## 基本原理

基于普通摄像头的视觉抓取，使用了视觉处理方法，通过识别定位框的位置和物块的相对位置，获得物块到机械臂的位置。

### 颜色识别原理

颜色的参数分别是：色调（H），饱和度（S），明度（V）。

#### 色调H

用角度度量，取值范围为0°～360°，从红色开始按逆时针方向计算，

#### 饱和度S

饱和度S表示颜色接近光谱色的程度。一种颜色，可以看成是某种光谱色与白色混合的结果。

其中光谱色所占的比例愈大，颜色接近光谱色的程度就愈高，颜色的饱和度也就愈高。

饱和度高，颜色则深而艳。

光谱色的白光成分为0，饱和度达到最高。

通常取值范围为0%～100%，值越大，颜色越饱和。

#### 明度V

明度表示颜色明亮的程度，

对于光源色，明度值与发光体的光亮度有关；

对于物体色，此值和物体的透射比或反射比有关。

通常取值范围为0%（黑）到100%（白）。

RGB和CMY颜色模型都是面向硬件的，而HSV（Hue Saturation Value）颜色模型是面向用户的。

### 文件说明

#### script

主程序main调用 arm.py和camera.py，相机模块调用颜色识别和定位框识别模块

main.py---->arm.py

----->camera.py---->color\_detection.py---> cvwin.py

----->plate\_detection.py---> cvwin.py

cvwin.py 单独的cv2相关的窗口显示线程

#### launch

aiarm-client-controller.launch 虚拟机上使用的

aiarm-controller.launch 网关上使用的，一般使用这个

### 参数说明

|  |
| --- |
| #定位板相关参数  loc\_plate=[141,192,465,386] #初始值，机械臂自己根据plate\_detection.py文件中自己查找并更新此参数 单位像素 不用设置  loc\_plate\_act=[0.153,0.173,0.092] #定位板实际长度 单位 m 定位板框实际长度 上底 下底 高[0.155,0.173,0.093]  loc\_plate\_act\_origin=[0.23,-0.0148] #定位板原点偏移 单位 m 定位板相对于base\_link的x,y方向偏移量  loc\_x\_off\_mx=25  loc\_x\_off\_mi=9  loc\_y\_off\_mx=0  loc\_opt\_par\_x=1.0 #X方向的优化参数  loc\_opt\_par\_y=1.1 #Y方向的优化参数  '''  loc\_plate 单位 pix 640\*480像素 定位板框 左上角x,y 和 右下角x,y  loc\_plate\_act 单位 m 定位板框实际长度 上底 下底 高[0.147,0.182,0.082]  loc\_x\_off\_mx 单位 pix 相机倾斜视角X像素偏移 最大值  loc\_y\_off\_mx 单位 pix 相机倾斜视角y像素偏移 最大值  loc\_plate\_origin 单位 pix cv2下像素原点  loc\_plate\_act\_origin 单位 m 实际定位原点 相对base\_link  '''  #机械臂  arm\_g\_height=0.1  #拍照位  # arm\_cam\_pose=[0.1572271238, 0.01332804092, 0.2300313884, 0.04640923861, 0.4895623458, 0.00796998209, 0.87069591217]  arm\_cam\_joint=[0.11535425216048287, -0.616704188215822, 1.40664924293023, 1.8002512210459807, 0.17097973936214012]  #过渡位  arm\_trans\_joint=[5.46315397666901e-05, 0.15632741105576997, 1.284452648106851, 0.9141770383249787, -9.986523001948356e-05] |

### 坐标系说明



## 操作步骤

### 基本步骤

#### 编译ROS环境

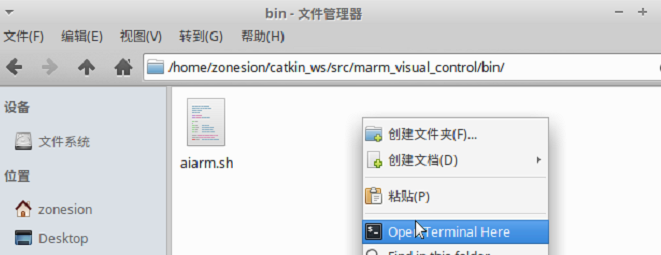
将两个实验包上传到/home/zonesion/catkin\_ws/src中，然后使用如下命令编译

|  |
| --- |
| **zonesion@rk3399:~$ cd ~/catkin\_ws/**  **zonesion@rk3399:~/catkin\_ws$ catkin\_make** |

