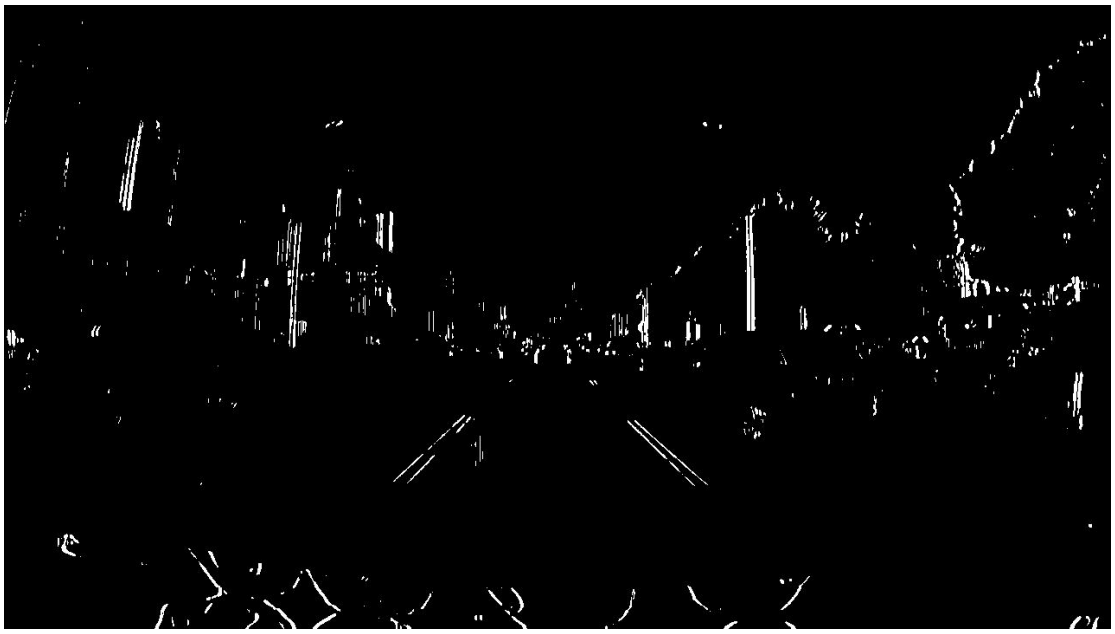
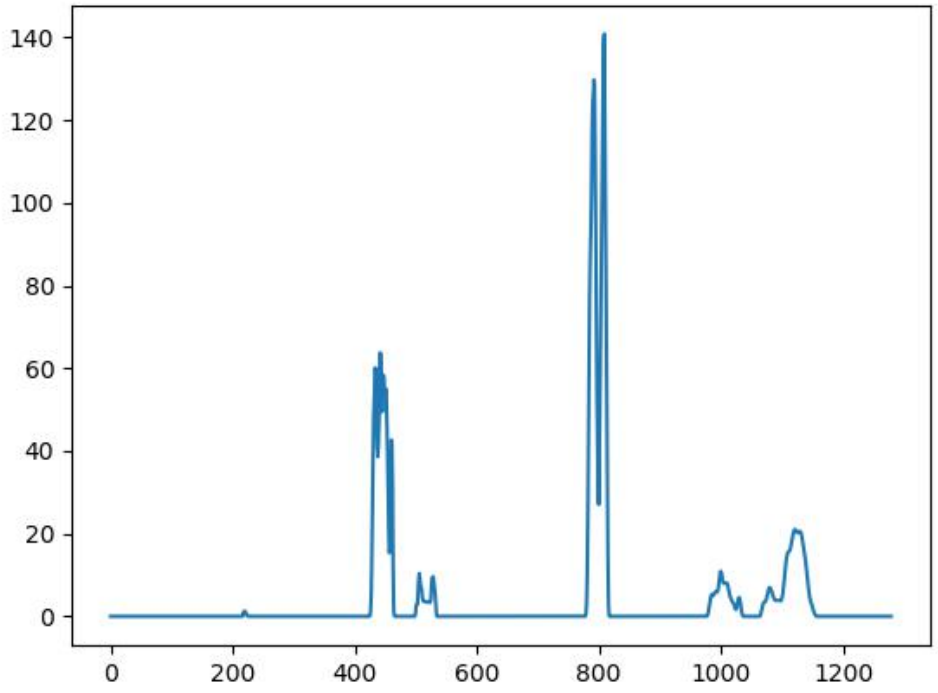
第一章 传统车道线检测

