

【手眼标定】ros / easy_handeye + ur5 + realsense d435i



丁洪凯

关注他

3 人赞同了该文章

0. 手眼标定原理

参考 flyqq 的文章：

3D 视觉之手眼标定
mp.weixin.qq.com



重点：

- **Eye-to-hand 眼在手外**：标定的是相机坐标系相对于**机器人基座坐标系**的位姿
- **Eye-in-hand 眼在手上**：标定的是相机坐标系相对于**机器人工具坐标系**的位姿
- 自己总结的标定原则
 - 确定哪些是**已知**的？哪些是**未知**的？哪些是**变化**的？哪些是**不变**的？哪些是**需要求解**的？哪些是**不需要求解**的？
 - 标定的是**未知且不变**的 TF（Transformation）
 - 恒等式关系
 - eye-to-hand：相机坐标系 <> 机器人基座坐标系（未知，不变，需要求解）
 - eye-in-hand：Marker坐标系 <> 机器人基座坐标系（未知，不变，不需要求解）

1. 安装配置

- 源码编译 ur5 & ur_modern_driver

```
cd ~/catkin_ws/src
git clone -b kinetic-devel https://github.com/ros-industrial/universal_robot.git
cd universal_robot
git clone -b kinetic-devel https://github.com/ros-industrial/ur_modern_driver.git

cd ~/catkin_ws
catkin_make
echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
source ~/.bashrc
```



· realsense-ros

丁洪凯: Ubuntu 16.04 + Realsense
D435i + ROS Kinetic 环境配置
zhuanlan.zhihu.com



· aruco_ros

```
cd ~/catkin_ws/src
git clone -b kinetic-devel https://github.com/pal-robotics/aruco_ros.git
cd ..
catkin_make
```

· vision_visp / visp_hand2eye_calibration

```
cd ~/catkin_ws/src
git clone -b kinetic-devel https://github.com/lagadic/vision_visp.git
cd ..
catkin_make --pkg visp_hand2eye_calibration
```

· easy_handeye

```
cd ~/catkin_ws/src
git clone https://github.com/IFL-CAMP/easy_handeye
cd ..
catkin_make
```

2. 眼在手外 eye-to-hand

· 2.1 修改标定 launch 文件

标定过程需启动 ur5 机械臂的相关节点, realsense 节点, aruco 节点, easy_handeye 节点, 可以写一个 launch 文件同时启动上述节点, 也可以分别启动。easy_handeye 包中给出了用一个 launch 文件实现的示例, 在如下的目录中:

xxx/easy_handeye/docs/example_launch/ur5_kinect_calibration.launch

我们可以在此基础上进行修改。

```
# 将 launch 文件拷贝到 easy_handeye 功能包的 launch 目录中, 顺便改个名字
cd ~/catkin_ws/src/easy_handeye/docs/example_launch
cp ur5_kinect_calibration.launch ~/catkin_ws/src/easy_handeye/easy_handeye/laun
```

