

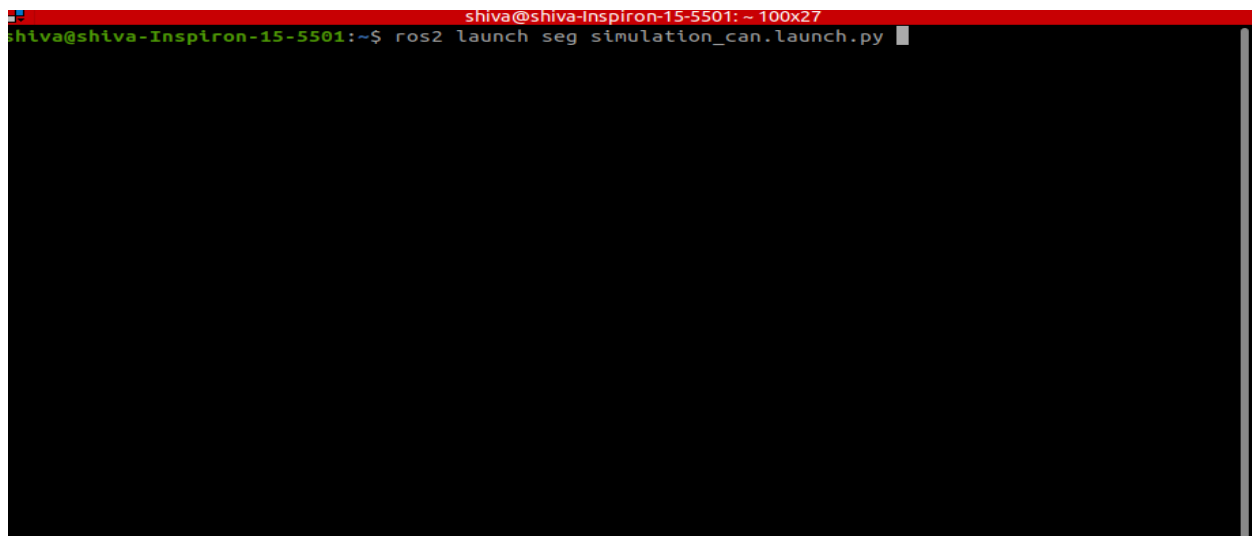
## DR Manual : Active Vision

Shiva Surya Lolla  
Saurabh Kashid  
Prarthana Sigedar  
Aditya Nisal

The below steps indicate how to use the package that generates object grasp points and stitches the object using PCL.

Step 1:

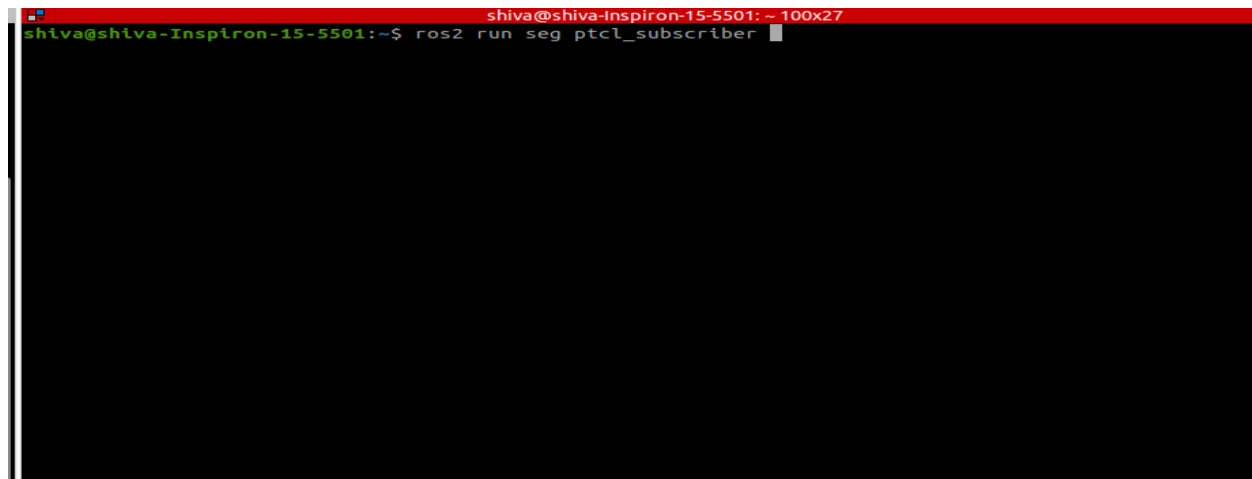
Open the terminal and run the following command to spawn the camera, object and table in Gazebo

