

# ROS 系统 Astra SDK 说明

## 目录

ROS 系统安装 .....	2
ROS-Astra-SDK 获取方式 .....	2
ROS-Astra-SDK 基本结构 .....	3
ROS-Astra-SDK 运行及数据获取方式 .....	3
1, Astra 运行 .....	3
2, 获取数据 .....	4
3, Image_view 显示深度数据 .....	6
4, RVIZ 显示深度数据 .....	7



ROS 是现在非常流行的机器人操作系统，作为世界领先的 RGBD 厂家，奥比中光也基于 OpenNI2 为开发者推出了 ROS 环境的 SDK，本文主要讲解如何在 ROS 下编译，运行并获取 Astra 系列相机的数据，以便开发者进行进一步开发。详细信息请访问 <http://wiki.ros.org/Sensors/OrbbecAstra>，本文假定用户了解并熟悉 Linux 及 ROS 的基本操作，所有操作以 Ros Indigo 系统为基本环境。

## ROS 系统安装

ROS 有详细的 WIKI 页面，如何安装并配置 ROS 环境，请访问 <http://wiki.ros.org/ROS/Installation> 页面，并按页面指导进行安装。目前 Astra 仅支持 Indigo 及以上的 ROS 系统版本。安装 ROS 系统请使用 wiki 说明中注明的相应 Ubuntu 版本，避免未知问题出现。例如 ROS Indigo -> Ubuntu 14.04, ROS Kenetic -> Ubuntu 16.04.

Eg. <http://wiki.ros.org/indigo/Installation/Ubuntu>  
<http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment>

## ROS-Astra-SDK 获取方式

- 1, 在 Ubuntu Indigo 及其以上版本可以直接通过 apt 的方式安装

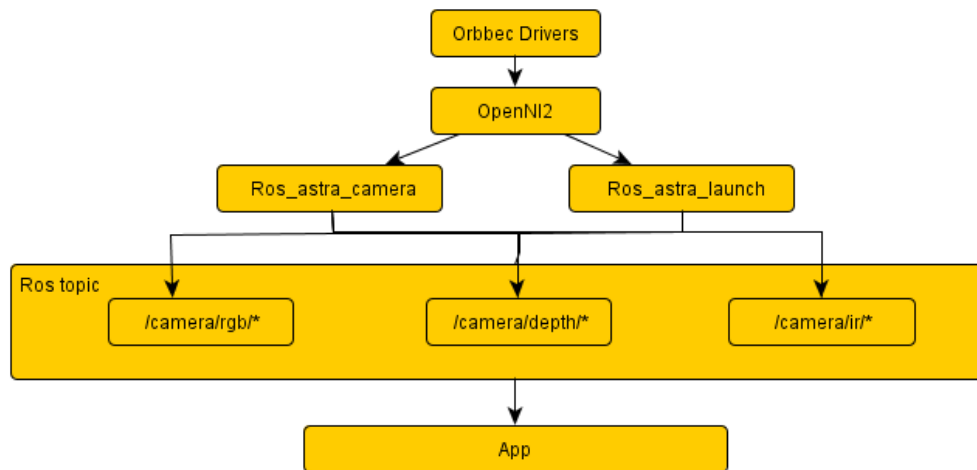
```
$ sudo apt-get install ros-indigo-astra-camera
$ sudo apt-get install ros-indigo-astra-launch
```
- 2, 通过编译源码方式获取
  - 1) GIT 地址为 [https://github.com/orbbec/ros\\_astra\\_camera](https://github.com/orbbec/ros_astra_camera) 和 [https://github.com/orbbec/ros\\_astra\\_launch](https://github.com/orbbec/ros_astra_launch)
  - 2) 安装 SDK 需要用到的 ros 系统包

```
$ sudo apt-get install ros-indigo-rgbd-launch
$ sudo apt-get install ros-indigo-camera-info-manager
```
  - 3) 编译 ros astra sdk

```
$ catkin_make --pkg astra_camera
```
  - 4) 安装 Astra 对应 UDEV 文件

```
$ roscd astra_camera && ./scripts/create_udev_rules
```

