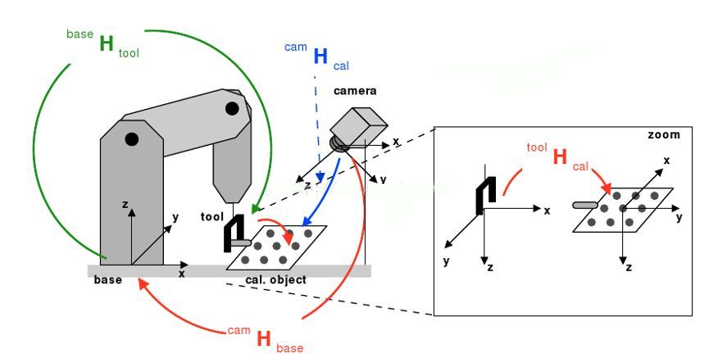
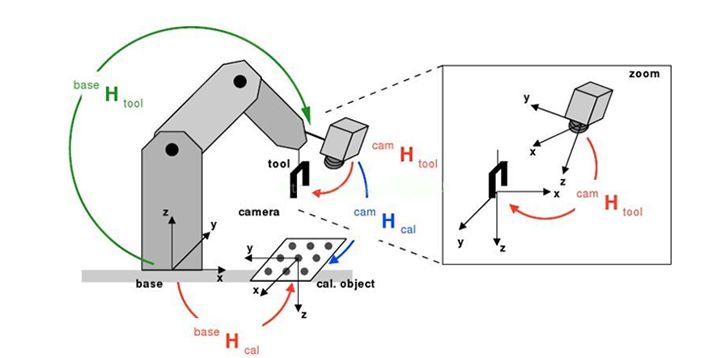
**基本原理：**

机器人手眼标定是机器人视觉的一项基本任务，它的任务目标就是估计机器人末端执行器与机器人相机坐标系之间的关系。是后续抓取的基础。根据相机安放的位置，又可以将手眼标定模型分为“眼在手上”（相机固定在机器人末端执行器上）与“眼在手外”（相机固定在机器人机身上）。 其原理模型如图所示：



机器人手眼标定的数学模型是求解 AX=XB这种形式的齐次矩阵方程，其中A表示机器人变换一次位姿后相机的相对运动（可由标定板读出）， B表示末端执行器的相对运动（可由tf读出），X也就是待求的相机到末端执行器的变换矩阵。

**标定程序简介:**

手眼标定模块已经封装成Ros Package， 名称为： hand\_eye\_calibration。在使用前请确保已经所有依赖都已经成功安装：

**catkin, actionlib, actionlib\_msgs, cv\_bridge, image\_transport, moveit\_core, moveit\_ros\_planning, moveit\_ros\_planning\_interface, roscpp, sensor\_msgs, std\_megs, tf**

手眼标定需要在抓取操作之前进行， 在演示的过程中如果相机位置出现变动，都需要重新标定。标定相机需要特定的标定板，并将标定板安装在机械臂末端，并调整机械臂到合适的构型使得标定板完全出现在摄像机视野内。手眼标定过程分为两步进行：

1.多次调整末端机械臂的姿态，并记录对应同一构型的 相机图片 和 机械臂末端到机器人基座的转换矩阵，获取足够多满足要求的 “图片-转换矩阵” 组，存入本地文件夹。