A terminal window with a red title bar. The title bar text is "shiva@shiva-Inspiron-15-5501: ~ 100x27". The terminal content shows the command "ros2 launch seg simulation\_can.launch.py" being entered at the prompt "shiva@shiva-Inspiron-15-5501:~\$".

```
shiva@shiva-Inspiron-15-5501:~$ ros2 launch seg simulation_can.launch.py
```

Step 2:

Open another terminal and run the below command. The below command will give the object point cloud from a particular camera view and grasp points for that view.

A terminal window with a red title bar. The title bar text is "shiva@shiva-Inspiron-15-5501: ~ 100x27". The terminal content shows the command "ros2 run seg ptcl\_subscriber" being entered at the prompt "shiva@shiva-Inspiron-15-5501:~\$".

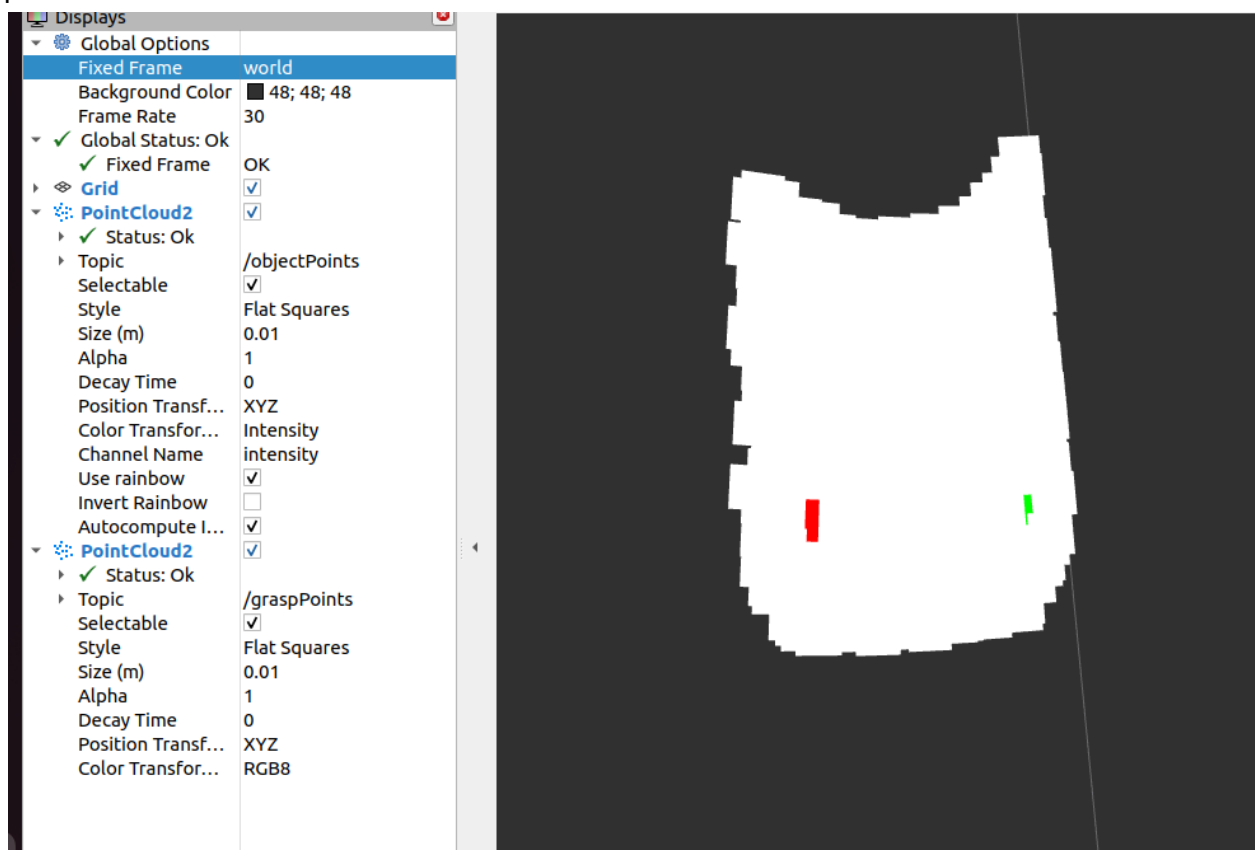
```
shiva@shiva-Inspiron-15-5501:~$ ros2 run seg ptcl_subscriber
```

