Some details/fixes for ROS-programming final project:

12/10/16

There are some important differences between the Baxter simulator and “Merry” that you will need to accommodate to make her work for you. Most of this is document in Ch 8 of our text.

Taking snapshots with the Kinect on Merry:
Snapshots of example block poses can be taken as follows. First, start up Merry, then open a terminal and run “baxter_master”. This will be necessary in every terminal that runs a ROS node, as this informs that terminal that the roscore is running on a remote computer (Merry's computer).

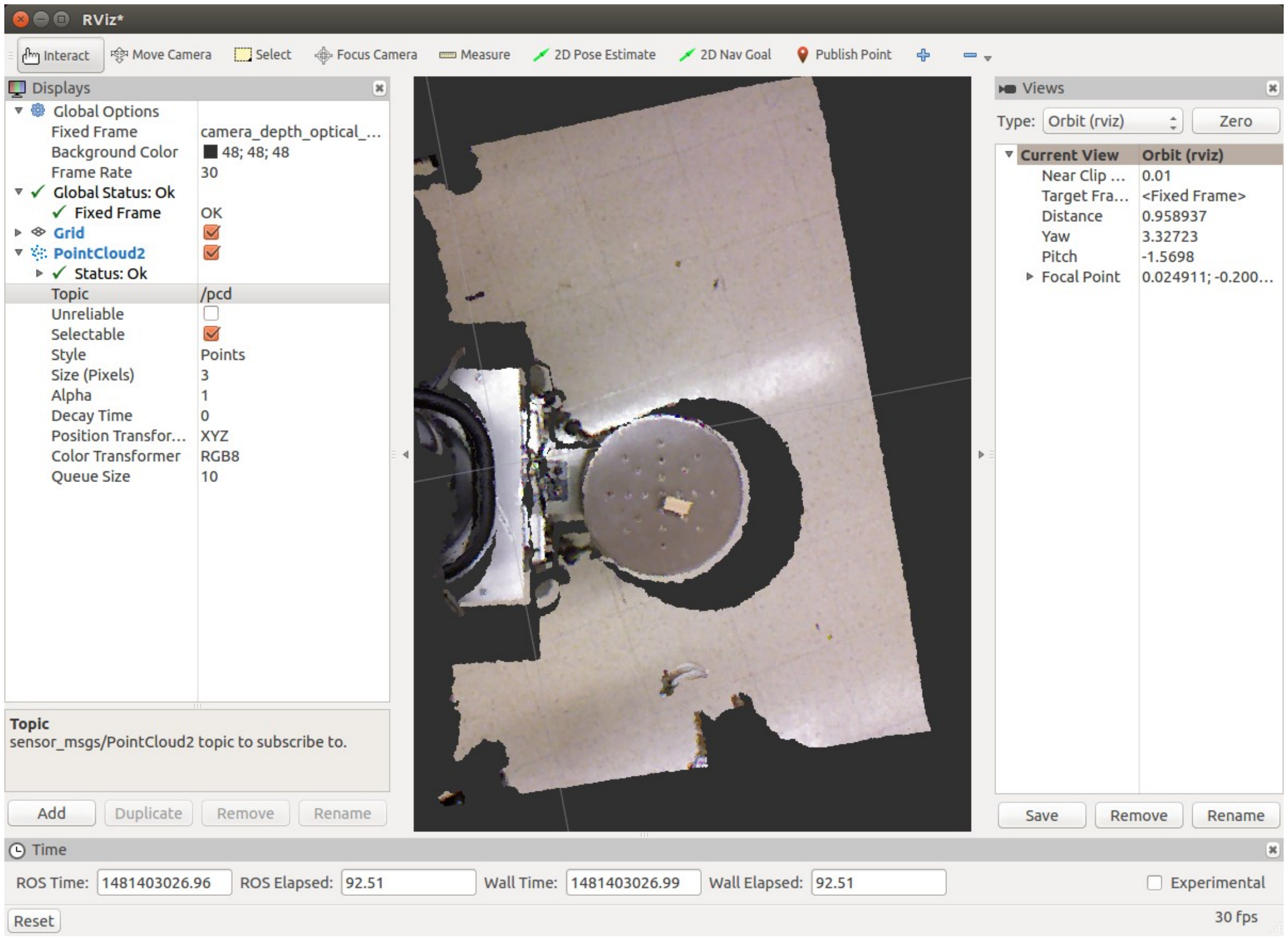
You can start up the Kinect with:
roslaunch freenect_launch freenect.launch

Then you can take a snapshot with:
rosrun pcl_utils pcd_snapshot

The filename of the new snapshot will be “kinect_snapshot.pcd.” It would be best to rename this file, so it does not get overwritten by future snapshots.

You can then view your PCD file via rviz by starting rviz and running:
rosrun pcl_utils display_pcd_file
then respond with the file name. (It will be easiest to run this node from the same directory that contains the snapshot file).

With the (metal) stool in front of Merry, here are a few example snapshots:



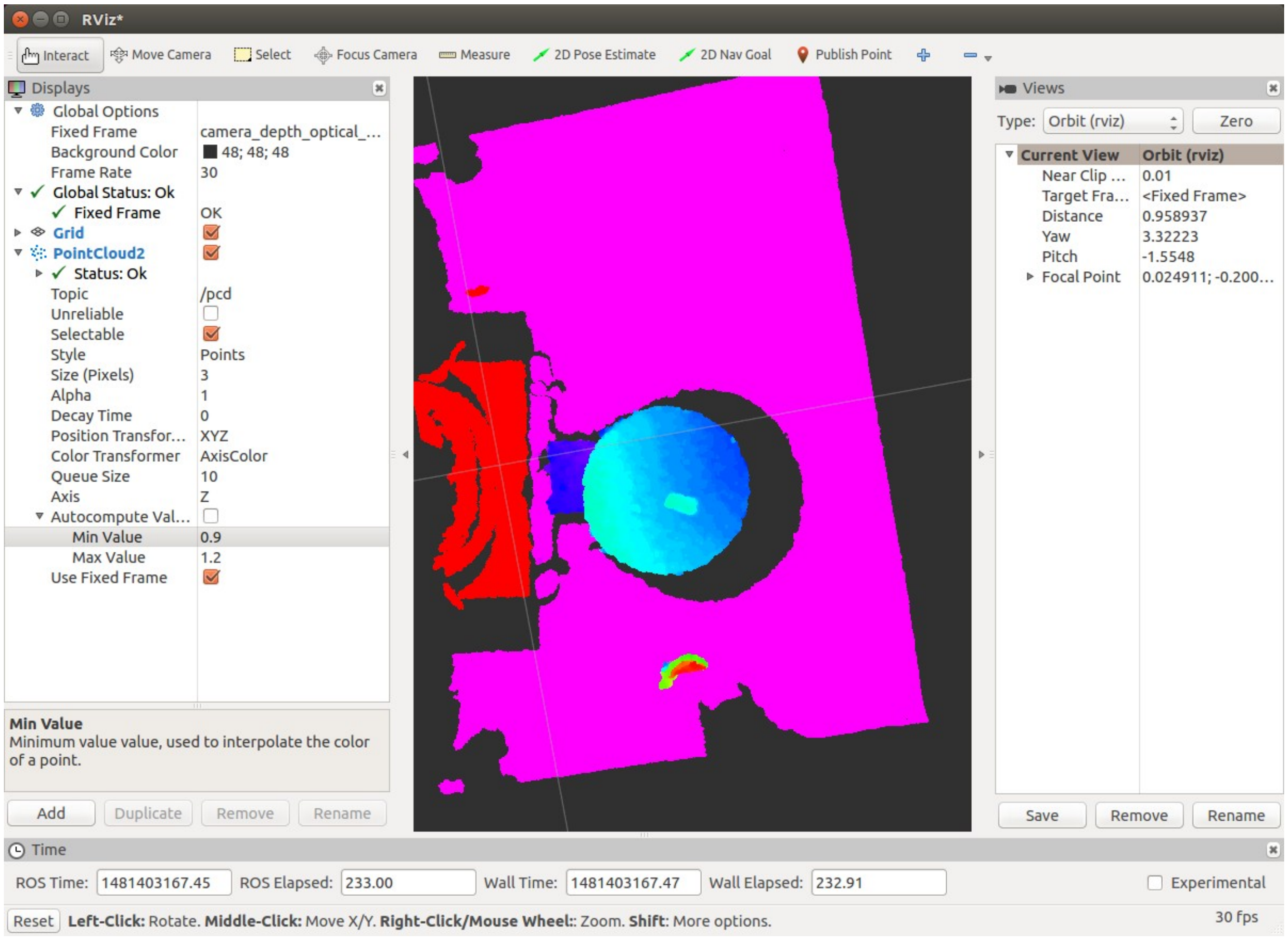
```
Global Options
  Fixed Frame            camera_depth_optical_...
  Background Color       48; 48; 48
  Frame Rate             30
Global Status: Ok
  Fixed Frame            OK
Grid                     ✓
PointCloud2              ✓
  Status: Ok
  Topic                  /pcd
  Unreliable             ☐
  Selectable             ✓
  Style                  Points
  Size (Pixels)          3
  Alpha                  1
  Decay Time             0
  Position Transfor...   XYZ
  Color Transformer      RGB8
  Queue Size             10
```

Topic
sensor_msgs/PointCloud2 topic to subscribe to.

```
Type: Orbit (rviz)
Current View             Orbit (rviz)
  Near Clip ...          0.01
  Target Fra...          <Fixed Frame>
  Distance               0.958937
  Yaw                    3.32723
  Pitch                  -1.5698
  Focal Point            0.024911; -0.200...
```

```
ROS Time: 1481403026.96   ROS Elapsed: 92.51   Wall Time: 1481403026.99   Wall Elapsed: 92.51   ☐ Experimental
30 fps
```

Illustration 1: view of rectangular block on stool



Global Options	
Fixed Frame	camera_depth_optical_...
Background Color	■ 48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	✓
PointCloud2	✓
✓ Status: Ok	
Topic	/pcd
Unreliable	☐
Selectable	✓
Style	Points
Size (Pixels)	3
Alpha	1
Decay Time	0
Position Transfor...	XYZ
Color Transformer	AxisColor
Queue Size	10
Axis	Z
Autocompute Val...	☐
Min Value	0.9
Max Value	1.2
Use Fixed Frame	✓

Min Value
Minimum value value, used to interpolate the color of a point.

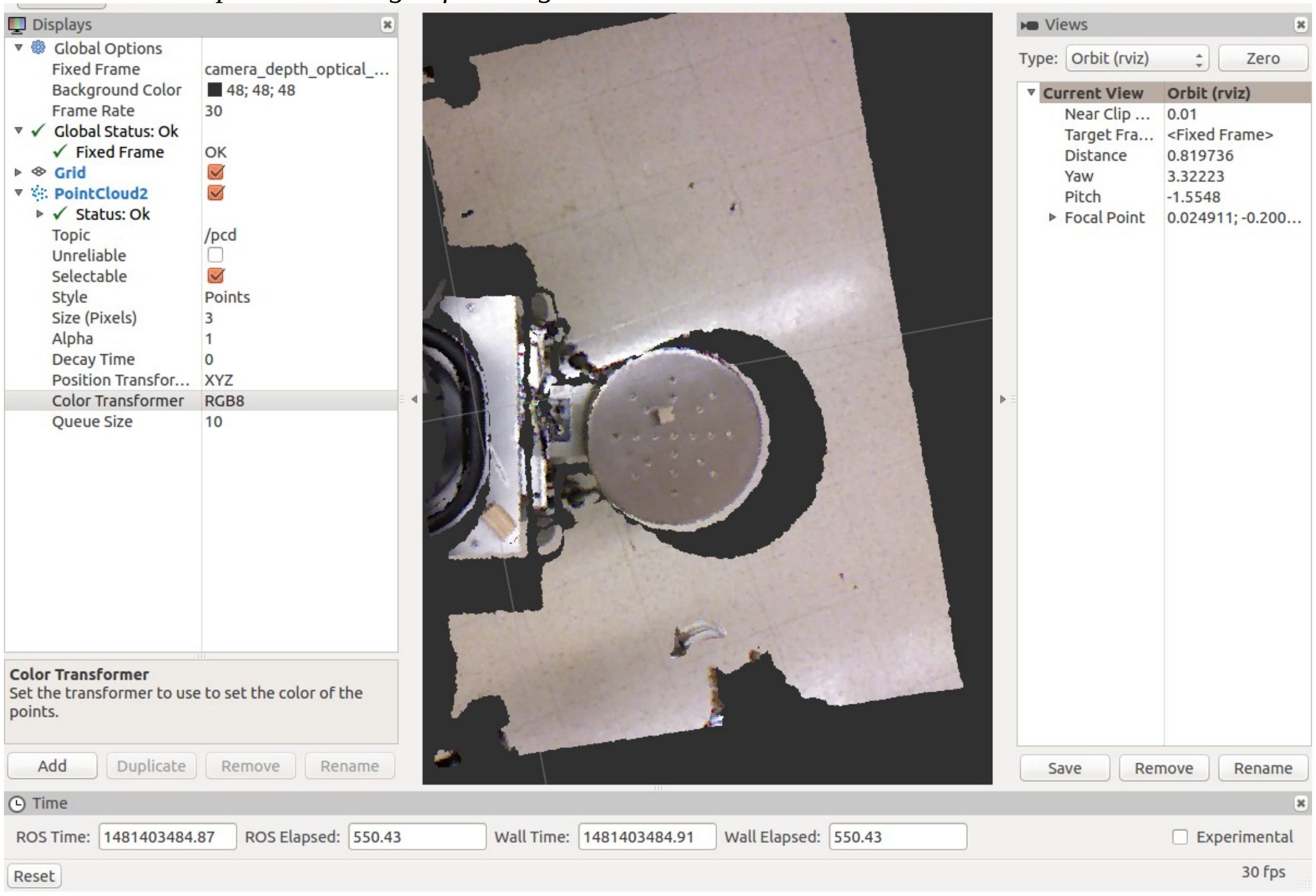
Type: Orbit (rviz)