2.计算相机坐标系到末端执行器的变换矩阵。

这两步分别由hand\_eye\_calibration 包中的getdata\_node 与 eyetohand\_calib\_node 完成。

**操作流程：**

1.确保机器人已经开启，相机usb线已经接入。

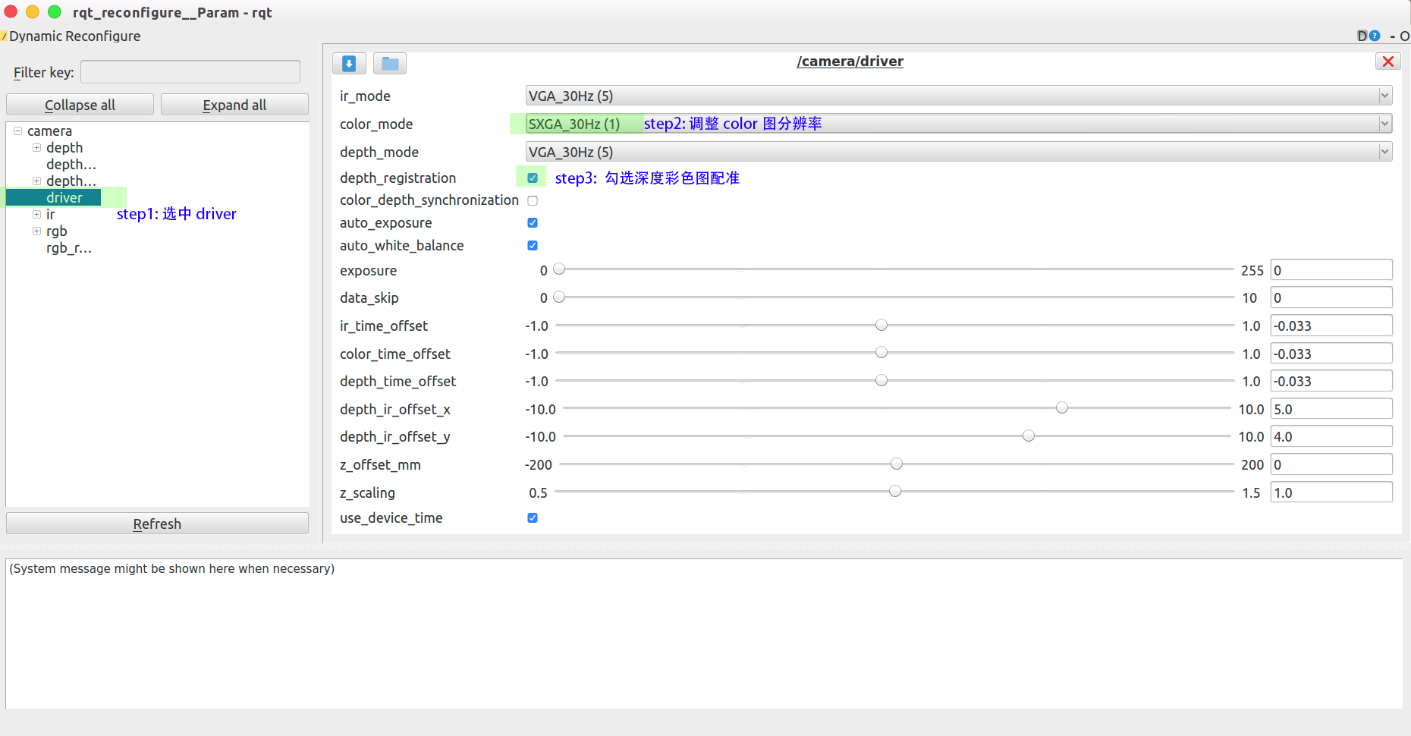
2.新建terminal，启动 openni2 相机驱动。

roslaunch openni2\_launch openni2.launch

3. 新建terminal 使用rqt\_reconfigure 调整相机图像分辨率，并使深度彩色图配准。

rosrun rqt\_reconfigure rqt\_reconfigure

配准过程参照下图。调整完后，可以关闭当前terminal，此步骤每次重新接入相机，启动 openni2 相机驱动 时都需要重新操作！。



4.新建ternimal，进入siasun\_ws工作空间 并 source工作空间

cd ~/siasun\_ws/

source devel/setup.bash

5.当前terminal中 运行机器人驱动：

roslaunch srd\_moveit\_config robot\_interface\_streaming.launch

6.新建terminal， 进入siasun\_ws工作空间，并source工作空间，如步骤4，完成后，在当前terminal中启动moveit可视化界面

roslaunch srd\_moveit\_config moveit\_planning\_execution.launch

此时moveit软件会打开，暂时不用管它。

7. 新建terminal，进入siasun\_ws工作空间，并source工作空间，如步骤4，完成后进入hand\_eye\_calibration包内：

roscd hand\_eye\_calibration

进入siasun文件夹:

cd siasun

这个文件夹将存放标定数据，确保camera\_data.xml文件在这个目录下且内容如下：

如要更换其他目录，务必将camera\_data.xml拷贝至目标目录下，此文件描述了相机内参数。

<?xml version="1.0"?>

<opencv\_storage>

<camera\_matrix type\_id="opencv-matrix">

<rows>3</rows>

<cols>3</cols>

<dt>f</dt>

<data>

1.0825505260489588e+03 0. 6.3950000000000000e+02 0.

1.0825505260489588e+03 5.1150000000000000e+02 0. 0. 1.

</data></camera\_matrix>

<distortion\_coeffs type\_id="opencv-matrix">

<rows>1</rows>

<cols>4</cols>

<dt>f</dt>

<data>

2.6674390223625083e-02 -4.3739405836745486e-03 0. 0.</data></distortion\_coeffs>

</opencv\_storage>

8. 在同一 ternimal 在siasun文件夹下运行 getdata\_node， 给定合适的参数，开始数据采集。此时会

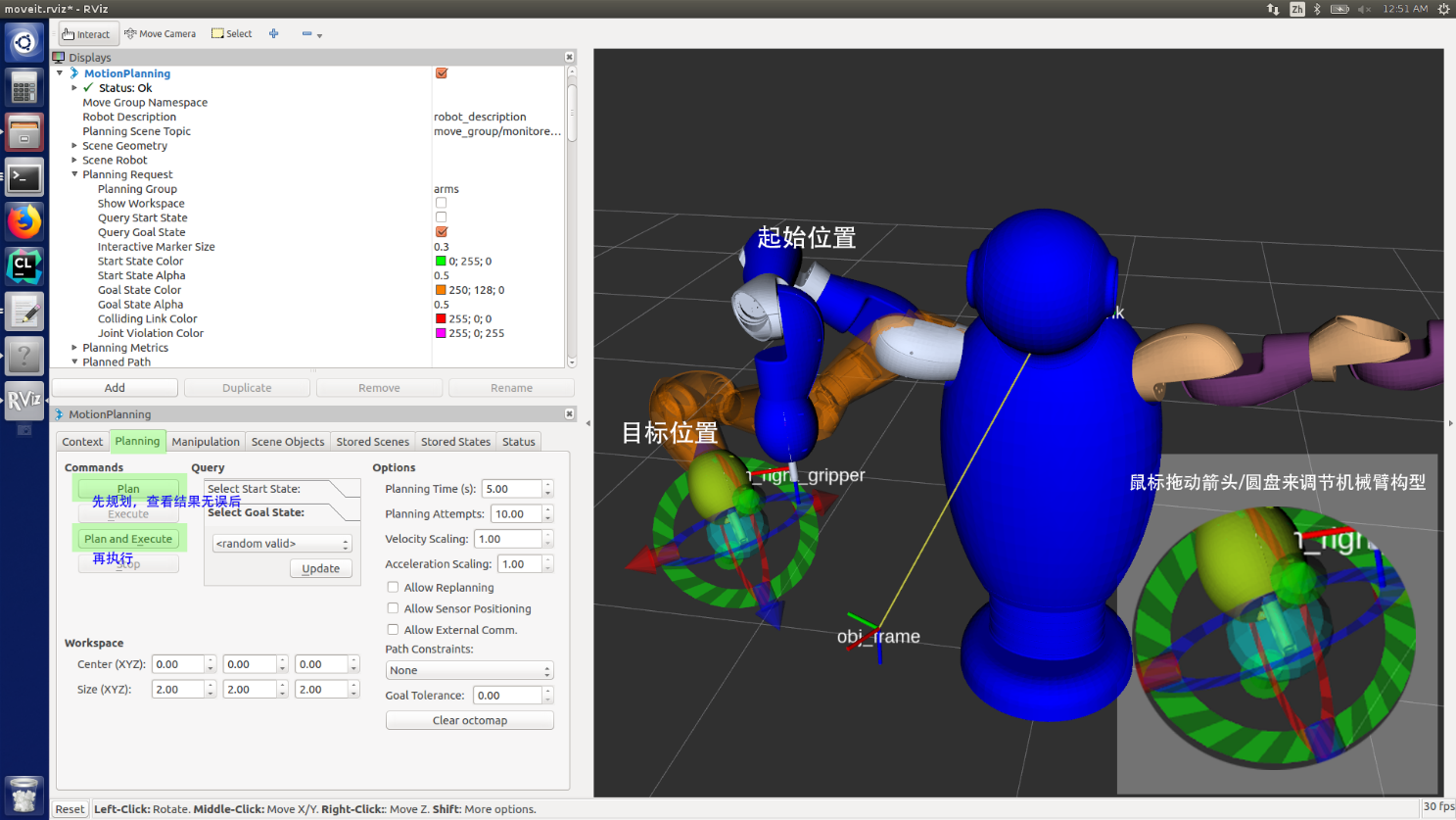
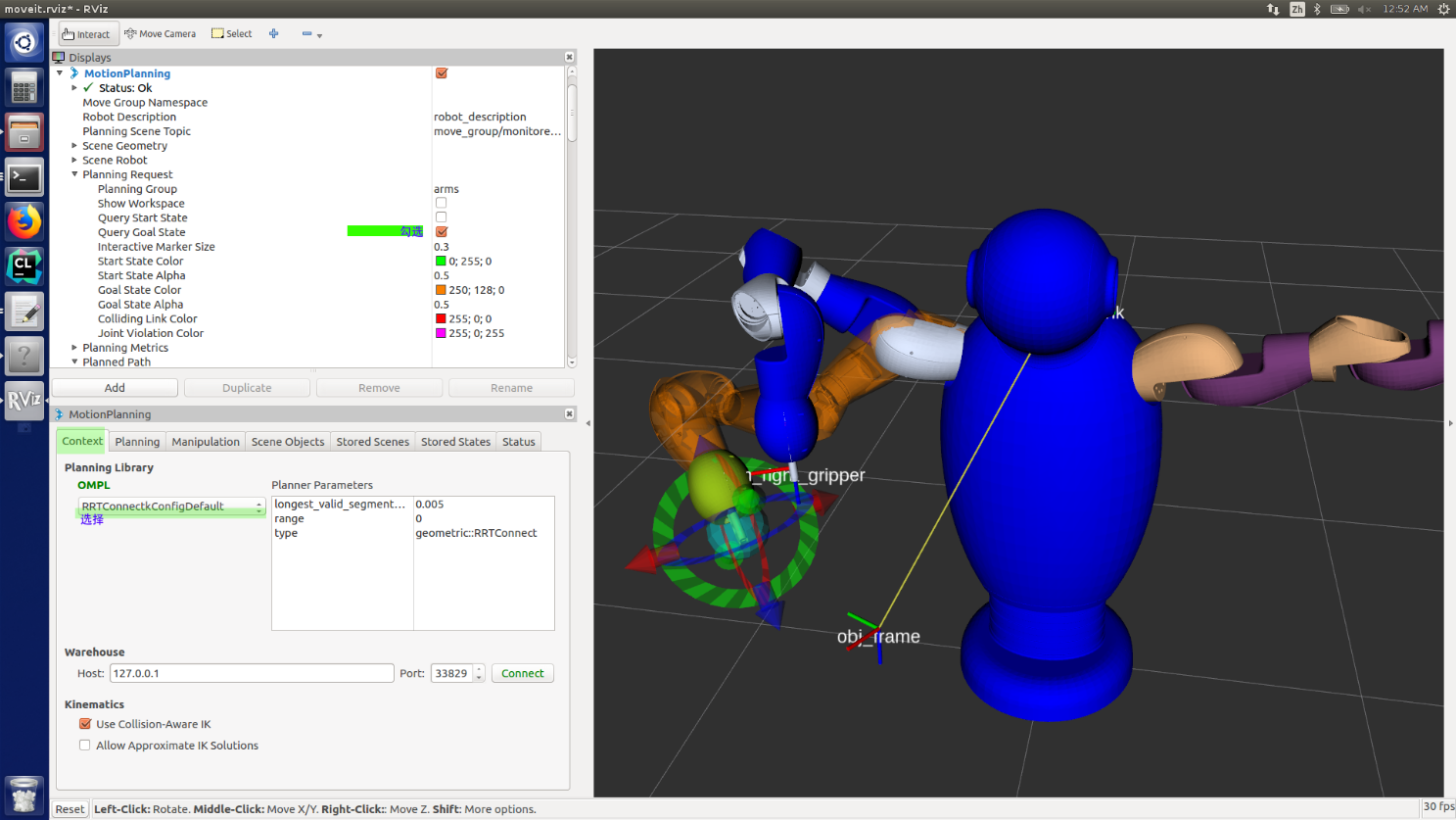
参数格式为：