Step 3:

Open Rviz2 for visualizing object point cloud and grasp points

```
shiva@shiva-Inspiron-15-5501: ~ 100x27
shiva@shiva-Inspiron-15-5501:~$ ros2 run rviz2 rviz2
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland to run on Wayland a
nyway.
[INFO] [1671124991.965899742] [rviz2]: Stereo is NOT SUPPORTED
[INFO] [1671124991.966298182] [rviz2]: OpenGL version: 4.6 (GLSL 4.6)
[INFO] [1671124992.080437375] [rviz2]: Stereo is NOT SUPPORTED
```

Select the topic /objectPoints to view object point cloud and /graspPoints to view the grasp points for that cloud



#### Step 4:

Run the below command to stitch point clouds for coke can. For other objects the code should be changed to below code (lines 194 and 196 index changed from 3 to 2)

```
shiva@shiva-Inspiron-15-5501: ~ 100x27
shiva@shiva-Inspiron-15-5501:~$ ros2 run seg stitch

void topic_callback1(const gazebo_msgs::msg::LinkStates msg1) //Get the message of /demo/link_states
{
    sleep(10); //delay of 10 seconds between processing messages for better point cloud stitching. We
    const std::vector<std::string> &names = msg1.name;
    std::string i = names[2]; //as there are different objects like camera, object, table in gazebo we
    std::cout<<"i:"<<i<<std::endl; //print to terminal to verify
    const geometry_msgs::msg::Pose camera_pose = msg1.pose[2]; //msg1.pose[3] needs to be used fof coke
    //input camera pose information into a new message
    message1.transform.translation.x = camera_pose.position.x;
    message1.transform.translation.y = camera_pose.position.y;
    message1.transform.translation.z = camera_pose.position.z;
    message1.transform.rotation.x = camera_pose.orientation.x;
    message1.transform.rotation.y = camera_pose.orientation.y;
    message1.transform.rotation.z = camera_pose.orientation.z;
```

