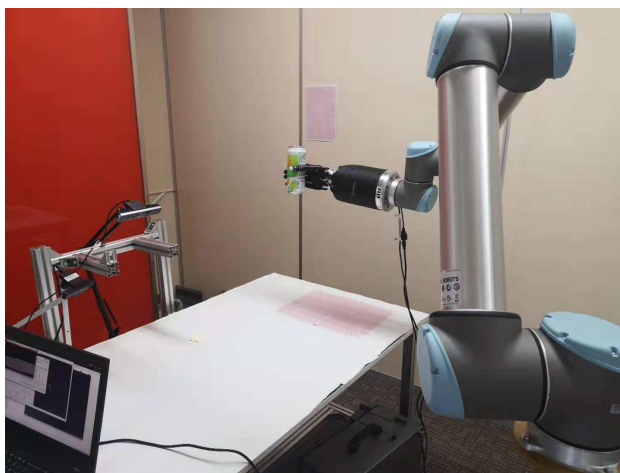


腾讯视觉抓取 Demo 说明及注意事项

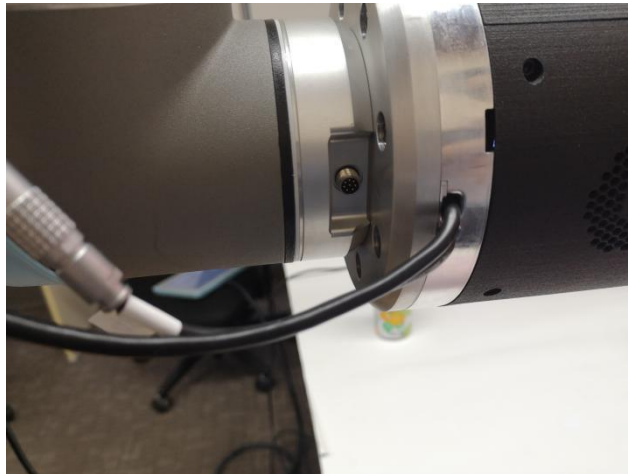


2018.11.11

一、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 Ethernet 接口与控制 PC 连接。UR10 机械臂通过 Ethernet 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 Ethernet 接口通讯,由于本次的 PC 只有一个 Ethernet 接口，所以我们在连接 UR10 与 PC 时使用了 Ethernet 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)：



二、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](#)) 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 链接](#)) 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。

三、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。

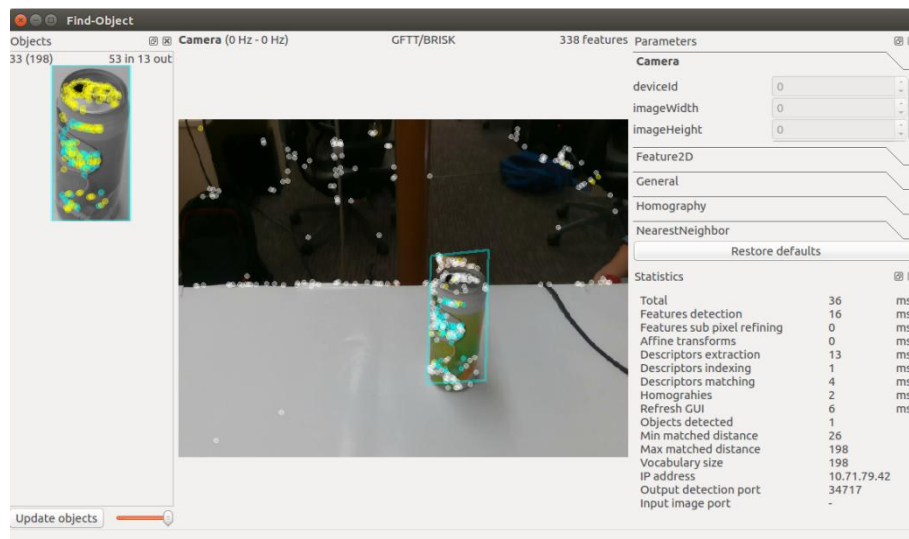
3.2 启动机械臂和 Shadow 手确定端口配置没有问题，`roslaunch sr_robot_launch sr_right_ur10arm_hand.launch sim:=false hand_serial:=2058 eth_port:=enp0s31f6`, eth_port 为 Shadow 手与 PC 连接的网络端口的名字。（Shadow 手启动后会进行初始化，有一小段时间的抖动，通过 RVIZ 查看，界面中显示的机械臂和手的状态必须和实际机械臂与手的状态一致，否则表示为启动失败）。确定能正常启动后，关闭启动文件。