修改 launch 文件如下:

```
<launch>
  <arg name="namespace_prefix" default="ur5_realsense_handeyecalibration" />

  <arg name="robot_ip" doc="The IP address of the UR5 robot" />

  <arg name="marker_size" doc="Size of the ArUco marker used, in meters" defa
  <arg name="marker_id" doc="The ID of the ArUco marker used" default="571"/>

  <!-- 1. start the Realsense435 -->
  <include file="$(find realsense2_camera)/launch/rs_camera.launch" />

  <!-- 2. start ArUco -->
  <node n
    <re
      ▲ 赞同 3 ▼
      ● 添加评论
      ➤ 分享
      ★ 收藏
      ...
```



```

<remap from="/image" to="/camera/color/image_raw" />
<param name="image_is_rectified" value="true"/>
<param name="marker_size" value="$(arg marker_size)"/>
<param name="marker_id" value="$(arg marker_id)"/>
<param name="reference_frame" value="camera_color_frame"/>
<param name="camera_frame" value="camera_color_frame"/>
<param name="marker_frame" value="camera_marker" />
</node>

<!-- 3. start the robot -->
<include file="$(find ur_modern_driver)/launch/ur5_bringup.launch">
  <arg name="limited" value="true" />
  <arg name="robot_ip" value="192.168.1.102" />
</include>
<include file="$(find ur5_moveit_config)/launch/ur5_moveit_planning_executi
  <arg name="limited" value="true" />
</include>

<!-- 4. start easy_handeye -->
<include file="$(find easy_handeye)/launch/calibrate.launch" >
  <arg name="namespace_prefix" value="$(arg namespace_prefix)" />
  <arg name="eye_on_hand" value="false" />

  <arg name="tracking_base_frame" value="camera_color_frame" />
  <arg name="tracking_marker_frame" value="camera_marker" />
  <arg name="robot_base_frame" value="base" />
  <arg name="robot_effector_frame" value="tool0_controller" />

  <arg name="freehand_robot_movement" value="false" />
  <arg name="robot_velocity_scaling" value="0.5" />
  <arg name="robot_acceleration_scaling" value="0.2" />
</include>

</launch>

```

(1) realsense 相机节点

```

<!-- 1. start the Realsense435 -->
<include file="$(find realsense2_camera)/launch/rs_camera.launch" />

```

(2) aruco_ros 节点

从下面的网站下载 aruco 二维码并打印出来

Online ArUco markers generator

[chev.me](#)



注意：

- Dictionary 一定要选 Original ArUco
- Marker ID 和 Marker size 自选，在launch 文件中做相应的修改
- 打印时，要选择原始大小，否则要测量一下打印出来的真实大小

```

<!-- 2. start ArUco -->
<node name="aruco_tracker" pkg="aruco_ros" type="single">
  <!-- /camera_info 和 /image 做修改 -->
  <remap from="/camera_info" to="/camera/color/camera_info" />
  <remap from="/image" to="/camera/color/image_raw" />
  <param name="image_is_rectified" value="true"/>
  <pa

```

▲ 赞同 3 ▼ ● 添加评论 ➤ 分享 ★ 收藏 ...

```

<param name="marker_id" value="$(arg marker_id)"/>
<!-- reference_frame 和 camera_frame 做修改 -->
<param name="reference_frame" value="camera_color_frame"/>
<param name="camera_frame" value="camera_color_frame"/>
<param name="marker_frame" value="camera_marker" />
</node>

```

(3) ur5 节点

注意：

- 这里用了 ur_modern_driver 包，而没有用原始的 ur_bringup 包
- 修改机器人的真实 ip

```

<!-- 3. start the robot -->
<include file="$(find ur_modern_driver)/launch/ur5_bringup.launch">
  <arg name="limited" value="true" />
  <arg name="robot_ip" value="192.168.1.102" />
</include>
<include file="$(find ur5_moveit_config)/launch/ur5_moveit_planning_executi
  <arg name="limited" value="true" />
</include>

