交通灯识别

本模块的主要功能为获取视频数据、准确识别视频中Aruco标签的位置，并以此为基础获取交通灯距小车的相对位置以及交通灯颜色，工作流程如图1-1所示：

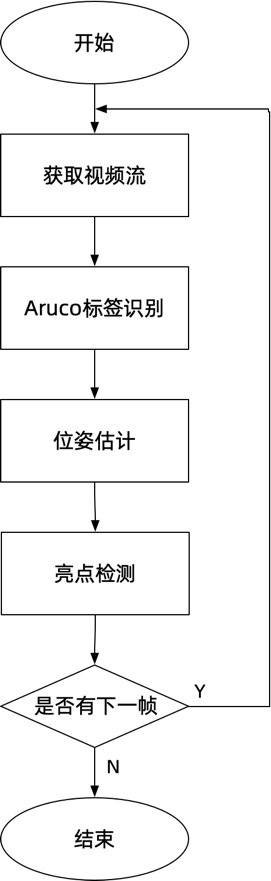


图1-1 工作流程图

本模块基于Ubuntu 18.04,OpenCV 4.0.0,使用C++语言以及Cmake工具进行工程构建，基于OpenCV和Eigen库进行二次开发，封装了Vision类。以下分四部分对技术细节进行介绍：

1. 获取视频流

模块的输入为来自摄像头或视频文件的视频流，使用VideoCapture类读取，之后通过流操作读取每一帧的图像，存储至缓冲区Frame中，送入识别Aruco标签的函数进行下一步处理。

1. Aruco标签识别

相机姿态估计在许多计算机视觉应用中极为重要，例如机器人导航、增强现实等方面，该过程基于查找真实环境中的点及其二维图像下的投影之间的关系，由于此过程难度较大，因此通常使用合成或基准标记进行定位。

Aruco是由宽黑色边框和确定其ID的内部二进制矩阵组成的方形标签，这些标签的主要优点在于单个标签就能够提供足够的位置信息以估计相机姿态。此外，其黑色的边框有利于提高检测的实时性，内部采用二进制编码，提高了检测的鲁棒性。

在本部分中，使用OpenCV contrib库中的检测函数detectMarkers对单个Aruco标签进行检测，即可得到其代表的数据及四个顶点在图像中的位置，将其传入位姿估计函数进行处理。

1. 位姿估计

在接收到来自上一部分的数据以及顶点位置后，即可根据预先标定得到的相机内参对其位姿进行估算。Vision类中封装了estimatePose函数，分别使用两种方式进行解算旋转和平移向量。

1. 使用Aruco::estimatePoseSingleMarkers解算

本函数接收检测到的Aruco标签，并依次返回单个标签相对于相机的位姿估计结果，即返回旋转和平移向量。本方法的特点是需预先测量出每个标签的长度，而无需知道每个标签在图像中的分布，此外，在OpenCV中，该方法使用多线程加速，输出位姿估计量的实时性较好。

1. 使用solvePnP解算

PnP(Perspective-n-Point)是一类根据图像中特征点二维坐标及其对应的三维坐标，来估计相机在参考系中位姿的算法。本函数将世界坐标系下的顶点坐标以及图像坐标系下对应顶点的坐标作为输入，通过计算相机外参（即旋转向量R、平移向量t）来描述相机坐标系c与世界坐标系w之间的位姿关系，如图1-2所示。