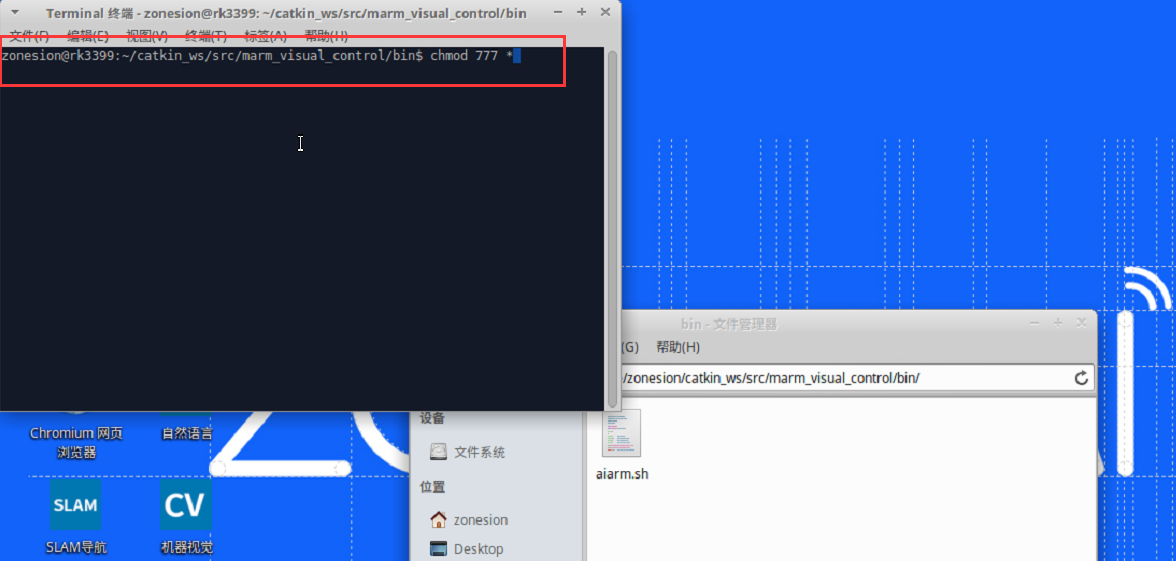
#### 桌面快捷方式

##### 1、添加执行权

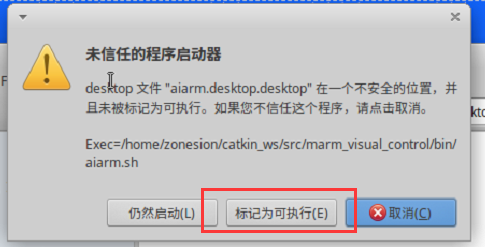
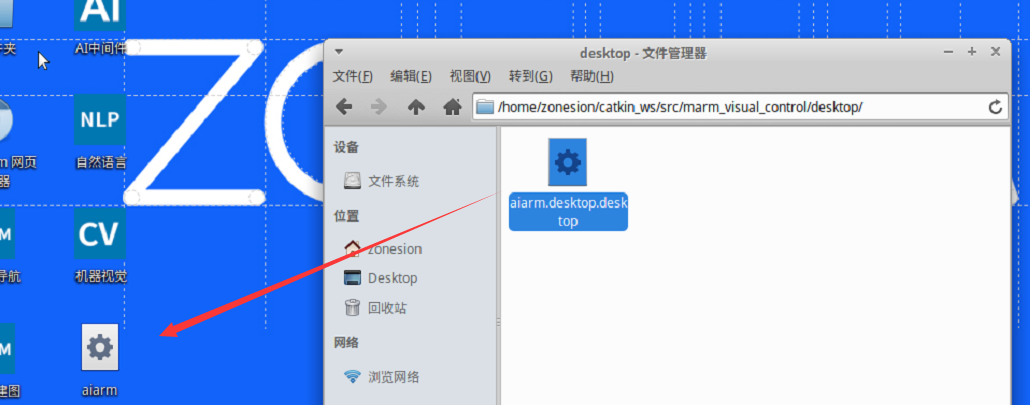
如果桌面上没有这个启动文件，先进入到/home/zonesion/catkin\_ws/src/marm\_visual\_control/bin/打开一个终端。

