



视觉识别 demo 手册



睿尔曼智能科技（北京）有限公司



一、 二维码识别定位及抓取

1.1 概述

该部分实现了机械臂识别并抓取目标物体的功能。目标物体上附着指定大小的二维码，通过 **realsense** 相机获取二维码在相机坐标系下的坐标，并通过相机与基坐标系的转换关系，求得目标物体在基坐标系下的位置，控制机械臂到达该位置并抓取，实现对目标物体的自动定位与抓取功能。

1.2 功能实现

在运行程序之前，需要先对相机进行标定工作，确定相机坐标系与基坐标系的转换关系（注意：目标识别时所使用的二维码的尺寸需与标定时使用的二维码尺寸一致）。标定过程详见《RM 机械臂与 Realsense D435 手眼标定教程》。

1.2.1 打开深度相机

打开新终端，输入以下指令打开 **realsense** 相机。

```
cd ~/catkin_ws
source devel/setup.bash
roslaunch realsense2_camera rs_camera.launch
```

```
/home/realman/catkin_ws/src/realsense-ros/realsense2_camera/launch/rs_camera.launch http://localhost:11311
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[INFO] [1639471966.582111583]: Built with LibRealSense v2.50.0
[INFO] [1639471966.582154314]: Running with LibRealSense v2.50.0
[INFO] [1639471966.608428472]:
[INFO] [1639471966.614174509]: Device with serial number 137322078962 was found.
[INFO] [1639471966.614247944]: Device with physical ID /sys/devices/pci0000:00/0000:00:14.0/usb3/3-1/3-1:1.0/video4linux/vi
bo0 was found.
[INFO] [1639471966.614293765]: Device with name Intel RealSense D435 was found.
[INFO] [1639471966.614935268]: Device with port number 3-1 was found.
[INFO] [1639471966.614991985]: Device USB type: 3.2
[INFO] [1639471966.618542300]: getParameters...
[INFO] [1639471966.672613411]: setupDevice...
[INFO] [1639471966.672663406]: JSON file is not provided
[INFO] [1639471966.672699541]: ROS Node Namespace: camera
[INFO] [1639471966.672738873]: Device Name: Intel RealSense D435
[INFO] [1639471966.672779843]: Device Serial No: 137322078962
[INFO] [1639471966.672797659]: Device physical port: /sys/devices/pci0000:00/0000:00:14.0/usb3/3-1/3-1:1.0/video4linux/video
[INFO] [1639471966.672826760]: Device FW version: 05.13.00.50
[INFO] [1639471966.672855684]: Device Product ID: 0x0B07
[INFO] [1639471966.672880771]: Enable PointCloud: Off
[INFO] [1639471966.672902753]: Align Depth: Off
[INFO] [1639471966.672941290]: Sync Mode: Off
[INFO] [1639471966.673001089]: Device Sensors:
[INFO] [1639471966.675687660]: Stereo Module was found.
[INFO] [1639471966.681077192]: RGB Camera was found.
[INFO] [1639471966.681142596]: (Confidence, 0) sensor isn't supported by current device! -- Skipping...
[INFO] [1639471966.681200985]: num_filters: 0
[INFO] [1639471966.681241109]: Setting Dynamic reconfig parameters.
[INFO] [1639471966.732648946]: Done Setting Dynamic reconfig parameters.
[INFO] [1639471966.735070792]: depth stream is enabled - width: 848, height: 480, fps: 30, Format: Z16
[INFO] [1639471966.735891283]: color stream is enabled - width: 640, height: 480, fps: 30, Format: RGB8
[INFO] [1639471966.735960930]: setupPublishers...
[INFO] [1639471966.739830103]: Expected frequency for depth = 30.00000
[INFO] [1639471966.786686451]: Expected frequency for color = 30.00000
[INFO] [1639471966.808855024]: setupStreams...
[INFO] [1639471966.936163645]: SELECTED BASE:Depth, 0
[INFO] [1639471966.954314706]: RealSense Node Is Up!
[WARN] [1639471967.007367169]:
```