## ROS-Astra-SDK 基本结构



## ROS-Astra-SDK 运行及数据获取方式

### 1, Astra 运行

1) 通过 roslaunch 方式 即可启动 Astra 相机

```
$ roslaunch astra_launch astra.launch
```

```
etting /run_id to c6b0043c-6eaa-11e7-9370-000c298ed66e
rocess[rosout-1]: started with pid [6538]
rted core service [/rosout]
rocess[camera/camera_nodelet_manager-2]: started with pid [6549]
rocess[camera/driver-3]: started with pid [6556]
rocess[camera/rgb_rectify_color-4]: started with pid [6557]
rocess[camera/depth_rectify_depth-5]: started with pid [6558]
rocess[camera/depth_metric_rect-6]: started with pid [6577]
rocess[camera/depth_metric-7]: started with pid [6591]
rocess[camera/depth_points-8]: started with pid [6595]
rocess[camera/register_depth_rgb-9]: started with pid [6610]
rocess[camera/points_xyzrgb_sw_registered-10]: started with pid [6625]
rocess[camera/depth_registered_sw_metric_rect-11]: started with pid [6633]
rocess[camera_base_link-12]: started with pid [6646]
rocess[camera_base_link1-13]: started with pid [6654]
rocess[camera_base_link2-14]: started with pid [6670]
INFO [1500706542.181969785]: Initializing nodelet with 4 worker threads.
rocess[camera_base_link3-15]: started with pid [6688]
INFO [1500706542.342949269]: Device "2bc5/0401@3/7" found.
arning: USB events thread - failed to set priority. This might cause loss of da
a...
```

2) 通过直接运行节点显示 Astra 相关信息

```
$ rosrn astra_camera astra_list_devices
```

```
orbbe@localhost:~$ roslaunch astra_camera astra_list_devices
[ INFO] [1500706596.935980289]: Device "2bc5/0401@3/7" found.
Found 1 devices:

Device #0:
Uri: 2bc5/0401@3/7 (Vendor: Orbbec, Name: Astra, Vendor ID: 2bc5, Product ID: 401)

Warning: USB events thread - failed to set priority. This might cause loss of data...
Serial number: 16030310030
```

## 2. 获取数据

- 1) 运行上面的命令 1 后，会启用 Astra 驱动，并按 ROS 规则输出相应的话题数据。可以通过 rostopic 的方式查看发布数据是否正常。

\$ rostopic list

```
orbbe@localhost:~$ rostopic list
/camera/camera_nodelet_manager/bond
/camera/depth/camera_info
/camera/depth/image
/camera/depth/image/compressed
/camera/depth/image/compressed/parameter_descriptions
/camera/depth/image/compressed/parameter_updates
/camera/depth/image/compressedDepth
/camera/depth/image/compressedDepth/parameter_descriptions
/camera/depth/image/compressedDepth/parameter_updates
/camera/depth/image/theora
/camera/depth/image/theora/parameter_descriptions
/camera/depth/image/theora/parameter_updates
/camera/depth/image_raw
/camera/depth/image_raw/compressed
/camera/depth/image_raw/compressed/parameter_descriptions
/camera/depth/image_raw/compressed/parameter_updates
/camera/depth/image_raw/compressedDepth
/camera/depth/image_raw/compressedDepth/parameter_descriptions
/camera/depth/image_raw/compressedDepth/parameter_updates
/camera/depth/image_raw/theora
/camera/depth/image_raw/theora/parameter_descriptions
/camera/depth/image_raw/theora/parameter_updates
/camera/depth/image_rect
/camera/depth/image_rect/compressed
/camera/depth/image_rect/compressed/parameter_descriptions
```

- 2) 上述发布的除了 image\_raw 部分由 Astra 驱动直接发布，大部分都是通过 rgbd\_launch 进行修改后发布。如果仅仅需要获取原始摄像头信息，主要关注下面 3 个 topic。

**/camera/rgb/image\_raw**

**/camera/depth/image\_raw**

**/camera/ir/image\_raw**

请注意由于 openni2 的限制，rgb 和 ir 无法同时输出。

如果需要获取点云数据，访问下面两个 topic

**/camera/depth/points**

**/camera/depth\_registered/points**

- 3) 查看 camera 信息

```
$ rostopic echo /camera/depth/camera_info
```

```
Header:
  seq: 226
  stamp:
    secs: 1500707483
    nsecs: 704247229
  frame_id: camera_depth_optical_frame
height: 480
width: 640
distortion_model: plumb_bob
D: [0.0, 0.0, 0.0, 0.0, 0.0]
K: [570.3422241210938, 0.0, 314.5, 0.0, 570.3422241210938, 235.5, 0.0, 0.0, 1.0]
R: [1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0]
P: [570.3422241210938, 0.0, 314.5, 0.0, 0.0, 570.3422241210938, 235.5, 0.0, 0.0,
0.0, 1.0, 0.0]
center_x: 0
center_y: 0
roi:
  x_offset: 0
  y_offset: 0
  height: 0
  width: 0
  do_rectify: False
--
```