Current View	Orbit (rviz)
Near Clip ...	0.01
Target Fra...	<Fixed Frame>
Distance	0.958937
Yaw	3.32223
Pitch	-1.5548
Focal Point	0.024911; -0.200...

ROS Time: 1481403167.45 ROS Elapsed: 233.00 Wall Time: 1481403167.47 Wall Elapsed: 232.91 ☐ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options. 30 fps

Illustration 2: depth-color image of rectangular block on stool



Global Options	
Fixed Frame	camera_depth_optical_...
Background Color	■ 48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	✓
PointCloud2	✓
✓ Status: Ok	
Topic	/pcd
Unreliable	☐
Selectable	✓
Style	Points
Size (Pixels)	3
Alpha	1
Decay Time	0
Position Transfor...	XYZ
Color Transformer	RGB8
Queue Size	10

Color Transformer
Set the transformer to use to set the color of the points.

Type: Orbit (rviz)

Current View	Orbit (rviz)
Near Clip ...	0.01
Target Fra...	<Fixed Frame>
Distance	0.819736
Yaw	3.32223
Pitch	-1.5548
Focal Point	0.024911; -0.200...

ROS Time: 1481403484.87 ROS Elapsed: 550.43 Wall Time: 1481403484.91 Wall Elapsed: 550.43 ☐ Experimental

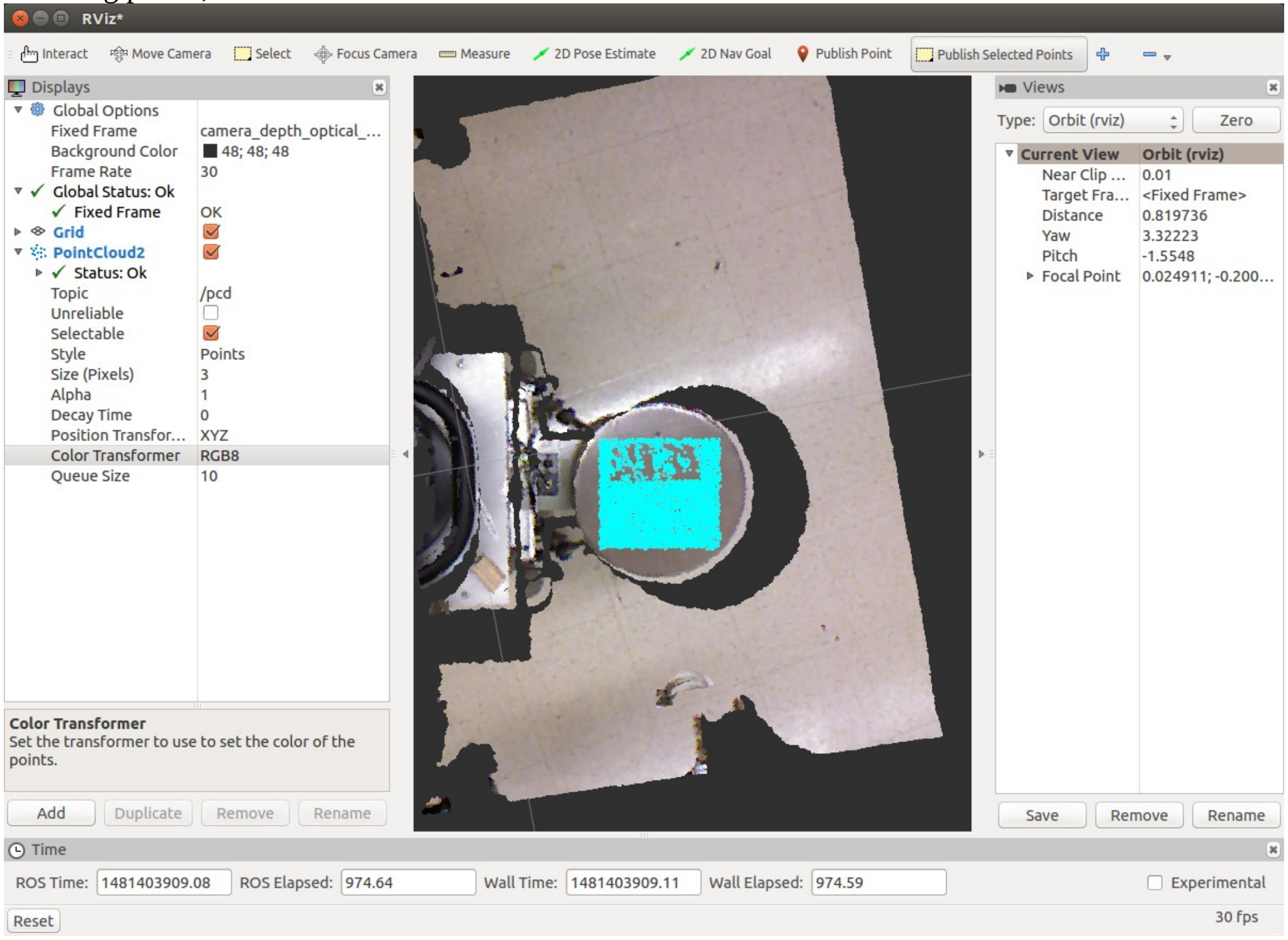
30 fps

Illustration 3: view of cube on stool

Finding Stool Height and Orientation:

It may also be useful to find the plane of the stool by capturing an image of the empty stool, then running:
rosrun pcl_utils find_plane_pcd_file
enter pcd file name: **empty_stool.pcd**

Selecting points, as shown below:



Global Options	
Fixed Frame	camera_depth_optical_...
Background Color	■ 48; 48; 48
Frame Rate	30
Global Status: Ok	
✓ Fixed Frame	OK
Grid	✓
PointCloud2	✓
✓ Status: Ok	
Topic	/pcd
Unreliable	☐
Selectable	✓
Style	Points
Size (Pixels)	3
Alpha	1
Decay Time	0
Position Transfor...	XYZ
Color Transformer	RGB8
Queue Size	10