```

(4) easy_handeye 节点

```

<!-- 4. start easy_handeye -->
<include file="$(find easy_handeye)/launch/calibrate.launch" >
  <arg name="namespace_prefix" value="$(arg namespace_prefix)" />
  <arg name="eye_on_hand" value="false" />

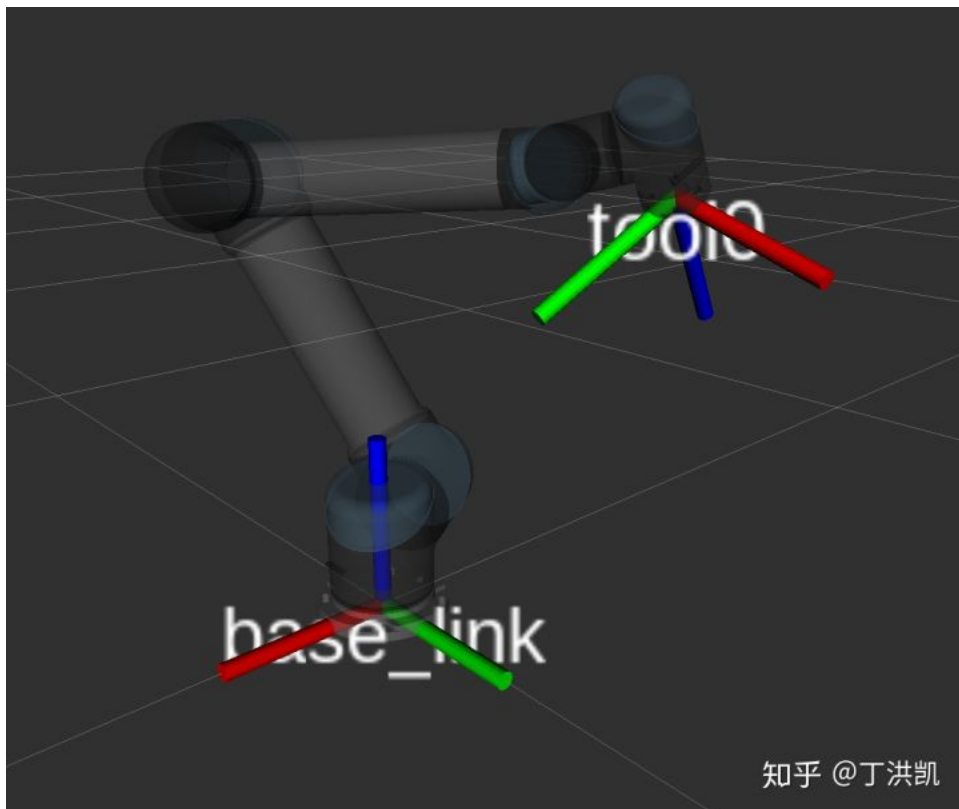
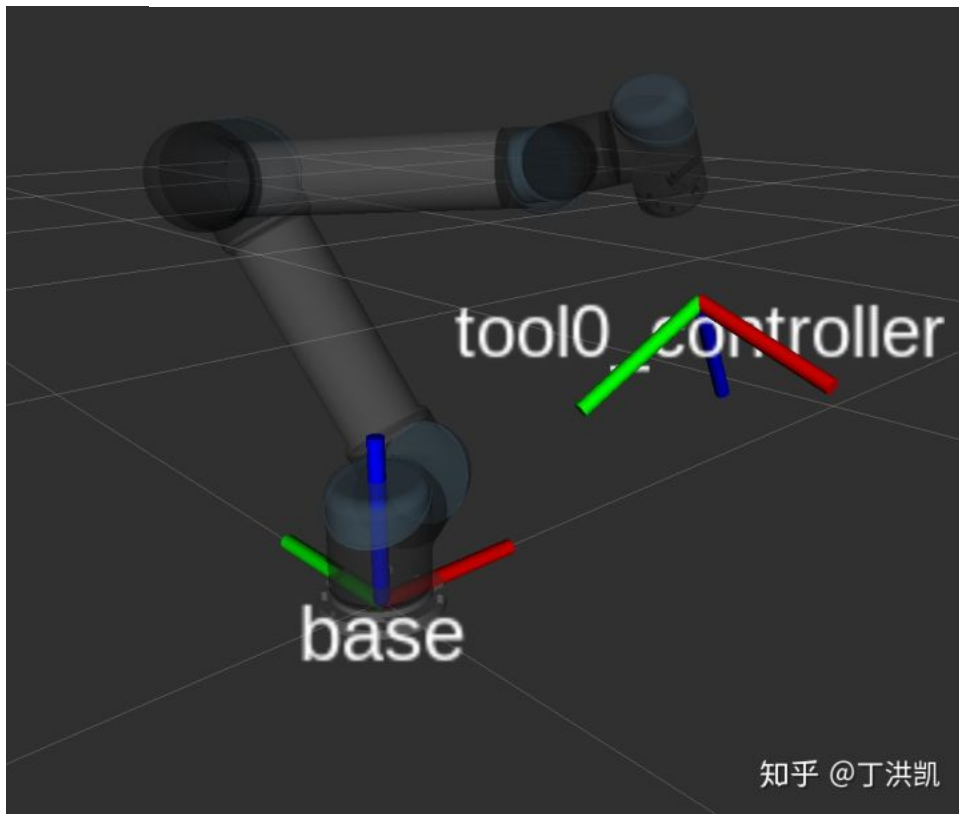
  <arg name="tracking_base_frame" value="camera_color_frame" />
  <arg name="tracking_marker_frame" value="camera_marker" />
  <arg name="robot_base_frame" value="base" />
  <arg name="robot_effector_frame" value="tool0_controller" />

  <arg name="freehand_robot_movement" value="false" />
  <arg name="robot_velocity_scaling" value="0.5" />
  <arg name="robot_acceleration_scaling" value="0.2" />
</include>