1.2.2 启动 aruco 节点识别 Marker 标签

打开一个新的终端，执行以下命令启动 **single_realsense.launch**:

```
cd ~/catkin_ws
source devel/setup.bash
roslaunch aruco_ros single_realsense.launch
```



```
/home/realman/catkin_ws/src/aruco_ros/aruco_ros/launch/single_realsense.launch http://localhost:11311
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 标签(B) 帮助(H)
/home/realman/catkin_ws/src/realsense-ros/realsense2_camera/launch/rs_came... * /home/realman/catkin_ws/src/aruco_ros/aruco_ros/launch/single_realsense.laun... *
realman@realman-laptop:~/catkin_ws$ cd ~/catkin_ws/
realman@realman-laptop:~/catkin_ws$ source devel/setup.bash
realman@realman-laptop:~/catkin_ws$ roslaunch aruco_ros single_realsense.launch
... logging to /home/realman/.ros/log/fdabfd16-5d88-11ec-9e13-b9364f7760b1/roslaunch-realman-laptop-7398.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://realman-laptop:46153/

SUMMARY
=====
PARAMETERS
 * /aruco_single/camera_frame: camera_color_frame
 * /aruco_single/corner_refinement: LINES
 * /aruco_single/image_is_rectified: True
 * /aruco_single/marker_frame: aruco_marker_frame
 * /aruco_single/marker_id: 582
 * /aruco_single/marker_size: 0.05
 * /aruco_single/reference_frame: camera_color_frame
 * /rostdistro: melodic
 * /rosversion: 1.14.12

NODES
 /
  aruco_single (aruco_ros/single)

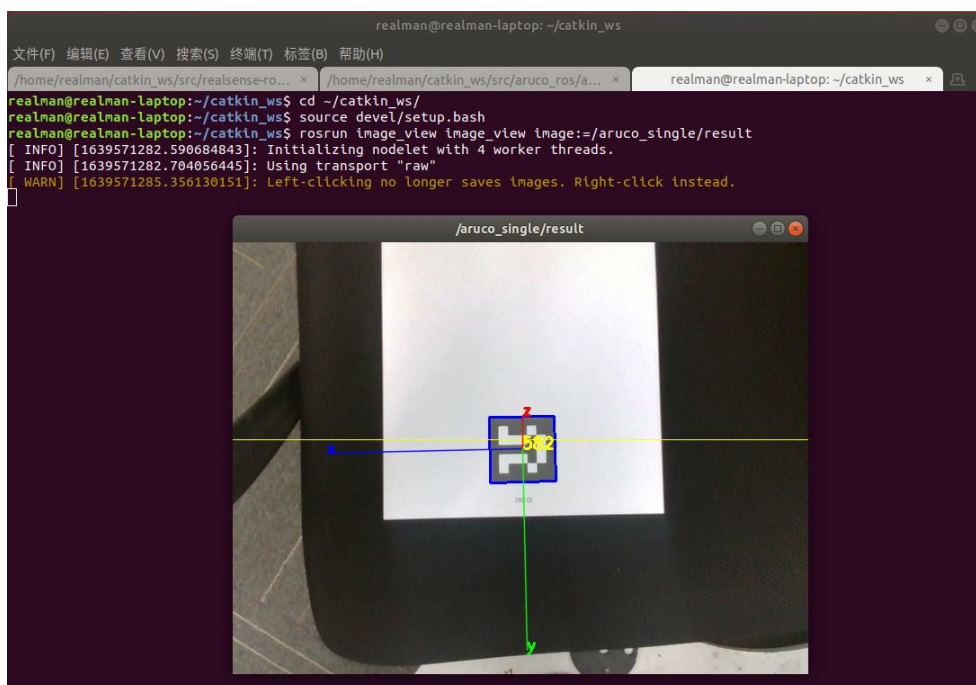
ROS_MASTER_URI=http://localhost:11311

process[aruco_single-1]: started with pid [7413]
[ WARN ] [1639560390.176964474]: Corner refinement options have been removed in ArUco 3.0.0, corner_refinement ROS parameter is deprecated
[ WARN ] [1639560390.245484926]: normalizeImageIllumination is unimplemented!
```

