu0-\mu1)^2$

采用遍历的方法得到使类间方差最大的阈值T,即为所求。

一、阈值分割

OTSU程序：

```
float u0, u1, w0, w1; int count0, t, maxT; float devi, maxDevi = 0; //方差及最大方差
int i, sum = 0;
for (i = 0; i < 256; i++){
        sum = sum + hist[i];
}
for (t = 0; t < 255; t++){
        u0 = 0; count0 = 0;
        for (i = 0; i <= t; i++) //阈值为t时，c0组的均值及产生的概率;
        {
                u0 += i * hist[i]; count0 += hist[i];
        }
        u0 = u0 / count0; w0 = (float)count0/sum;
        for (i = t + 1; i < 256; i++) //阈值为t时，c1组的均值及产生的概率
        {
                u1 += i * hist[i];
        }
        u1 = u1 / (sum - count0); w1 = 1 - w0;
        devi = w0 * w1 * (u1 - u0) * (u1 - u0); //两类间方差
        if (devi > maxDevi) //记录最大的方差及最佳位置
        {
                maxDevi = devi;
                maxT = t;
        }
}
```

一、阈值分割

阈值分割效果

原始灰度图

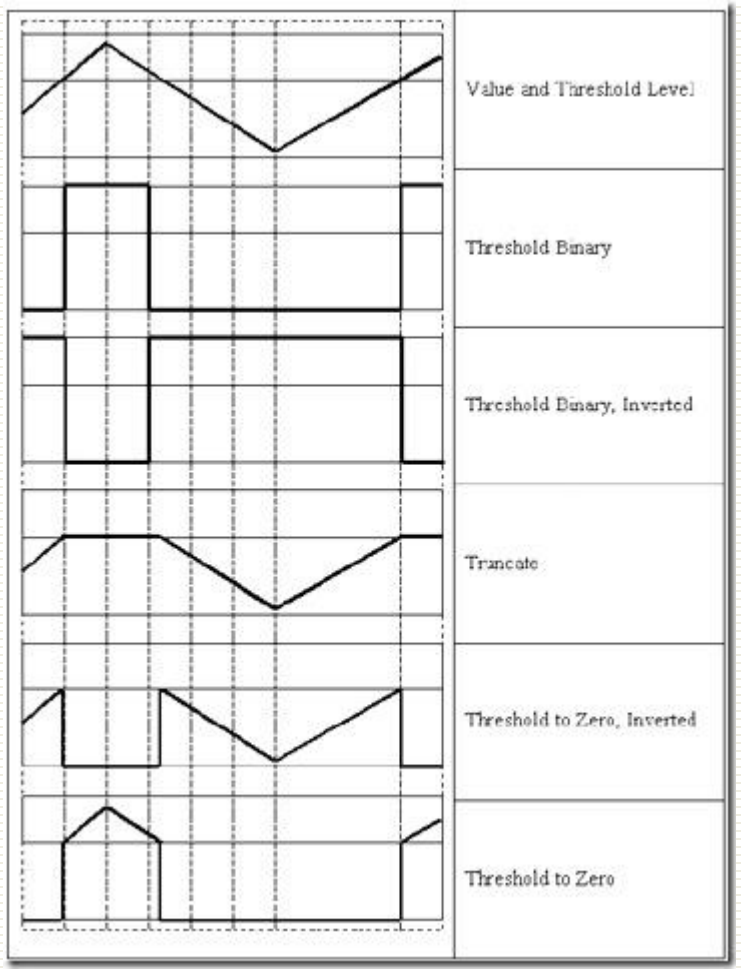
OTSU 阈值117

一、阈值分割

（3）OpenCV程序实现

Double threshold(CvArr* src, CvArr* dst,double threashold, double max_value, int threshold_type)

阈值类型	操作
CV_THRESH_BINARY	dst(x,y) =(src(x,y)>T)?M:0
CV_THRESH_BINARY_INV	dst(x,y) =(src(x,y)>T)?0:M
CV_THRESH_TRUNC	dst(x,y) =(src(x,y)>T)?M:src
CV_THRESH_TOZERO_INV	dst(x,y) =(src(x,y)>T)? 0:src
CV_THRESH_TOZERO	dst(x,y) =(src(x,y)>T)? src:0



CV_THRESH_OTSU

使用大律法OTSU得到的全局自适应阈值来进行二值化图片，而参数中的threshold不再起作用

InRange(InputArray src, InputArray lowerb, InputArray upperb , InputArray dst)

二、颜色特征提取

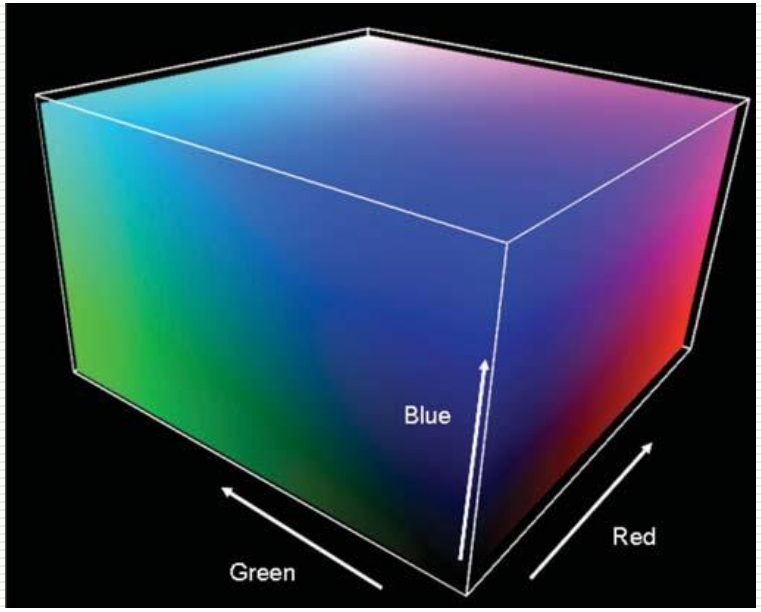
根据图像间的相关信息来检测目标，即在两幅或者多幅不同视点的图像中寻找同一目标的特征。