```

注意：

- <arg name="eye_on_hand" value="false"/> 眼在手外：value 为 false
- tracking_base_frame 为相机坐标系 camera_color_frame
- robot_base_frame 为机器人基座坐标系，示例里写的是 base_link，我在 rviz 中查看 base 才是真实的基座坐标系
- robot_effector_frame 为工具坐标系，因为我安装了 RG2 夹具，TCP 改变了

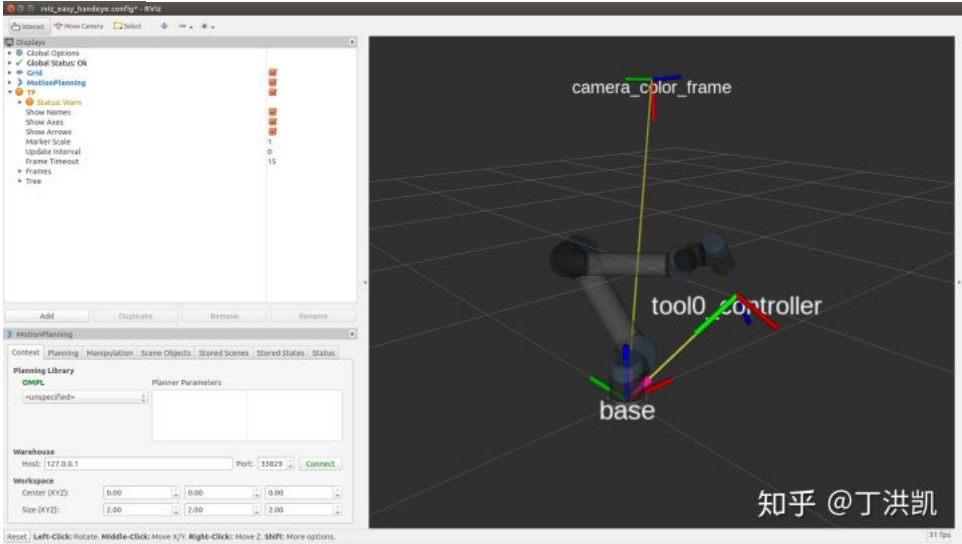


· 2.2 启动 launch 文件，开始标定

(1) 启动 launch 文件

```
roslaunch easy_handeye eye_to_hand_calibration.launch
```

会同时打开三个界面



界面 1

界面 2



界面 3

在界面 2 中，点击菜单栏的 Plugins -> Visualization -> Image View，选择 /aruco_tracker/result 话题，界面会如下所示。

(2) 标定步骤

- 手动调节机械臂，使 aruco 二维码移动至相机视野中心处附近，作为 home config。在界面 3 中，点击 check starting pose，若检查成功，界面会出现：0/17，ready to start
- 界面 3 中依次点击 Next Pose，Plan，Execute，机械臂会移动至新的位置，若二维码在相机视野范围内，且能检测成功，则进行下一步
- 界面 2 中点击 Take Sample，若 Samples 对话框中出现有效信息，说明第一个点标定成功
- 重复执行步骤 2 和步骤 3，直至 17 个点全部标定完毕
- 界面 2 中点击 Compute，则 Result 对话框中会出现结果
- 界面 2 中 Save，会将结果保存为一个 YAML 文件，路径为 ~/.ros/easy_handeye