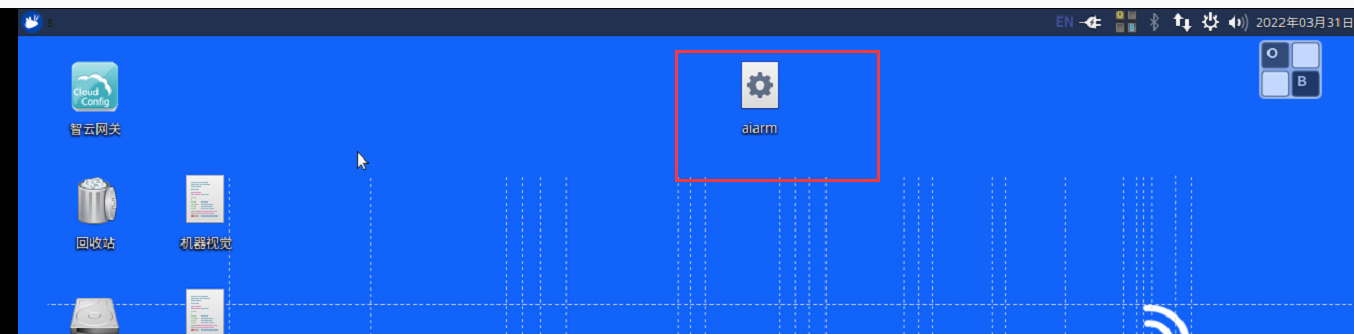


使用chmod 777 \* 命令



然后进入到/home/zonesion/catkin\_ws/src/marm\_visual\_control/desktop/目录，将其拖到桌面。双击后点击标记执行即可。然后再启动





**关闭所有终端，使用启动程序启动以上程序**

### 进阶步骤

#### 机械臂走到相机拍照位

关闭如上

###### 1、启动launch启动文件

启动一个ssh终端

|  |
| --- |
| **$ roslaunch marm\_visual\_control aiarm-controller.launch**  ... logging to /home/zonesion/.ros/log/24f02082-b08b-11ec-885a-b1e0c15c78ec/roslaunch-rk3399-9153.log  Checking log directory for disk usage. This may take awhile.  Press Ctrl-C to interrupt  Done checking log file disk usage. Usage is <1GB.  started roslaunch server http://rk3399:46050/  SUMMARY  ========  PARAMETERS  \* /grasp\_filter\_test/arm: manipulator  [ INFO] [1648688509.308670709]: MoveGroup context using planning plugin ompl\_interface/OMPLPlanner  [ INFO] [1648688509.309991661]: MoveGroup context initialization complete  You can start planning now! |

###### 2、运行到拍照位

启动一个ssh终端

|  |
| --- |
| zonesion@rk3399:**~$ cd catkin\_ws/src/marm\_visual\_control/script/**  zonesion@rk3399:~/catkin\_ws/src/marm\_visual\_control/script**$ python main.py**  Failed to import pyassimp, see https://github.com/ros-planning/moveit/issues/86 for more info  [ INFO] [1648688601.580013535]: Loading robot model 'marm'...  [ INFO] [1648688601.580419243]: No root/virtual joint specified in SRDF. Assuming fixed joint  [ INFO] [1648688604.750564185]: Ready to take commands for planning group manipulator.  ('red color\_par', (156, 51, 190, 190, 255, 255))  ('green color\_par', (50, 102, 158, 71, 200, 200))  libpng warning: Incorrect sBIT chunk length  set cam number 0  [INFO] [1648688606.222780]: Open the fixture  ^CEXIT APP  zonesion@rk3399:~/catkin\_ws/src/marm\_visual\_control/script$ |