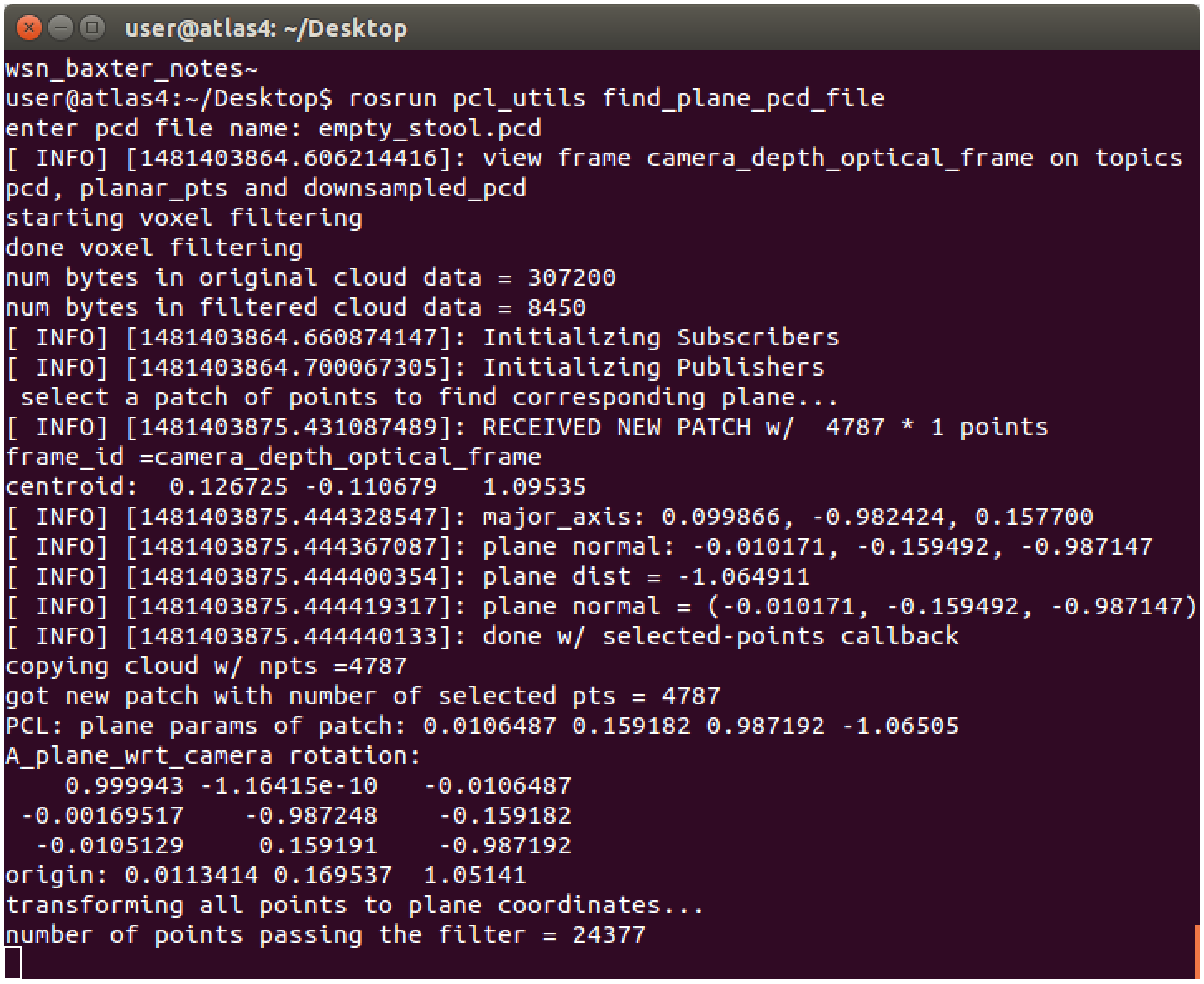
Color Transformer
Set the transformer to use to set the color of the points.

Type: Orbit (rviz)

Current View	Orbit (rviz)
Near Clip ...	0.01
Target Fra...	<Fixed Frame>
Distance	0.819736
Yaw	3.32223
Pitch	-1.5548
Focal Point	0.024911; -0.200...

ROS Time: 1481403909.08 ROS Elapsed: 974.64 Wall Time: 1481403909.11 Wall Elapsed: 974.59 ☐ Experimental
30 fps

results in terminal output from “find_plane_pcd_file” as follows:



```
user@atlas4: ~/Desktop
wsn_baxter_notes~
user@atlas4:~/Desktop$ rosrun pcl_utils find_plane_pcd_file
enter pcd file name: empty_stool.pcd
[ INFO] [1481403864.606214416]: view frame camera_depth_optical_frame on topics
pcd, planar_pts and downsampled_pcd
starting voxel filtering
done voxel filtering
num bytes in original cloud data = 307200
num bytes in filtered cloud data = 8450
[ INFO] [1481403864.660874147]: Initializing Subscribers
[ INFO] [1481403864.700067305]: Initializing Publishers
 select a patch of points to find corresponding plane...
[ INFO] [1481403875.431087489]: RECEIVED NEW PATCH w/  4787 * 1 points
frame_id =camera_depth_optical_frame
centroid:  0.126725 -0.110679   1.09535
[ INFO] [1481403875.444328547]: major_axis: 0.099866, -0.982424, 0.157700
[ INFO] [1481403875.444367087]: plane normal: -0.010171, -0.159492, -0.987147
[ INFO] [1481403875.444400354]: plane dist = -1.064911
[ INFO] [1481403875.444419317]: plane normal = (-0.010171, -0.159492, -0.987147)
[ INFO] [1481403875.444440133]: done w/ selected-points callback
copying cloud w/ npts =4787
got new patch with number of selected pts = 4787
PCL: plane params of patch: 0.0106487 0.159182 0.987192 -1.06505
A_plane_wrt_camera rotation:
    0.999943 -1.16415e-10   -0.0106487
 -0.00169517    -0.987248    -0.159182
  -0.0105129     0.159191    -0.987192
origin: 0.0113414 0.169537  1.05141
transforming all points to plane coordinates...
number of points passing the filter = 24377
```

This data may be useful in tuning your camera-to-torso transform values. We see that the stool is approximately 1 meter below the camera frame. The z-axis of the stool surface is nearly vertical (from bvec = (-0.01, -0.16, -0.987). The y-component of this surface normal is tilted slightly (about 0.16 rad), as seen by the camera.