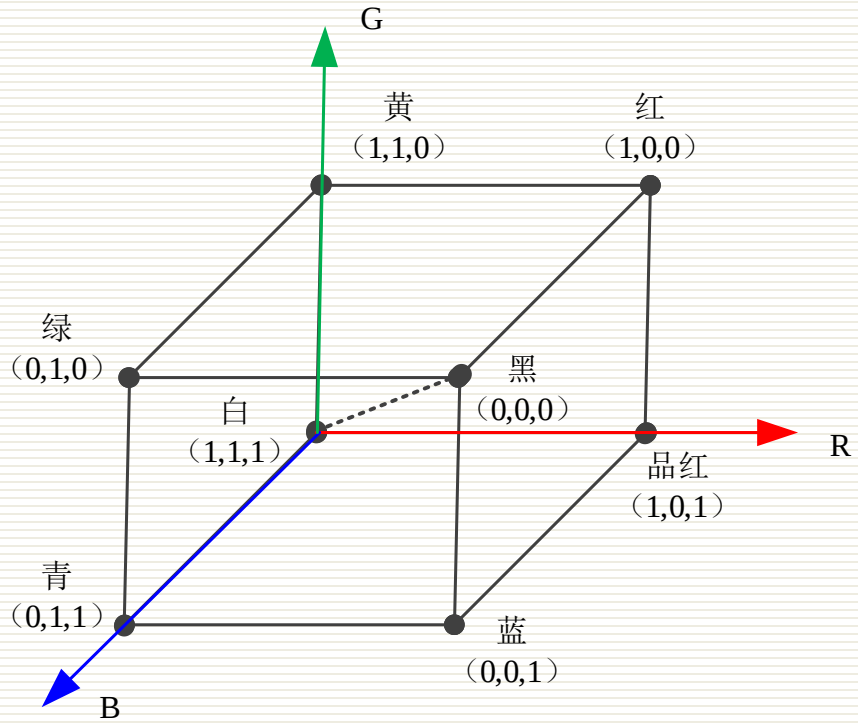
- 颜色特征
  - RGB颜色空间
  - HSV颜色空间
  - HLS颜色空间

二、颜色特征提取

1、RGB颜色空间（三维立方体）

- 任意色光都可以用R、G、B三色不同分量的相加混合而成；
- 当三基色分量都为0（最弱）时混合为黑色光，当三基色分量都为k（最强）时混合为白色光。

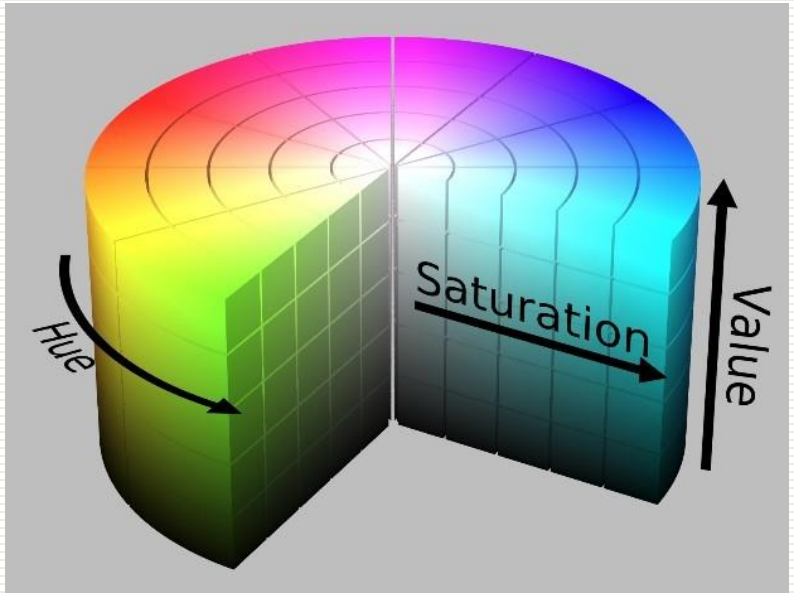


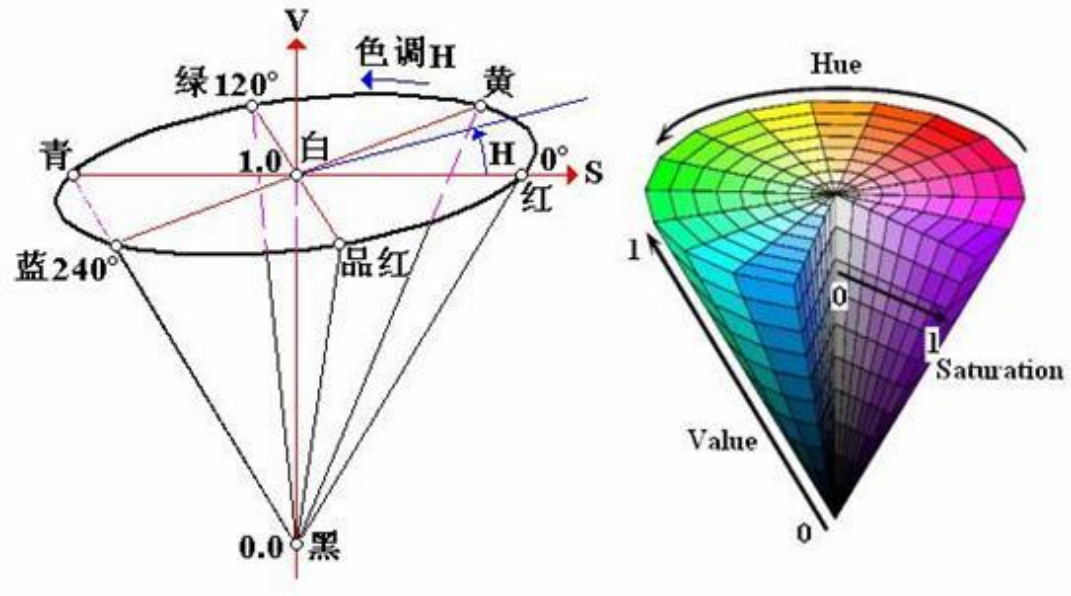


二、颜色特征提取

2、HSV颜色空间（H色调 S饱和度 V色明度）

- 色调H由绕V轴的旋转角给定（0°～360°），红色为0°，绿色为120°,蓝色为240°
- 饱和度S表示颜色接近光谱色的程度，取值从0到1 。一种颜色可看成是某种光谱色与白色混合的结果。饱和度高，颜色则深而艳。
- 色明度V表示色彩的明亮程度，范围从0到1