运行main.py脚本，当机械臂动作到照相机拍照位后，退出程序

#### 颜色识别标定

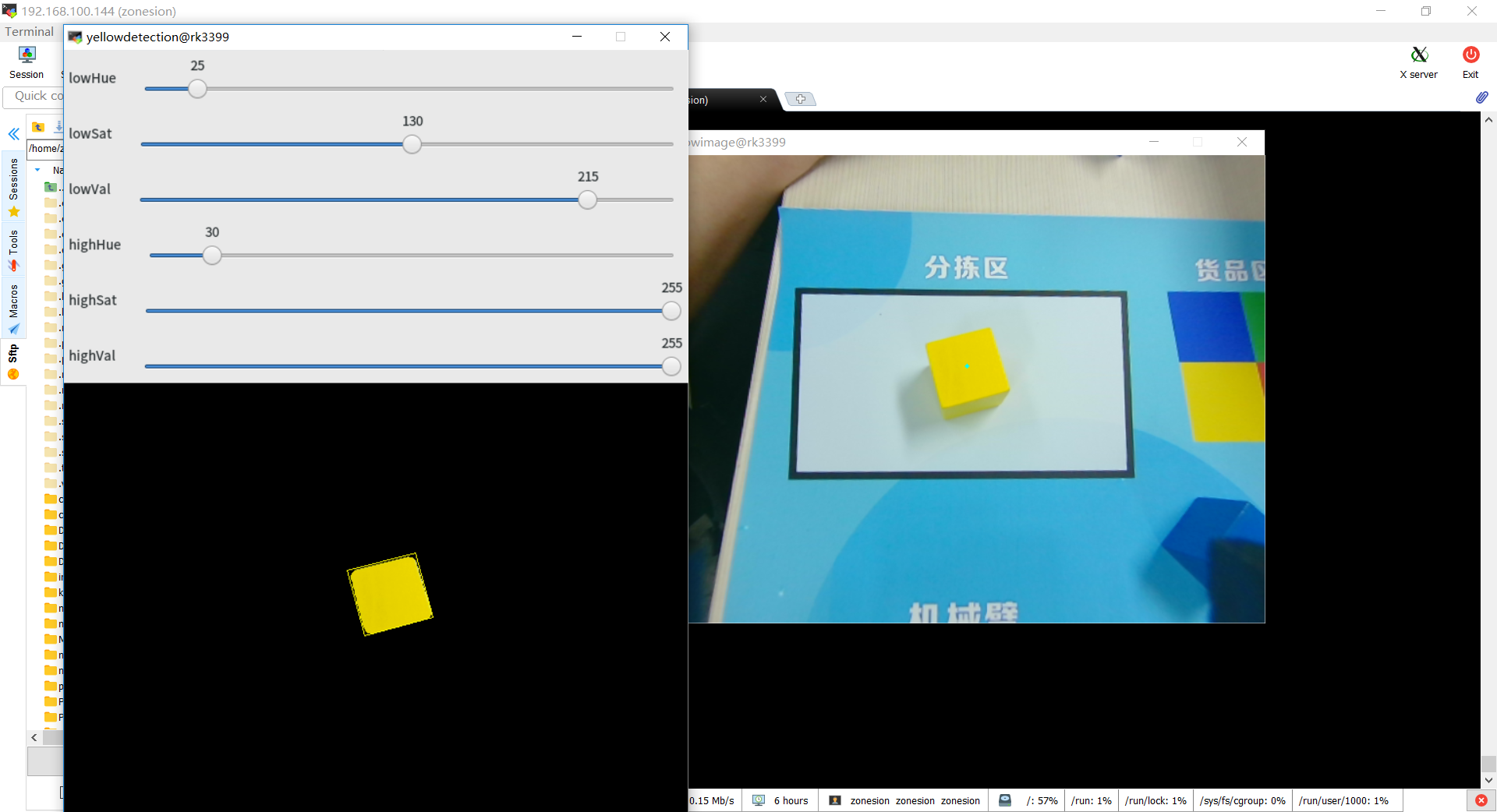
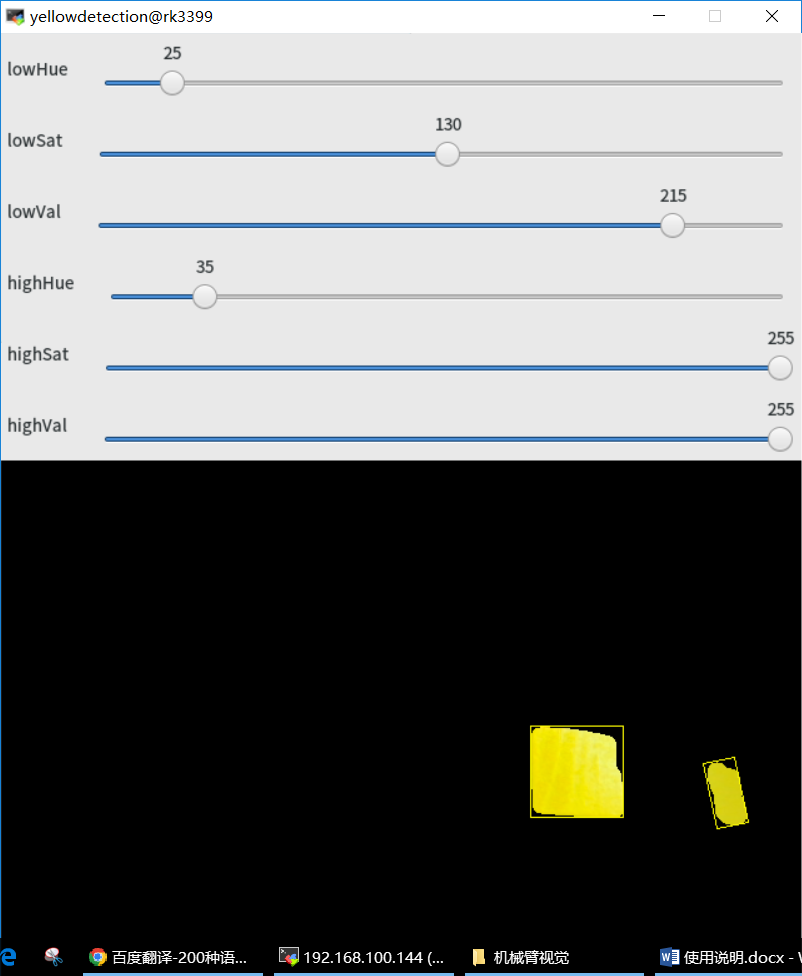
使用color\_detection.py文件