Publishing Camera Transforms:

To get the Kinect camera to work with the object_finder node, you will need to get the frame names and transforms correct. With the Kinect running, you can see the sensor's reference frame by running:

rostopic echo /kinect/depth/points/header

which responds with:

frame_id: camera_rgb_optical_frame

The frame "camera_rgb_optical_frame" needs to be related to the robot's kinematic tree. However, this frame is already a child of parent "camera_link," as can be seen from running:

rosrun tf tf_echo camera_link camera_rgb_optical_frame
which outputs:

```
At time 1481405822.657
- Translation: [0.000, -0.045, 0.000]
- Rotation: in Quaternion [-0.500, 0.500, -0.500, 0.500]
            in RPY (radian) [-1.571, -0.000, -1.571]
            in RPY (degree) [-90.000, -0.000, -90.000]
```

Therefore, we need to relate "camera_link" to the robot. Note that the Baxter model defines a frame "base" that is synonymous with "torso". Thus, data transformed to the "base" frame will have a zero z-height at the level of the torso mounting plate (about 1m above the floor).

To relate the camera_rgb_optical_frame to baxter's torso (or base), include a line in your launch file, something like:

```
<node name="from_torso_to_camera_link_static_tf" pkg="tf" type="static_transform_publisher"
args="0.2 0.1 0.8 0 1.5 9 base camera_link 50" />
```

The numbers above are grossly correct, but they need to be tuned.

The node "object_finder_as.cpp" will need a couple of important changes. First, near line 237, the object finder attempts to find a published transform from camera to base:

```
tfListener.lookupTransform("base", "camera_link", ros::Time(0), stf_kinect_wrt_base);
```

Change this line to lookup the transform of "camera_rgb_optical_frame" to base (or torso):

```
tfListener.lookupTransform("base", "camera_rgb_optical_frame", ros::Time(0), stf_kinect_wrt_base);
```

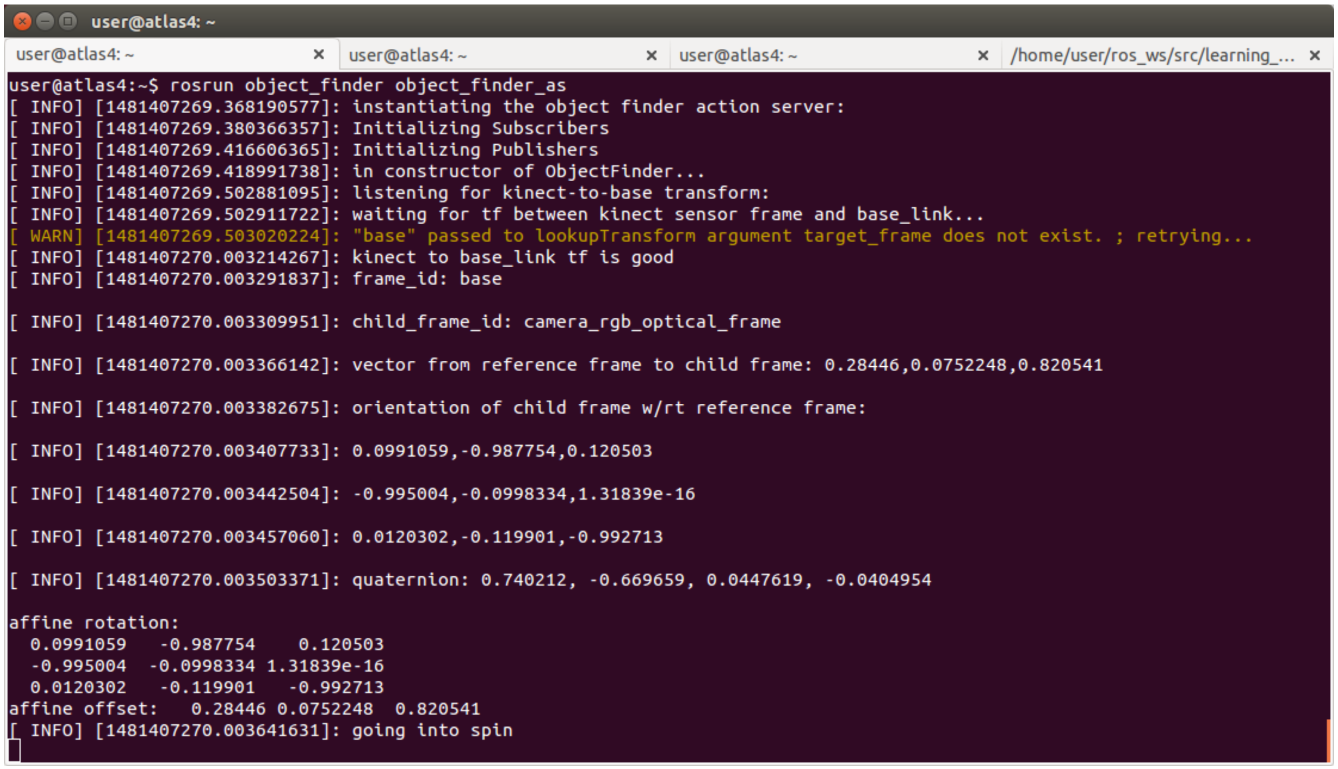
Another necessary change is the search range for the table height. The object_finder code assumed table heights relative to the floor. However, since we will transform sensor data to the torso frame, the stool height is actually *negative* relative to the torso frame. Therefore, near line 154, edit the table-height search range to:

```
 table_ht = pclUtils_.find_table_height(0.5, 1, -0.5, 0.5, -0.3, 0.3, 0.005);
```

This will enable the search from -0.3m height to +0.3m height. With this change (and with the camera-frame transform being published), running:

rosrun object_finder object_finder_as

will result in output such as:

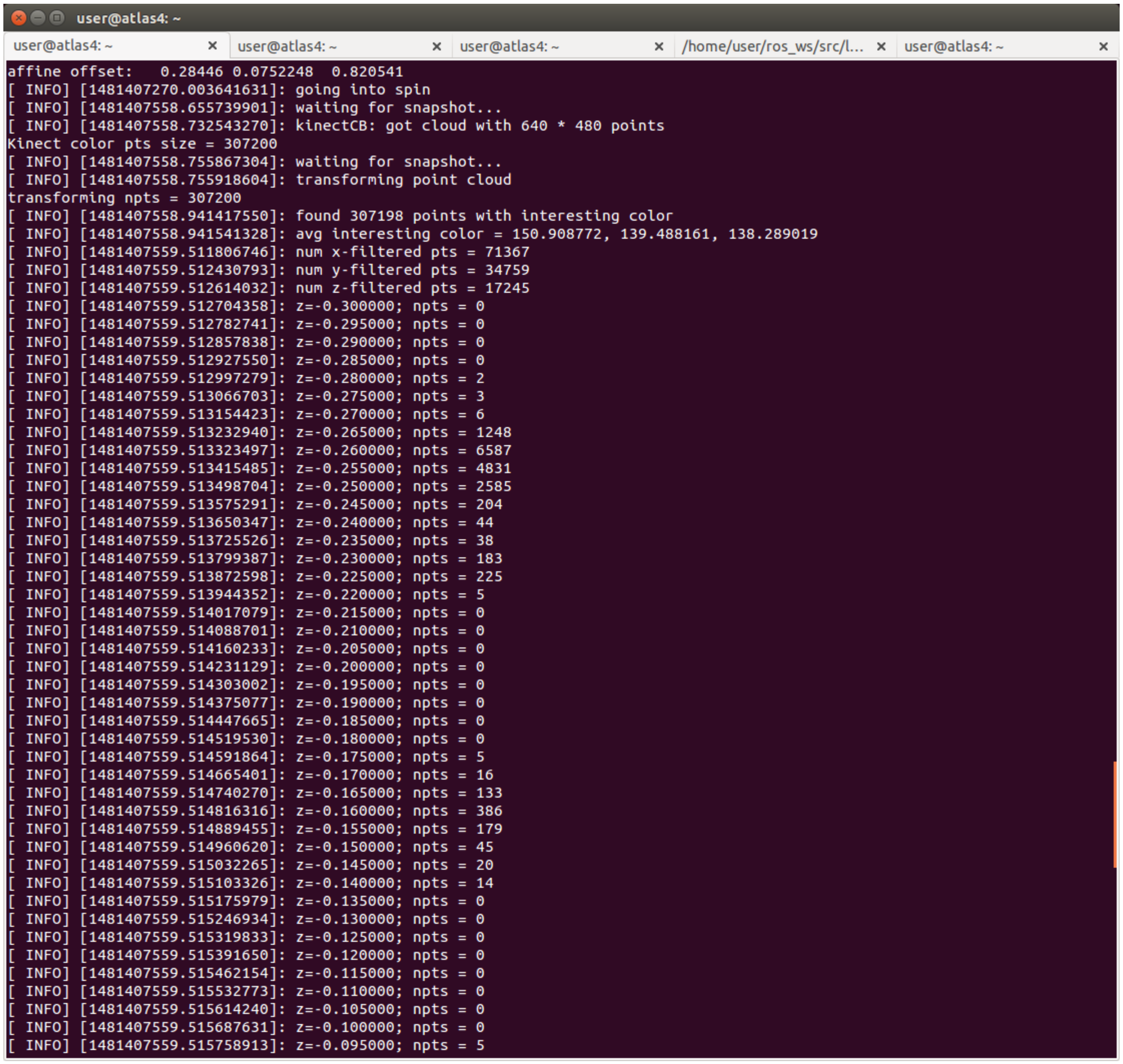


```
user@atlas4:~$ rosrun object_finder object_finder_as
[ INFO] [1481407269.368190577]: instantiating the object finder action server:
[ INFO] [1481407269.380366357]: Initializing Subscribers
[ INFO] [1481407269.416606365]: Initializing Publishers
[ INFO] [1481407269.418991738]: in constructor of ObjectFinder...
[ INFO] [1481407269.502881095]: listening for kinect-to-base transform:
[ INFO] [1481407269.502911722]: waiting for tf between kinect sensor frame and base_link...
[ WARN] [1481407269.503020224]: "base" passed to lookupTransform argument target_frame does not exist. ; retrying...
[ INFO] [1481407270.003214267]: kinect to base_link tf is good
[ INFO] [1481407270.003291837]: frame_id: base

[ INFO] [1481407270.003309951]: child_frame_id: camera_rgb_optical_frame

[ INFO] [1481407270.003366142]: vector from reference frame to child frame: 0.28446,0.0752248,0.820541

[ INFO] [1481407270.003382675]: orientation of child frame w/rt reference frame:

[ INFO] [1481407270.003407733]: 0.0991059,-0.987754,0.120503

[ INFO] [1481407270.003442504]: -0.995004,-0.0998334,1.31839e-16

[ INFO] [1481407270.003457060]: 0.0120302,-0.119901,-0.992713

[ INFO] [1481407270.003503371]: quaternion: 0.740212, -0.669659, 0.0447619, -0.0404954

affine rotation:
  0.0991059   -0.987754    0.120503
  -0.995004  -0.0998334 1.31839e-16
  0.0120302   -0.119901   -0.992713
affine offset:   0.28446 0.0752248  0.820541
[ INFO] [1481407270.003641631]: going into spin
```

The object finder is now ready to accept commands.

Running the block-finder client:
 rosrun object_finder example_object_finder_action_client