1. 预备知识
   (1) 车道线检测的框架
      ① 传感器数据（单目或者双目相机）
      ② 数据去噪，增强
      ③ 车道线提取（传统的图像边缘提取， 以机器学习，深度学习）
      ④ 车道线拟合
      ⑤ 车道线时间维度上的追踪与融合
   (2) 编程基础
      ① 系统：Linux， 例如 Ubuntu，Debian 等等
      ② 机器人操作系统 ROS
      ③ C++
      ④ Python：numpy
      ⑤ 第三方库例如 opencv
   (3) 数学知识：
      ① 基础的矩阵运算
      ② 基础的高等数学，例如导数，微分等
2. 车道线检测代码说明
   整个处理流程函数为 process_image
   (1) 图片载入
   (2) BGR 图片转换为 HSV 通道图片，并取出 V 通道进行后续处理
   (3) 边缘提取（作业）：canny 算子，sobel 算子等
   (4) 将前视视角下的边缘图片通过透视变换转换成俯视视角下的边缘图片，方便后续车道线的检测和提取



(5) 在俯视视角下的边缘图片中，对每一列像素值累加，可以发现会出现两个波峰值，分别为左边和右边的车道线，取出这两个波峰对应的像素横坐标 u



(6) 跟上步骤 5 中提取的两个像素横坐标，分别在附近搜素可能为车道线的像素坐标

(7) 将上述找到的像素坐标再通过透视逆变换转化成前视视角

(8) 分别对前视视角下的左右车道线进行三次多项式拟合（作业）

$y = c_0 + c_1 x + c_2 x^2 + c_3 x^3$，写成矩阵形式为

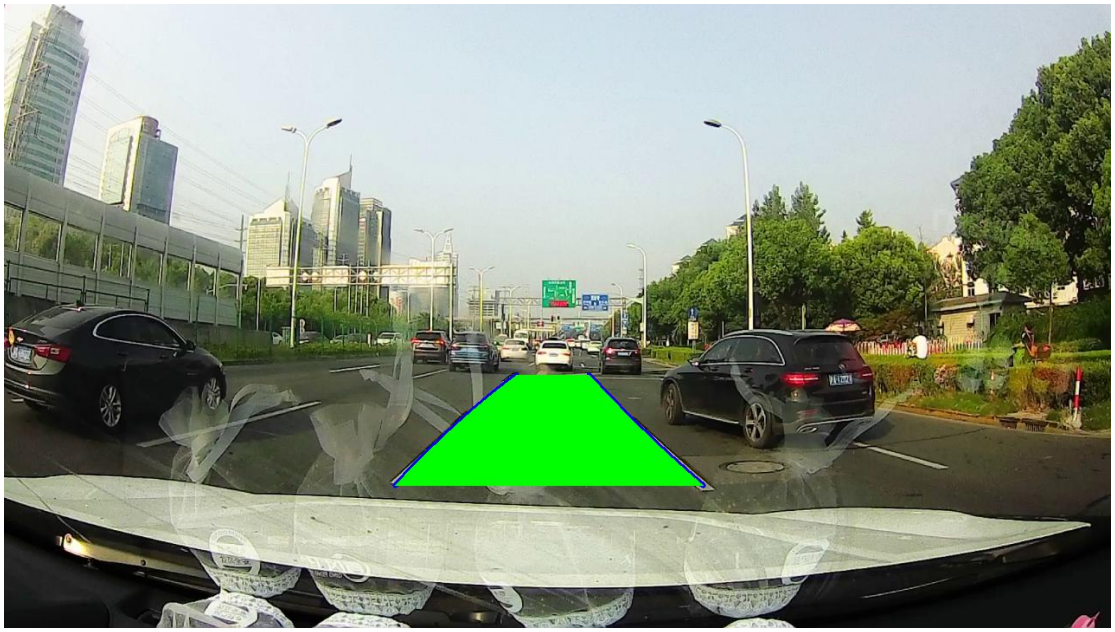
$y=[1,x,x^2,x^3]\begin{bmatrix}c_0\\c_1\\c_2\\c_3\end{bmatrix}$，对于多组观测数据，我们可以写成

$$\begin{bmatrix}y_1\\y_2\\\cdots\\y_n\end{bmatrix}=\begin{bmatrix}1&x_1&x_1^2&x_1^3\\1&x_2&x_2^2&x_2^3\\\cdots&\cdots&\cdots&\cdots\\1&x_n&x_n^2&x_n^3\end{bmatrix}\begin{bmatrix}c_0\\c_1\\c_2\\c_3\end{bmatrix}$$

，记包含 y 的向量为 b，记包含 x 幂的矩阵为系数矩阵 A， 记待求解的 c 向量为 x，则原方程变为 $b=Ax$. 该方程的最小二乘解为

$x=(A^TA)^{-1}A^Tb$

(9) 可视化车道线以及可行驶区域



3. 作业

(1) 编程实现：

① 边缘提取，编程实现 detect_edges 函数，函数输出为一个二值图，其中 1 表示提取到的边缘，0 表示背景

② 多项式拟合，编程实现 PolynomialCurve 类中 fit 函数，函数输出拟合的三次多项式的各个系数

(2) 证明：

对于超静定线性方程组 $Ax=b$，其最小二乘解

$x=(A^TA)^{-1}A^Tb$，为原方程的最优解