1.2.3 启动 image_view 节点显示图像

打开一个新的终端，执行以下命令启动 image_view 订阅/aruco_single/result 显示图像：

```
cd ~/catkin_ws
source devel/setup.bash
roslaunch image_view image_view image:=/aruco_single/result
```



1.2.4 启动 RM 机械臂的 control 和 driver 节点

打开一个新的终端，执行以下命令启动 control 节点：



```
cd ~/catkin_ws
source devel/setup.bash
roslaunch rm_control rm_control.launch
```

```
/home/realman/catkin_ws/src/rm_65_robot/rm_control/launch/rm_control.launch http://localhost:11311
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
realman@ubuntu:~$ cd catkin_ws/
realman@ubuntu:~/catkin_ws$ source devel/setup.bash
realman@ubuntu:~/catkin_ws$ roslaunch rm_control rm_control.launch
.. logging to /home/realman/.ros/log/38da116a-408d-11ed-891c-000c2998aff3/roslaunch-ubuntu-16592.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ubuntu:46711/

SUMMARY
=====

PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.13

NODES
/
  rm_control (rm_control/rm_control)

ROS_MASTER_URI=http://localhost:11311

process[rm_control-1]: started with pid [16607]
```

再打开一个新的终端，执行以下命令启动 driver 节点：

```
cd ~/catkin_ws
source devel/setup.bash
roslaunch rm_bringup rm_robot.launch
```

1.2.5 启动 easy_handeye 节点

再打开一个新的终端，执行以下命令发布手眼标定后的 TF：

```
roslaunch easy_handeye publish.launch
```

```
/home/realman/catkin_ws/src/easy_handeye/easy_handeye/launch/publish.launch http://localhost:11311
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 标签(B) 帮助(H)
/home/realman/... x /home/realman/... x realman@realman... x /home/realman/... x /home/realman/... x /home/realman/... x
realman@realman-laptop:~/catkin_ws$ roslaunch easy_handeye publish.launch
.. logging to /home/realman/.ros/log/b9a2478a-5da9-11ec-9e13-b9364f7760b1/roslaunch-realman-laptop-14505.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://realman-laptop:41179/

SUMMARY
=====

PARAMETERS
* /rm_rs_d435_eye_on_hand/calibration_file:
* /rm_rs_d435_eye_on_hand/eye_on_hand: True
* /rm_rs_d435_eye_on_hand/inverse: False
* /rm_rs_d435_eye_on_hand/robot_effector_frame:
* /rm_rs_d435_eye_on_hand/tracking_base_frame:
* /rostdistro: melodic
* /rosversion: 1.14.12

NODES
/
  /rm_rs_d435_eye_on_hand/
    handeye_publisher_realman_laptop_14505_503373985637918066 (easy_handeye/publish.py)

ROS_MASTER_URI=http://localhost:11311

process[rm_rs_d435_eye_on_hand/handeye_publisher_realman_laptop_14505_503373985637918066-1]: started with pid [14520]
[INFO] [1639576159.067217]: Loading the calibration from file: /home/realman/.ros/easy_handeye/rm_rs_d435_eye_on_hand
.yaml
[INFO] [1639576159.084798]: loading calibration parameters into namespace /rm_rs_d435_eye_on_hand/
[INFO] [1639576159.087175]: Storing calibration /rm_rs_d435_eye_on_hand/ into the parameters server
[INFO] [1639576159.095454]: Storing parameters for calibration /rm_rs_d435_eye_on_hand/ into the parameters server
```