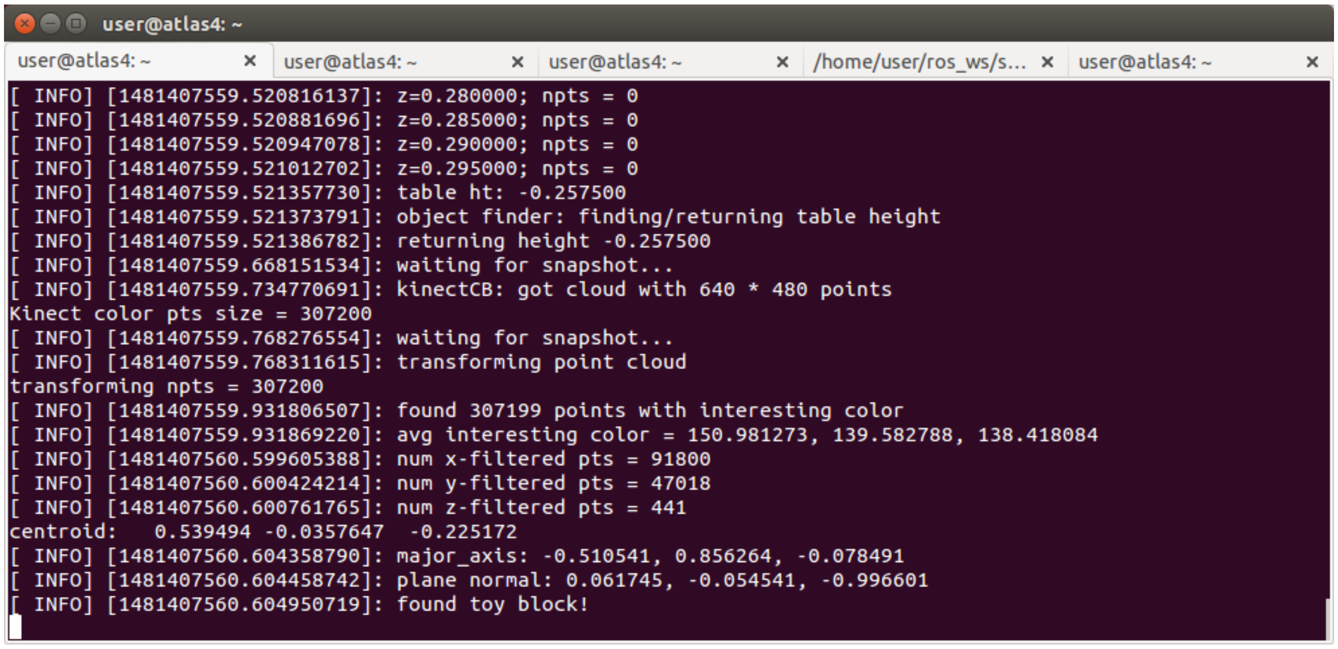
results in the object-finder displaying the following (in part):



```
affine offset:   0.28446 0.0752248  0.820541
[ INFO] [1481407270.003641631]: going into spin
[ INFO] [1481407558.655739901]: waiting for snapshot...
[ INFO] [1481407558.732543270]: kinectCB: got cloud with 640 * 480 points
Kinect color pts size = 307200
[ INFO] [1481407558.755867304]: waiting for snapshot...
[ INFO] [1481407558.755918604]: transforming point cloud
transforming npts = 307200
[ INFO] [1481407558.941417550]: found 307198 points with interesting color
[ INFO] [1481407558.941541328]: avg interesting color = 150.908772, 139.488161, 138.289019
[ INFO] [1481407559.511806746]: num x-filtered pts = 71367
[ INFO] [1481407559.512430793]: num y-filtered pts = 34759
[ INFO] [1481407559.512614032]: num z-filtered pts = 17245
[ INFO] [1481407559.512704358]: z=-0.300000; npts = 0
[ INFO] [1481407559.512782741]: z=-0.295000; npts = 0
[ INFO] [1481407559.512857838]: z=-0.290000; npts = 0
[ INFO] [1481407559.512927550]: z=-0.285000; npts = 0
[ INFO] [1481407559.512997279]: z=-0.280000; npts = 2
[ INFO] [1481407559.513066703]: z=-0.275000; npts = 3
[ INFO] [1481407559.513154423]: z=-0.270000; npts = 6
[ INFO] [1481407559.513232940]: z=-0.265000; npts = 1248
[ INFO] [1481407559.513323497]: z=-0.260000; npts = 6587
[ INFO] [1481407559.513415485]: z=-0.255000; npts = 4831
[ INFO] [1481407559.513498704]: z=-0.250000; npts = 2585
[ INFO] [1481407559.513575291]: z=-0.245000; npts = 204
[ INFO] [1481407559.513650347]: z=-0.240000; npts = 44
[ INFO] [1481407559.513725526]: z=-0.235000; npts = 38
[ INFO] [1481407559.513799387]: z=-0.230000; npts = 183
[ INFO] [1481407559.513872598]: z=-0.225000; npts = 225
[ INFO] [1481407559.513944352]: z=-0.220000; npts = 5
[ INFO] [1481407559.514017079]: z=-0.215000; npts = 0
[ INFO] [1481407559.514088701]: z=-0.210000; npts = 0
[ INFO] [1481407559.514160233]: z=-0.205000; npts = 0
[ INFO] [1481407559.514231129]: z=-0.200000; npts = 0
[ INFO] [1481407559.514303002]: z=-0.195000; npts = 0
[ INFO] [1481407559.514375077]: z=-0.190000; npts = 0
[ INFO] [1481407559.514447665]: z=-0.185000; npts = 0
[ INFO] [1481407559.514519530]: z=-0.180000; npts = 0
[ INFO] [1481407559.514591864]: z=-0.175000; npts = 5
[ INFO] [1481407559.514665401]: z=-0.170000; npts = 16
[ INFO] [1481407559.514740270]: z=-0.165000; npts = 133
[ INFO] [1481407559.514816316]: z=-0.160000; npts = 386
[ INFO] [1481407559.514889455]: z=-0.155000; npts = 179
[ INFO] [1481407559.514960620]: z=-0.150000; npts = 45
[ INFO] [1481407559.515032265]: z=-0.145000; npts = 20
[ INFO] [1481407559.515103326]: z=-0.140000; npts = 14
[ INFO] [1481407559.515175979]: z=-0.135000; npts = 0
[ INFO] [1481407559.515246934]: z=-0.130000; npts = 0
[ INFO] [1481407559.515319833]: z=-0.125000; npts = 0
[ INFO] [1481407559.515391650]: z=-0.120000; npts = 0
[ INFO] [1481407559.515462154]: z=-0.115000; npts = 0
[ INFO] [1481407559.515532773]: z=-0.110000; npts = 0
[ INFO] [1481407559.515614240]: z=-0.105000; npts = 0
[ INFO] [1481407559.515687631]: z=-0.100000; npts = 0
[ INFO] [1481407559.515758913]: z=-0.095000; npts = 5
```

We can see that the largest number of points in a slab was found at height -0.257. That is, the stool height is about 0.25m *below* the torso frame.

This is verified towards the end of the object_finder output:

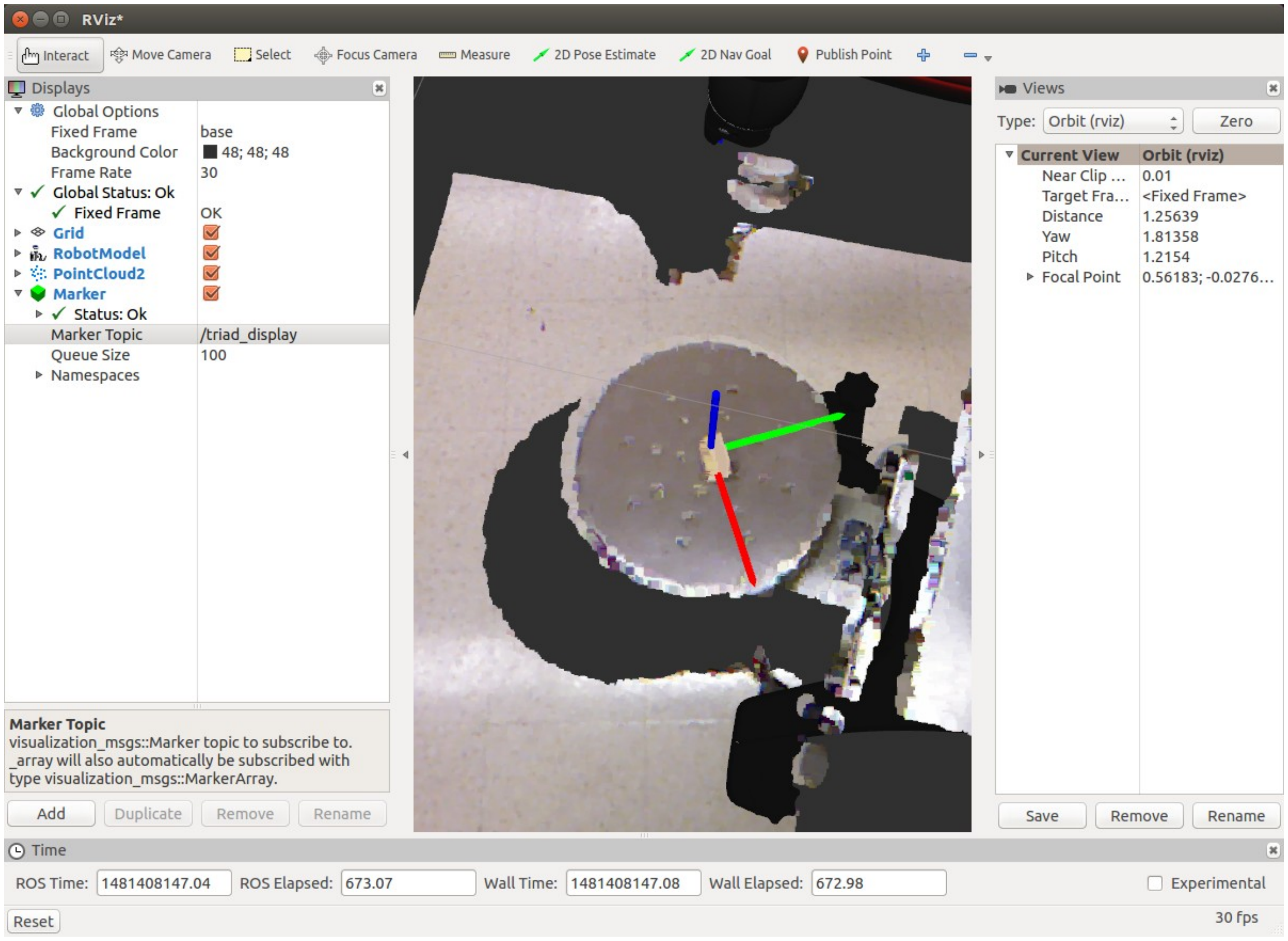


```
[ INFO] [1481407559.520816137]: z=0.280000; npts = 0
[ INFO] [1481407559.520881696]: z=0.285000; npts = 0
[ INFO] [1481407559.520947078]: z=0.290000; npts = 0
[ INFO] [1481407559.521012702]: z=0.295000; npts = 0
[ INFO] [1481407559.521357730]: table ht: -0.257500
[ INFO] [1481407559.521373791]: object finder: finding/returning table height
[ INFO] [1481407559.521386782]: returning height -0.257500
[ INFO] [1481407559.668151534]: waiting for snapshot...
[ INFO] [1481407559.734770691]: kinectCB: got cloud with 640 * 480 points
Kinect color pts size = 307200
[ INFO] [1481407559.768276554]: waiting for snapshot...
[ INFO] [1481407559.768311615]: transforming point cloud
transforming npts = 307200
[ INFO] [1481407559.931806507]: found 307199 points with interesting color
[ INFO] [1481407559.931869220]: avg interesting color = 150.981273, 139.582788, 138.418084
[ INFO] [1481407560.599605388]: num x-filtered pts = 91800
[ INFO] [1481407560.600424214]: num y-filtered pts = 47018
[ INFO] [1481407560.600761765]: num z-filtered pts = 441
centroid:   0.539494 -0.0357647  -0.225172
[ INFO] [1481407560.604358790]: major_axis: -0.510541, 0.856264, -0.078491
[ INFO] [1481407560.604458742]: plane normal: 0.061745, -0.054541, -0.996601
[ INFO] [1481407560.604950719]: found toy block!
```

Which declares the stool (table) height to be -0.257m.

The object-finder continues with searching for a block above the stool, and it returns computed coordinates. The frame assigned to the found block is shown in rviz by running the helper node:
rosrun example_rviz_marker triad_display

and viewing the result in rviz, as below:



Global Options	
Fixed Frame	base
Background Color	48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	✓
RobotModel	✓
PointCloud2	✓
Marker	✓
✓ Status: Ok	
Marker Topic	/triad_display
Queue Size	100
Namespaces	

Marker Topic
visualization_msgs::Marker topic to subscribe to. _array will also automatically be subscribed with type visualization_msgs::MarkerArray.

Type: Orbit (rviz) Zero

Current View	Orbit (rviz)
Near Clip ...	0.01
Target Fra...	<Fixed Frame>
Distance	1.25639
Yaw	1.81358
Pitch	1.2154
Focal Point	0.56183; -0.0276...

ROS Time: 1481408147.04 ROS Elapsed: 673.07 Wall Time: 1481408147.08 Wall Elapsed: 672.98 Experimental

30 fps

This view shows that the centroid of the block has been found reasonably well, as well as the orientation of the block (identifying the major axis as the x-axis in red).

In this example, the centroid of the block—as printed out by the object-finder—is:
centroid: 0.539494 -0.0357647 -0.225172

This is reasonable. The x-value is about 0.5m in front of the robot's torso frame, the block is approximately centered left/right (y-direction),and the z-value is -0.22, which is below the torso but above the stool surface.

It will require some tuning to get the coordinates calibrated adequately with respect to the torso frame. With valid block coordinates, the robot can be commanded to acquire the block.

You will need to modify the PCL code (or interpret the existing results) to distinguish rectangular blocks from cubes.

Object grabber:
Once valid block coordinates are found from vision, grabbing blocks can largely re-use the object_grabber action server. However, it will be necessary to modify the code to utilize the Yale hand.

In the “object_manipulation_properties” package, node “object_manipulation_query_svc.cpp”, there is provision to accommodate a Yale gripper. However, it will be easiest to simply pretend that Merry has a ReThink gripper (and set this gripper-ID on the parameter server) and re-use this code. *However*, it will be necessary to find (and publish) a suitable transform for “generic_gripper_frame” with respect to the “right_hand” frame of the Baxter model. (The right_hand frame is actually the right tool-flange frame).

You should publish a transform on “tf” that expresses a gripper frame (called “generic_gripper_frame”) that is defined as:

- origin between the fingertips
- z-axis pointing out from the tool flange
- x-axis orientation in the “slide” direction, i.e. parallel to the major axis of a grasped, rectangular block.

The numerical values in the object-manipulation query service have NOT been tested with Merry and the Yale gripper, so some necessary tuning/edits are expected.

The Yale gripper needs some repair yet. Additional information on the gripper will follow.