#### 4) 查看话题发布频率

```
$ rostopic hz /camera/depth/image_raw
```

```
orbbe@localhost:~$ rostopic hz /camera/depth/image_raw
subscribed to [/camera/depth/image_raw]
average rate: 28.326
    min: 0.015s max: 0.050s std dev: 0.00682s window: 21
average rate: 27.338
    min: 0.015s max: 0.100s std dev: 0.01116s window: 48
average rate: 26.411
    min: 0.015s max: 0.100s std dev: 0.01197s window: 74
average rate: 26.985
```

#### 5) 查看深度数据 (ir/rgb 类似)

```
$ rostopic echo /camera/depth/image_raw
```

```
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
159, 2, 156, 2, 156, 2, 156, 2, 155, 2, 132, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
4, 9, 161, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 235, 5, 235, 5, 229, 5, 229, 5,
22, 5, 222, 5, 216, 5, 216, 5, 216, 5, 209, 5, 209, 5, 209, 5, 209, 5, 209, 5,
09, 5, 203, 5, 203, 5, 203, 5, 203, 5, 203, 5, 203, 5, 203, 5, 203, 5, 203, 5,
03, 5, 203, 5, 203, 5, 203, 5, 203, 5, 203, 5, 203, 5, 203, 5, 203, 5, 203, 5,
03, 5, 196, 5, 196, 5, 196, 5, 196, 5, 196, 5, 190, 5, 190, 5, 190, 5, 190, 5,
84, 5, 184, 5, 184, 5, 184, 5, 184, 5, 184, 5, 184, 5, 184, 5, 184, 5, 184, 5,
84, 5, 184, 5, 184, 5, 184, 5, 184, 5, 184, 5, 184, 5, 177, 5, 177, 5, 177, 5,
77, 5, 177, 5, 177, 5, 171, 5, 171, 5, 171, 5, 171, 5, 171, 5, 171, 5, 165, 5,
65, 5, 165, 5, 165, 5, 159, 5, 159, 5, 159, 5, 153, 5, 153, 5, 153, 5, 153, 5,
63, 5, 153, 5, 153, 5, 153, 5, 153, 5, 159, 5, 159, 5, 159, 5, 159, 5, 159, 5,
69, 5, 159, 5, 159, 5, 159, 5, 159, 5, 165, 5, 165, 5, 165, 5, 159, 5, 159, 5,
69, 5, 159, 5, 159, 5, 159, 5, 159, 5, 159, 5, 153, 5, 153, 5, 153, 5, 147, 5,
47, 5, 147, 5, 147, 5, 141, 5, 141, 5, 141, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 115, 11, 140, 11, 140, 11, 166, 11, 166, 11, 166, 11,
166, 11, 166, 11, 192, 11, 192, 11, 192, 11, 192, 11, 192, 11, 192, 11, 192, 11,
192, 11, 192, 11, 192, 11, 219, 11, 219, 11, 219, 11, 0, 0, 219, 11, 219, 11,
```