#### Step 5:

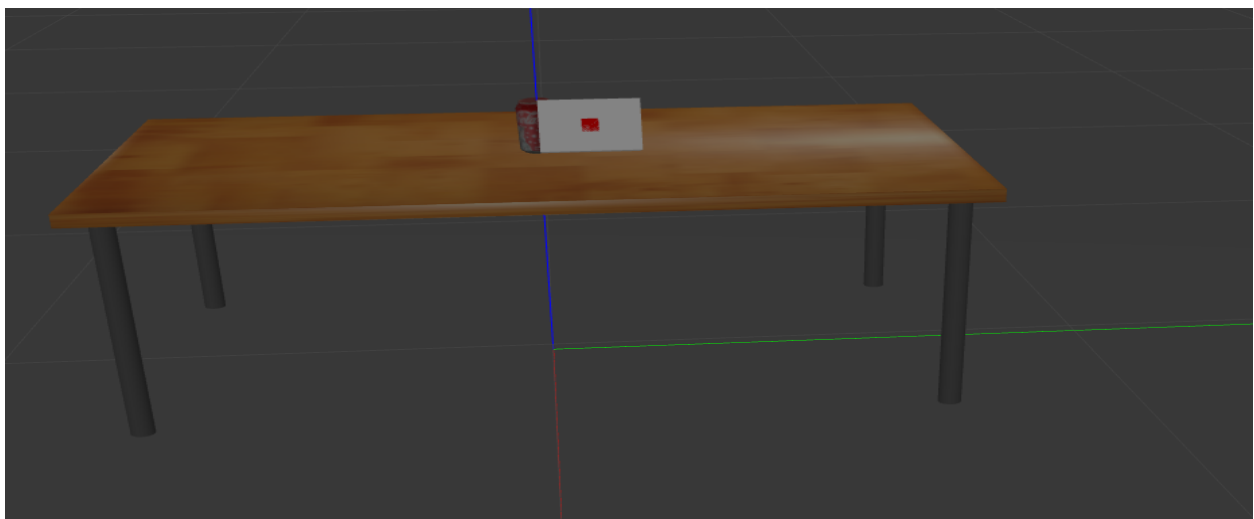
Start moving the camera in gazebo. First give a small translation to the right after getting terminal feedback that transform has changed

```
shiva@shiva-Inspiron-15-5501: ~ 100x27
message1.transform.rotation.x:1.11996e-06
message1.transform.rotation.y:0.989391
message1.transform.rotation.z:1.11996e-06
message1.transform.rotation.w:0.145281
[INFO] [1671125710.326882572] [stitcher]: got transform: 0.000796327 0.000796326 0.999999
0
-1 6.34136e-07 0.000796326 0
2.22045e-16 -1 0.000796327 0
0 0 0 1
[INFO] [1671125710.327171007] [stitcher]: transform was not initialized
i:camera::camera_link
message1.transform.translation.x:0.5
message1.transform.translation.y:0
message1.transform.translation.z:0.7
message1.transform.rotation.x:1.11996e-06
message1.transform.rotation.y:0.989391
message1.transform.rotation.z:1.11996e-06
message1.transform.rotation.w:0.145281
[INFO] [1671125720.330508565] [stitcher]: got transform: 0.000796327 0.000796326 0.999999
0
-1 6.34136e-07 0.000796326 0
2.22045e-16 -1 0.000796327 0
0 0 0 1
[INFO] [1671125720.331837507] [stitcher]: transform has changed, SERIOUS MODE -_-
[INFO] [1671125720.342538848] [stitcher]: stitched cloud, it's gonna rain
i:camera::camera_link
message1.transform.translation.x:0.5
```