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）。

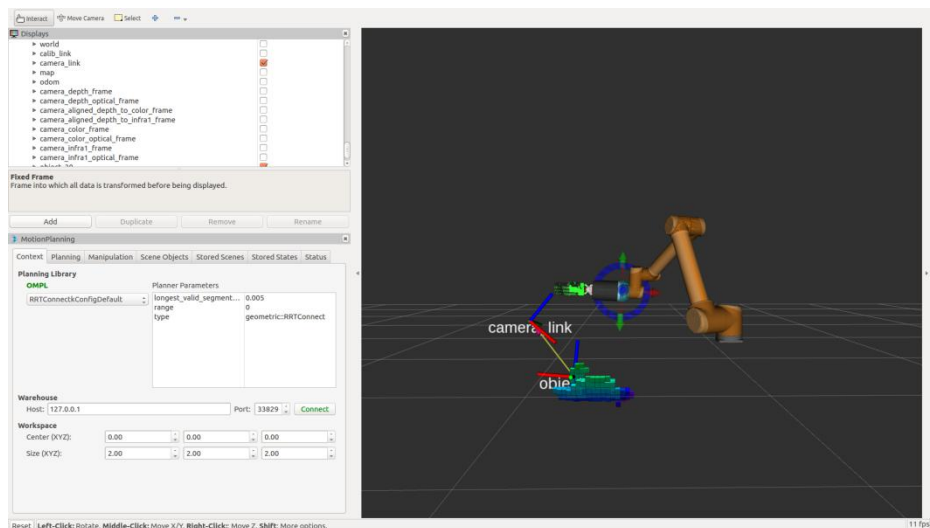
3.4 确定相机已启动，通过 rostopic 和 rqt_image_view 确定以下三个相应话题 `rgb/image_rect_color`（`sensor_msgs/Image`），`depth_registered/image_raw`（`sensor_msgs/Image`），`depth_registered/camera_info`（`sensor_msgs/CameraInfo`）。本次 Demo 中我们使用的三个话题分别是 `camera/color/image_raw`, `camera/aligned_depth_to_color/image_raw`, `camera/camera/aligned_depth_to_color/image_info`。

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：

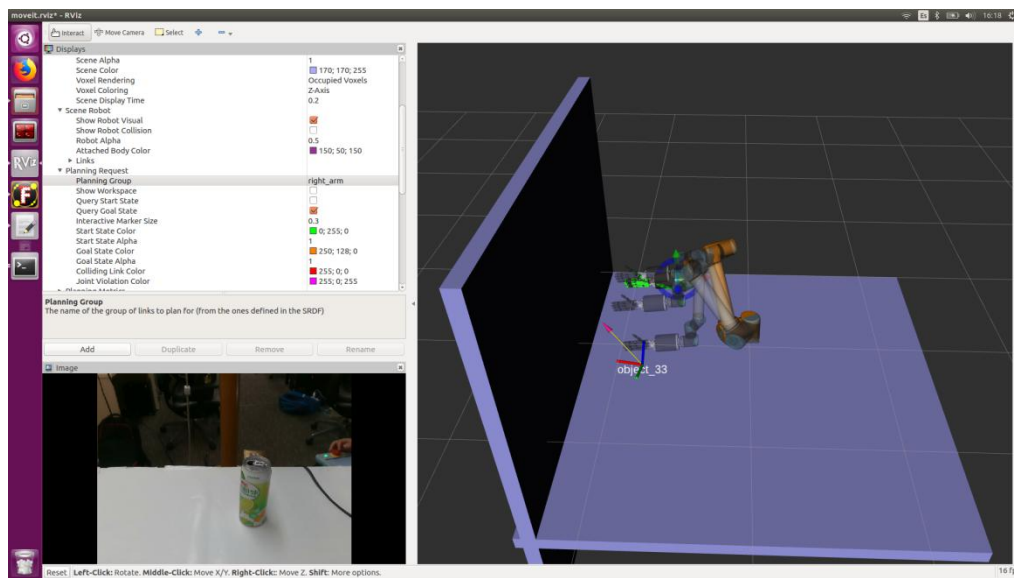
- A. 手进入准备状态
- B. 臂回到 Home 位置（我们设置的一个固定位置）
- C. 臂到达拾取目标的位置(该位置通过订阅/tf_transform/tf_result 话题得到，然后通过测试做一些数值修改，以便之后能进行更好的抓取)
- D. 手抓取目标
- E. 臂到 Home 位置
- F. 臂到放置目标的位置（我们设置的一个固定位置）
- G. 手进入准备状态
- H. 手进入打开状态
- I. 臂回到 Home 位置

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全过程。

五、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 Ethernet 转 USB 转换器），可以使用以下步骤进行快速启动：

- A. 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）。
- B. 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。
- C. 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



六、存在的问题

1、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。

2、本次使用的 UR 的 ROS 驱动包为 ur_driver（[Github 连接](#)，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`。

3、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。

七、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：

- A. 请在使用设备前仔细阅读相关的操作手册和说明书。
- B. 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。
- C. 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)。
- D. 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。
- E. 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。
- F. 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。
- G. 请不要使用 Shadow 手抓取超过其最大负载限制的物体（4Kg）。
- H. 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。