**kinova\_gazebo使用说明**

目录

[1. 安装依赖项 1](#_Toc522131223)

[2. 安装kinova包和识别包 1](#_Toc522131224)

[3. 文件夹结构 2](#_Toc522131225)

[4. 训练模型 5](#_Toc522131226)

[5. 查看识别物体 6](#_Toc522131227)

[6. kinova\_gazebo 仿真 8](#_Toc522131228)

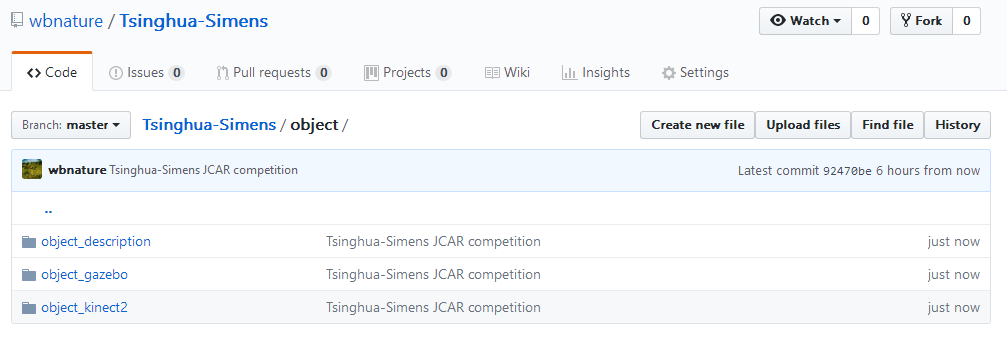
[7. 用相机构建三维模型 9](#_Toc522131229)

[8. 实物运行检测 10](#_Toc522131230)

[9. 存在问题 11](#_Toc522131231)

代码的改动：

在 object下新增了 object\_kinect2 包，用来实物启动 Kinect2，下载下来重新编译。,Git 地址：<https://github.com/wbnature/Tsinghua-Simens.git>



## 安装依赖项

* OpenCV：目前用的OpenCV版本是OpenCV-2.4.13.6；OpenCV-3.x的版本不能用；OpenCV-2.x的版本应该都可以用，没试过其他版本，如果不能用，就装OpenCV-2.4.13.6。安装过程可参考官网：

<https://docs.opencv.org/2.4/doc/tutorials/introduction/linux_install/linux_install.html>

* Moveit：sudo apt-get install ros-版本号-moveit即可。
* Kinect2驱动：doc文件夹下有注意事项,参照网上安装教程即可：

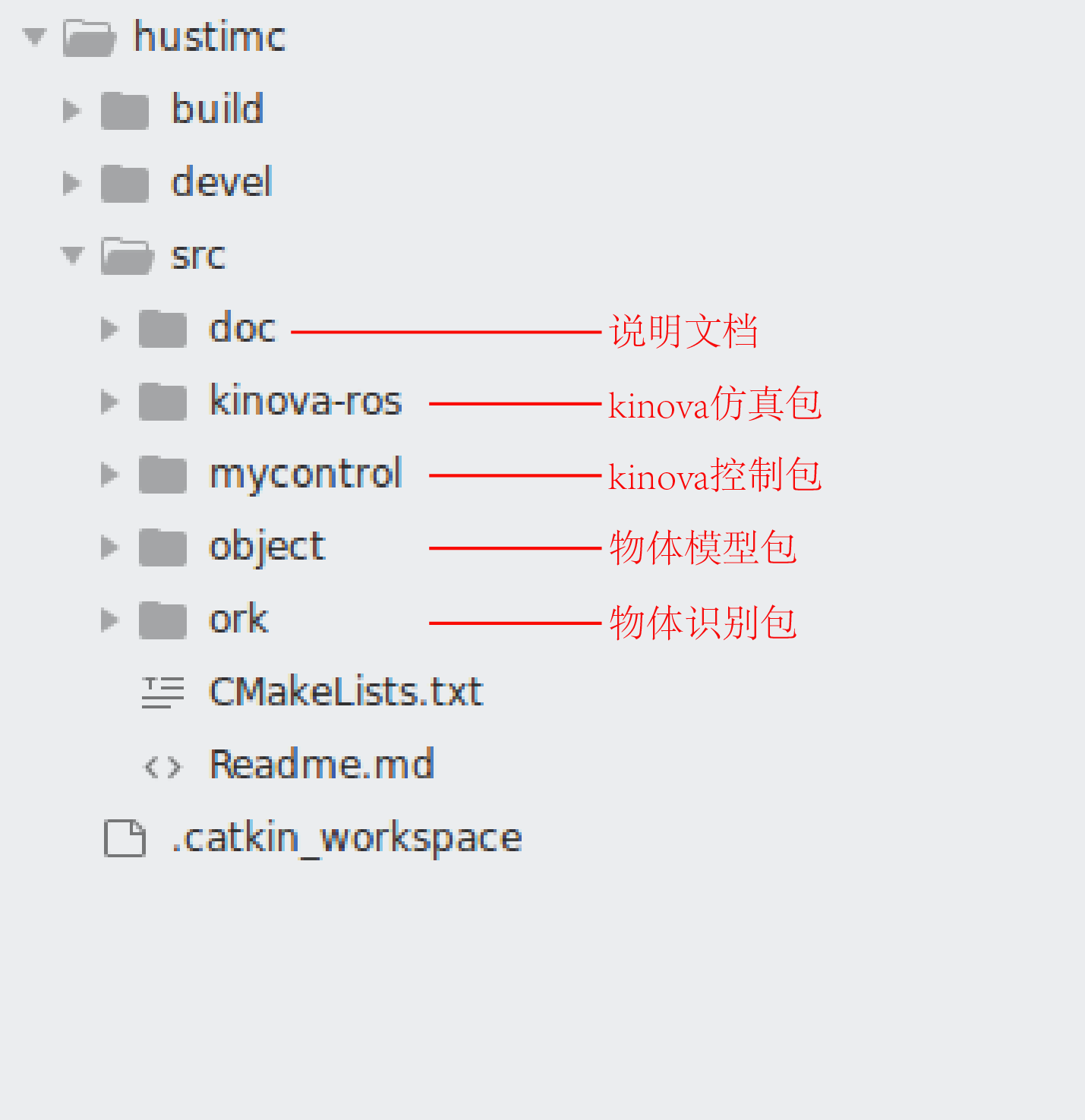
<http://blog.csdn.net/sunbibei/article/details/51594824>