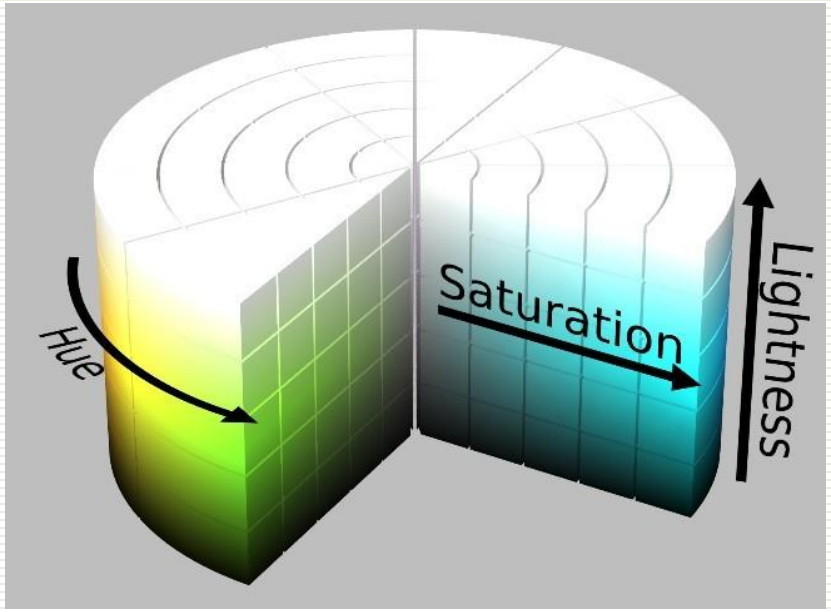
cvtColor(image, hsv, COLOR_BGR2HSV);

cv2.cvtColor(img, cv2.COLOR_RGB2HSV)

二、颜色特征提取

3、HLS颜色空间

- hue（色度）、saturation（饱和度）、lightness（亮度）
- HLS中L分量为亮度，亮度为100%，表示白色，亮度为0，表示黑色；HSV 中的 V 分量为明度，明度为100%，表示光谱色，明度为0，表示黑色。
- 饱和度S取值从0到1，所以圆锥顶面的半径为1。



- HSV中的Hue里没有白色，白色需要由S和V共同决定（S=0, V=100%）。
- HLS 中，白色仅由亮度L一个分量决定。
- 检测白色时使用HLS颜色空间更准确。

cvtColor(image, hsv, COLOR_BGR2HLS);

cv2.cvtColor(img, cv2.COLOR_RGB2HLS)

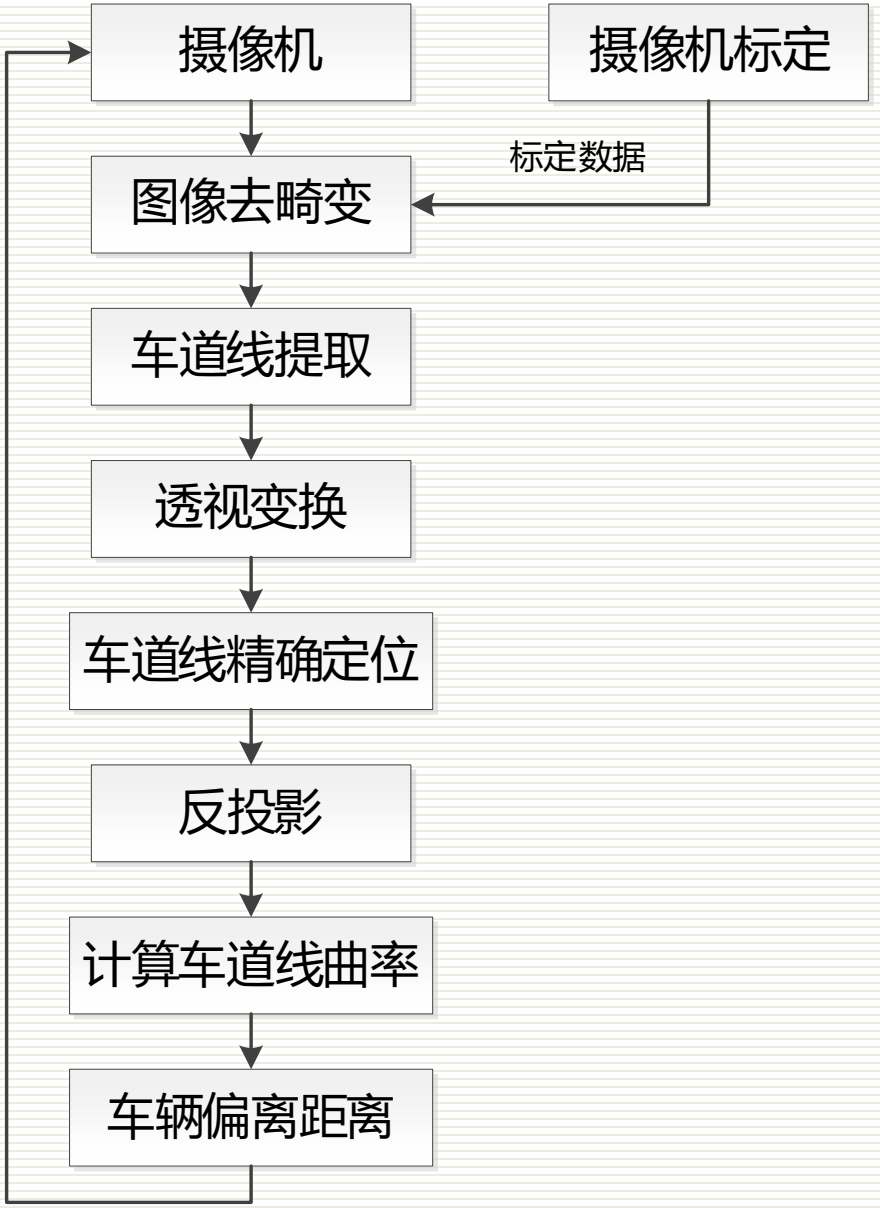
编程作业

针对车道线图像，将RGB图像转为HLS图像，并提取L和S通道，对L通道作sobel边缘检测，对S通道作阈值分割，并将两个通道融合。

s_thresh=(170,255)
sx_thresh=(20, 100)

