1. 车道线检测

智能场景中需要实时对车道线进行检测，以实现车道线保持功能。车道线检测的目标在于实时确定前方车道线中间可行驶区域范围，同时返回车辆离车道中心线的偏移距离。其主要步骤如下：

（1） 利用 Opencv 确定摄像头内参矩阵和畸变矩阵。

（2） 确定透视变换矩阵：其主要流程图如图 1 所示，首先进行图像矫正，选出场景中需要关注的区域范围（图 2），然后进行 Scanny 边缘检测，勾勒出此区域范围内的边缘轮廓（图 3），利用霍夫变换找出图 3 中所有直线的交点坐标，即消失点坐标。最后利用消失点坐标选出需要透视变换的包含车道线的梯形区域，根据目标图像尺寸 400x500 计算透视变换矩阵。

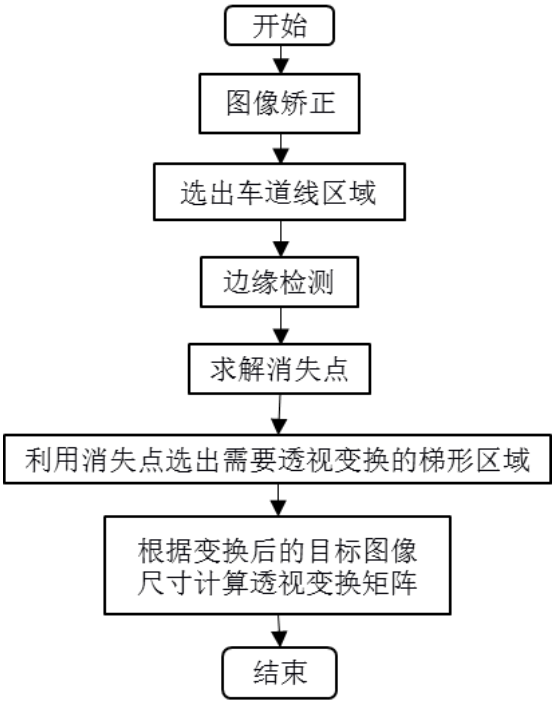


图 1 确定透视变换矩阵流程图

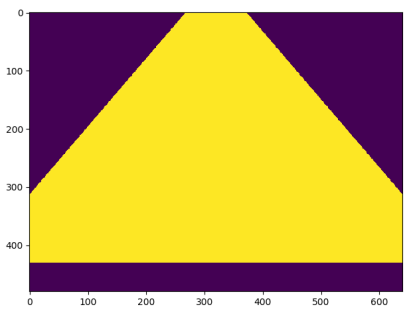


图 2 关注区域

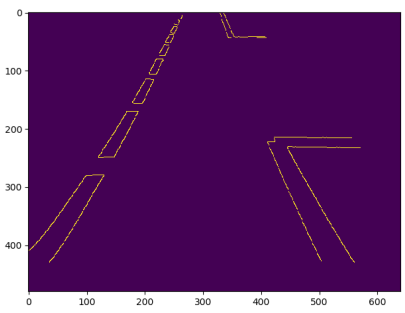


图 3 Scanny 边缘检测结果

（3） 确定像素比例尺:求得透视变换矩阵之后，还需要根据车道线的实际宽度求解摄像头返回图像像素象征实际场景中的尺寸，用于视觉定位。具体流程如图 4 所示，将矫正后的图像进行透视变换得到图 5，再进行二值化处理，可得到两条车道线的实际位置(图 6)，把两条车道线像素的质心横向的像素距离，除以其车道线的实际宽度，便可得到图像宽度方向的像素比例尺(像素/米)，再根据摄像头内参矩阵求解高度方向上的像素比例尺。