## 安装kinova包和识别包

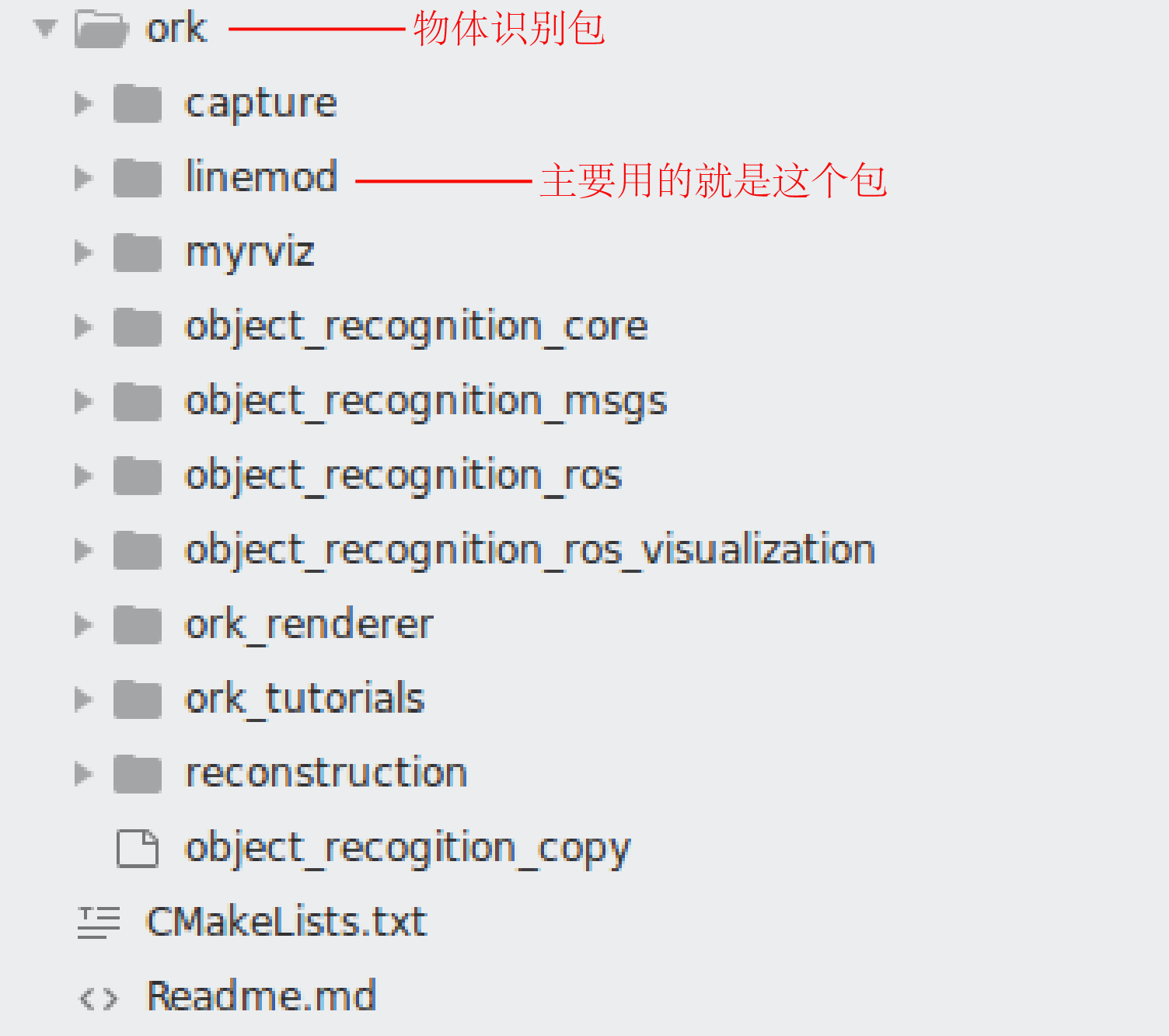
