**腾讯视觉抓取Demo说明及注意事项**

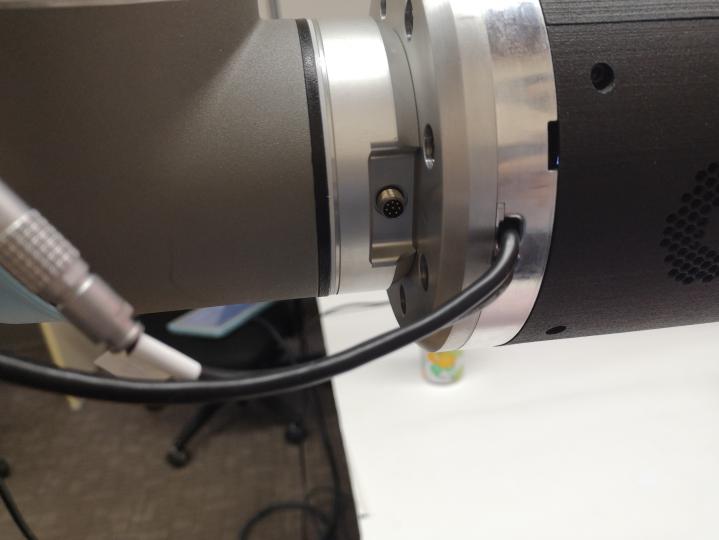
****

**2018.11.11**

1. **硬件组成与连接**

硬件由四部分组成，RGBD摄像头，5指Shadow手，UR10机械臂以及一台移动PC。本次Demo使用的相机的型号为Realsense SR300,通过USB与控制PC通讯。5指Shadow手通过Ehternet接口与控制PC连接。UR10机械臂通过Ehternet转USB与控制PC通讯。这里需要说明的是Shadow手与UR机械臂都是要通过Ehternet接口通讯，由于本次的PC只有一个Ehternet接口，所以我们在连接UR10与PC时使用了Ehternet转USB转换器。（Shadow手必须通网口直接与控制PC相连，不能使用转换器）。

Shadow手与机械臂的硬件固定请参看下图所示安装（机械臂末端的I/O口须与Shadow出料的两根线大体在同一水平位置上）：



1. **软件**

本次Demo软件部分主要由四部分组成，分别是相机标定，物体识别，tf转换以及机械臂规划抓取。

2.1 相机标定：此部分使用了handeye进行相机的标定，通过此包建立相机与机械臂之间的tf连接。

2.2 物体识别：此部分使用了ROS包find\_object\_2d，（[Github链接](https://github.com/introlab/find-object.git)）通过彩色图像获取特定目标的特征点，通过深度图像建立目标与相机之间的tf连接。

2.3 tf转换：此部分使用了tf 的lookupTransform, （[Github连接](https://github.com/DongMuJi/tf_transfom)）实现目标与机械臂底座之间的tf转换。

2.4机械臂规划抓取：此部分使用了机械臂与手的Moveit包，调用高级接口库，实现对目标的抓取。

1. **实现过程**

3.1硬件连接后，使用UR10示教器设置UR的IP, 此Demo中设置的UR臂IP为192.168.1.1,Netmask为255.255.255.0。设置控端PC的网络，与Shadow连接的网口定义为DHCP，与机械臂连接的网口定义为静态IP，此Demo中静态IP设置为192.168.1.100，Netmask为255.255.255.0。（此Demo中我们修改了电脑的网络配置文件/etc/network/interfaces,用来满足网络配置要求),可以改变机械臂与PC的IP（同网段），这个IP定义文件在sr\_config/sr\_ur\_arm\_config/config/ur\_arm\_1\_robot\_hw.yaml中，其中robot\_ip\_address为机械臂的IP，control\_pc\_ip\_address为控制端PC的IP。