车道线检测流程

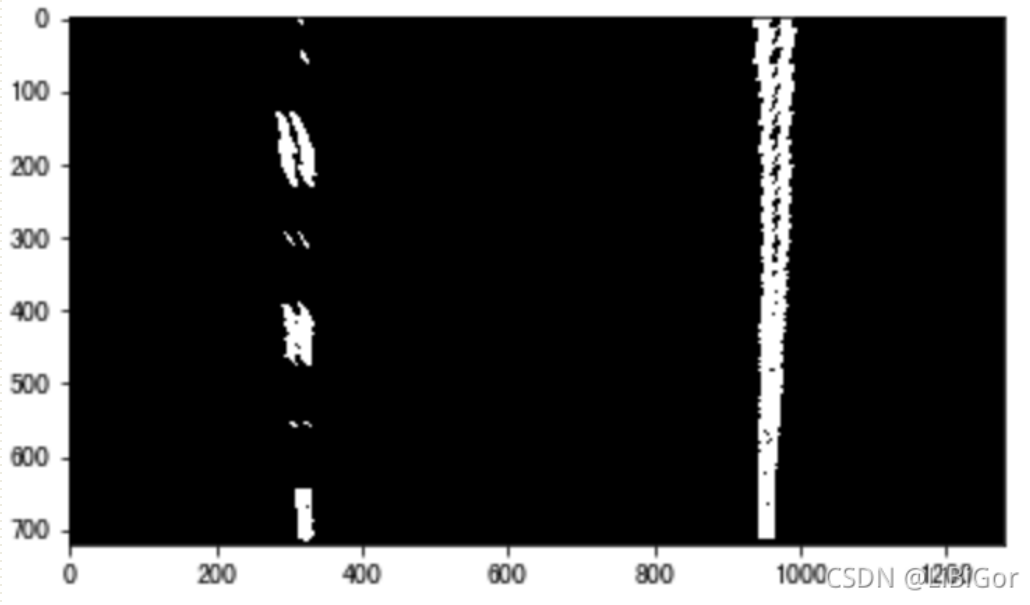

读取车道线视频/图像
图像去畸变
• 相机校正
• 图像校正
车道线提取
• 边缘检测
• 梯度阈值
• 颜色阈值
透视变换
• 转换成鸟瞰图
精确定位车道线
• 利用直方图确定左右车道线
• 最小二乘拟合车道线
• 绘制车道线多边形
透视变换
• 车道线检测结果
计算曲率及车辆偏离中心距离
结果展示

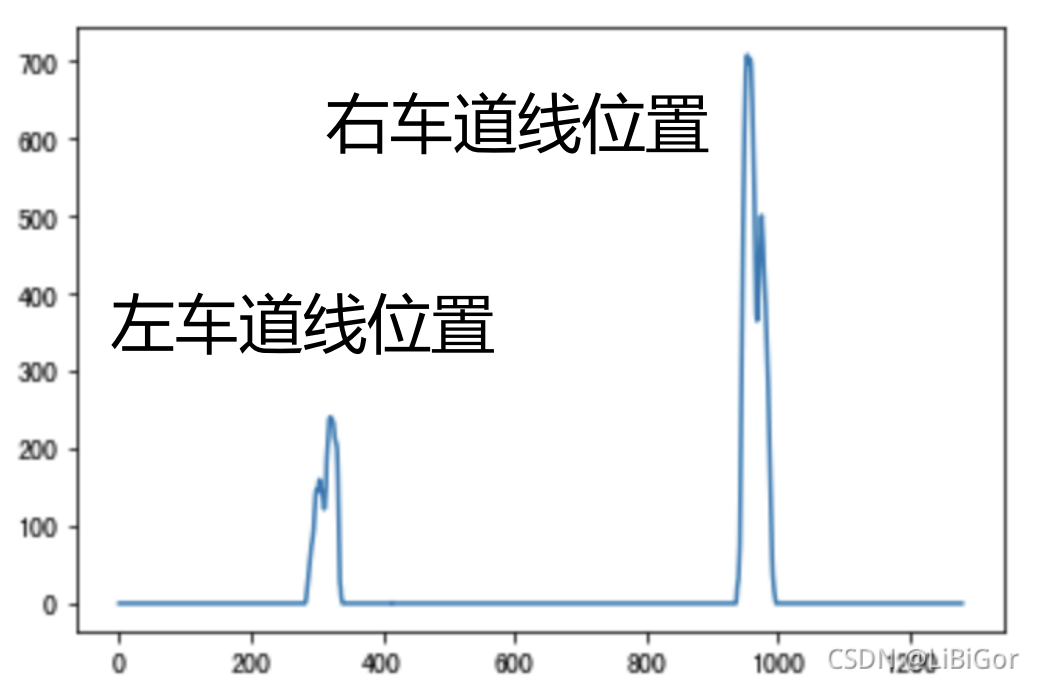

摄像机
摄像机标定
标定数据
图像去畸变
车道线提取
透视变换
车道线精确定位
反投影
计算车道线曲率
车辆偏离距离

三、车道线定位

1、求取图像直方图

沿x轴方向统计每一列中白色像素点的个数，横坐标是图像的列数，纵坐标表示每列中白色点的数量，即为这幅图的“直方图”。





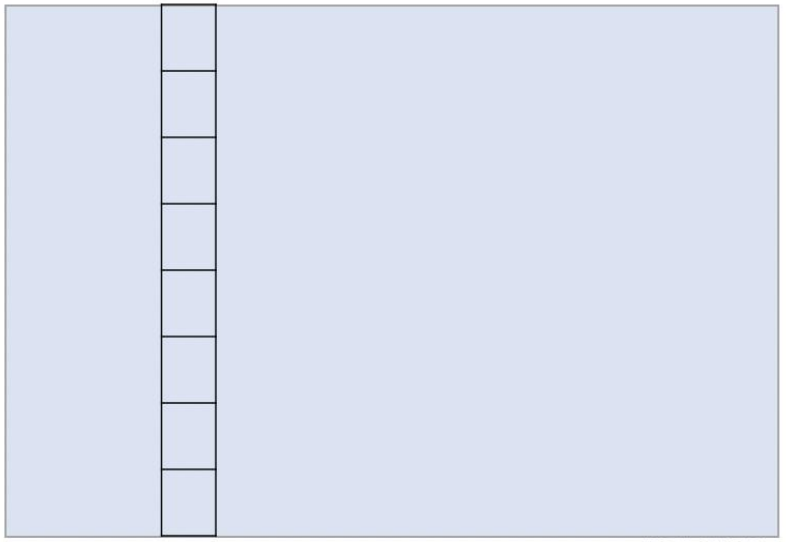
确定左右车道线的大致位置后，使用“滑动窗口”法，在图中对左右车道线的点进行搜索。