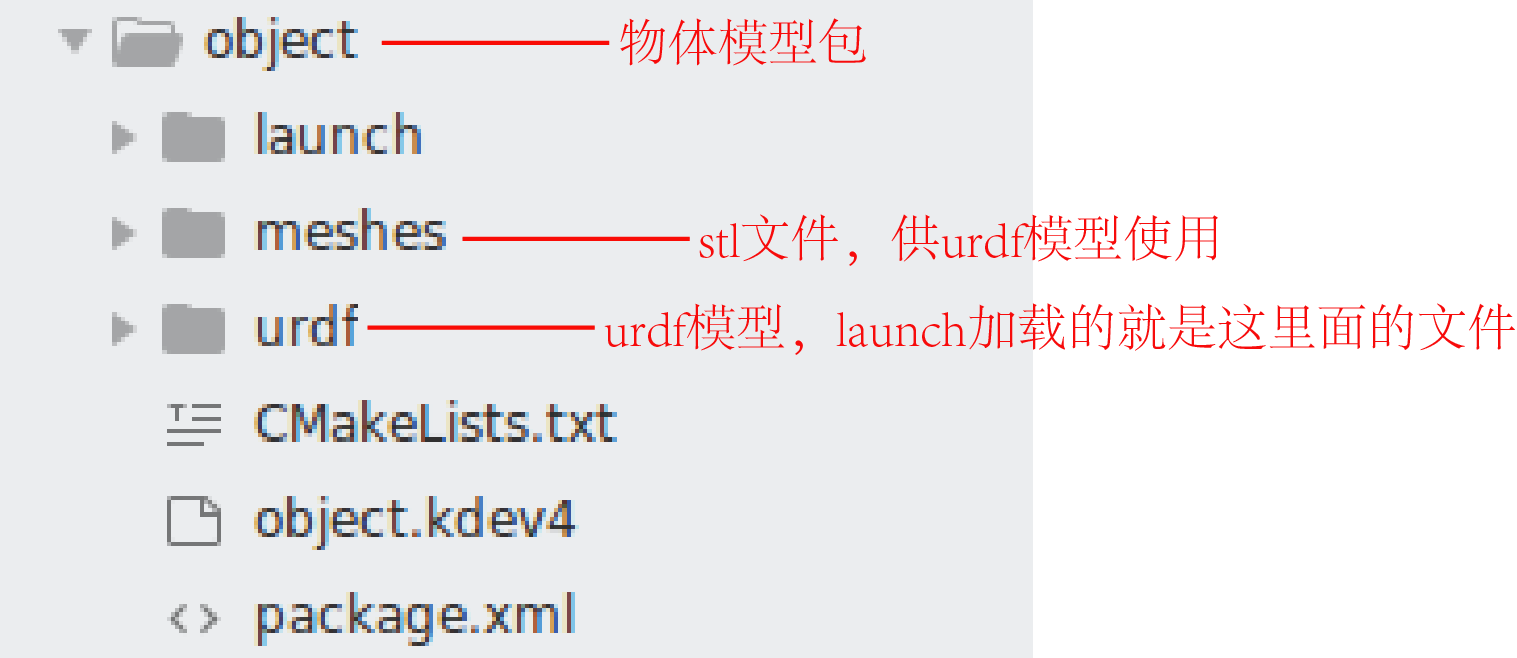
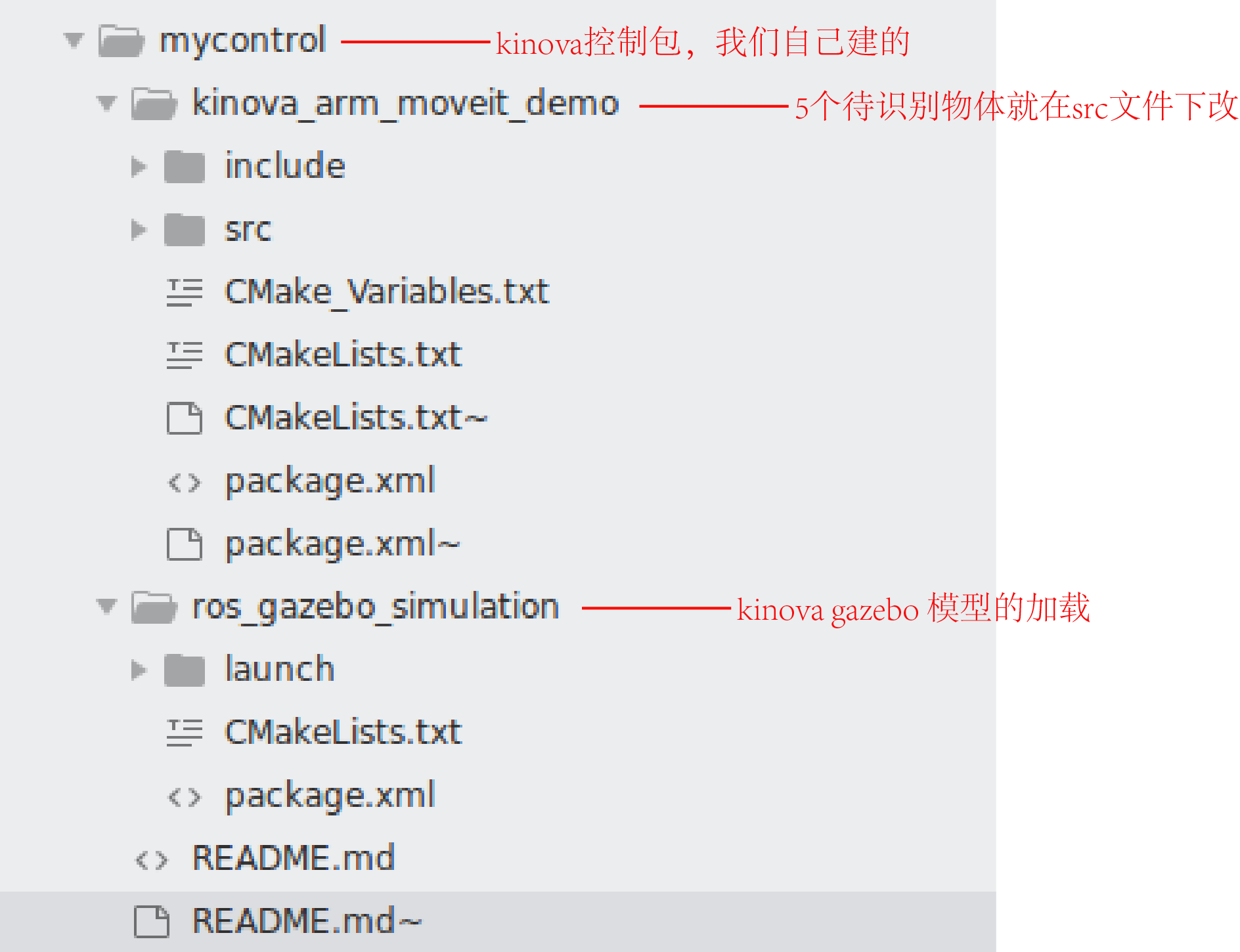
操作流程:

* 在home下新建文件夹，比如mkdir hustimc
* cd hustmic
* mkdir src
* cd src
* git clone https://github.com/wbnature/Tsinghua-Simens.git
* cd ..
* catkin\_make
* 不要忘了source，可以在.bashrc文件中加入，也可以在每个终端都 source一下。

## 文件夹结构







## 训练模型

我们的stl模型在 object/object\_description/meshes文件夹下,要训练的10个物体名字如下：

ball：球体；centrum：圆锥；cuboid\_3：三棱柱；

squre\_50：正方体；cylinder\_75: 75mm高圆柱，50mm高的未用；

prisms\_6：六棱柱；centrum\_4：三棱锥；cuboid\_4：四棱柱，长方体

centrum\_5：四棱锥；centrum\_6：四棱台。

操作流程：

* 在home下，cd hustimc/src/object/object\_description/meshes
* rosrun object\_recognition\_core object\_add.py -n "ball" -d "ball" --commit

ball 更换成相应的stl文件

* rosrun object\_recognition\_core mesh\_add.py 171966a56d345d9d31d4bc0df500bf0c ball.stl –commit

171966a56d345d9d31d4bc0df500bf0更换为上一步生成的id，ball 更换成相应的stl文件

* 将10个物体都添加进去，再执行下面的训练

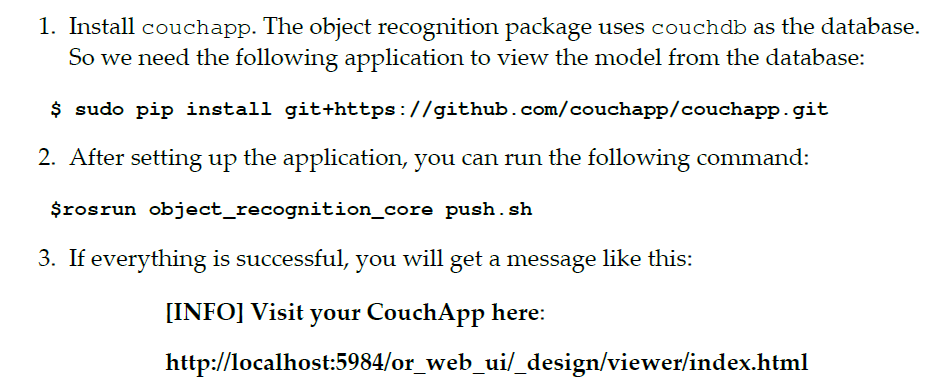
rosrun object\_recognition\_core training -c `rospack find object\_recognition\_linemod`/conf/training.ork