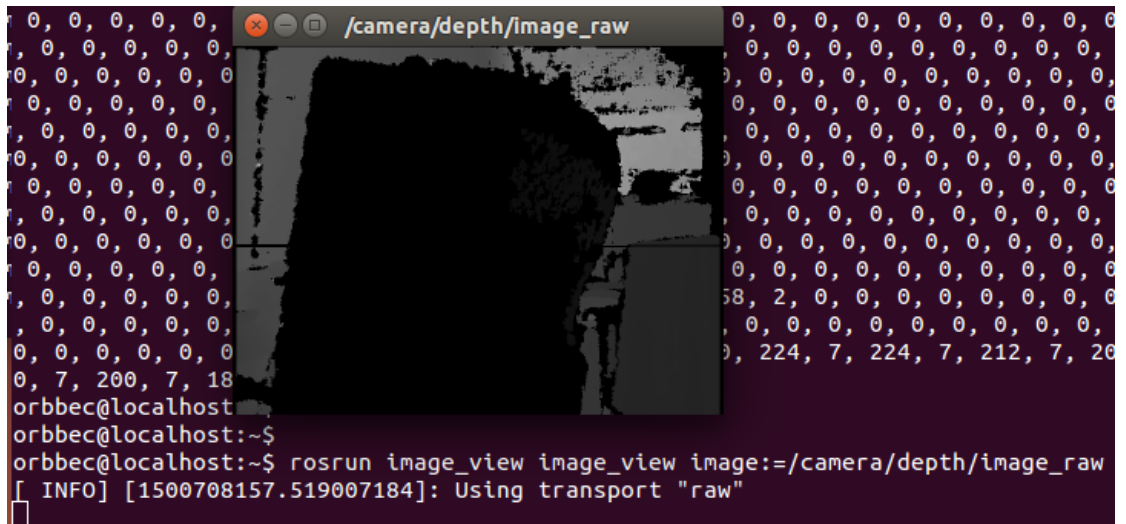
### 3, Image\_view 显示深度数据

可以通过安装 image\_view 工具直观查看摄像头发布的数据。

```
$ sudo apt-get install ros-indigo-image-view
```

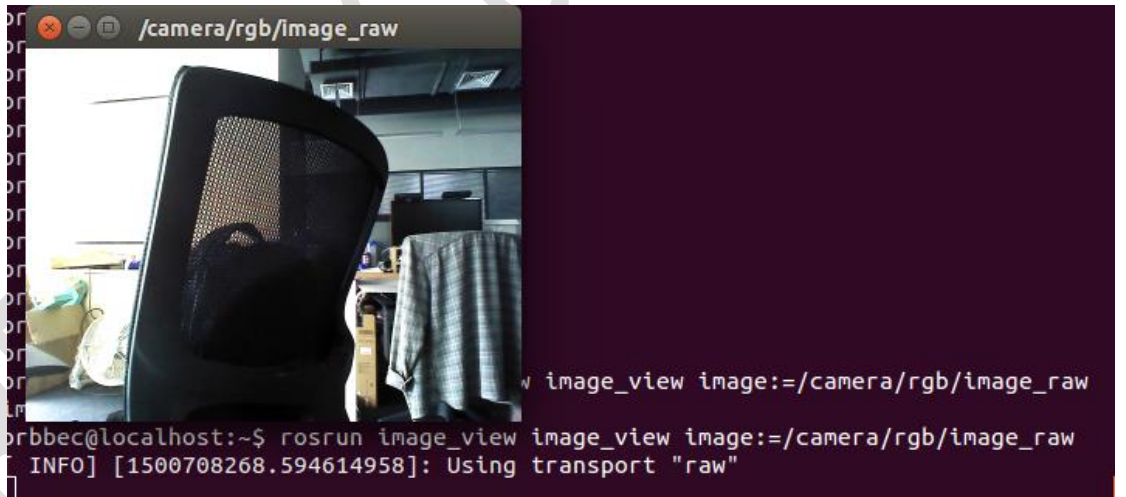
#### 1) 查看深度图

```
$ roslaunch image_view image_view image:=/camera/depth/image_raw
```



#### 2) 查看彩色图

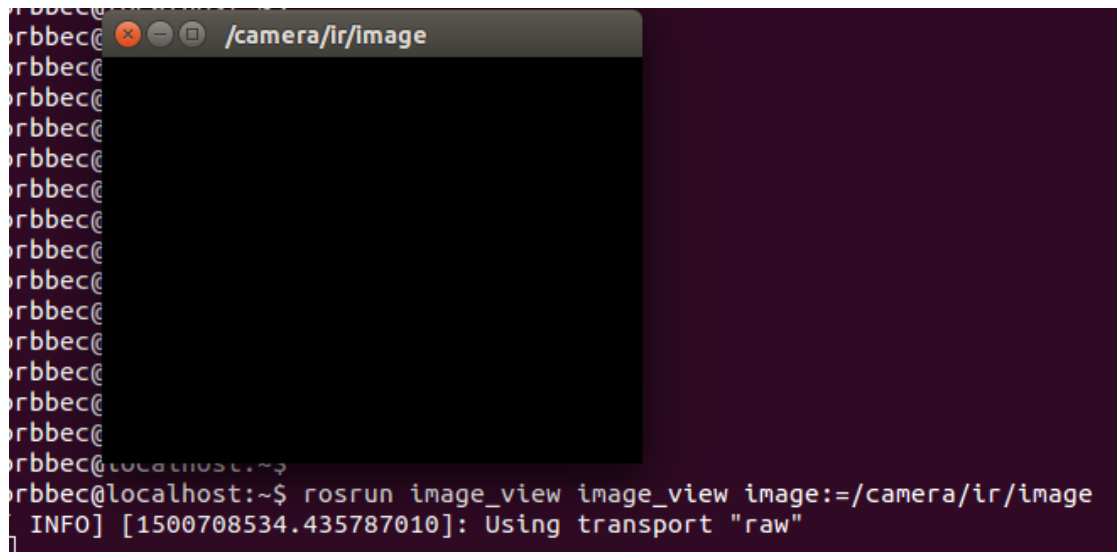
```
$ roslaunch image_view image_view image:=/camera/rgb/image_raw
```