三、车道线定位

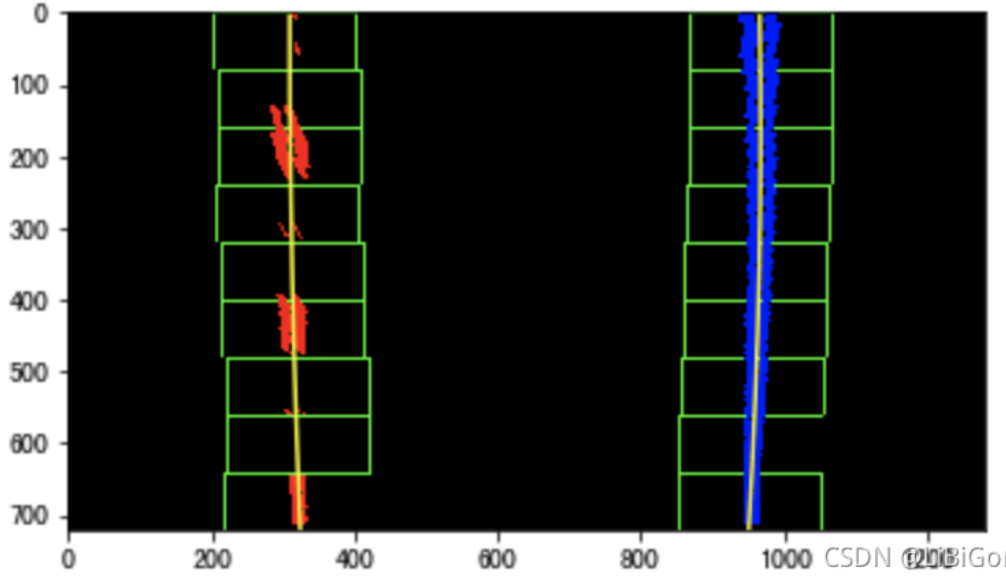
2、滑窗法确定车道线的点

滑动窗口的搜索过程：

- 设置搜索窗口大小（width和height）：一般情况下width为手工设定，height为图片大小除以设置搜索窗口数目计算得到。
- 以搜寻起始点作为当前搜索的基点，并以当前基点为中心，做一个网格化搜索。



- 对每个搜索窗口分别做水平和垂直方向直方图统计，统计在搜索框区域内非零像素个数，并过滤掉非零像素数目小于50的框。
- 计算非零像素坐标的均值作为当前搜索框的中心，并对这些中心点做二阶的多项式拟合，得到当前搜寻对应的车道线曲线参数。



计算零阶距： $M_{00} = \sum_x \sum_y I(x,y)$

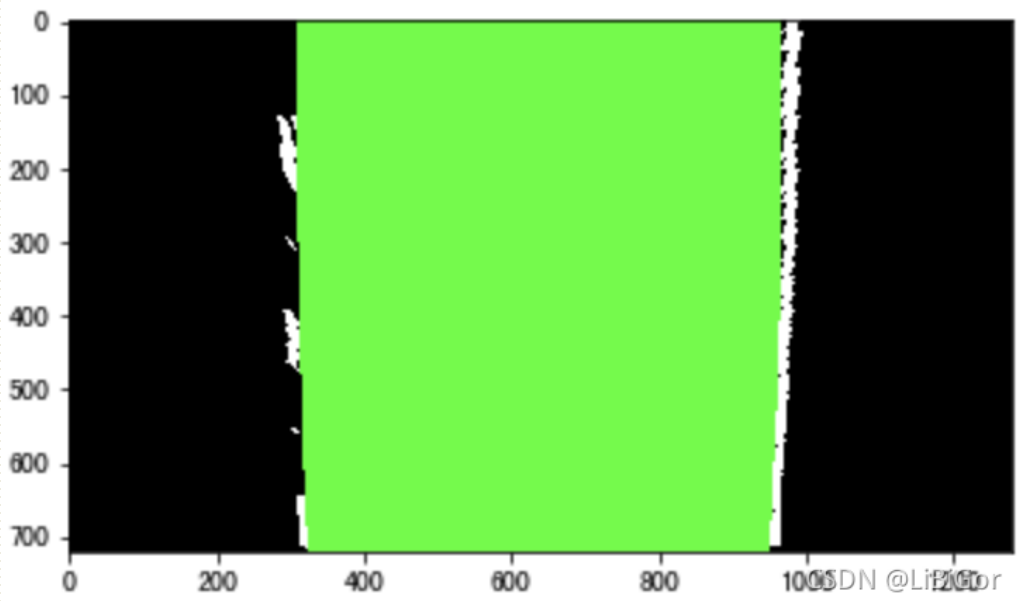
计算一阶距： $M_{10} = \sum_x \sum_y xI(x,y)$ $M_{01} = \sum_x \sum_y yI(x,y)$