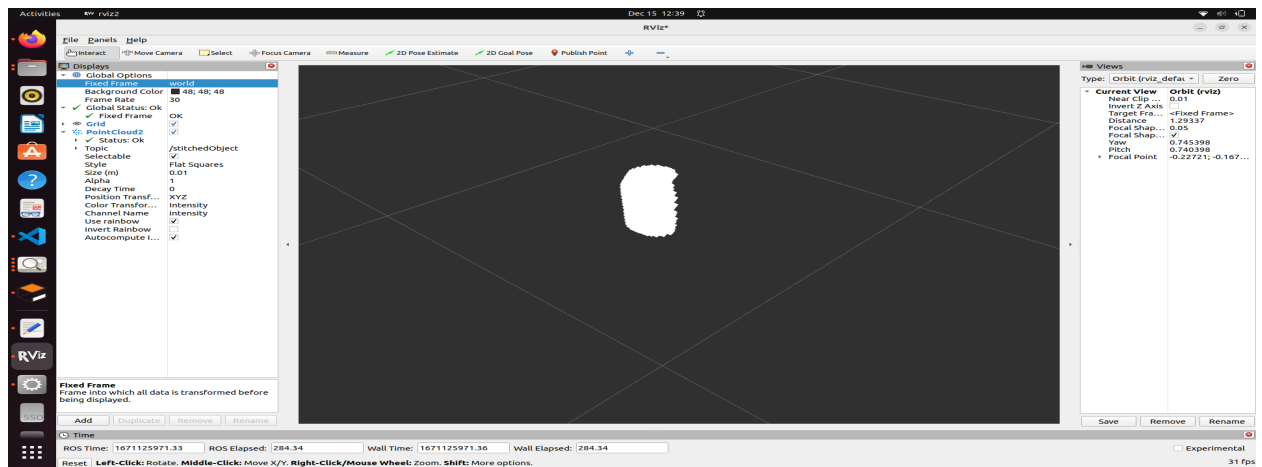
Initial camera pose



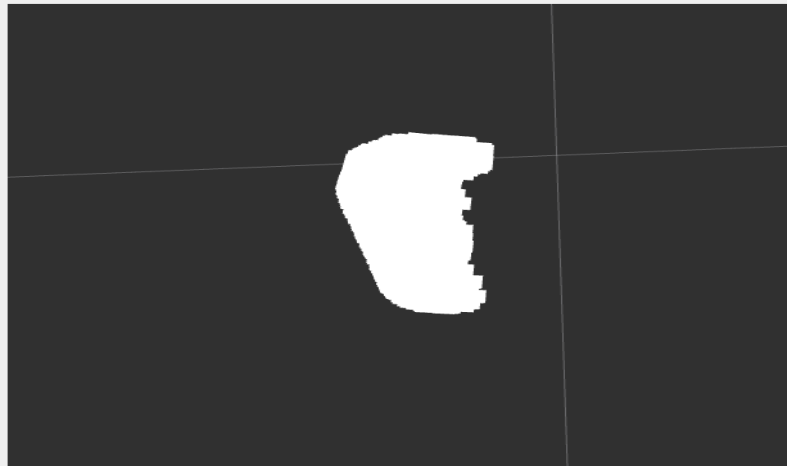
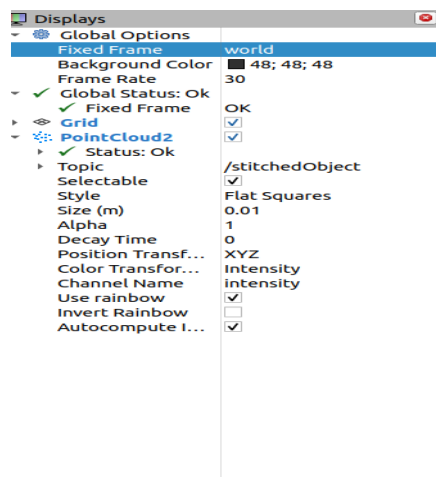
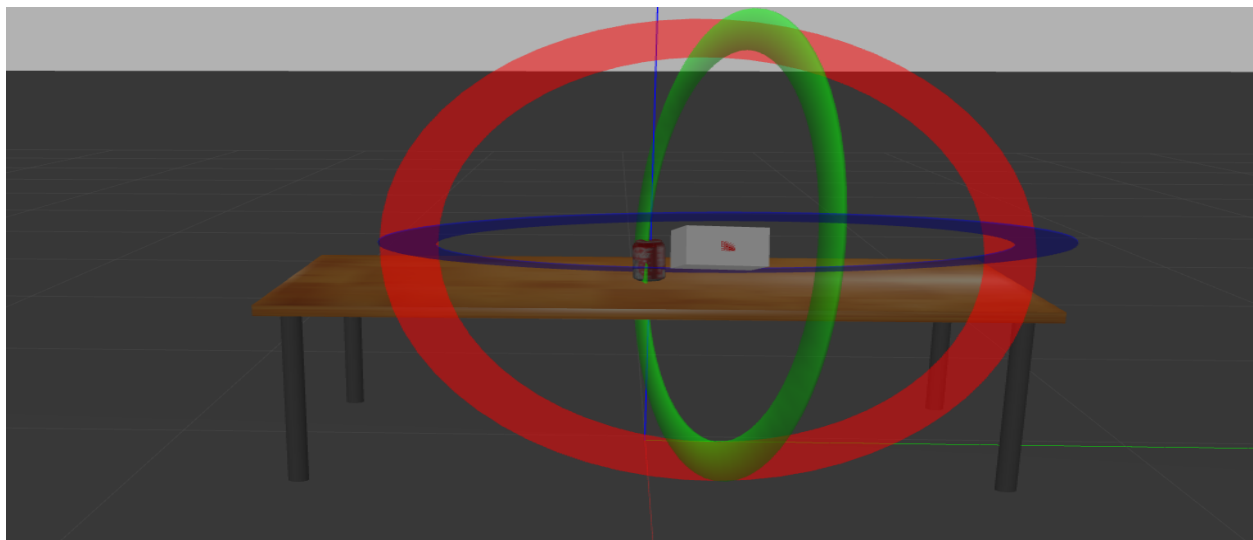
Next camera pose



