# Displaying point cloud with ROS

Launch the OpenNI driver: roslaunch openni\_launch openni.launch

To visualize in rviz: rosrun rviz rviz  
Set the Fixed Frame (top left of rviz window) to /camera\_depth\_optical\_frame.

Add a PointCloud2 display, and set the topic to /camera/depth/points. Turning the background to light gray can help with viewing. This is the unregistered point cloud in the frame of the depth (IR) camera. It is not matched with the RGB camera images.

Now let’s look at a registered point cloud, aligned with the RGB data. Open the dynamic reconfigure GUI:  
rosrun rqt\_reconfigure rqt\_reconfigure

And select /camera/driver from the drop-down menu. Enable the depth\_registration checkbox.

Now go back to rviz, and change your PointCloud2 topic to /camera/depth\_registered/points. Set Color Transformer to RGB8. You should see a color, 3D point cloud of your scene.

• list connected usb devices:

lsusb

# Openni\_launch package

**• start openni kinect driver :**

* roslaunch openni\_launch openni.launch

• **visualization with rviz:**

* rosrun rviz rviz

• **registered point cloud:**

* *rosrun rqt\_reconfigure rqt\_reconfigure*
* select driver : check « *depth\_registration* »
* go back to rviz : PointCloud2 : select Topic « */camera/depth\_registered/points* », select Color Transformer « *RGB8*»

# ROS system commands

|  |  |
| --- | --- |
| **Stacks: (Ansammlung von Packages)** | **Listing running nodes:** |
| rosstack | Rosnode list |
| **Packages:** | **Node über Package name starten:** |
| rospack | rosrun [package\_name] [node\_name] |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# ROS package creation

|  |
| --- |
|  |
| • **visualization with ros:**  **Color image :**  rosrun image\_view image\_view image:=/camera/rgb/image\_color  **depth image:**  rosrun image\_view disparity\_view image:=/camera/depth\_registered/disparity  **• alle Bilddaten anzeigen:**  rostopic list  **• Überprüfen ob Kamera Bilder sendet:**  rostopic echo /camera/rgb/image\_color |
|  |
| Registration: matching depth to color (point clouds) rosrun rqt\_reconfigure rqt\_reconfigure  driver > check: register depth |

## From openni\_launch:

## Quick start

Launch the OpenNI driver. This will launch the full constellation of nodelets:

roslaunch openni\_launch openni.launch

To visualize in [rviz](http://wiki.ros.org/rviz):

rosrun rviz rviz

Set the Fixed Frame (top left of rviz window) to /camera\_depth\_optical\_frame.

Add a PointCloud2 display, and set the topic to /camera/depth/points. Turning the background to light gray can help with viewing. This is the unregistered point cloud in the frame of the depth (IR) camera. It is not matched with the RGB camera images.

Alternatively you can view the disparity image:

rosrun image\_view disparity\_view image:=/camera/depth/disparity

Now let's look at a registered point cloud, aligned with the RGB data. Open the dynamic reconfigure GUI:

rosrun rqt\_reconfigure rqt\_reconfigure

And select /camera/driver from the drop-down menu. Enable the depth\_registration checkbox.

Now go back to rviz, and change your PointCloud2 topic to /camera/depth\_registered/points. Set Color Transformer to RGB8. You should see a color, 3D point cloud of your scene.

Again, you can view the registered disparity image:

rosrun image\_view disparity\_view image:=/camera/depth\_registered/disparity

To view the color image from the RGB camera outside of rviz:

rosrun image\_view image\_view image:=/camera/rgb/image\_color

or to view the grayscale image:

rosrun image\_view image\_view image:=/camera/rgb/image\_mono

# Checking camera/nodes :

To see if the Kinect is publishing RGB images, use:

rostopic echo /camera/rgb/image\_color

Since rviz isn't displaying the image though, this will most likely fail too. You can also try:

rosnode list

and check for the "openni\_camera" node.