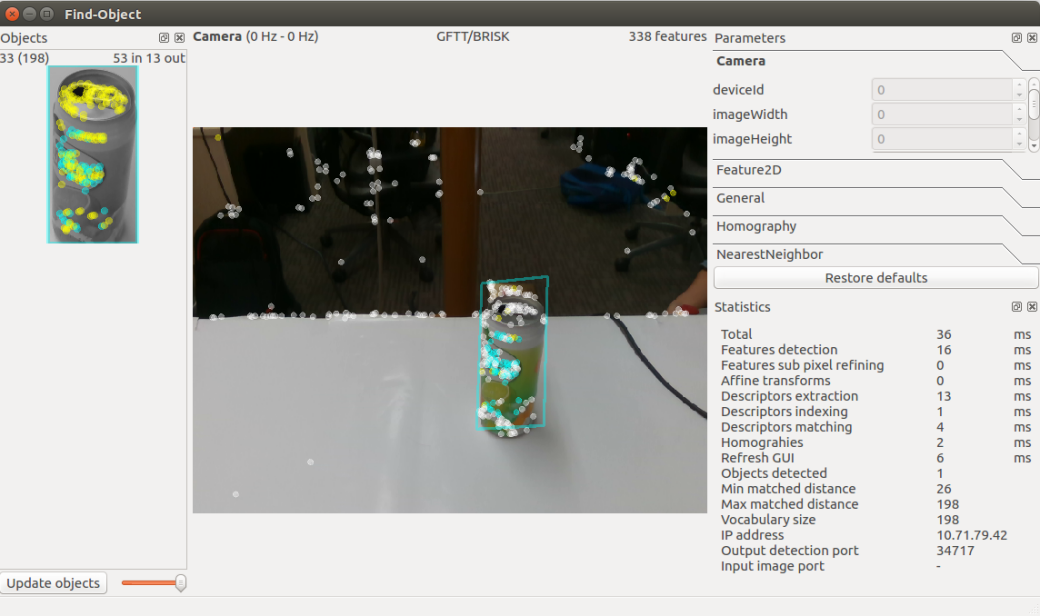
3.2启动机械臂和Shadow手确定端口配置没有问题，roslaunch sr\_robot\_launch sr\_right\_ur10arm\_hand.launch sim:=false hand\_serial:=2058 eth\_port:=enp0s31f6，eth\_port为Shadow手与PC连接的网络端口的名字。（Shadow手启动后会进行初始化，有一小段时间的抖动，通过RVIZ查看，界面中显示的机械臂和手的状态必须和实际机械臂与手的状态一致，否则表示为启动失败）。确定能正常启动后，关闭启动文件。

3.3 启动相机和机械臂，使用handeye进行相机的标定，得到相机基于机械臂底座的Pose。单独机械臂的启动可以使用src/sr\_ur\_arm/sr\_ur\_launch/launch文件夹下的right\_arm.launch。请注意将启动文件中的use\_moveit的值改为false。然后调用service使机械臂进入示教模式，命令为rosservice call /ra\_sr\_ur\_robot\_hw/set\_teach\_mode “teach\_mode: true” ,然后启动handeye中的start\_calibration文件，手动拖动机械臂进行相机标定。完成后请结束相机与机械臂的进程。（该部分相机标定，也可单独使用ur\_morden\_driver来驱动机械臂来实现对机械臂的控制）。

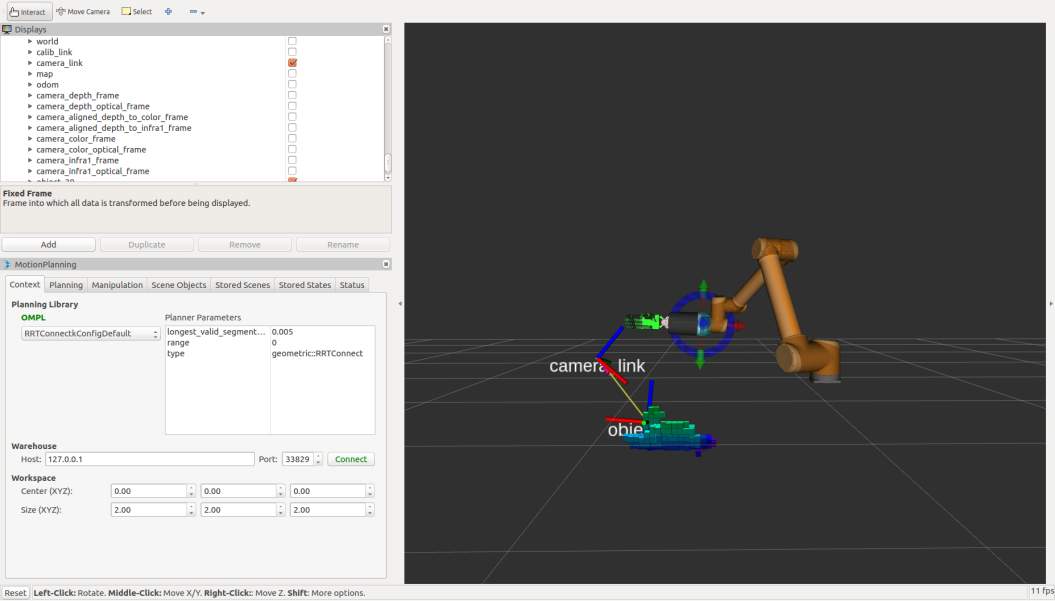
3.4确定相机已启动，通过rostopic和rqt\_image\_view确定以下三个相应话题rgb/image\_rect\_color（[sensor\_msgs/Image](http://docs.ros.org/api/sensor_msgs/html/msg/Image.html)），depth\_registered/image\_raw（[sensor\_msgs/Image](http://docs.ros.org/api/sensor_msgs/html/msg/Image.html)），depth\_registered/camera\_info（[sensor\_msgs/CameraInfo](http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html)）。本次Demo中我们使用的三个话题分别是camera/color/image\_raw, camera/aligned\_depth\_to\_color/image\_raw,camera/camera/aligned\_depth\_to\_color/image\_info。