#### 3) 查看 IR 图

```
$ roslaunch image_view image_view image:=/camera/ir/image
```



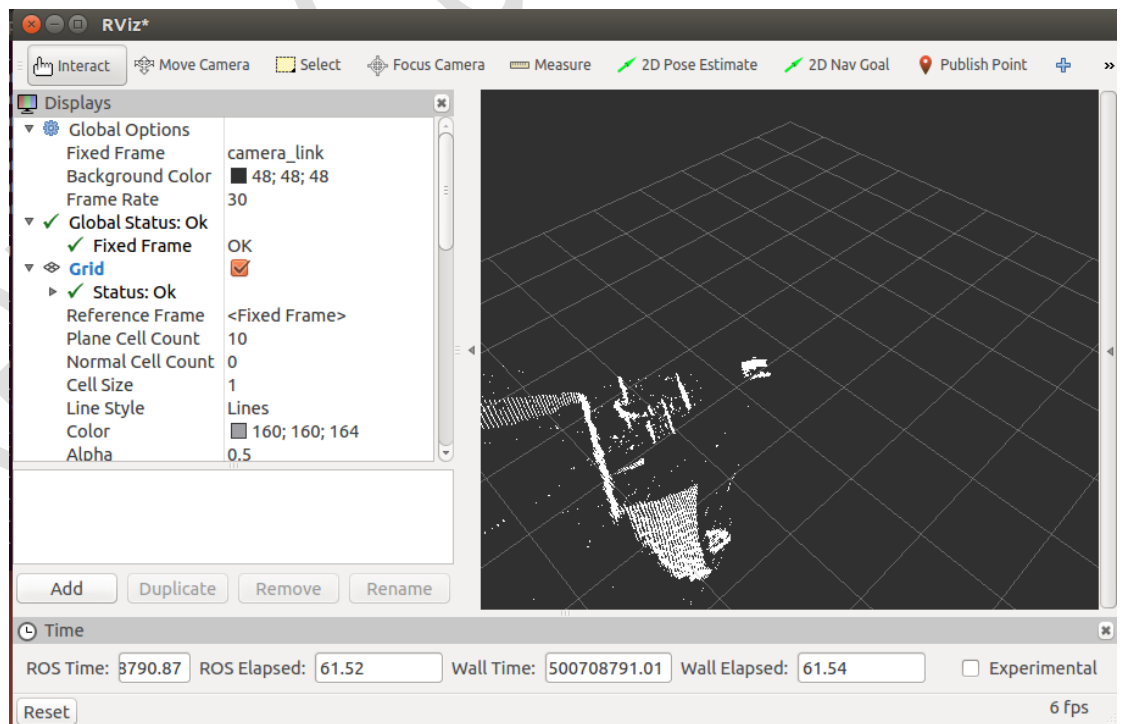


## 4，RVIZ 显示深度数据

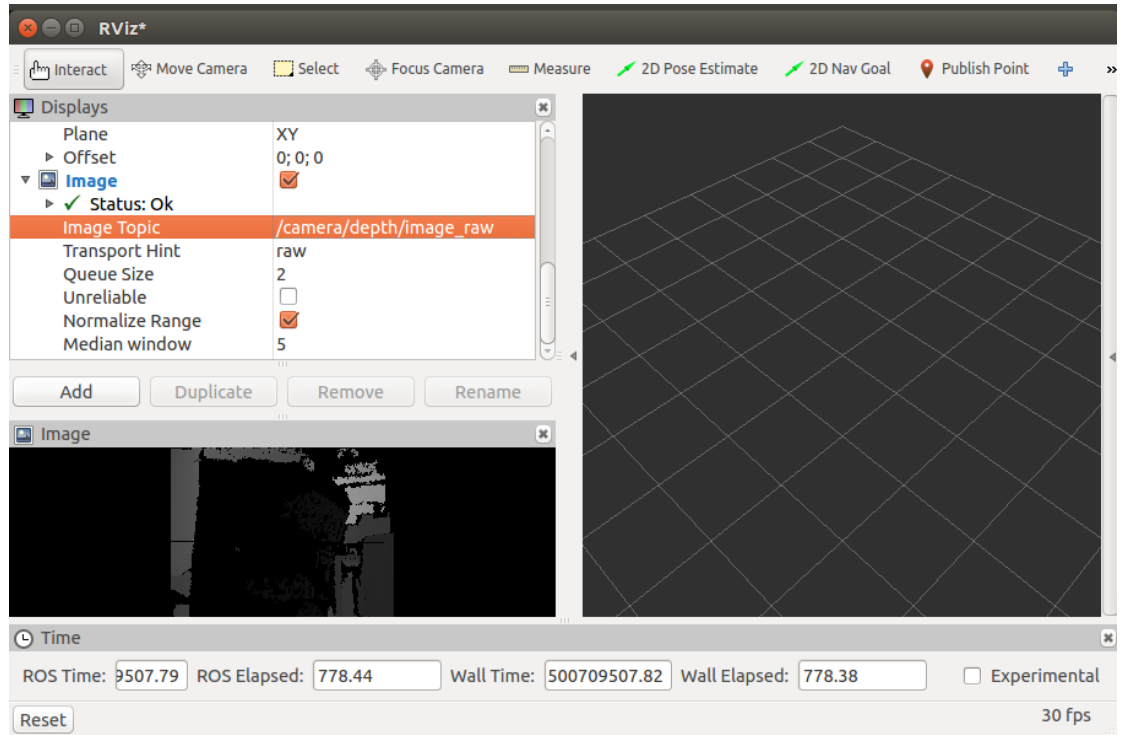
Rviz 是 ROS 下面的一个可视化开发工具，在 rviz 中我们除了可以加载上一步的 image 或者 camera 来实现图像数据可视化分析，还可以直接显示点云数据。