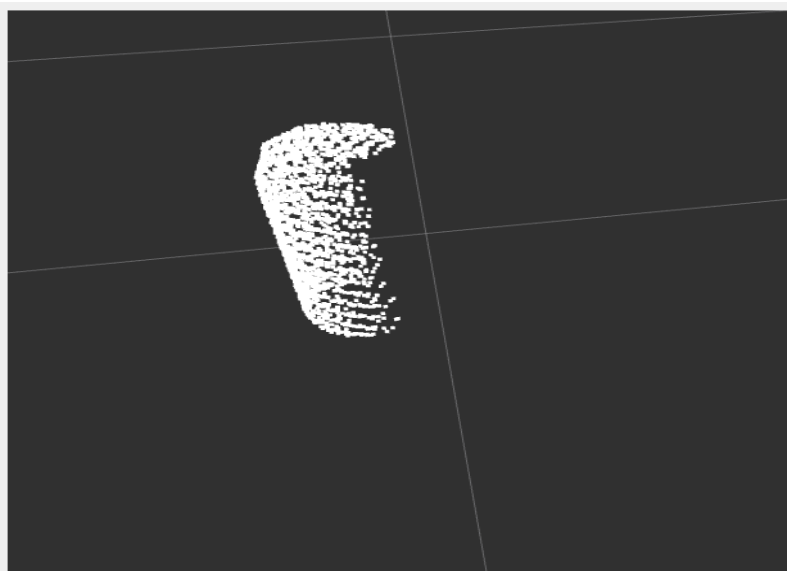
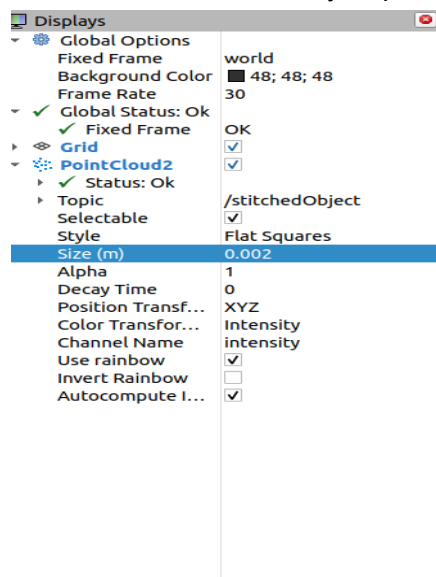
Check for /stitchedObject topic in Rviz and notice that the object got a little stitched