计算搜索窗的质心： $x_c = M_{10}/M_{00}$ $y_c = M_{01}/M_{00}$

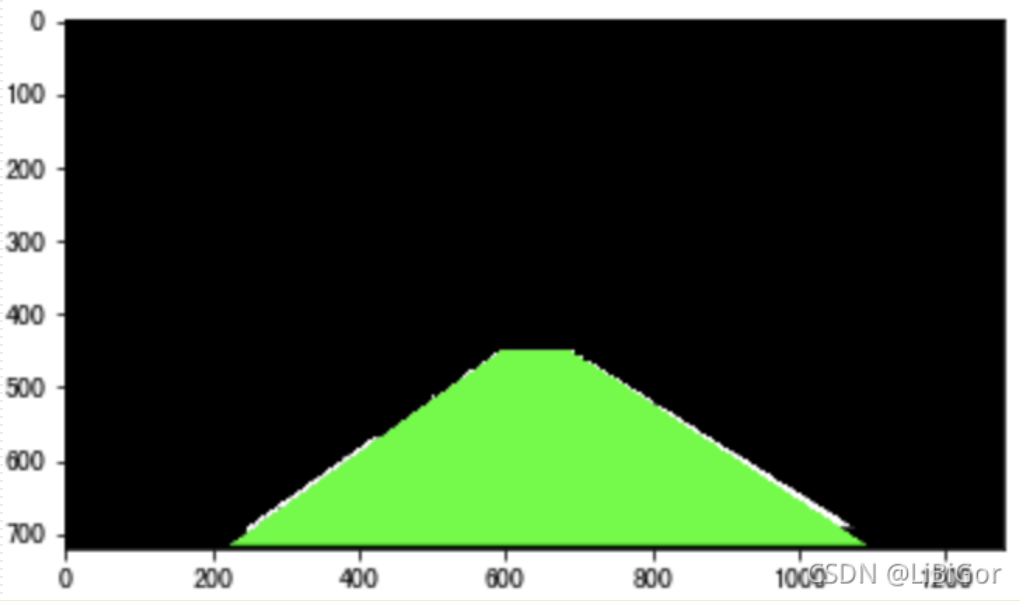
三、车道线定位

3、确定车道线区域

将车道区域绘制处理，即在检测出的车道线中间绘制多边形

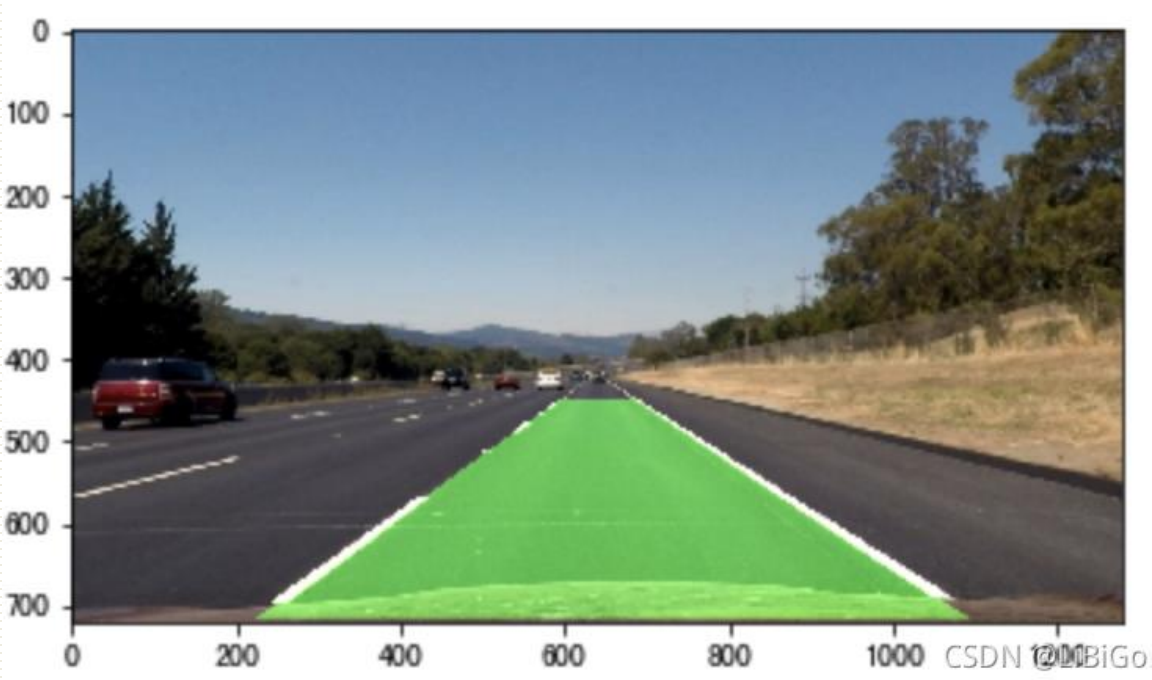


将检测出的车道逆投影到原始图像，直接调用透视变换的方法



三、车道线定位

4、在原图投影



三、车道线定位

车道线精确定位思想：

首先利用直方图的方法确定左右车道线的位置，然后利用滑动窗口的方法，搜索车道线位置，并进行多项式拟合，然后绘制车道区域，并进行反投影，得到原始图像中的车道区域。

直线提取思想：

Hough 变换(Hough Transform)是一种常用的检测图形的算法。

通过搜索特定形状(如直线,圆,椭圆等)在参数空间的累加器中的局部最大值来检测形状。Hough 变换主要用于检测图像中的基本形状,如直线,圆等。

三、车道线定位

霍夫变换（Hough Transform）换于1962年由Paul Hough 首次提出，后于1972年由Richard Duda和Peter Hart推广使用，是图像处理中从图像中检测几何形状的基本方法之一。

经典霍夫变换用来检测图像中的直线，后来霍夫变换经过扩展可以进行任意形状物体的识别，例如圆和椭圆。

Hough变换基本原理：

Hough直线检测的基本原理在于利用点与线的对偶性，在直线检测任务中，即图像空间中的直线与参数空间中的点是一一对应的，参数空间中的直线与图像空间中的点也是一一对应的。