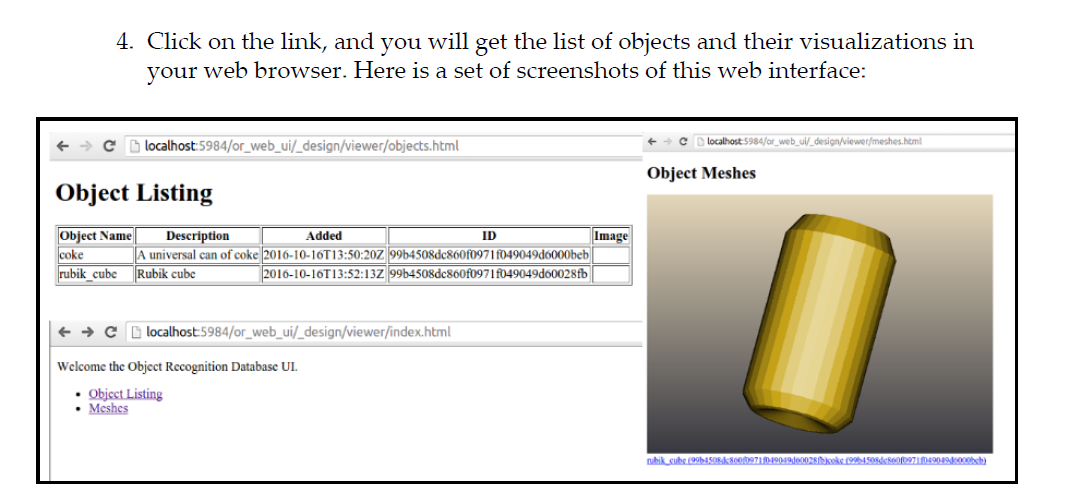
* 训练完成后，可查看，需要安装couchapp,下面有代码和说明：

sudo pip install git+https://github.com/couchapp/couchapp.git

rosrun object\_recognition\_core push.sh

http://localhost:5984/or\_web\_ui/\_design/viewer/index.html





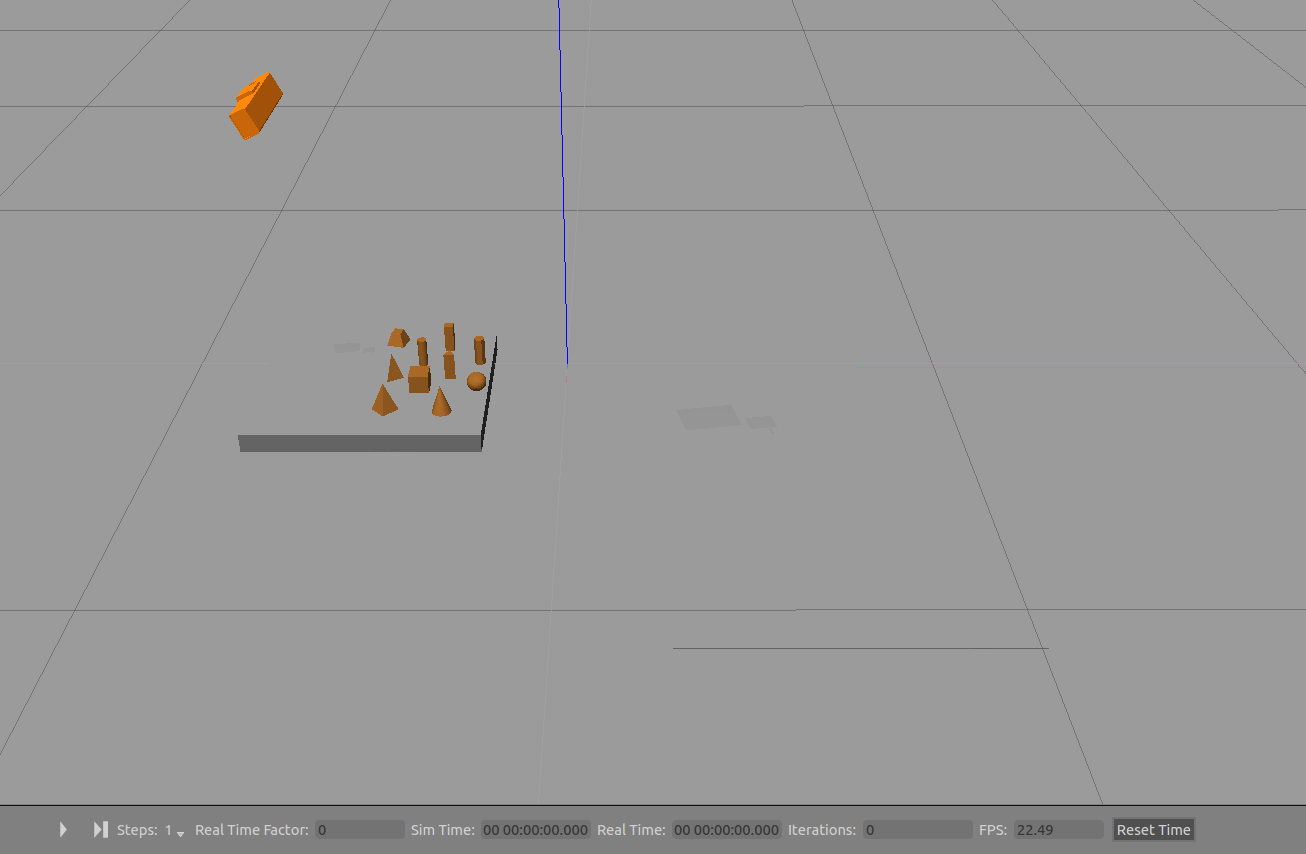
如果浏览器不能查看，说明浏览器不支持。

## 查看识别物体

文件在object文件夹的object\_gazebo和object\_description的launch中操作步骤如下：

* roslaunch object\_gazebo object\_gazebo\_module.launch

出现下面界面，若gazebo报错 process has died，多运行几次，

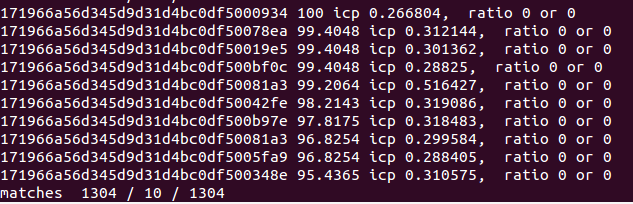


注意一定要等加载完成后点图中红色部分开始，一定要点，一定要点！！！

在object\_gazebo\_module.launch中可更改每个识别物体的位置，注意若要在kinvao仿真时用，需要在object\_gazebo.launch中改，因为kinvao仿真时加载的是这个文件。

