Exercise 1, 2 and 3 pipeline

1. Complete Exercise 1 steps. Pipeline for filtering and RANSAC plane fitting implemented.
2. Complete Exercise 2 steps: Pipeline including clustering for segmentation implemented.
3. Complete Exercise 3 Steps. Features extracted and SVM trained. Object recognition implemented.

For the three exercises I followed the instructions from the lessons. I only tweaked the parameters:
For voxel: LEAF_SIZE = 0.01
For passthrough: axis_min = 0.76, axis_max = 1.1
For ransac: max_distance = 0.01

The results were as expected.

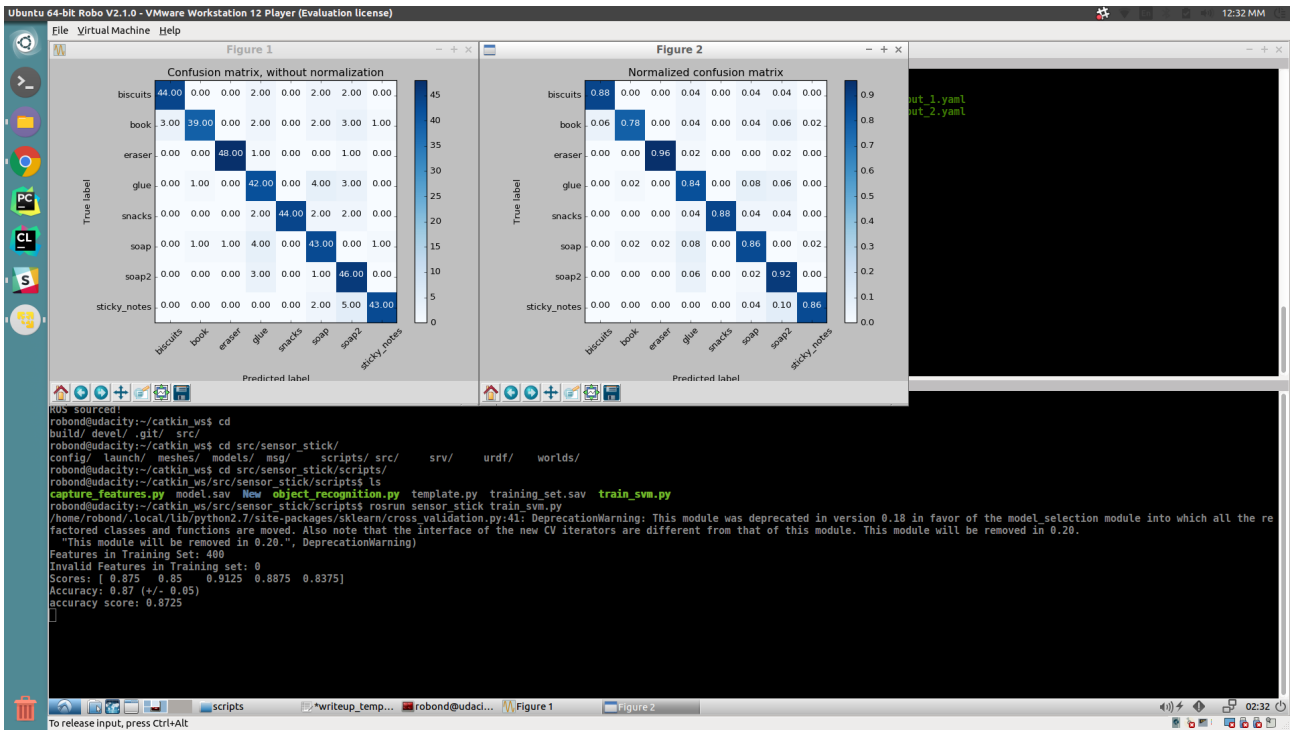
Because a large part of the exercises was used in the project I will comment it bellow.

Pick and Place Setup

First I have to create a supervised machine learning model for object classification. I used the model from the lessons with linear kernel. In order to generate a training set I modified the capture_features.py file and include to it the object models of the project:

```
models = [\
    'biscuits',
    'soap',
    'soap2',
    'book',
    'glue',
    'sticky_notes',
    'snacks',
    'eraser']
```

In the same file I tried different training set sizes: 15, 20, 25, 50 and 100. I found that 50 was good enough with 87.25% accuracy. For 100 loops the accuracy wasn't improving much and I have the feeling that it created an over-fitted model.



For 15 iterations, 32bin size I got 78.3% accuracy.
For 50 iterations with 64bin size I got 80%.
For 20 iterations with RGB colorspace included I got 81.25%.
For 100 iteration with RGB colorspace included and 64bin size I got 90%.

Features for model training

The features I decided to use where:

1. color histogram on the hsv colorspace :
chists = compute_color_histograms(sample_cloud, using_hsv=True)

2. normals histogram :
nhists = compute_normal_histograms(normals)

I also tried including both the hsv and rgb colorspaces in the same feature vector but the accuracy was only improving around 3% and I didn't think it was worth the extra computations and time.

Histogram bins

In features.py file I used 32bins for the histogram creation. I also tried 64 bins but even though the accuracy was better, the object recognition in the project was worse.
The range of color histogram was (0,256) and for normals histogram (-1,1).