###### 1、机械臂走到拍照位

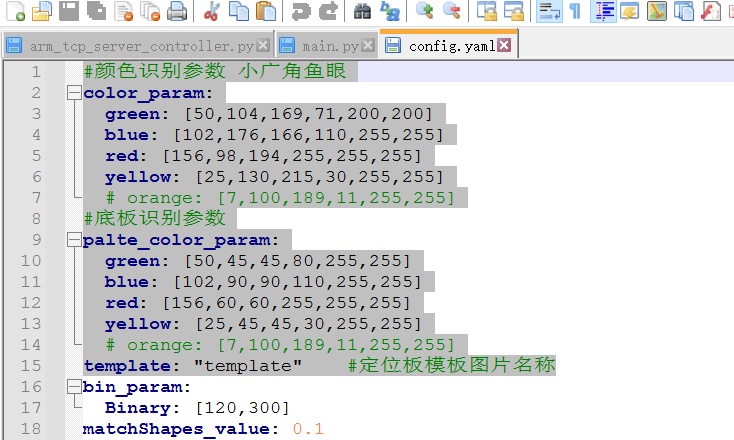
###### 2、标定颜色

zonesion@rk3399:~/catkin\_ws/src/marm\_visual\_control/script$ **python color\_detection.py -c yellow**

如标定黄色，后面跟随参数为yellow即可。如下这样标定所有颜色，

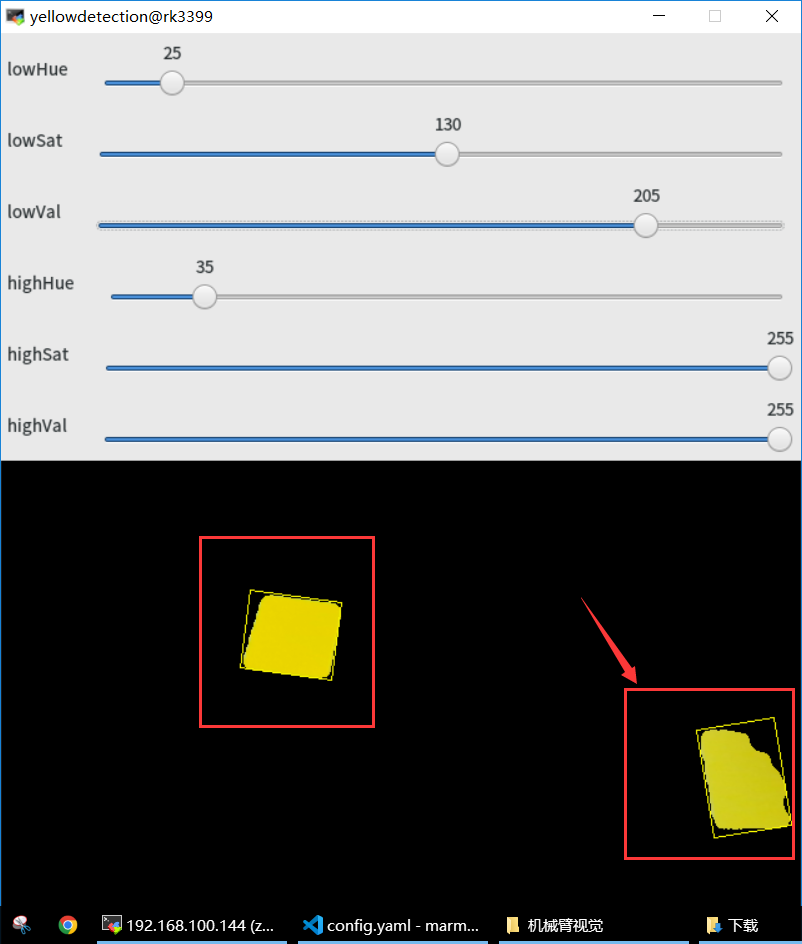
右边这样也不影响，主要黄色的木块能够识别出来即可。后面的camera应用中会将不在定位框内的进行清楚



将颜色识别参数写入到color\_param.