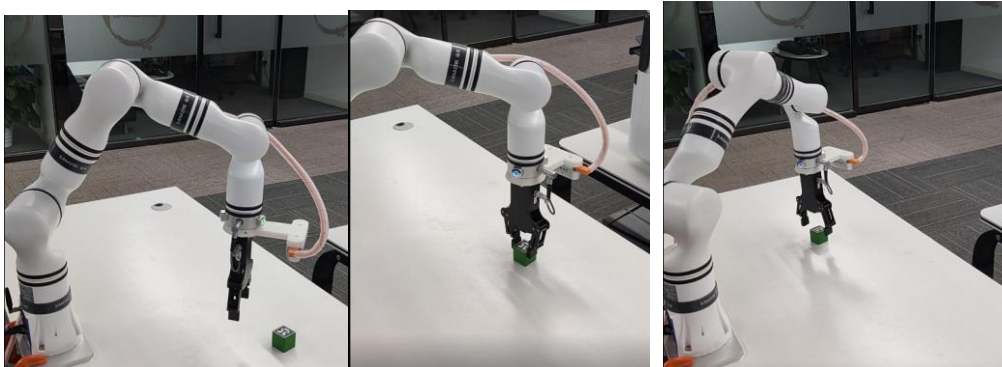


1.2.6 启动二维码识别定位及抓取节点

再打开一个新的终端，执行以下命令实现目标物体定位及抓取功能：

```
cd ~/catkin_ws
source devel/setup.bash
roslaunch rm_visual_demo realsense_recognize_and_pick_demo
```

运行结果如下图所示，机械臂首先运行到初始位姿，然后通过 **realsense** 相机识别并定位附着二维码的目标物体，机械臂到达目标物体位姿后，通过夹爪夹取物体，并将其放置到目标位置。



1.3 关键代码

1) 在机械臂抓取之前，必须先打开夹爪

```
//确保抓手打开才能进行下一步操作
if (!isGripperOpen)
{
    ROS_INFO("*****Set Gripper Open*****");

    gripperSet.position = 1000;
    gripper_set_pub.publish(gripperSet); //打开抓手

    ros::Duration(3.0).sleep();
    isGripperOpen = true;
}
```

2) 打开夹爪之后，机械臂运行到初始位姿，并在该位姿下识别目标二维码

```
/**
 * 2.控制机械臂运动到识别抓取的初始位姿
 */
//判断是否在起始位姿，如果不在则先控制机械臂移动到初始位姿
while (!isBeginPose)
{
    ROS_INFO("*****MoveJ to BeginPose Start*****");
    plan_state = PLAN_ING;
    //发布空间规划指令运动到起始位姿
    moveJ_pub.publish(moveJ_BeginPose);

    //等待规划运动完成返回
    while (plan_state == PLAN_ING && wait_count < 10)
    {
        ros::Duration(1.0).sleep();
        wait_count++;
    }
    wait_count = 0;

    if (plan_state == PLAN_OK)
    {
        isBeginPose = true;
    } else {
        plan_state = PLAN_FAIL;
    }
}
```



- 3) 机械臂识别物体，该过程若为识别到物体，则一直等待，直到识别到目标物体。

```

    ros::Duration(1.0).sleep();
    /**
     * 3.等待2秒,然后判断是否识别到抓取物,如果没有识别到marker标记物则一直等待直到识别到
     */
    findMarkerObj = false;
    updateMarkerPos_flag = true;
    ros::WallDuration(1.0).sleep(); //等待1秒
    //如果没有识别到marker标记物则一直等待直到识别到
    while (!findMarkerObj) {
        ros::WallDuration(2.0).sleep(); //等待2秒
    }
    updateMarkerPos_flag = false;