1）图像空间中的每条直线在参数空间中都对应着单独一个点来表示；

2）图像空间中的直线上任何一部分线段在参数空间对应的是同一个点。

Hough直线检测算法就是把在图像空间中的直线检测问题转换到参数空间中对点的检测问题，通过在参数空间里寻找峰值来完成直线检测任务。

三、车道线定位

图像处理中Hough变换主要步骤：

1、边缘检测：检测出图像中的曲线特征点(局部最大,局部最小,递增点等)。

2、参数空间生成：对每条可能的曲线,确定其在参数空间中的坐标。

3、投票：对每一个特征点,判断出其对应的参数空间中的所有可能的曲线,并对相应的参数空间坐标进行投票(累加)。

4、检测曲线：在参数空间中检测出投票数较大的局部最大值点,这些点对应的参数值即为图像中存在的曲线的参数。

5、群聚分析：连接符合要检测的曲线定义的特征点，形成较完整的曲线。

Hough 变换通过在参数空间中的投票统计,实现了对图像中基本形状的检测,它广泛应用于图像处理和计算机视觉中曲线检测的问题。

三、车道线定位

1、对偶性

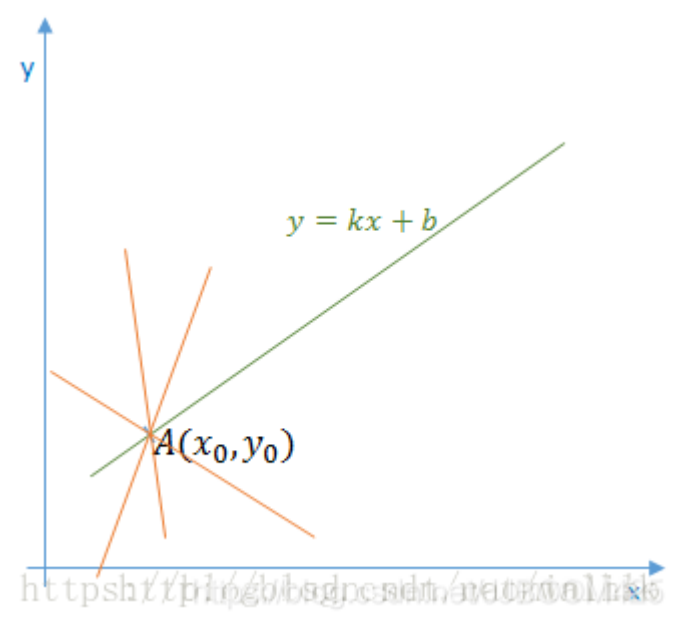
（1）图像空间中的点与参数空间中的直线一一对应

图像空间x-y中一条直线在直角坐标系下可以表示为y=k*x+b,其中k和b是参数，对应表示斜率和截距。

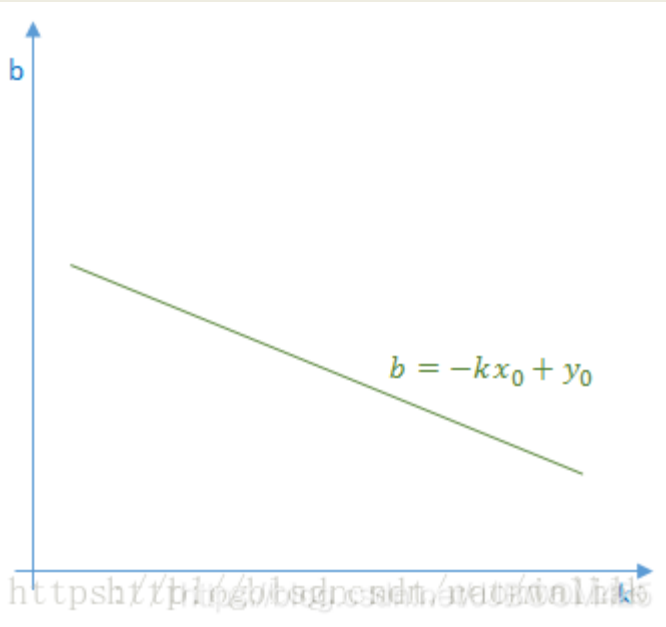
过某一点A(x0, y0)的所有直线的参数均满足方程y0=k*x0+b，即点A(x0, y0)确定了一族直线。

可将方程改写为： b=-k*x0+y0

那么该方程在参数空间k-b中就对应了一条直线：



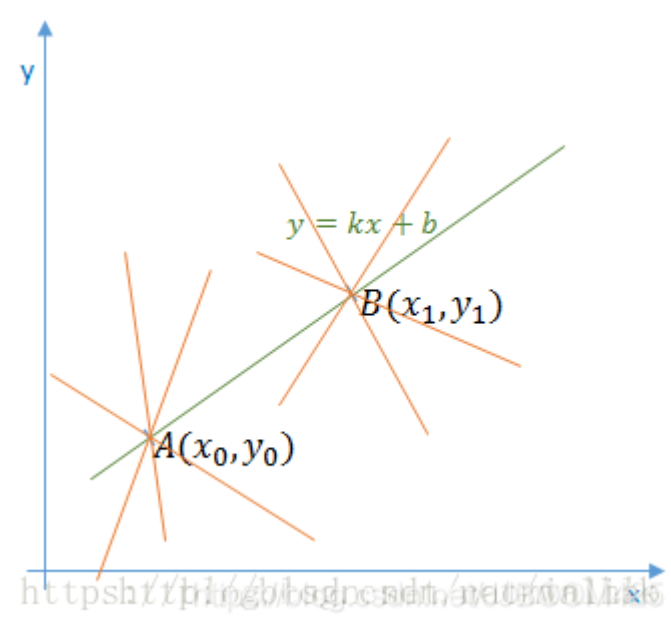
图像空间x-y中的点(x0,y0)对应了参数空间k-b中的直线b=-k*x0+y0。即图像空间中的点与参数空间中的直线一一对应。



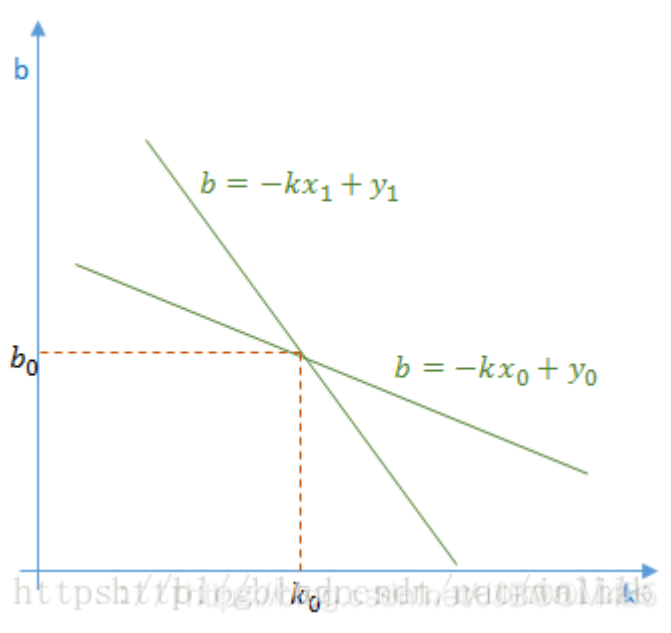
三、车道线定位

（2）图像空间中的直线与参数空间中的点一一对应

在直线y=k*x+b上再增加一个点B(x1, y1)　　点B(x1, y1)在参数空间同样对应了一条直线



图像空间x-y中的点A和点B在参数空间k-b中对应的直线相交于一点，也就是说AB所确定的直线，在参数空间中对应着唯一一个点，这个点的坐标值（k0, b0）也就是直线AB的参数