* rosrun object\_recognition\_core detection -c `rospack find object\_recognition\_linemod`/conf/linemod\_detection.ros.ork

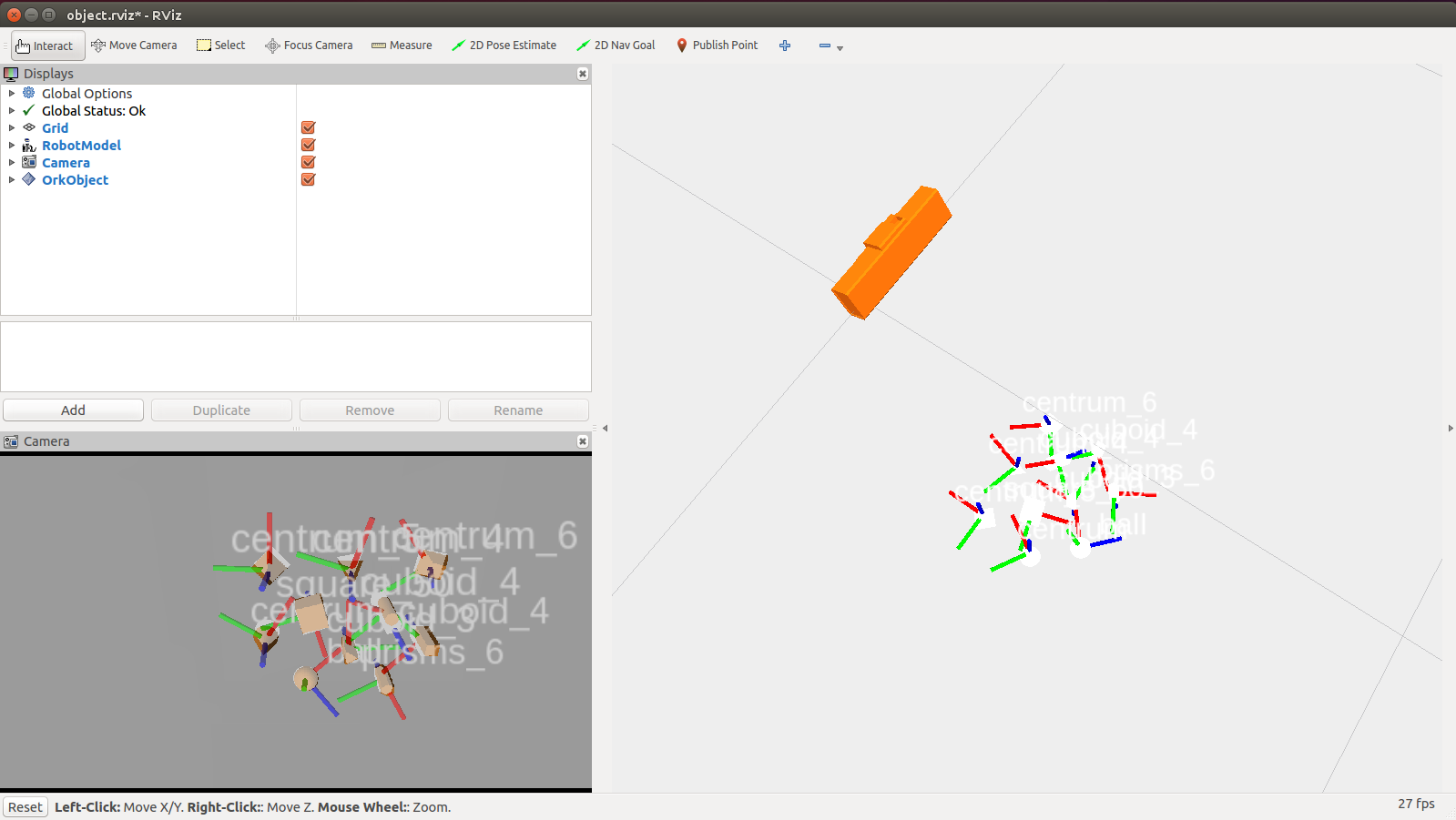
这一步是检测，时间有点长，当出现下面内容是，说明检测出来了。



上图中的10说明检测出来10个物体，其他两个不用管。