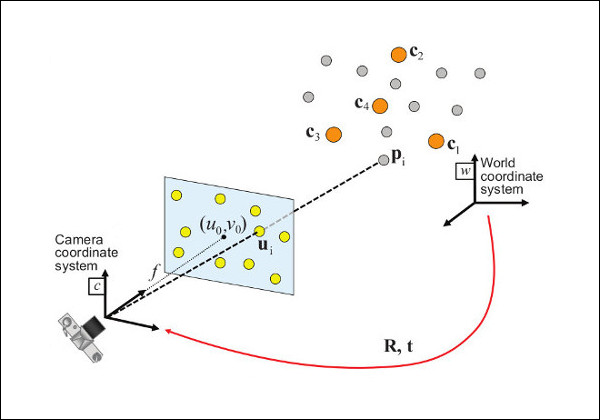


图1-2 solvePnP原理

随后，将旋转向量根据Rodrigues旋转公式转换为旋转矩阵，根据公式

可求出相机对物体的距离信息，其中代表物体在相机坐标系下的坐标，代表物体在世界坐标系下的坐标。

令,即物体转换到了相机坐标系的原点，表示相机在坐标系中的三维位置，可得公式（1.2）

1. 交通灯检测

交通灯检测部分的流程为：首先根据上一模块识别得到的Aruco数据以及顶点，确定信号灯的位置并裁剪得到图像。将信号灯图像转换为灰度图，取阈值后作开操作，以去除图像中可能存在的细连接以及噪点，可得到当前图像中最亮的部分，即此时点亮的信号灯。之后使用findContours函数对图像中的轮廓进行检测，选择最大面积的轮廓。最后使用minAreaRect函数寻找检测到最大面积的轮廓的最小外接矩形，矩形的中心点即为当前信号灯中点亮的位置，根据红、黄、绿在信号灯中的相对位置，即可判断当前点亮信号灯的颜色。由于环境光照、拍摄角度等问题，经多次实验发现信号灯在不同环境中的颜色，在RGB或HSV颜色模型下，均会发生变化，于是舍弃了基于颜色阈值的检测方案，采用前文所述的检测亮点的方法。

图1-3，1-4，1-5，1-6为本模块对输入视频的测试结果。其中pos\_x，pos\_y，pos\_z分别为相机到Aruco标签的真实距离，单位为mm。

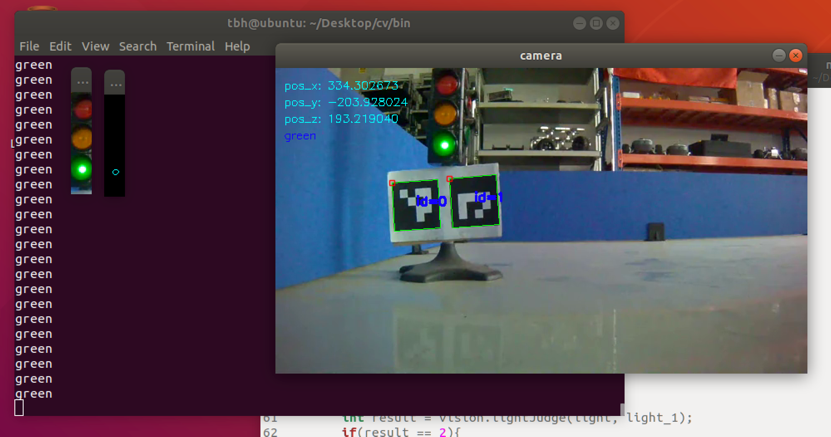


图1-3 输入视频绿灯测试结果

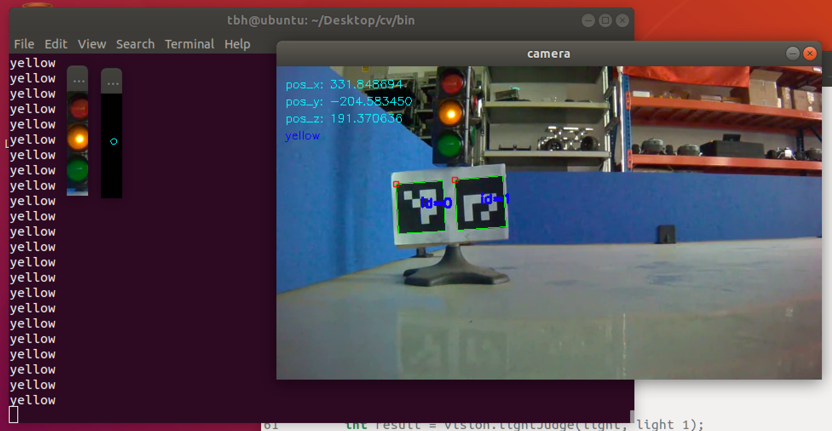


图1-4 输入视频黄灯测试结果

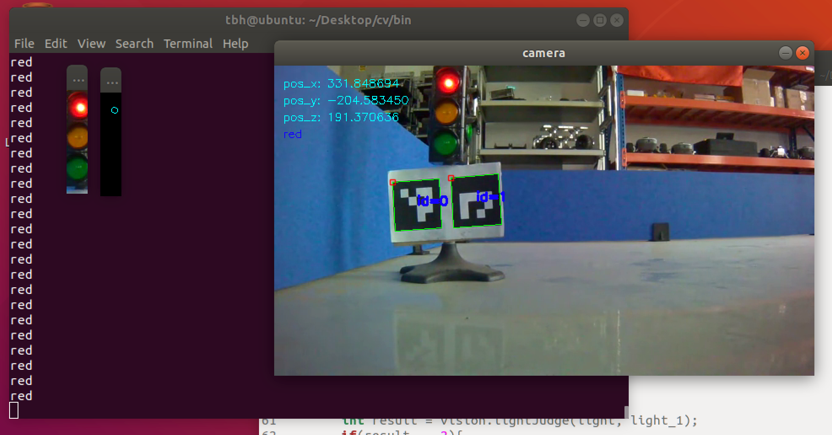


图1-5 输入视频红灯测试结果

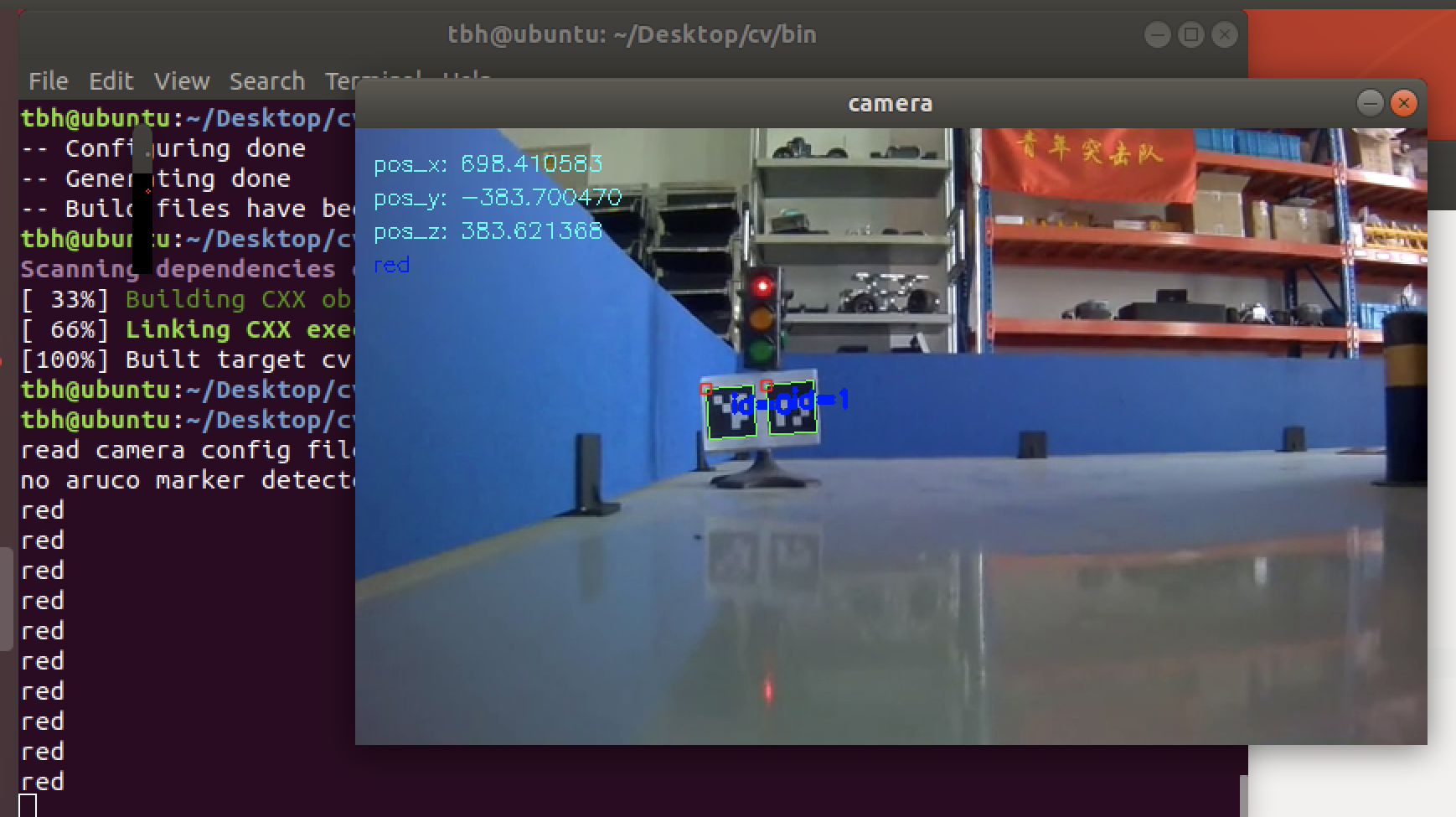


图1-6 输入视频测试结果