```

- 4) 识别到物体后，将目标物体在相机坐标系下的位姿转换为其在基坐标系下的位姿。

```

/**
 * 4.开始转换坐标系并控制机械臂抓取放置操作
 */
try {

    //获取base_link和camera_color_frame之间的关系，也就是手眼标定的结果
    ROS_INFO("waitForTransform ok [base_link, camera_color_frame]");
    tfListener.waitForTransform("/base_link", "/camera_color_frame", ros::Time(0), ros::Duration(50.0));

    tfListener.transformPose("base_link", marker_pose, transed_pose);
    ROS_INFO("Transed pose->position[%f, %f, %f]", transed_pose.pose.position.x, transed_pose.pose.position.y,
        transed_pose.pose.position.z);
    ROS_INFO("Transed pose->orientation[%f, %f, %f]", transed_pose.pose.orientation.x,
        transed_pose.pose.orientation.y, transed_pose.pose.orientation.z, transed_pose.pose.orientation.w);
}

```

- 5) 机械臂运行到目标位姿并进行抓取。

```

//发布直线规划指令运动到抓取位姿
ROS_INFO("*****MoveL to PickPose Start*****");
plan_state = PLAN_ING;
rm_msgs::MoveL moveL_pickPose;
moveL_pickPose.Pose = pickPose;
moveL_pickPose.speed = 0.5;
moveL_pub.publish(moveL_pickPose);

```

- 6) 机械臂运行至目标位姿，并松开夹爪，放置物体。

```

if (plan_state == PLAN_OK) {
    //发布空间规划指令运动到放置位姿
    ROS_INFO("*****MoveL to PlacePose Start*****");
    plan_state = PLAN_ING;
    moveJ_pub.publish(moveJ_PlacePose);

    while (plan_state == PLAN_ING && wait_count < 10) {
        ros::Duration(1.0).sleep();
        wait_count++;
    }
    wait_count = 0;

    plan_state = (plan_state == PLAN_OK) ? PLAN_OK : PLAN_FAIL;
}

```

至此，对物体的识别以及抓取完成，在使用过程中需要注意夹爪的长度，并对识别位姿在 z 方向的数值做补偿，放置机械臂下降高度过大，与桌面等发生碰撞。



二、颜色识别及无序分拣

2.1 概述

该部分机械臂通过颜色识别并抓取目标物体，然后根据颜色对物体进行分类。目标物体为颜色不同的小木块，通过 **realsense** 相机获取 **RGB** 图像以及深度图像，确定物体在在相机坐标系下的坐标，并通过相机与基坐标系的转换关系，求得目标物体在基坐标系下的位置，控制机械臂到达该位置并抓取，然后根据识别到物体的颜色，将物体放置到指定位置，实现对目标物体的颜色识别及无序分拣功能。在程序执行过程中，可以通过发布话题，控制执行程序进行开始、暂停、继续以及结束操作。