* roslaunch object\_description object\_rviz.launch

出现下面界面，可能要等一会才出来识别结果，



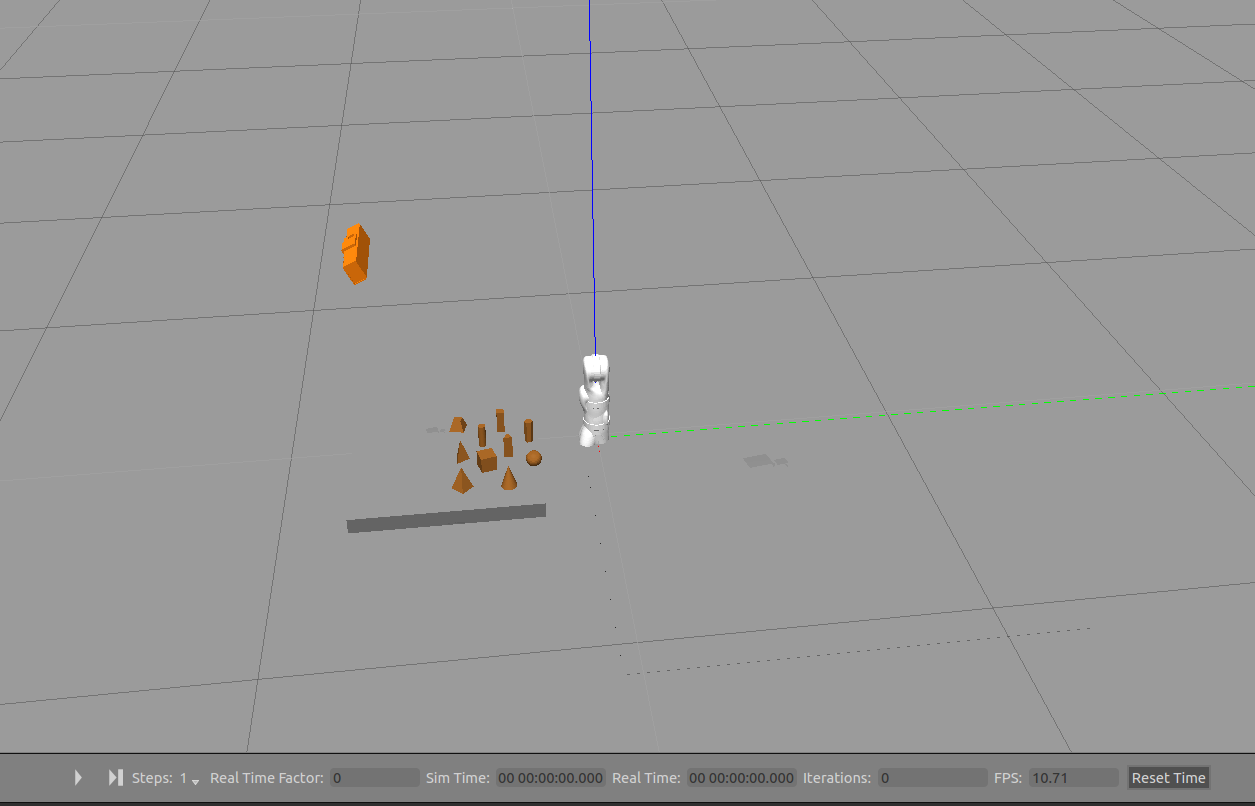
## kinova\_gazebo 仿真

演示的就是这一步，上面的都是准备工作

操作步骤：

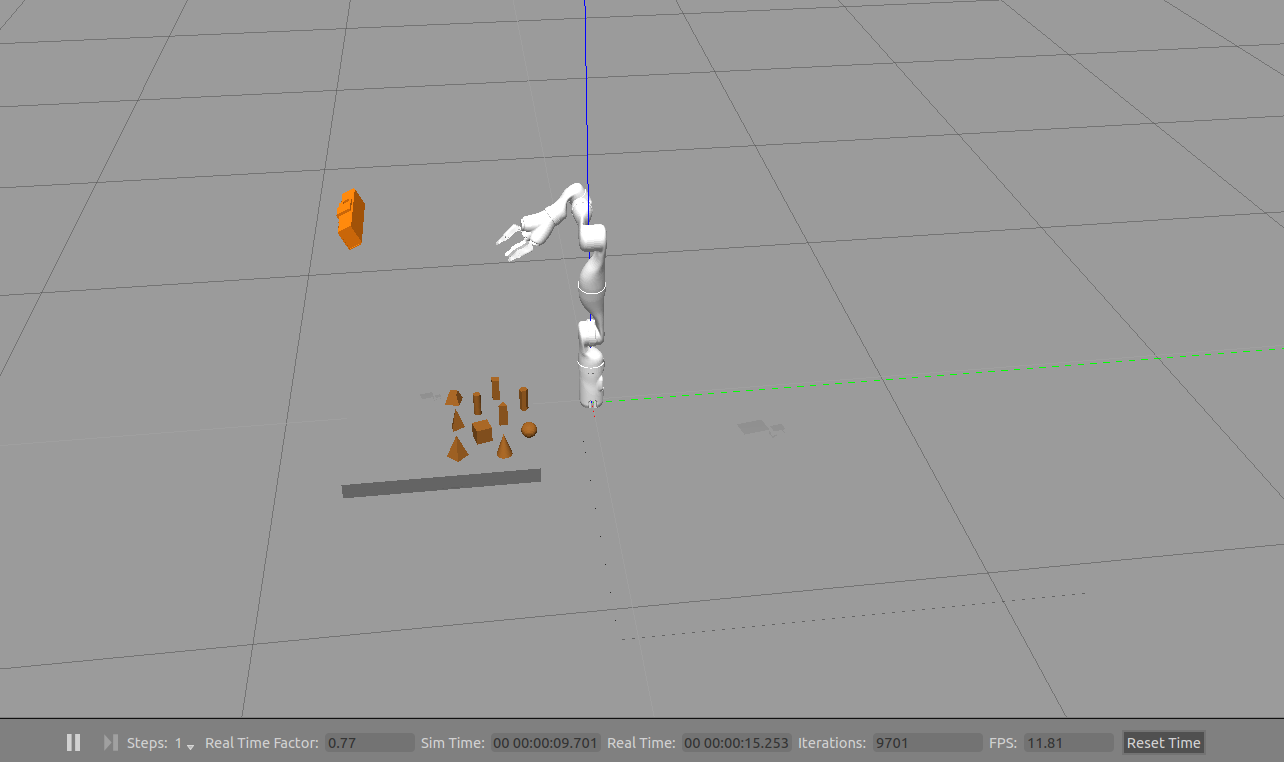
* roslaunch ros\_gazebo\_simulation motion\_control.launch