3.5 修改find\_object\_2d包中的find\_object\_3d.launch文件，remap三个与相机相关的话题，添加相机与机械臂base之间的静态tf转换，注意相机的摄像头朝向为相机的Frame的x方向，满足右手定则，如果相机标定后的Frame结果不满足要求，则需要再进行一次转换，此次Demo中，我们做了相关的转换（ra\_base\_link to 。

3.6 确保相机已启动，启动find\_object\_3d.launch,关于这个包的使用，请查看Computer Vision Task for Manipulation.pdf文档，在选取相应的目标后，该包自动给目标进行编号，其发布相应的Frame ID，类似/object\_34, 如果不对选取的Object进行保存，则在每次选取目标的时候，其编号依次累加，可以通过保存Object,在下次进行目标识别时，加载保存好的Object(一张png图像文件）来保证编号不变。（图像的编号即为其发布相应的Frame ID）。如若检测到相关目标，则会如下图所示，目标物上出现一个矩形框。



3.7 修改tf\_listener.launch文件中的original的value值，其为目标物的Frame ID。该文件默认情况下以1Hz的频率向/tf\_transform/tf\_result话题发布从original\_frame到destination\_frame的tf转换信息。启动tf\_listener.launch。如果一切设置正确，则应该在RVIZ中看到相应目标物的Frame ID的tf连接到了机械臂的tf树中，如下图所示：



3.8 确保Shadow已启动，启动rqt, 加载Shadow的控制插件，利用控制插件控制记录输出手的每个关节位置。关于该插件的使用，请查看saving object pose.odt

3.9 调用相应的python接口库（其接口库在sr\_interface / sr\_robot\_commander / src / sr\_robot\_commander文件夹下，其调用示例在sr\_interface/sr\_robot\_commander/doc/tutorial文件夹下），订阅话题/tf\_transform/tf\_result，实现对目标物的抓取。目前该代码每次只能进行一次抓取（代码为pose\_sr\_srm.py)，整个过程分为9个状态：