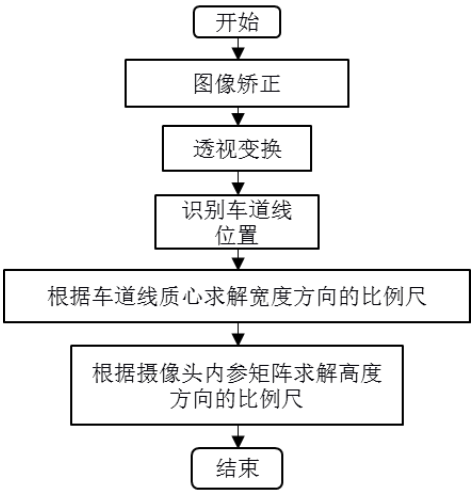


图 4 确定像素比例尺

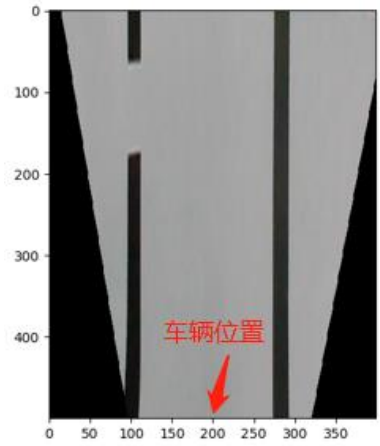


图 5鸟瞰图

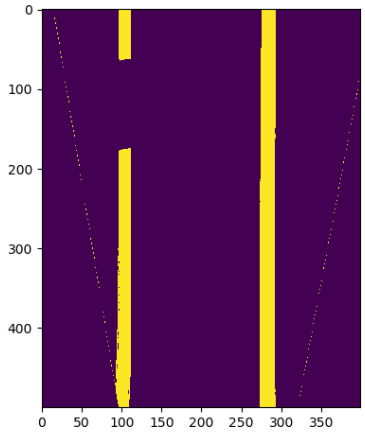


图 6 二值化

（4） 车道线识别：上述步骤确定好摄像头的内参矩阵、畸变矩阵、透视变换矩阵、像素比例尺后，便可进行车道线识别，识别流程图如图 7 所示，前期处理与刚才求解像素比例尺基本一致，在识别车道线位置方面，需要逐步框出车道线的像素点（图 8），然后根据框出的像素点进行二项式拟合（图 9），根据像素点最多侧车道线的拟合曲线得出两侧最终车道的鸟瞰图（图 10），再将鸟瞰图中车道线之间的可行驶区域进行反透视变换，便可得到车道线识别的最终图像结果（图 11），此外，车辆一直位于鸟瞰图中的(200,0)像素点，根据拟合的二项式，可以预估车辆正前方十厘米处车道线中间的位置，进而得出车辆与此位置的横向偏移距离用于进行车道线保持控制。