第一组颜色识别参数用于定位物块的位置

第二组颜色识别参数用于定位定位板框位置，为了提高定位框识别精度用的。



同时保证两个识别框数据稳定，不然会造成定位数据失败

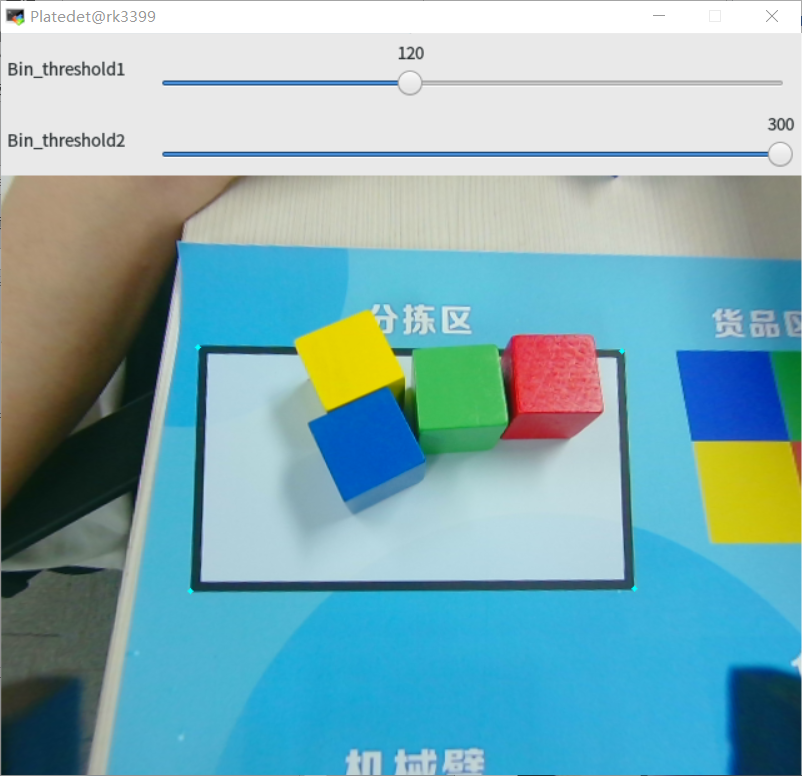
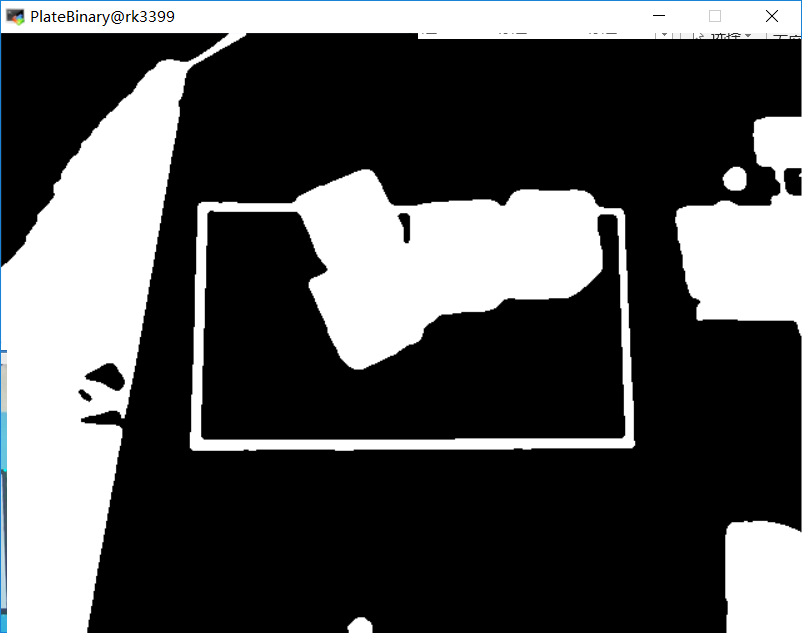
#### 定位框识别标定

同样机械臂走到拍照位。

##### 1、运行plate\_detection.py

|  |
| --- |
| zonesion@rk3399:~/catkin\_ws/src/marm\_visual\_control/script**$ python plate\_detection.py**  libpng warning: Incorrect sBIT chunk length |