2.2 功能实现

在运行程序之前，需要先对相机进行标定工作，确定相机坐标系与基坐标系的转换关系。标定过程详见《RM 机械臂与 Realsense D435 手眼标定教程》。

2.2.1 打开深度相机

打开新终端，输入以下指令打开 **realsense** 相机。

```
cd ~/catkin_ws
source devel/setup.bash
roslaunch realsense2_camera rs_camera.launch
```

```
/home/realman/catkin_ws/src/realsense-ros/realsense2_camera/launch/rs_camera.launch http://localhost:11311
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[INFO] [1639471966.582111583]: Built with LibRealSense v2.50.0
[INFO] [1639471966.582154314]: Running with LibRealSense v2.50.0
[INFO] [1639471966.608428472]: 
[INFO] [1639471966.614174509]: Device with serial number 137322078962 was found.
[INFO] [1639471966.614247944]: Device with physical ID /sys/devices/pci0000:00/0000:00:14.0/usb3/3-1/3-1:1.0/video4linux/vid
bo0 was found.
[INFO] [1639471966.614293765]: Device with name Intel RealSense D435 was found.
[INFO] [1639471966.614935268]: Device with port number 3-1 was found.
[INFO] [1639471966.614991985]: Device USB type: 3.2
[INFO] [1639471966.618542300]: getParameters...
[INFO] [1639471966.672613411]: setupDevice...
[INFO] [1639471966.672663406]: JSON file is not provided
[INFO] [1639471966.672699541]: ROS Node Namespace: camera
[INFO] [1639471966.672738873]: Device Name: Intel RealSense D435
[INFO] [1639471966.672779843]: Device Serial No: 137322078962
[INFO] [1639471966.672797659]: Device physical port: /sys/devices/pci0000:00/0000:00:14.0/usb3/3-1/3-1:1.0/video4linux/video
[INFO] [1639471966.672826760]: Device FW version: 05.13.00.50
[INFO] [1639471966.672855684]: Device Product ID: 0x0B07
[INFO] [1639471966.672880771]: Enable PointCloud: Off
[INFO] [1639471966.672902753]: Align Depth: Off
[INFO] [1639471966.672941290]: Sync Mode: Off
[INFO] [1639471966.673001089]: Device Sensors:
[INFO] [1639471966.675687660]: Stereo Module was found.
[INFO] [1639471966.681077192]: RGB Camera was found.
[INFO] [1639471966.681142596]: (Confidence, 0) sensor isn't supported by current device! -- Skipping...
[INFO] [1639471966.681200985]: num_filters: 0
[INFO] [1639471966.681241109]: Setting Dynamic reconfig parameters.
[INFO] [1639471966.732648946]: Done Setting Dynamic reconfig parameters.
[INFO] [1639471966.735070792]: depth stream is enabled - width: 848, height: 480, fps: 30, Format: Z16
[INFO] [1639471966.735891283]: color stream is enabled - width: 640, height: 480, fps: 30, Format: RGB8
[INFO] [1639471966.735960930]: setupPublishers...
[INFO] [1639471966.739830103]: Expected frequency for depth = 30.00000
[INFO] [1639471966.786686451]: Expected frequency for color = 30.00000
[INFO] [1639471966.808855024]: setupStreams...
[INFO] [1639471966.936163645]: SELECTED BASE:Depth, 0
[INFO] [1639471966.954314706]: RealSense Node Is Up!
[WARN] [1639471967.007367169]:
```

2.2.2 启动 RM 机械臂的 control 和 driver 节点

打开一个新的终端，执行以下命令启动 **control** 节点：

```
cd ~/catkin_ws
source devel/setup.bash
```



```
roslaunch rm_control rm_control.launch
```

```

/home/realman/catkin_ws/src/rm_65_robot/rm_control/launch/rm_control.launch http://localhost:11311
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
realman@ubuntu:~$ cd catkin_ws/
realman@ubuntu:~/catkin_ws$ source devel/setup.bash
realman@ubuntu:~/catkin_ws$ roslaunch rm_control rm_control.launch
.. logging to /home/realman/.ros/log/38da116a-408d-11ed-891c-000c2998aff3/roslaunch-ubuntu-16592.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ubuntu:46711/

SUMMARY
=====

PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.13

NODES
/
  rm_control (rm_control/rm_control)

ROS_MASTER_URI=http://localhost:11311

process[rm_control-1]: started with pid [16607]
```