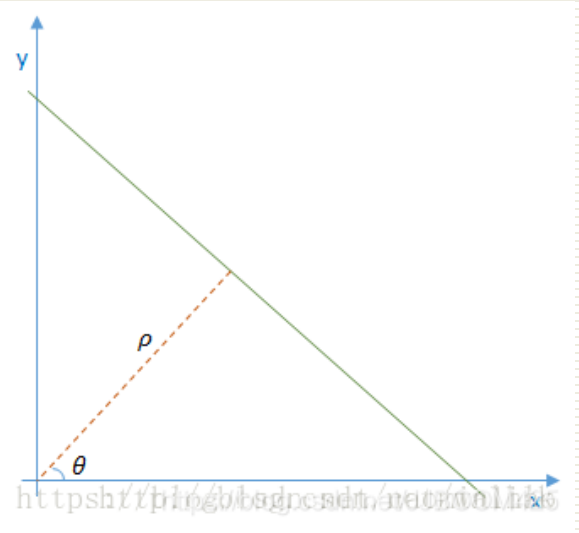


三、车道线定位

2、参数空间的选择

在实际应用中，参数空间采用极坐标系ρ-θ

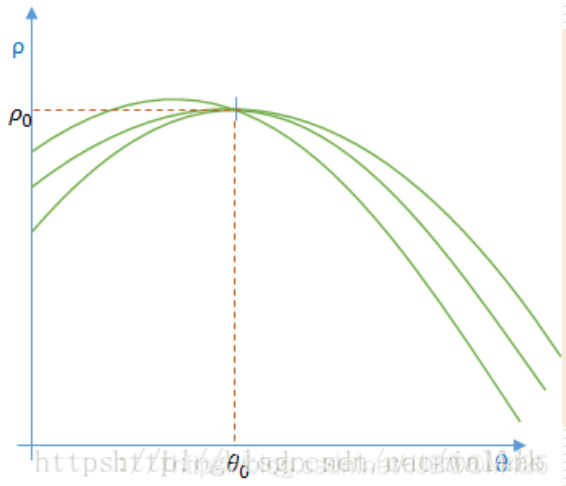
直线表达式为 $y=\left(-\frac{cos\theta}{sin\theta}\right)x+\left(\frac{\rho}{sin\theta}\right)$ 化简可得： $\rho=xcos\theta+ysin\theta$



对于直线上的点(x0, y0)，可以将通过该点的直线族定义为

$$\rho=x_0cos\theta+y_0sin\theta$$

参数空间的每个点(ρ,θ)都对应了图像空间的一条直线。参数空间采用极坐标系，这样就可以在参数空间表示图像空间中的所有直线。



图像空间（直角坐标系x-y）上的一条直线对应到参数空间（极坐标系ρ-θ）上是一个点

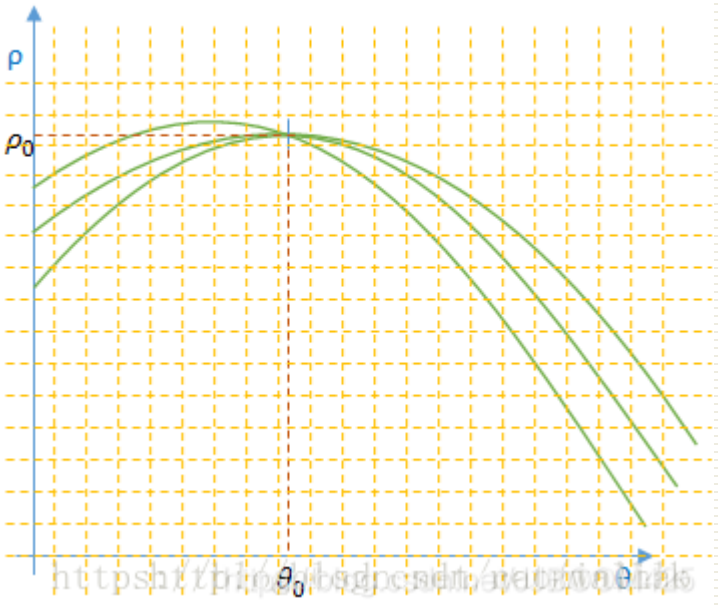
三、车道线定位

3、利用霍夫变换检测直线

霍夫直线检测就是把图像空间中的直线变换到参数空间中的点，通过统计特性来解决检测问题。

具体来说，如果一幅图像中的像素构成一条直线，那么这些像素坐标值（x, y）在参数空间对应的曲线一定相交于一个点，所以只需要将图像中的所有像素点（坐标值）变换成参数空间的曲线，并在参数空间检测曲线交点就可以确定直线了。

- 将直线的方向θ离散化为有限个等间距的离散值，参数ρ也就对应离散化为有限个值，于是参数空间不再是连续的，而是被离散量化为一个个等大小网格单元。
- 将图像空间（直角坐标系）中每个像素点坐标值变换到参数空间（极坐标系）后，所得值会落在某个网格内，使该网格单元的累加计数器加1。
- 当图像空间中所有的像素都经过霍夫变换后，对网格单元进行检查，累加计数值最大的网格，其坐标值（ρ0, θ0）就对应图像空间中所求的直线。



三、车道线定位

3、利用霍夫变换检测直线

优点：Hough直线检测的优点是抗干扰能力强，对图像中直线的残缺部分、噪声以及其它共存的非直线结构不敏感。

缺点：Hough变换算法的特点导致其时间复杂度和空间复杂度都很高，并且在检测过程中只能确定直线方向，丢失了线段的长度信息。

```
lines = cv2.HoughLines(image, rho, theta, threshold)
image: 单通道的二进制图像
rho: (ρ，θ)中ρ的精度。
theta: (ρ，θ)中θ的精度。
threshold: 阈值，(ρ，θ)对应的最低投票数。>=threshold被检测为一条线。
```

```
vector<Vec2f>lines;
HoughLines(image,lines，rho, theta, threshold)
```

编程作业

1、针对车道线图像，将RGB图像转为HLS图像，并提取L和S通道，对L通道作sobel边缘检测，对S通道作阈值分割，并将两个通道融合。

2、针对预处理后的车道线图像，进行透视变换，统计列直方图，提取车道线位置，通过滑窗确定车道线的像素值，并进行多项式拟合，绘制出车道区域并逆投影到原始图像中。

3、针对车道线图像，进行边缘检测，通hough变换检测出车道线，并显示。