[[[288 318 288 158]]

[[140 333 181 242]]

[[142 333 197 211]]

[[284 318 284 249]]

[[348 81 415 136]]

[[355 168 376 221]]

[[ 7 338 66 266]]

[[ 10 338 71 262]]

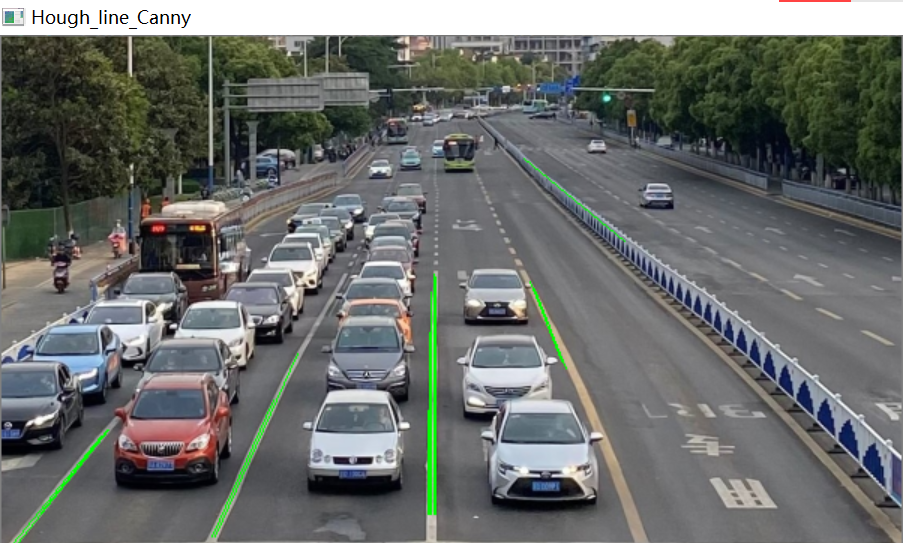
[[352 163 373 216]]

[[286 318 286 170]]

[[285 318 285 197]]

[[289 318 289 160]]

[[ 26 317 70 262]]]



(627, 1109, 3)

width: 627

height: 1109

思路：

1. 求出每一条线的方程，同一个车道线进行算数平均，得到车道方程
2. 根据斜率讲排序
3. 两两相邻斜率的直线求中线
4. 中线像素点的灰度值连续不变（变化小于某一阈值）判定为没有车

检测思路：

1. 获取中线方程
2. 随着y值减小，输出灰度值（二值图）到一个一维数组
3. 如果连续n个为0、255就判定此处无车，那么这一串中y值最大的纵坐标为车尾

检测起点：

1. 从图像宽度的2/3处开始检测，因为有车不往前停

将中线上的灰度按顺序（339->0 即：排头->排尾）赋值给一个一维数组

检测到连续黑色为没有车：

按顺序对每个0开始检测，如果连续n个点为0，那么这个0为队尾

然后返回这个0所在的纵坐标，就是拥塞长度

1. 先转化像素点代表长度，再转化为检测长度