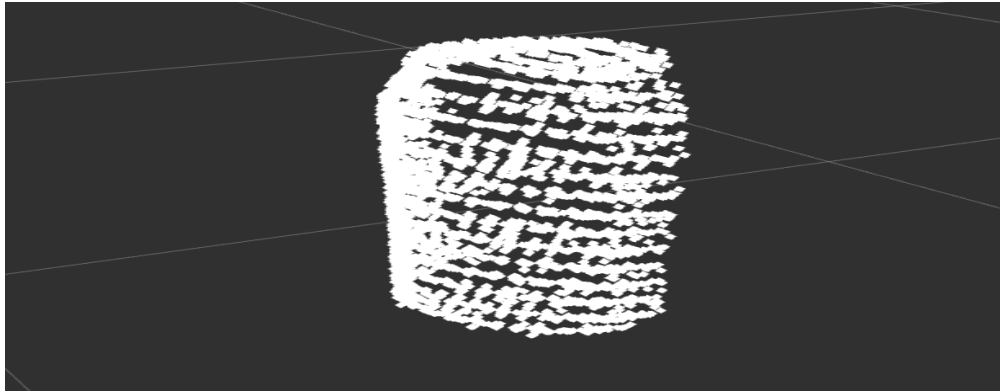
Keep moving or rotating the camera with little increments to stitch the point cloud



You can decrease density of point cloud in Size to see points more clearly



Like this we can stitch the entire object

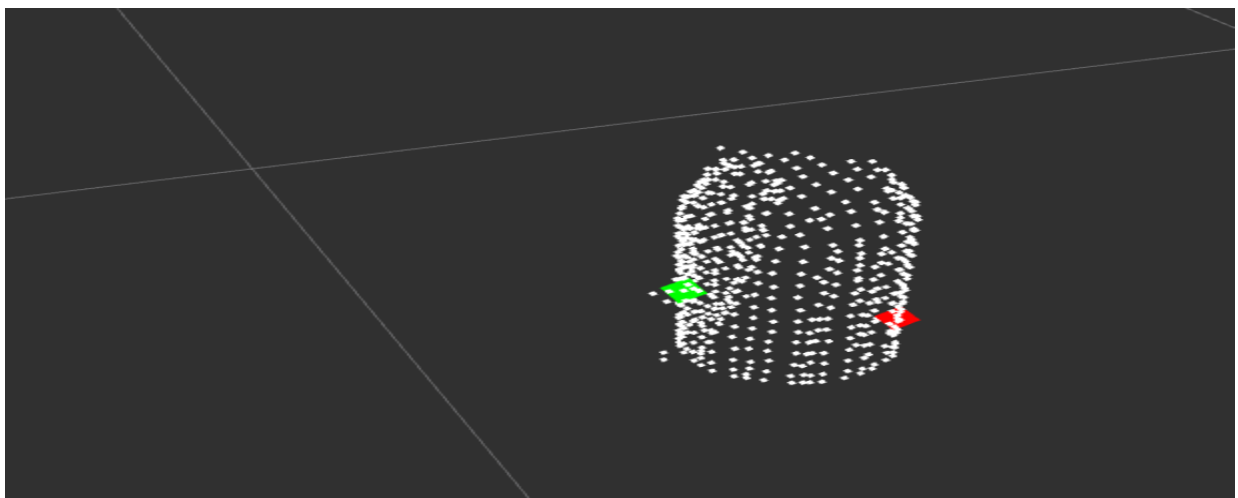


Step 6:

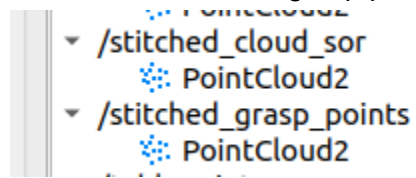
After stitching sufficiently run the following command in terminal to visualize grasp points of the stitched point cloud

```
shiva@shiva-Inspiron-15-5501: ~ 100x27
shiva@shiva-Inspiron-15-5501:~$ ros2 run seg gp_points
```

Below is the stitched point cloud with its grasp points



Please ensure only the below two topics are selected for visualization in Rviz to view the stitched cloud and its grasp points



Above was the example for coke can but it also applies to other objects