2.2.3 启动 easy_handeye 节点

再打开一个新的终端，执行以下命令发布手眼标定后的 TF：

```
roslaunch easy_handeye publish.launch
```

```

/home/realman/catkin_ws/src/easy_handeye/easy_handeye/launch/publish.launch http://localhost:11311
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 标签(B) 帮助(H)
/home/realman/... x /home/realman/... x realman@real... x /home/realman/... x /home/realman/... x /home/realman/... x
realman@realman-laptop:~/catkin_ws$ roslaunch easy_handeye publish.launch
.. logging to /home/realman/.ros/log/b9a2478a-5da9-11ec-9e13-b9364f7760b1/roslaunch-realman-laptop-14505.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://realman-laptop:41179/

SUMMARY
=====

PARAMETERS
* /rm_rs_d435_eye_on_hand/calibration_file:
* /rm_rs_d435_eye_on_hand/eye_on_hand: True
* /rm_rs_d435_eye_on_hand/inverse: False
* /rm_rs_d435_eye_on_hand/robot_effector_frame:
* /rm_rs_d435_eye_on_hand/tracking_base_frame:
* /rostdistro: melodic
* /rosversion: 1.14.12

NODES
/
  rm_rs_d435_eye_on_hand/
    handeye_publisher_realman_laptop_14505_503373985637918066 (easy_handeye/publish.py)

ROS_MASTER_URI=http://localhost:11311

process[rm_rs_d435_eye_on_hand/handeye_publisher_realman_laptop_14505_503373985637918066-1]: started with pid [14520]
[INFO] [1639576159.067217]: Loading the calibration from file: /home/realman/.ros/easy_handeye/rm_rs_d435_eye_on_hand.yaml
[INFO] [1639576159.084798]: loading calibration parameters into namespace /rm_rs_d435_eye_on_hand/
[INFO] [1639576159.087175]: Storing calibration /rm_rs_d435_eye_on_hand/ into the parameters server
[INFO] [1639576159.095454]: Storing parameters for calibration /rm_rs_d435_eye_on_hand/ into the parameters server
```

2.2.4 启动无序分拣功能节点

再打开一个新的终端，执行以下命令打开无序分拣功能节点。

```

cd ~/catkin_ws
source devel/setup.bash
roslaunch rm_visual_demo disorderly_sorting
```



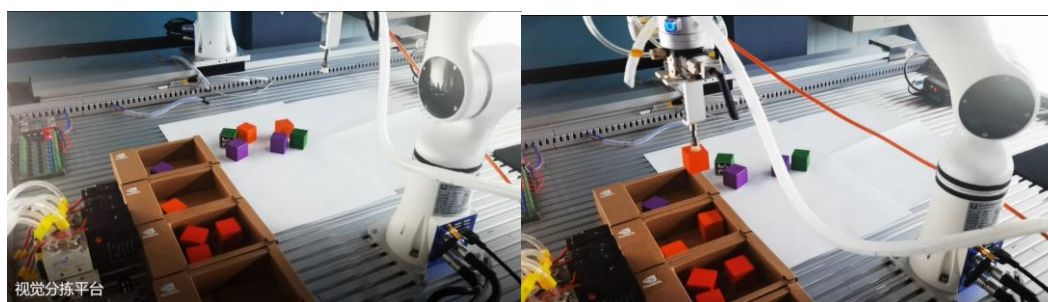

2.2.5 启动控制节点

再打开一个新的终端，执行以下命令发送控制指令，控制程序的开始、暂停、继续以及结束。其中，1 代表开始，2 代表暂停，3 代表继续，4 代表结束。执行该步骤之前需要在 rm_msg 文件夹中创建 Socket_Command.msg 消息文件，该消息文件中包含 int 类型的变量 command，创建完成后需要编译 rm_msg 功能包。

```
cd ~/catkin_ws
source devel/setup.bash
rostopic pub -1 TcpCommand rm_msgs/Socket_Command 1
```

```
realman@ubuntu: ~/catkin_ws
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
realman@ubuntu:~$ cd catkin_ws/
realman@ubuntu:~/catkin_ws$ source devel/setup.bash
realman@ubuntu:~/catkin_ws$ rostopic pub -1 TcpCommand rm_msgs/Socket_Command 1
publishing and latching message for 3.0 seconds
realman@ubuntu:~/catkin_ws$
```

程序执行时，通过发送指令，控制机械臂的执行过程。机械臂在执行过程中通过颜色识别，识别特定颜色的小木块，并将其抓取放置到指定位置，某一颜色木块抓取完成后，继续抓取并放置下一颜色木块。