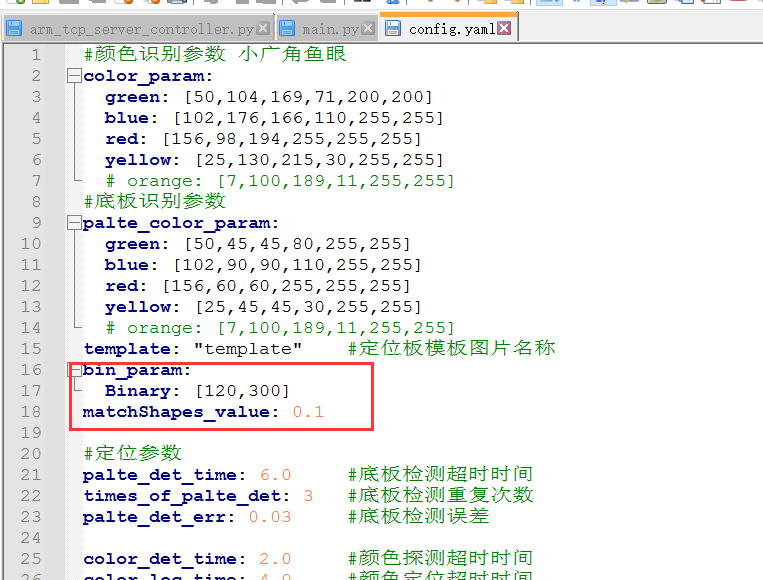
##### 2、调整参数

识别出四个角，同时物体显示为白色，说参数就可以

通常阈值2不用调，只需要调整第一个值

如上参数为：120,300

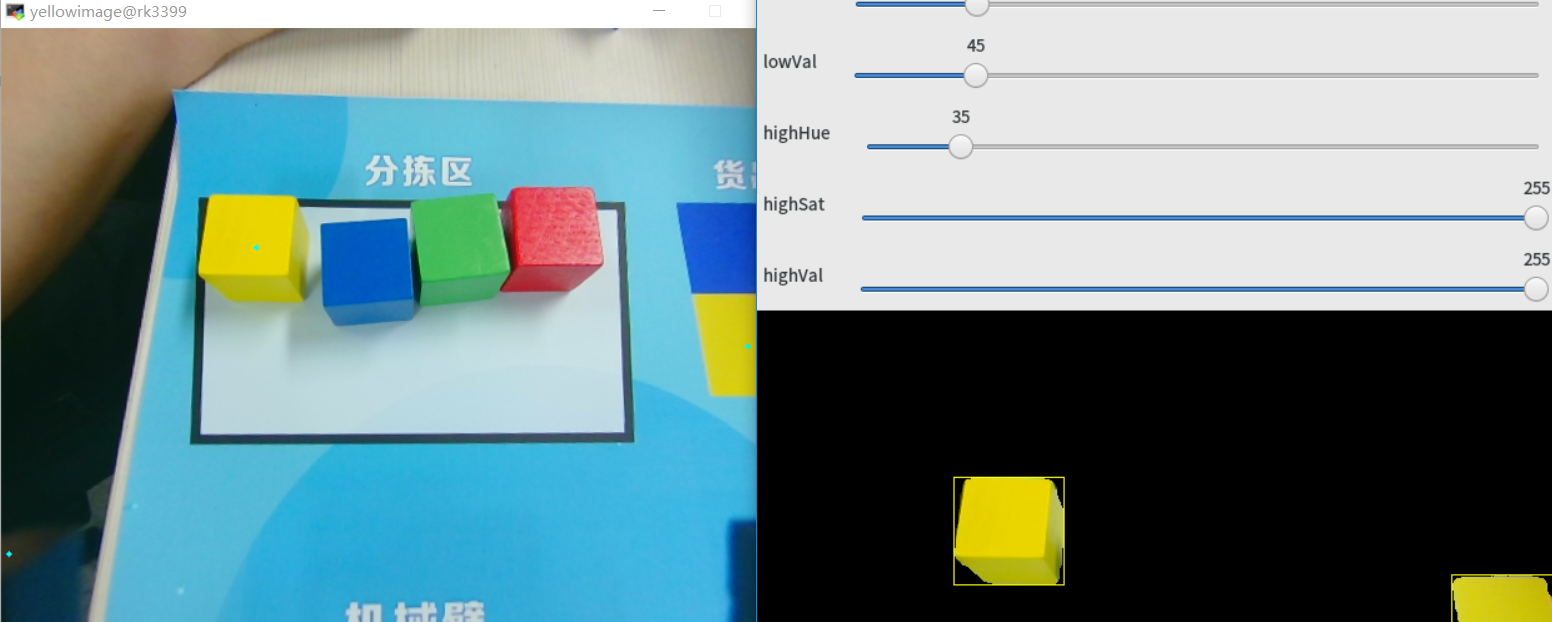


##### 3、调整色块色域参数

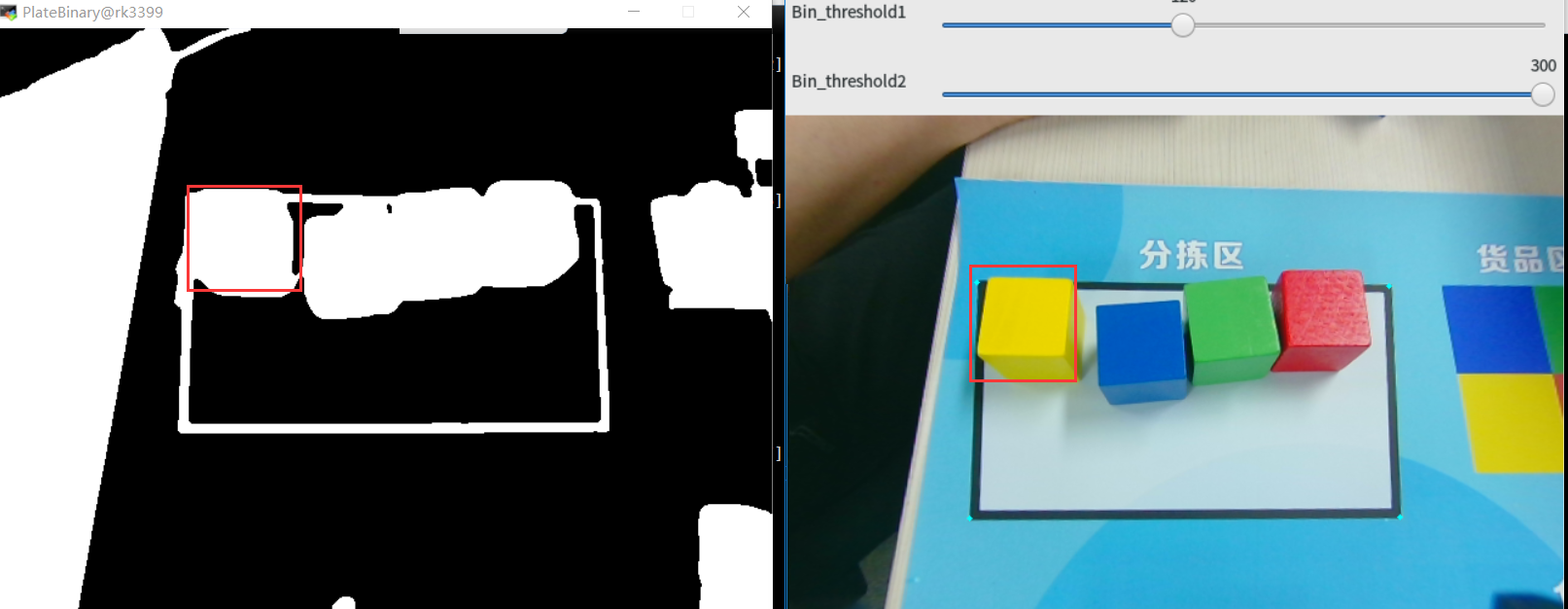
zonesion@rk3399:~/catkin\_ws/src/marm\_visual\_control/script$ **python color\_detection.py -c yellow -t 1**

('yellow color\_par', [25, 45, 45, 30, 255, 255])

('yellow', [[598, 255, 0.0], [277, 149, 24.4439544



主要黄色的木块能够整个识别即可。这样



这样就可以进行相应的取反颜色。**而不影响定位框的识别。**

#### 在window端使用虚拟机联机调试并运行综合程序

##### 1、3399启动服务

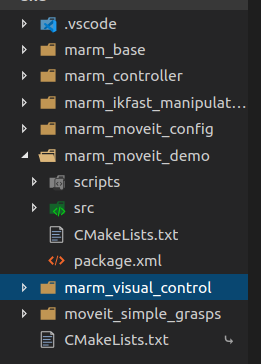
编译完成后在3399上启动如下命令

zonesion@rk3399:~$ roslaunch marm\_controller arm-tcp-server-controller-rviz.launch

server\_ipaddress='192.168.100.144' #IP地址根据实际情况设定

注意这个文件中共的IP设置要和小车当前IP一致

##### 2、虚拟机端启动客户端



虚拟机这里启动机械臂控制的客户端，

