# 直道车道线检测

## 主要代码

Camera.py 中实现了不断拍照以及点到直线距离的函数

Detect.py 中实现了输入一张图，返回检测到的两条车道线，左右车道线的区分暂时在camera.py中按照距离简单实现

Trans.py 中实现了对一张图做透视变换，被detect.py调用

Kmeans.py 中实现了聚类的k-means算法，被detect.py调用

每个代码可以在没被调用时独立运行，运行可查看在测试图片 5.jpg上的效果

## Detect的主要流程

参考<https://zhuanlan.zhihu.com/p/25354571?utm_source=weibo&utm_medium=social>

* 灰度化
* 高斯模糊
* 边缘检测
* 截取roi
* Hough变换获取直线，暂定以极坐标形式表示直线方程
* 对所得直线做聚类成两类，分别求平均得两条车道线

## 需要调参的地方

主要的需要调参的地方代码中都以形如

#-------------------------------------------

blur\_ksize = 19 # Gaussian blur kernel size

#-------------------------------------------

这样标出

在 trans.py 中

透视变换的四个标定点的坐标以及标定点变换后应在的坐标，顺序为(left\_bottom,right\_bottom,left\_top,right\_top)，坐标以左上角为原点，向右向下为正方向

参考 https://blog.csdn.net/guduruyu/article/details/72518340

在detect.py中

* 高斯模糊的kernel\_size，确定每个点做高斯模糊时高斯分布的大小，暂定为19，表现还不错，能有效的把赛道布上的褶皱在边缘检测中不被检测到
* Roi的区域，roi的目的是为了仅仅关注于一部分重要的区域，如中央，而把其他如四周的图像给去掉，四个顶点的顺序不同于上一项，是(left\_bottom,left\_top,right\_top,right\_bottom)
* 霍夫变换求直线的threshold表示霍夫空间下多少条直线（对应原图中多少个点）可以确定一条直线的存在，暂定为20，表现还不错