2.3 关键代码

- 1) 通过订阅话题，实时改变任务阶段，实现对程序执行过程的控制



```

void TcpComCallback(const rm_msgs::Socket_Command::ConstPtr& msg)
{
    //开始
    if(msg->command == 1)
    {
        task = 3;
        ROS_INFO("start");
    }
    //暂停
    else if(msg->command == 2)
    {
        pause_sort = 1;
        ROS_INFO("pause");
    }
    //继续
    else if(msg->command == 3)
    {
        pause_sort = 0;
        ROS_INFO("continue");
    }
    //结束
    else if(msg->command == 4)
    {
        task = 4;
        ROS_INFO("end");
    }
}

```

2) 同步订阅 RGB 图像与深度图像话题，实现图像的同步处理。

//同步处理RGB图与深度图

```

message_filters::Subscriber<sensor_msgs::Image> img_sub(nh, "/camera/color/image_raw", 1);
message_filters::Subscriber<sensor_msgs::Image> depth_sub(nh, "/camera/depth/image_rect_raw", 1);
TimeSynchronizer<sensor_msgs::Image, sensor_msgs::Image> sync(img_sub, depth_sub, 10);
sync.registerCallback(boost::bind(&callback, _1, _2));

```

3) 在执行抓取之前，机械臂首先通过监听坐标变换获取初始位姿下，相机坐标系至基坐标系的变换关系。

```

Sophus::SE3d get_transpose()
{
    //监听坐标变换
    tf::TransformListener listener;
    tf::StampedTransform transform;

    try
    {
        listener.waitForTransform("/base_link", "/camera_color_frame", ros::Time(0), ros::Duration(30));
        listener.lookupTransform("/base_link", "/camera_color_frame", ros::Time(0), transform);

        get_trans = 1;

        ROS_INFO("Transed pose->position[%f,%f, %f, %f, %f, %f, %f]",
            transform.getRotation().getW(),
            transform.getRotation().getX(),
            transform.getRotation().getY(),
            transform.getRotation().getZ(),
            transform.getOrigin().x(),
            transform.getOrigin().y(),
            transform.getOrigin().z());
    }
    catch (tf::TransformException &ex)
    {
        ROS_ERROR("[adventure_tf]: (wait) %s", ex.what());
        ros::Duration(1.0).sleep();
    }
}

```

3) 根据 `task_step` 的值判断机械臂执行抓取过程中的步骤。在图像识别过程中，首先按顺序依次识别不同颜色，然后获取特定颜色下各个小物块的中心位置。然后通过相机内参将图像坐标系下的点转换到相机坐标系下，然后通过转换矩阵，将相机坐标系下的点转换到基坐标系下，再控制机械臂到达指定位置并吸取小物块。



```
//绘制轮廓
drawContours(img_BGR, contours, i, CV_RGB(255, 0, 0), 2, 8, hierarchy, 0);

// 计算每个轮廓所有矩
// 创建一个vector,元素个数为contours.size(),获得轮廓的所有最高达三阶所有矩
mu[i] = moments(contours[i], false );

// 计算轮廓的质心,质心的 x,y 坐标: (m10/m00, m01/m00)
mc[i] = Point2f(static_cast<float>(mu[i].m10/mu[i].m00), static_cast<float>(mu[i].m01/mu[i].m00));
//ROS_INFO("area is : %f", area);

//计算世界坐标
int u = (int)mc[i].x;
int v = (int)mc[i].y;
// 深度值
unsigned int depth_point = img_depth.at<uint16_t>(u, v);

//相机坐标系下坐标值
Eigen::Vector3d point;
point[2] = double(depth_point) / depthScale;
point[0] = (u - cx) * point[2] / fx;
point[1] = (v - cy) * point[2] / fy;

//世界坐标系下坐标值
Eigen::Vector3d pointWorld = trans_camera2base * point;
```

4) 依次执行识别、抓取以及放置动作，直至所有小木块被放置到指定位置。