3. 眼在手上 eye-in-hand

- 方法与“眼在手外”基本相同，需要修改一下 launch 文件，注意两个 launch 文件不要重名
- 唯一修改的地方是 easy_handeye 功能包 calibrate.launch 文件中 "eye_on_hand" 参数改成 true，如下所示

```
<launch>
  <arg name="namespace_prefix" default="ur5_realsense_handeyecalibration" />

  <arg name="robot_ip" doc="The IP address of the UR5 robot" />

  <arg name="marker_size" doc="Size of the ArUco marker used, in meters" defa
  <arg name="marker_id" doc="The ID of the ArUco marker used" default="571"/>

  <!-- start the Realsense435 -->
  <include file="$(find realsense2_camera)/launch/rs_camera.launch" />

  <!-- start ArUco -->
  <node name="aruco_tracker" pkg="aruco_ros" type="single">
    <remap from="/camera_info" to="/camera/color/camera_info" />
    <remap from="/image" to="/camera/color/image_raw" />
    <param name="image_is_rectified" value="true"/>
    <param name="marker_size" value="$(arg marker_size)"/>
    <param name="marker_id" value="$(arg marker_id)"/>
    <param name="reference_frame" value="camera_color_frame"/>
    <param name="camera_frame" value="camera_color_frame"/>
    <param name="marker_frame" value="camera_marker" />
  </node>

  <!-- start the robot -->
  <include file="$(find ur_modern_driver)/launch/ur5_bringup.launch">
    <arg name="limited" value="true" />
    <arg name="robot_ip" value="192.168.1.102" />
  </include>
  <include file="$(find ur5_moveit_config)/launch/ur5_moveit_planning_executi
    <arg name="limited" value="true" />
  </include>

  <!-- start easy_handeye -->
  <include file="$(find easy_handeye)/launch/calibrate.launch" >
    <arg name="namespace_prefix" value="$(arg namespace_prefix)" />
    <!-- 此处改成 true -->
    <arg name="eye_on_hand" value="true" />

    <arg name="tracking_base_frame" value="camera_color_frame" />
    <arg name="tracking_marker_frame" value="camera_marker" />
    <arg name="robot_base_frame" value="base" />
    <arg name="robot_effector_frame" value="tool0_controller" />

  </ar
  </ar
```



```
<arg name="robot_acceleration_scaling" value="0.2" />
</include>
```

```
</launch>
```

4. 发布 TF

easy_handeye 功能包提供了 publish.launch 文件，可以将标定好的 TF 发布

· 眼在手外 eye-to-hand

注：要修改 "namespace_prefix" 参数，与眼在手外标定 launch 文件中的 "namespace_prefix" 一致，这样才能找到标定好的 YAML 文件

```
<?xml version="1.0"?>
<launch>
  <arg name="eye_on_hand" doc="eye-on-hand instead of eye-on-base" />
  <arg name="namespace_prefix" default="ur5_realsense_handeyecalibration" />
  <arg if="$(arg eye_on_hand)" name="namespace" value="$(arg namespace_prefix)" />
  <arg unless="$(arg eye_on_hand)" name="namespace" value="$(arg namespace_prefix)" />

  <!--it is possible to override the link names saved in the yaml file in cas
  <arg if="$(arg eye_on_hand)" name="robot_effector_frame" default="" />
  <arg unless="$(arg eye_on_hand)" name="robot_base_frame" default="" />
  <arg name="tracking_base_frame" default="" />

  <arg name="inverse" default="false" />

  <!--publish hand-eye calibration-->
  <group ns="$(arg namespace)">
    <param name="eye_on_hand" value="$(arg eye_on_hand)" />
    <param unless="$(arg eye_on_hand)" name="robot_base_frame" value="$(arg namespace_prefix)" />
    <param if="$(arg eye_on_hand)" name="robot_effector_frame" value="$(arg namespace_prefix)" />
    <param name="tracking_base_frame" value="$(arg tracking_base_frame)" />
    <param name="inverse" value="$(arg inverse)" />
    <node name="$(anon handeye_publisher)" pkg="easy_handeye" type="publish" />
  </group>
</launch>
```