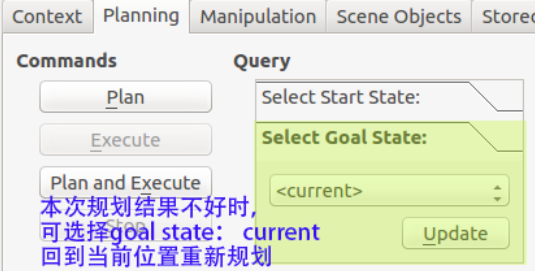
**rosrun 包名 节点名 相机图像topic名 末端机械臂link(tf) 基座link(tf)**

rosrun hand\_eye\_calibration getdata\_node /camera/rgb/image\_raw arm\_right\_link\_7\_t base\_link

有窗口弹出，显示相机图像流。

9. 使用之前步骤6中打开的moveit可视化界面，使用moveit调整末端机械臂位姿，使标定板集中在相机图像中心附近且能清楚看见棋盘格。 在terminal中按下空格采集当前帧“图片-转换矩阵”组，图片以jpg文件存在当前文件夹内，转换矩阵以四元数与平移向量存在当前文件夹内的robotpose.xml中。 采集参考标准见后文。





10. 采集到足够数据组后， 在当前terminal按下 Esc， 结束getdata\_node, 仍在当前文件夹下 运行： eyetohand\_calib\_node， 给定合适的参数，开始计算外参。

参数格式为：

**rosrun 包名 节点名 文件夹内本次标定采集的图像数 标定板棋盘格节点长方向个数 标定板棋盘格节点宽方向个数 每个棋盘格大小mm**

rosrun hand\_eye\_calibration eyetohand\_calib\_node #(number——of\_images)# 5 4 25

11. 程序会将每张图片标记上棋盘格的位置后输出， 按下空格查看下一张，等全部图像显示完， 在terminal中查看标定结果。

12. 标定结果给

~/siasun\_ws/src/siasun\_obj/launch/rgbd\_image.launch

在标定数据采集过程中，应确保每次机械臂末端都至少同时有一定量的平移与旋转，并且使标定板集中在相机图像中心附近且能清楚看见棋盘格，采样12~15组左右不同的数据，合格采样如下图：