### 1) 运行 rviz

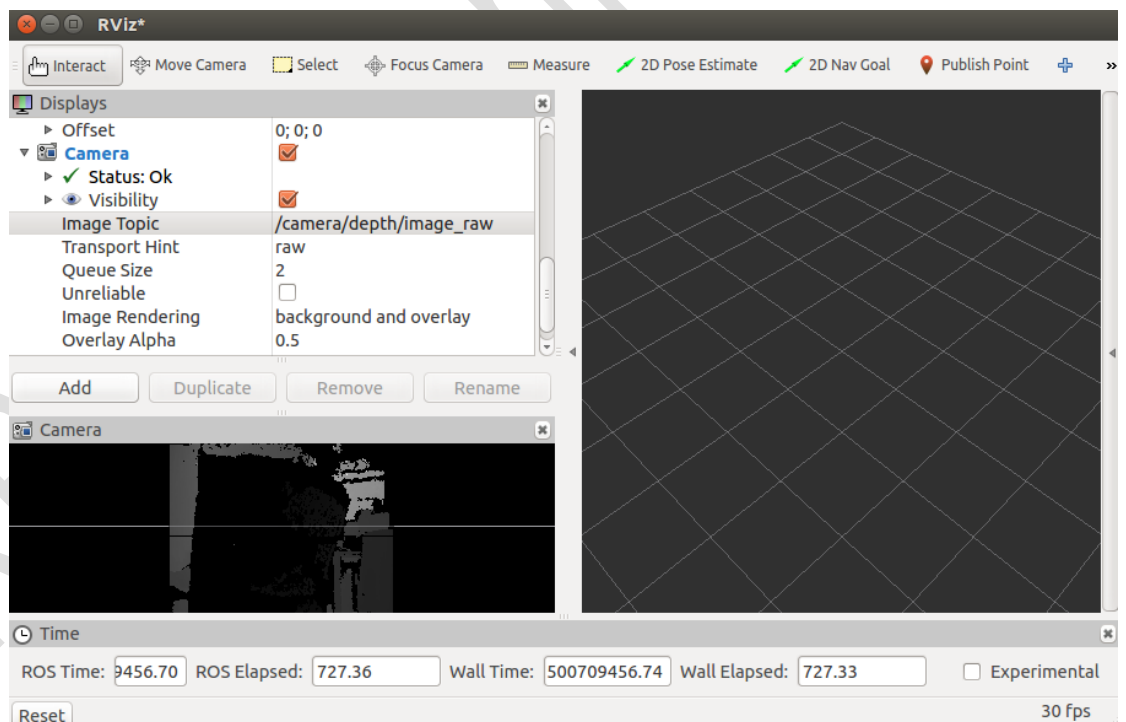
```
$ rviz
```



### 2) 增加一个 image ,点击 Add -> Image ,并在 image 属性页中 指定 topic 为 /camera/depth/image\_raw, 可以看到 image 窗口输出深度图。显示彩色和 ir 图类似，指定对应 topic 即可。

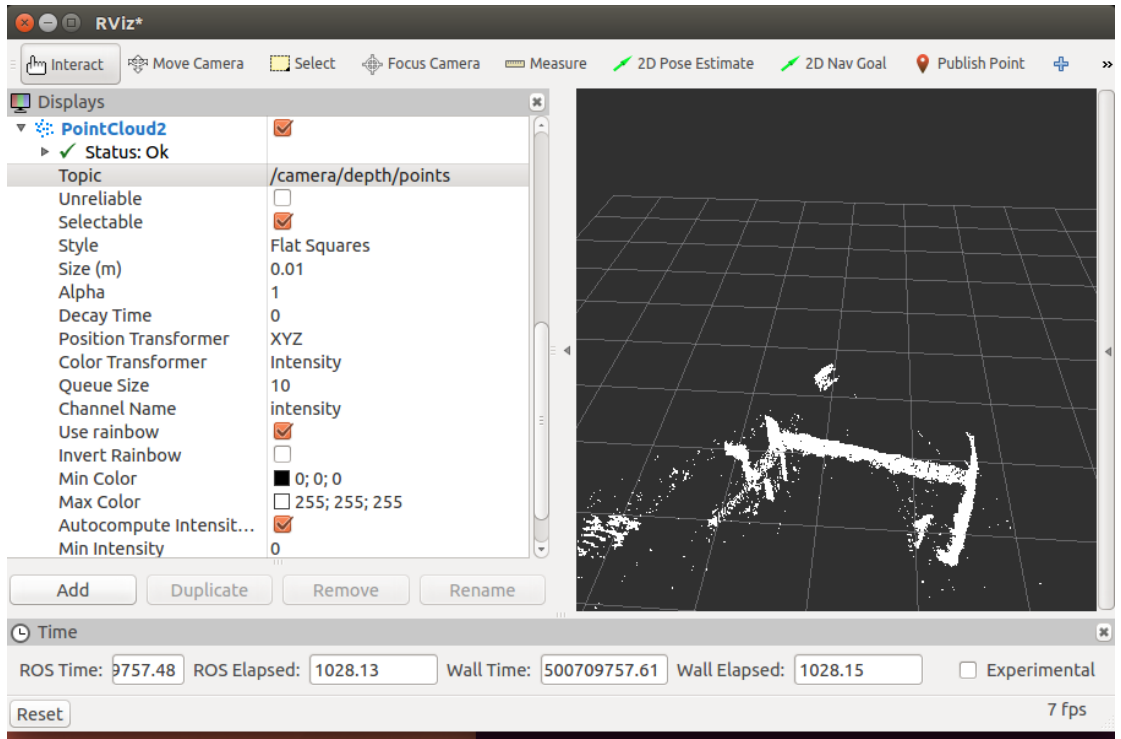


- 3) Camera 与 image 相同，不过可以调整 camera 一些其他参数。



- 4) 显示点云，点击 Add -> PointCloud2 增加点云显示，topic 设置为 /camera/depth/points。Add 中还有一个 Depthcloud 是 rviz 实时从 /camera/depth/image\_raw 转换而来，显示效果与 PointCloud 类似。





- 5) 显示彩色点云，PointCloud2 中 topic 设置为/camera/depth\_registered/points, color transformer 设置为 rgb8.