启动仿真，出现下面界面，



注意一定要等加载完成后点图中红色部分开始，一定要点，一定要点！！！

点过之后变成这样：



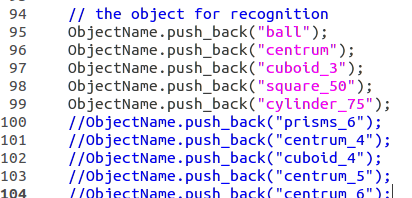
* 运行检测

rosrun object\_recognition\_core detection -c `rospack find object\_recognition\_linemod`/conf/linemod\_detection.ros.ork

* 开始抓取

rosrun kinova\_arm\_moveit\_demo objectplan

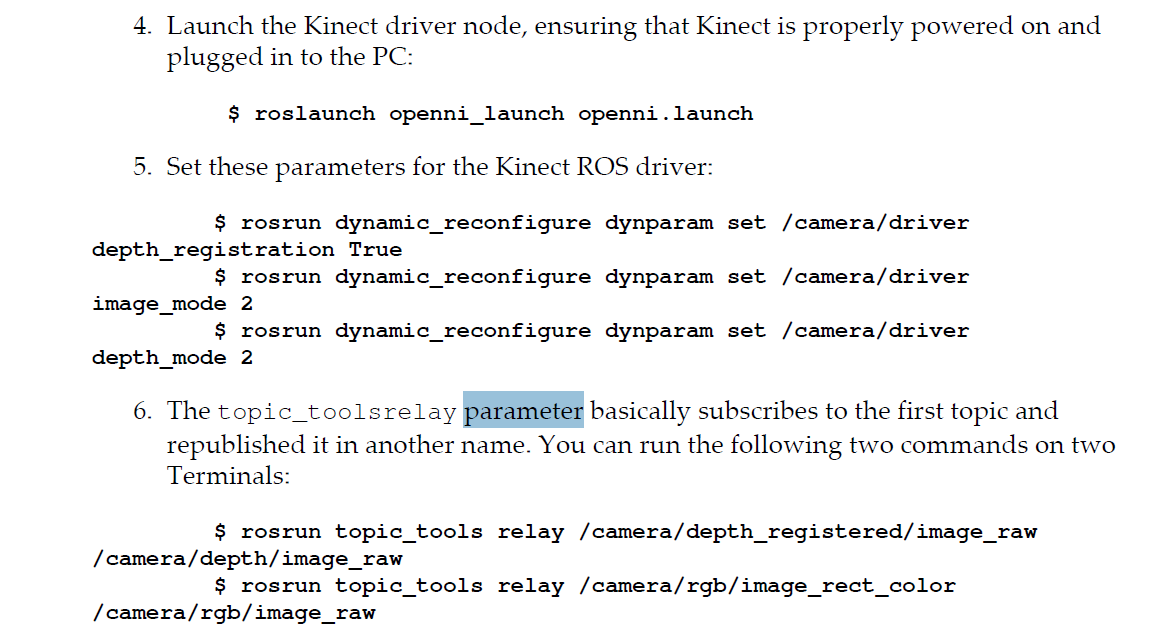
在kinova\_arm\_moveit\_demo/src/ objectplan.cpp中更改识别物体，注释相应代码即可，并需要编译：



## 用相机构建三维模型

参考为书籍ROS\_ROBOTICS\_PROJECTS.pdf 的191页Training from captured 3D models。

由于书上是 Kinect1 版本的，第4,5,6三个步骤使用kinect1需要更改，其他步骤的不变。



将4,5,6步骤换成运行

roslaunch object\_kinect2 object\_kinect2.launch

即可，在launch文件中已经将kinect2的话题（topic）映射到kinect1上，不需要再映射。

## 实物运行检测

操作步骤：

* 开启Kinect2

roslaunch object\_kinect2 object\_kinect2.launch

在launch文件中已经将kinect2的话题（topic）映射到kinect1上，因为现在使用的算法 linemod 是基于kinec1的，如果不用现在的算法是不需要映射的。

* 运行检测

rosrun object\_recognition\_core detection -c `rospack find object\_recognition\_linemod`/conf/linemod\_detection.ros.ork

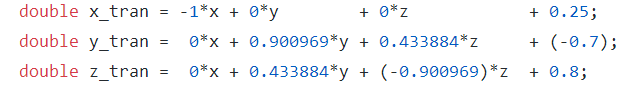
* 开启 kinova 的实物驱动

这一部分不会，不仅有 kinova 的实物驱动，还有用move\_it进行轨迹规划。

* 开始抓取

rosrun kinova\_arm\_moveit\_demo objectplan

需要注意的是识别物体的更改（上面已写）和相机在世界坐标系下的位置，在kinova\_arm\_moveit\_demo/src/ objectplan.cpp中更改识别物体和相机的位置，并需要编译：



上面的 x,y,z 前面的系数就是变换矩阵，单位是米（m）。

写成矩阵形式为：[x y z 1]世界坐标系下 = T \* [x y z 1] 相机坐标系下 ，T为：

-1 0 0 0.25

0 0.900969 0.433884 -0.7

1. 0.433884 0.900969 0.8

0 0 0 1

## 存在问题

* 篮子模型太难建，未建；
* 圆柱识别不出来，会被识别成长方体或六棱柱；