A. 手进入准备状态

B. 臂回到Home位置（我们设置的一个固定位置）

C. 臂到达拾取目标的位置(该位置通过订阅/tf\_transform/tf\_result话题得到，然后通过测试做一些数值修改，以便之后能进行更好的抓取）

D. 手抓取目标

E. 臂到Home位置

F. 臂到放置目标的位置（我们设置的一个固定位置）

G. 手进入准备状态

H. 手进入打开状态

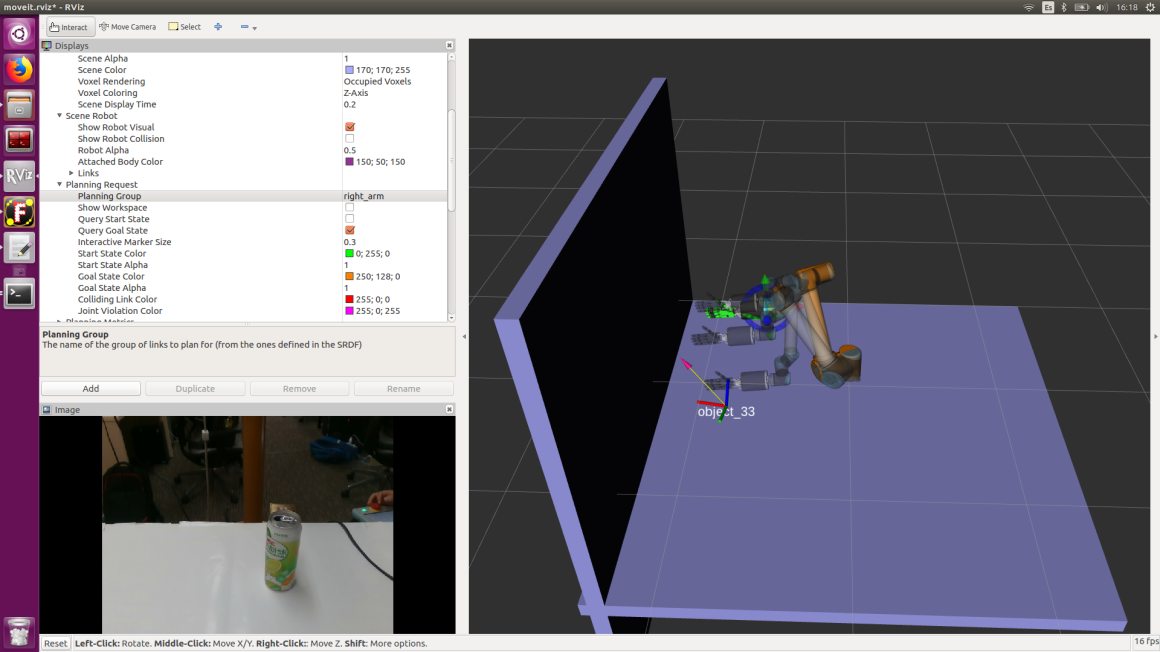
I. 臂回到Home位置

3.10 启动相机，机械臂和shadow，启动find\_object\_3d.launch,选取目标物后，修改tf\_listener.launch参数,启动tf\_listeber.launch,启动pose\_sr\_srm.py，实现物体识别和抓取的整个过程。

1. **快速启动**

在没有改变相关参数的情况下（没有改变相关参数表示，使用Shadow配置的电脑，使用我们调试时使用的相机及其位置，使用我们Demo演示时所用的识别物体，使用我们对于机械臂选择的固定位置，使用我们提供的Ehternet转USB转换器），可以使用以下步骤进行快速启动：

1. 启动Tencent\_demo.launch文件（该文件启动机械臂，Shadow手，相机以及识别模块）。
2. 加载已有的标物，object\_33，启动tf\_listener.launch文件，确保在RVIZ中看到Object\_33的tf。
3. 启动pose\_sr\_arm.py文件，进行机械臂与手的规划与抓取。会看到如下图所示的RVIZ界面



1. **存在的问题**
2. 由于本次使用了Moveit的move\_group接口进行机械臂的运动规划，机械臂的运动规划的路径存在不确定性，特别是在使用move\_to\_target\_pose时，Moveit有时并不会选取IK解算器得出的最短的路径，目前我们选择添加外部障碍物来局限其规划路径，其有一定的效果，但对机械臂的运动局限比较大并不是最可取的方式。
3. 本次使用的UR的ROS驱动包为ur\_driver（[Github连接](https://github.com/shadow-robot/universal_robot)，UR的Scop版本为3.5），配合该驱动包使用Moveit进行机械臂运动规划时，机械臂的示教器有时会出现Protect Stop，目前通过网上资料发现许多人有同样的问题，这是由于驱动包本身存在的Bug，使用ur\_modern\_driver可改善这个问题（对于是否能完全解决这个问题还是需要讨论的，因为在Github的ur\_morden\_driver的Issue中也有人说同样遇到了这个问题）。由于目前的Shadow的驱动与UR的驱动做了比较完全的结合，目前我们无法更换现有的ur\_driver。
4. 机械臂在规划运动时，有时机械臂在运动过程中会出现卡顿的现象，这个问题的产生是由于给机械臂发布的命令的频率小于125Hz，这个问题也是由于目前的UR机械臂的驱动包造成的。目前无法进行改善解决。
5. **注意事项**

为了更加安全的运行Demo，特做以下几点注意事项说明，请在使用过程中注意：

1. 请在使用设备前仔细阅读相关的操作手册和说明书。
2. 在使用过程中，请时刻注意机械臂的状态，确保有人能及时对臂进行急停操作，以防危险发生。
3. 请在机械臂初始化时设置机械臂的Current payload为4.55Kg(虽然在启动机械臂和手的launc文件时会自动设置，但我们建议最好在机械臂使用示教器初始化时就进行相应的设置)。
4. 在运行Demo前请确保机械臂和Shadow处在一个合适的抓取物体的rotation。
5. 在机械臂急停或者出现警告（例如Protect Stop)，再次使能机械臂后，需要重新启动相关的launch文件，否则无法再次使用ROS控制机械臂。
6. 目前机械臂无法同时通过ROS和示教器控制，在启动相关的launch文件后，如果期间使用示教器控制机械臂，ROS控制就会失效，此时需要退出示教器的控制，重新启动相关的launch文件。
7. 请不要使用Shadow手抓取超过其最大负载限制的物体（4Kg)。
8. Shadows手在使用一段时间后，其肌腱线可能会发生松动（这属于正常现象），请参照手册的教程对其进行重新拉紧操作。可能需要使用软件进行校准，请参考Shadow的使用手册说明进行相应操作。