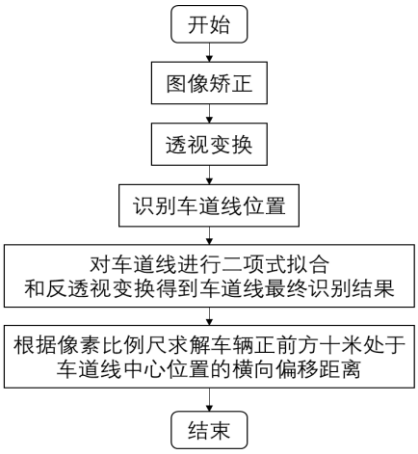


图 7 车道线识别

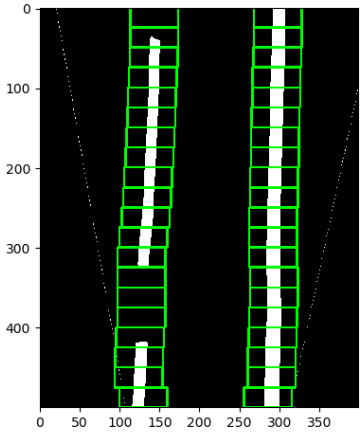


图 8 车道线

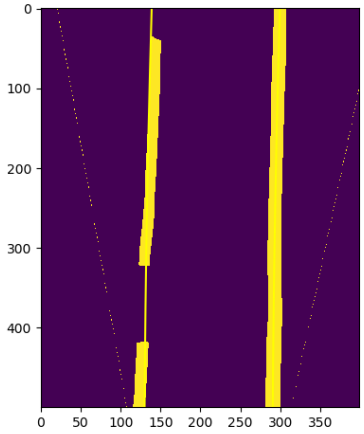


图 9 二项式拟合

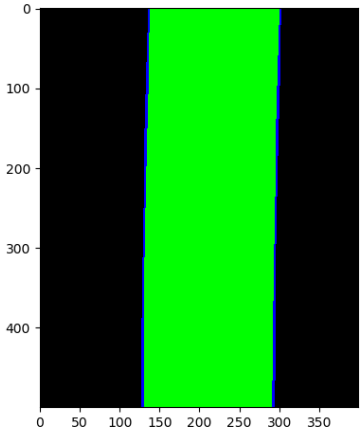


图 10 绿色填充

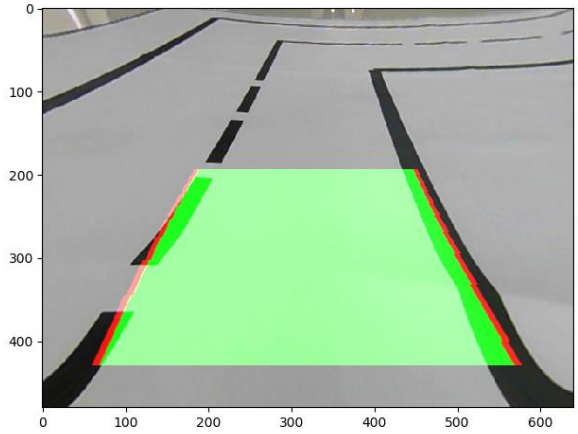
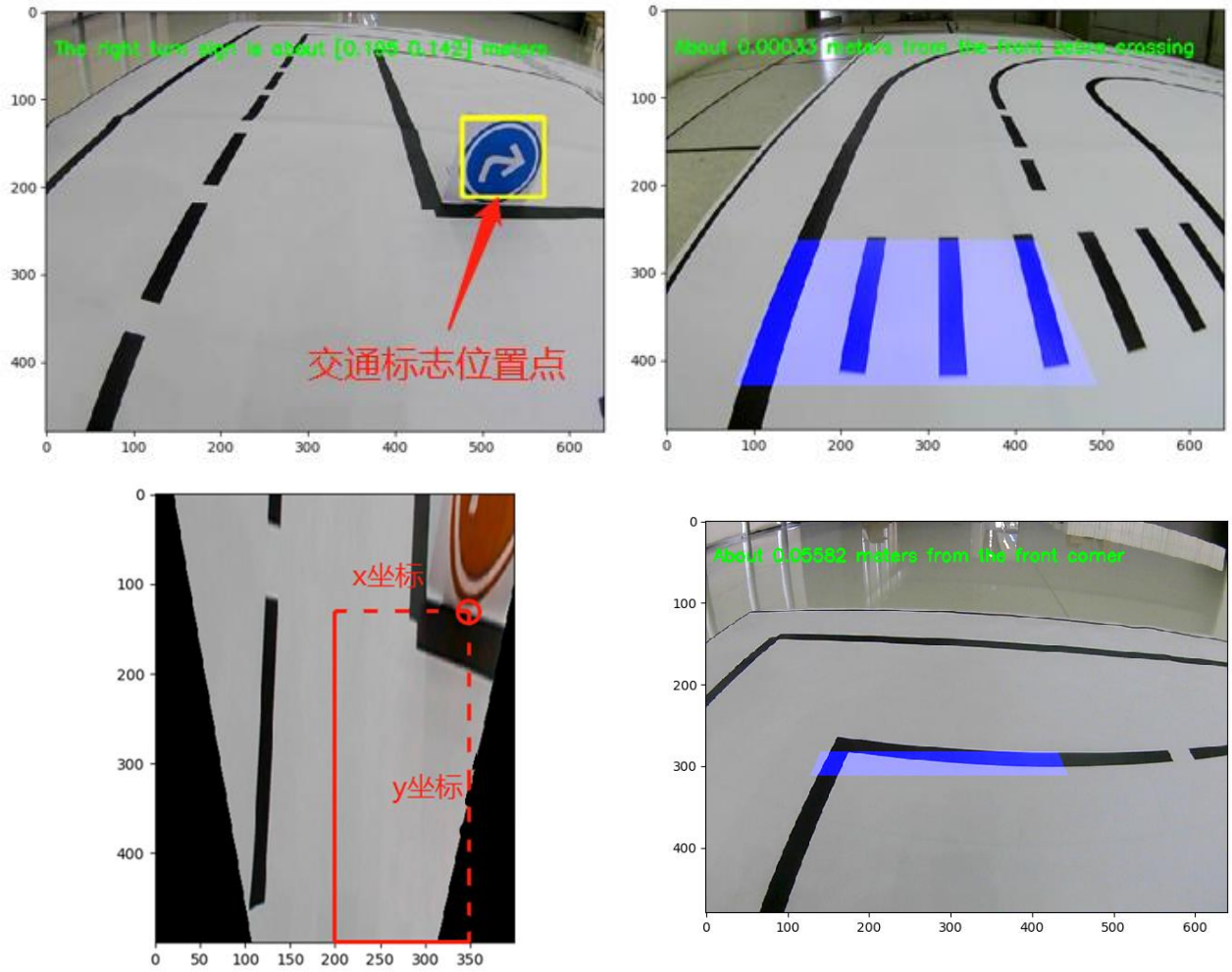


图 11 绿色填充反透视变换(利用单侧车道线进行车道区域识别)

2. 交通标志识别及单目摄像头视觉定位

本小组基于目标检测算法进行交通标志识别，先采集智能车道路图像信息，运用 Tensorflow Oject Dextection API 对 SSD 模型进行深度学习，并根据测试集结果，提取难例集进行强化学习，最终得到目标检测模型。此外，交通标志和斑马线识别则是基于传统方法，根据不同标志像素点特点进行提取。在俯视图中，根据鸟瞰图像素矩阵，利用每一横行的斑马线黑色像素点与直角弯处黑色像素点个数不同的特点来进行目标检测，并进行绘图。



在视觉定位时，需要确定识别目标像素位置。对于交通标志，我们讲交通标志识别框，底边线段中点像素坐标作为交通标志的位置点，进行透视变换得到俯视图中，交通标志的像素位置，再根据俯视图的像素比例尺，计算该点到俯视图最底端中点像素点的距离，进而得到交通标志相对于智能车的前方以及侧方的偏移距离，斑马线视觉定位类似道理。

内容	代码
车道线检测、交通标志识别、单目摄像头视觉定位	lane_detec_main.py