· 眼在手上 eye-in-hand

注：除了要修改 "namespace_prefix" 外（同上），还要将 "eye_on_hand" 参数设为 true

```
<?xml version="1.0"?>
<launch>
  <arg name="eye_on_hand" doc="eye-on-hand instead of eye-on-base" value="true" />
  <arg name="namespace_prefix" default="ur5_realsense_handeyecalibration" />
  <arg if="$(arg eye_on_hand)" name="namespace" value="$(arg namespace_prefix)" />
  <arg unless="$(arg eye_on_hand)" name="namespace" value="$(arg namespace_prefix)" />

  <!--it is possible to override the link names saved in the yaml file in cas
  <arg if="$(arg eye_on_hand)" name="robot_effector_frame" default="" />
  <arg unless="$(arg eye_on_hand)" name="robot_base_frame" default="" />
  <arg name="tracking_base_frame" default="" />

  <arg name="inverse" default="false" />

  <!--publish hand-eye calibration-->
  <group ns="$(arg namespace)">
    <param name="eye_on_hand" value="true" />
    <param unless="$(arg eye_on_hand)" name="robot_base_frame" value="$(arg namespace_prefix)" />
    <param if="$(arg eye_on_hand)" name="robot_effector_frame" value="$(arg namespace_prefix)" />
    <param name="tracking_base_frame" value="$(arg tracking_base_frame)" />
    <param name="inverse" value="$(arg inverse)" />
    <node name="$(anon handeye_publisher)" pkg="easy_handeye" type="publish" />
  </group>
</launch>
```

```
<param if="$(arg eye_on_hand)" name="robot_effector_frame" value="$(arg
<param name="tracking_base_frame" value="$(arg tracking_base_frame)" />
<param name="inverse" value="$(arg inverse)" />
<node name="$(anon handeye_publisher)" pkg="easy_handeye" type="publish
</group>
</launch>
```



· 测试

```
# 开启 publish.launch 文件，以眼在手上为例
roslaunch easy_handeye publish.launch

# 查看 TF (改成自己的工具坐标系与相机坐标系)
roslaunch tf tf_echo /tool0_controller /camera_color_frame
```

RG2 夹具实际安装构型如上图所示，红绿蓝（RGB）对应坐标轴 XYZ（TCP 工具坐标系），坐标轴符合右手定则。X 轴（红色箭头）垂直于两个 tip 的连线，Y 轴（绿色箭头）指向图中右侧的 tip，Z 轴指向图中下方。



相机实际的安装位置如上图所示。



综合夹具的安装位置以及标定参数的 Translation 值 和 Rotation 欧拉角值，判断标定基本正确，具体的标定精度，后面再测试。

参考：

没趣啊：UR5 kinect2 eye on base calibration
zhuanlan.zhihu.com



古月：古月私房课 | “手眼” 结合完成物体抓取应用
zhuanlan.zhihu.com



https://blog.csdn.net/txy12029047/article/details/99323620
blog.csdn.net



https://blog.csdn.net/ecH0o0/article/details/95245858
blog.csdn.net



编辑于 2019-11-27

「真诚赞赏，手留余香」

赞赏

还没有人赞赏，快来当第一个赞赏的人吧！

RealSense 机器人操作平台 (ROS) 手眼标定

文章被以下专栏收录



运动规划

关注专栏

▲ 赞同 3 ▼ ● 添加评论 ➦ 分享 ★ 收藏 ...



ROS机器人操作系统相关书籍、资料和学习路径

Top L... 发表于开始学习机...



提神醒脑方法专题-男女老少出门居家必备

黄燕婷

重要的话

感谢这些话，铭记在心。

芬尼根狂欢... 发表于艺术观察与...



”发现“你的发

Hezhe

还没有评论

写下你的评论...

😊