zonesion@zonesion:~/catkin\_ws/src$ roslaunch "/home/zonesion/catkin\_ws/src/marm\_visual\_control/launch/aiarm-client-controller.launch"

虚拟机则运行如上程序即可

##### 3、将鱼眼摄像头链接到电脑

##### 4、mian中定义为varm

然后在虚拟机中main程序中这里一定要定义为varm

class AiArm(Arm,AiCamera):

def \_\_init\_\_(self,g\_open,color,win=[],loc\_plate=[141,192,465,386],loc\_plate\_act=[0.147,0.173,0.092] ,

loc\_plate\_act\_origin=[0,0],loc\_x\_off\_mx=25,loc\_x\_off\_mi=9,loc\_y\_off\_mx=15,color\_par=None,bin\_param=None,):

super(AiArm,self).\_\_init\_\_(g\_open,xarm="varm")

super(Arm,self).\_\_init\_\_(color,win,loc\_plate,loc\_plate\_act,loc\_plate\_act\_origin,

loc\_x\_off\_mx,loc\_x\_off\_mi,loc\_y\_off\_mx,color\_par,bin\_param)

这样表示使用虚拟机控制机械臂，

## 注意事项

1、颜色标定时注意货品区会存在一定的干扰，标定颜色时要保证数据稳定，camera.py中的无法准确获取颜色数据的。

**2、如果摄像头位置不正确，会导致识别定位板位置不对，理想状态下照相机的拍照位拍摄的定位框是居中并且水平的。**

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |