（说明：下文为本人毕业设计论文中图像识别程序部分的截取）

5 图像识别程序编写

5.1图形处理程序编写思路

图像识别程序编写阶段使用C++语言调用OpenCV视觉库的库函数实现特定颜色物体的提取及位置标定，并了解、学习相关图像处理算法的原理。在程序编写的前一阶段，首先在ubuntu环境下实现目标，当程序在电脑上测试通过时，再将图像识别程序移植到嵌入式平台运行，这样可接加快调试速度，通过查阅书籍和文献资料，明确了根据颜色识别目标的的具体步骤，如图5-1所示。

5.2图像的读取与显示

在新版本OpenCV中，调用摄像头进行图像采集的方法非常简单，只需将代码VideoCapture(“video\_path”)中“video\_path”（video\_path指的是视频文件的路径）换为0就表示调用摄像头采集图像而不是从文件中读取视频。将图像载入到新版本的图像存储数据结构Mat类中，然后用imshow函数显示即可[2]，图形读取与现实的核心代码片段如表5-1，完整程序见附录十三。

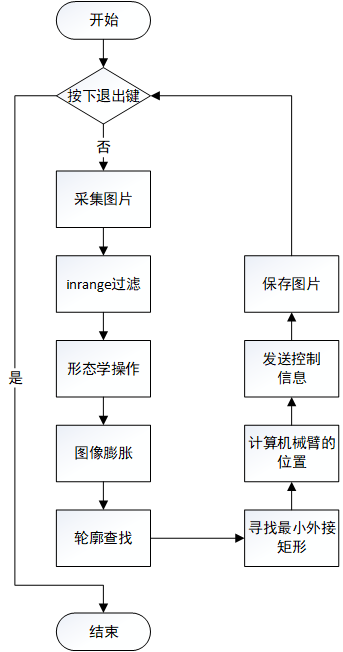


图5-1图形处理程序编写流程图

表5-1图像的读取与显示核心代码

VideoCapture capture **(**0**);**//从摄像头读取视频

…  
        **while(**1**)**  
        **{**  
        capture**>>**frame**;**//读取当前帧  
        imshow**(**"读取视频"**,**frame**);**//显示当前帧

…  
       **}**

执行程序后，运行结果如图5-2：



图5-2通过摄像头读取视频

5.3 inrange过滤

OpenCV中的inRange()函数可实现二值化功能，函数原型void inRange(InputArray src, InputArray lowerb, InputArray upperb, OutputArray dst)，该函数的输出dst是一幅二值化之后的图像。可以同时针对多通道进行操作，使用起来非常方便。，官方文档中对函数功能的解释是“Checks if array elements lie between the elements of two other arrays”。即检查数组元素是否在另外两个数组元素值之间。这里的数组通常也就是矩阵Mat或向量。

滑动条(Trackbar)是OpenCV动态调节参数非常好用的一种工具，它依附于窗口而存在。使用函数createTrackbar可以创建一个可以调整数值的滑动条（常常也被称作轨迹条），并将滑动条附加到指定的窗口上，使用起来很方便，在下面的程序中使用滑动条可以动态改变inRange函数颜色过滤的阈值，通过观察过滤效果选择出合适的阈值[2]。使用inrange函数进行颜色过滤的核心代码片段如表5-2，完整程序见附录十四。

表5-2inrange过滤核心代码

…  
     cvCreateTrackbar**(**"bh"**,** "control"**, &**b\_h**,** 255**);**   
     cvCreateTrackbar**(**"bl"**,** "control"**, &**b\_l**,** 255**);**  
 …  
        //1. inRange过滤  
        inRange**(**src**,**Scalar**(**b\_l**,**g\_l**,**r\_l**),**Scalar**(**b\_h**,**g\_h**,**r\_h**),**dst**);**  
  …

执行程序后，使用鼠标拖动滑动条改变过滤参数，运行结果如图5-3。

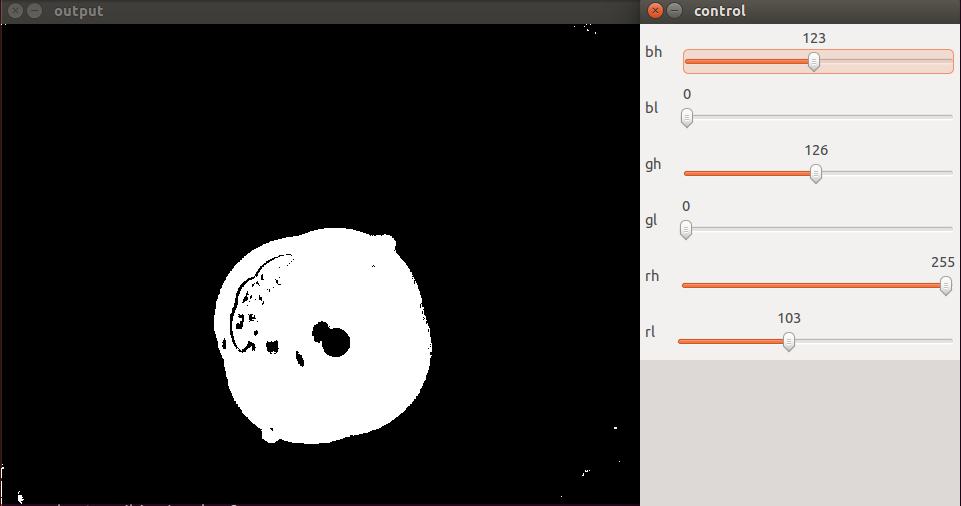


图5-3 inrange过滤图像

5.4形态学操作

观察左图和右图可以看出，当二值化之后的图形含有小的噪声时，可以通过形态学操作滤除噪声，左图中的噪声是由于灯光的原因产生的。通过形态学操作滤除噪声的核心代码片段如表5-3，完整程序见附录十五。

表5-3形态学操作核心代码

…  
 //2.形态学操作

morphologyEx(dst,dst2,MORPH\_OPEN,kernel1,Point(-1,-1),5);  
  …

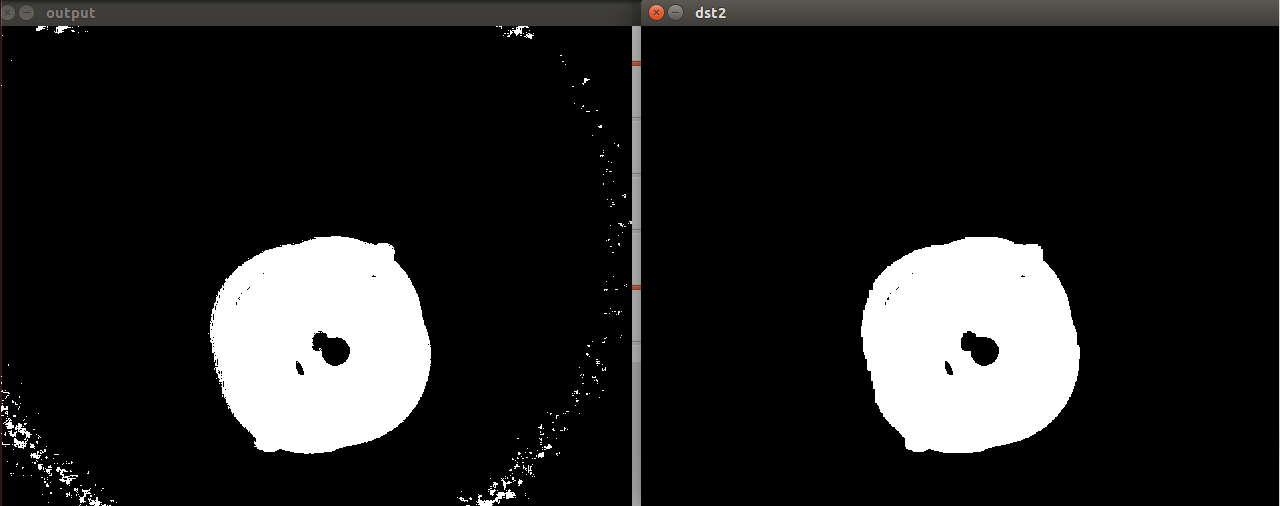


图5-4 形态学操作

5.5图像膨胀

膨胀和腐蚀操作的最基本组成部分，用于测试输出图像，通常要比待处理的图像小还很多。膨胀的作用就是将目标的边缘或者是内部的坑填掉。膨胀的算法：用3x3的结构元素，扫描图像的每一个像素 用结构元素与其覆盖的二值图像做“与”操作 如果都为0，结果图像的该像素为0。否则为1 结果：使二值图像扩大一圈。

二维平面结构元素由一个数值为0或1的矩阵组成。结构元素的原点指定了图像中需要处理的像素范围，结构元素中数值为1的点决定结构元素的邻域像素在进行膨胀或腐蚀操作时是否需要参与计算[3]。图像膨胀的核心代码片段见表5-4，完整程序见附录十六。

表5-4图像膨胀核心代码

Mat kernel2=getStructuringElement(MORPH\_RECT,Size(3,3),Point(-1,-1));

…

//膨胀

dilate(dst,dst2,kernel2,Point(-1,-1),8);

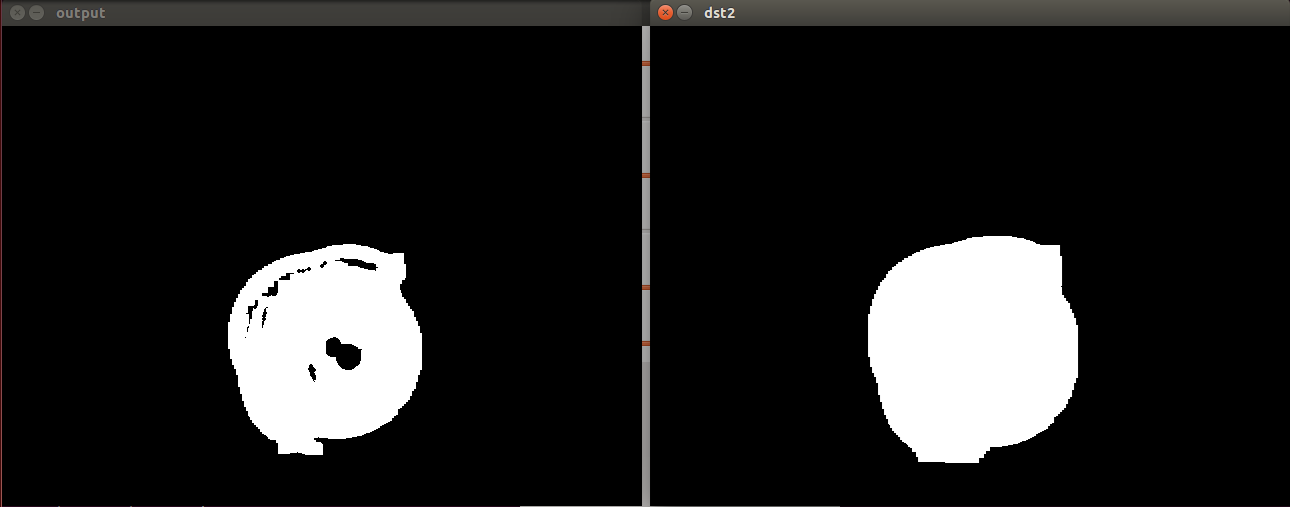


图5-5 图像膨胀

5.6轮廓查找与位置标定

查找图像的轮廓在图像处理及应用中扮演着重要的角色，本次实验的轮廓查找是调用了OpenCVfindContours函数来实现的，它是从二值图像中来计算轮廓的，核心代码片段见表5-5，完整程序见附录十七。图5-6为轮廓查找及位置标定显式结果。

表5-5轮廓查找与位置标定核心代码

findContours(mask,contours,hireachy,RETR\_EXTERNAL,CHAIN\_APPROX\_SIMPLE,Point(0,0));

for (int i = 0; i<contours.size(); i++) { //绘制轮廓

drawContours(mask, contours, i, Scalar(255),1,8,hireachy);

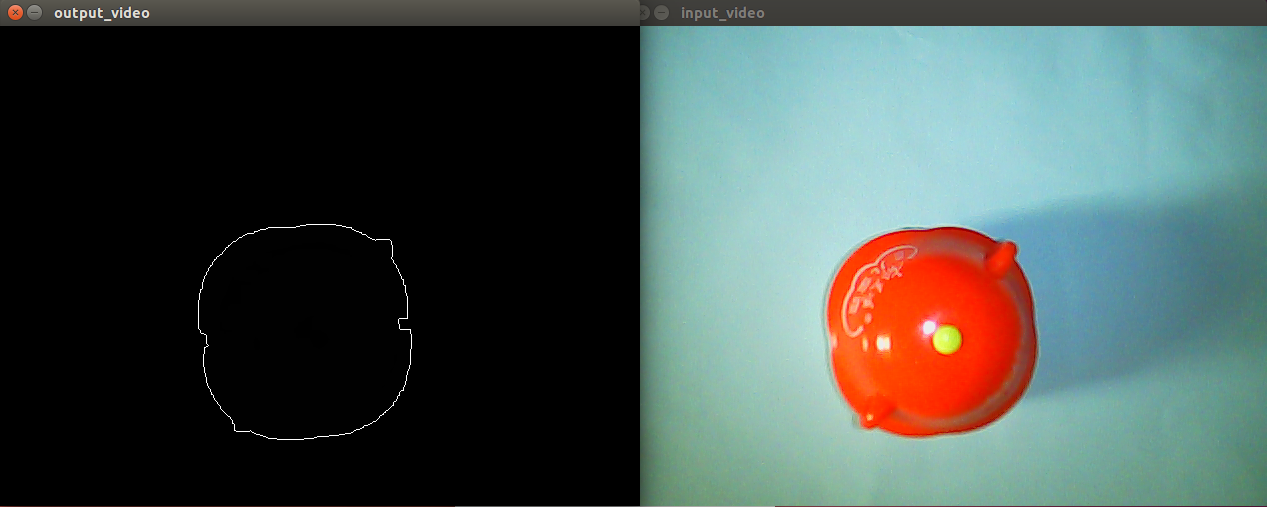


图5-6 轮廓查找与位置标定

5.7图形绘制及文字标注

在5.6中，已经找到了目标物体的轮廓和中心的坐标，接下来要做的是在图像中绘制图形，圈出目标物体，并在图像中添加目标的标号和计数文本信息。除此之外，还可以在图像中用线段模拟机械臂，作为真实机械臂的参考。关于机械臂建模及位置计算方法见6.2节。图形绘制及文字标注核心代码片段如表5-6，完整程序见附录十八，图5-7为识别单个目标的结果，图5-8为识别多个目标的结果。

表5-6图形绘制和文字标注核心代码

for (int j = 0; j <= 3; j++)

{

line(frame, P[j], P[(j + 1) % 4], Scalar(0,0,255), 3);

line(mask, P[j], P[(j + 1) % 4], Scalar(111), 3); }

pxx.x=(P[0].x+P[2].x)/2;

pxx.y=(P[0].y+P[2].y)/2;

//画实心点

circle(frame, pxx, 3,Scalar(255,0,255),-1);//第五个参数设为-1，表明这是个实点。

sprintf(buf,"A%d",i);

putText(frame,buf,pxx,FONT\_HERSHEY\_PLAIN,3,Scalar(0,255,0),1,8);

}

line(frame, Point(0,0), sec, Scalar(0,0,255), 4);

line(frame, sec, pxx, Scalar(255,0,255), 4);

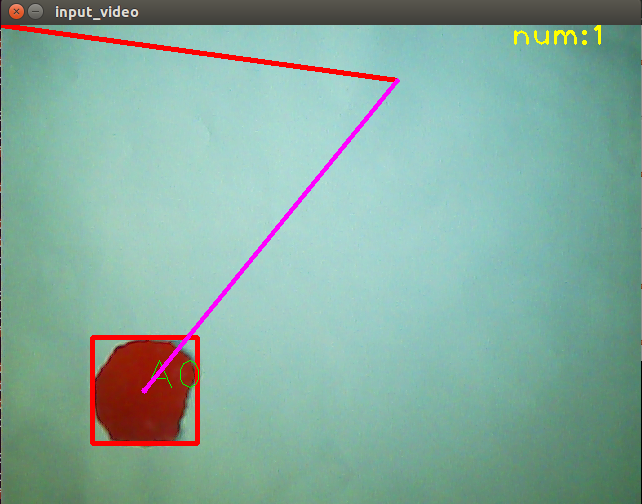


图5-7识别单个目标

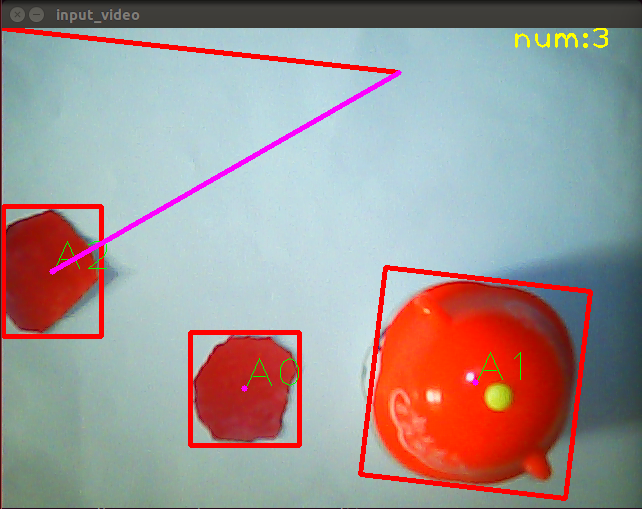


图5-8识别多个目标

5.8 OpenCV与视频流服务器对接

当图像采集和处理完成之后，接下来就是图形的显示，由于现有的嵌入式开发板没有配带LCD液晶屏，所以选择使用mjpg-streamer视频流服务器来显示图像，而且可以在不影响作品最终效果的情况下简化工作。

使用mjpg-streamer视频流服务器来显示图像，首先要去除程序中有关GUI和用户交互相关的代码，因为视频流服务器显示的图像不能反馈按键和鼠标的值，如果不去除这部分代码会在程序运行时报错，修改完成之后使用arm编译器重新编译程序，编译完成后放到nfs目录中。修改之后的代码见附录十九。

mjpg-streamer视频流服务器支持多种方式的视频源输入，比如摄像头设备或文件，这里选择图片文件作为视频流服务器的输入，那么设置OpenCV的输出为图片，就完成了两者的对接。因此需要执行以下操作：

（1）使用OpenCV视觉库将处理完的图片保存到指定目录。需要用到imwrite函数，函数原型如下：imwrite(const string& filename, InputArray img, const vector<int>& params= vecto r<int>() )，其中filename为需要写入的文件名，包含文件路径，img为要保存的图像源，params:表示为特定格式保存的参数编码。

（2）设置mjpg-streamer视频流服务器从指定目录下读文件，如mjpg\_streamer -i "input\_file.so -f /pic -n 1.jpg -r " -o "output\_http.so -w /www" 表示mjpg-streamer视频流服务器以/pic目录下的文件作为输入源，-n用来指定文件的名称，-r表示读取文件之后将文件移除，以便OpenCV视觉库将新的图像放置到/pic目录下[4]。

对视频流服务器的网页文件进行修改，适配到本毕业设计。网页文件目录在编译成功的目录下的mjpg-streamer/www中，文件类型包含html、css、js文件，只需要对html文件中的标签和文字进行简单的修改，并修改css文件中对于标签的架构设置，就可以使网页更美观，稍作修改后的网页如图5-11所示。

网页文件修改完成之后保存，然后上电启动开发板，依次加载uboot，内核，根文件系统，然后执行命令启动视频流服务器，紧接着启动OpenCV程序进行图像采集。程序正常启动打印信息如图5-9，网页端显示效果如图5-10。