Project pipeline

In the project_template.py file I implemented the pipeline used for object recognition for the project.

First in the __main__ part I initialize the ros node “object recognition”.

Then I create the subscriber that reads the raw point cloud from the rgbd camera of the robot.
Then I have five publishers:
1. For publishing the object point clouds.
2. For publishing the table point cloud after ransac.
3. For publishing the object clusters in different colors.
4. For publishingthe object markers with the name of the object.
5. For publishing the list of detected objects.

Then I load the machine learning model I trained before and I start running the node.

When the subscriber receives a point cloud it calls the pcl_callback function. In that function the raw point cloud is filtered with a statistical outlier filter. I used the values from the lessons for the filter mean and std.

Then the new cloud is downsampled with voxel grid filter in order to decrease resolution and for faster computations. With a LEAF_SIZE = 0.005 I got good results. First I tried 0.001 but it was too small and for 0.01 information was lost from the cloud.

Then I had to remove irrelevant information from the cloud since we know where the object are placed in front of the robot.
I used a Z axis passthrough filter to isolate the object and table clouds. However it wasn't enough and I got wrong object detection from the left and right trays. Therefore I also used a Y axis passthrough filter to remove these outliers too.

Then I was ready for RANSAC plane segmentation. The max_distance parameter was set to 0.01. By the way that was the easiest parameter to tune from the project.

The new cloud included all the objects so I had to separate them with Euclidean clustering.
The parameters I chose were:
Tolerance = 0.01
MinClusterSize = 50
MaxClusterSize = 5000

These challenge here was to chose the correct parameters. Because the object of the project were different from the exercises, I had to tune the parameters again.

From the cloud of each object that was created I extracted the same features I used for the model training (HSV histogram and Normals histogram), concatenated them into one feature vector and feed it to the trained classifier in order to identify the object. Finally I created a list of the detected objects and passed it to the pr2_mover function.

For creating the yaml files in pr2_mover function I followed the project instruction. What I mainly did was iterate through the pick object list and through the identified object list and once I found an object in both lists a wrote its information to the yaml file.

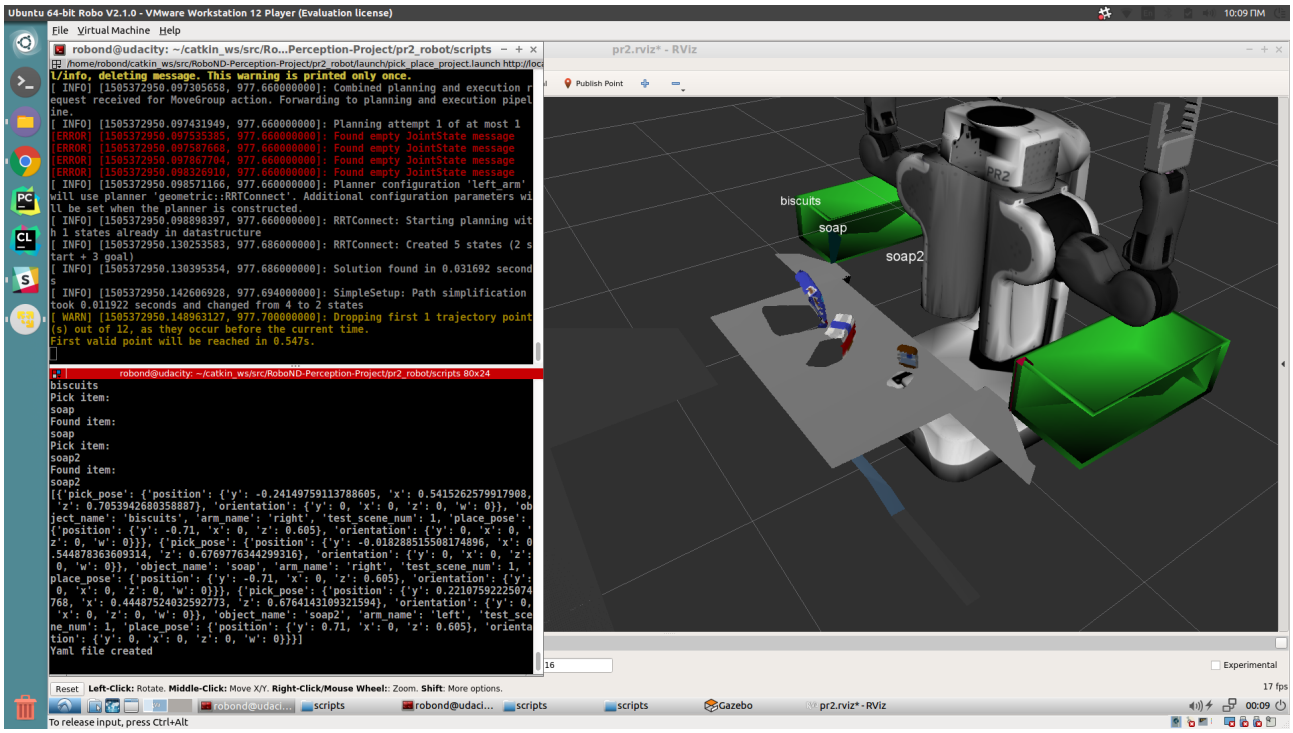
I created all three yaml files for the thee picking lists.

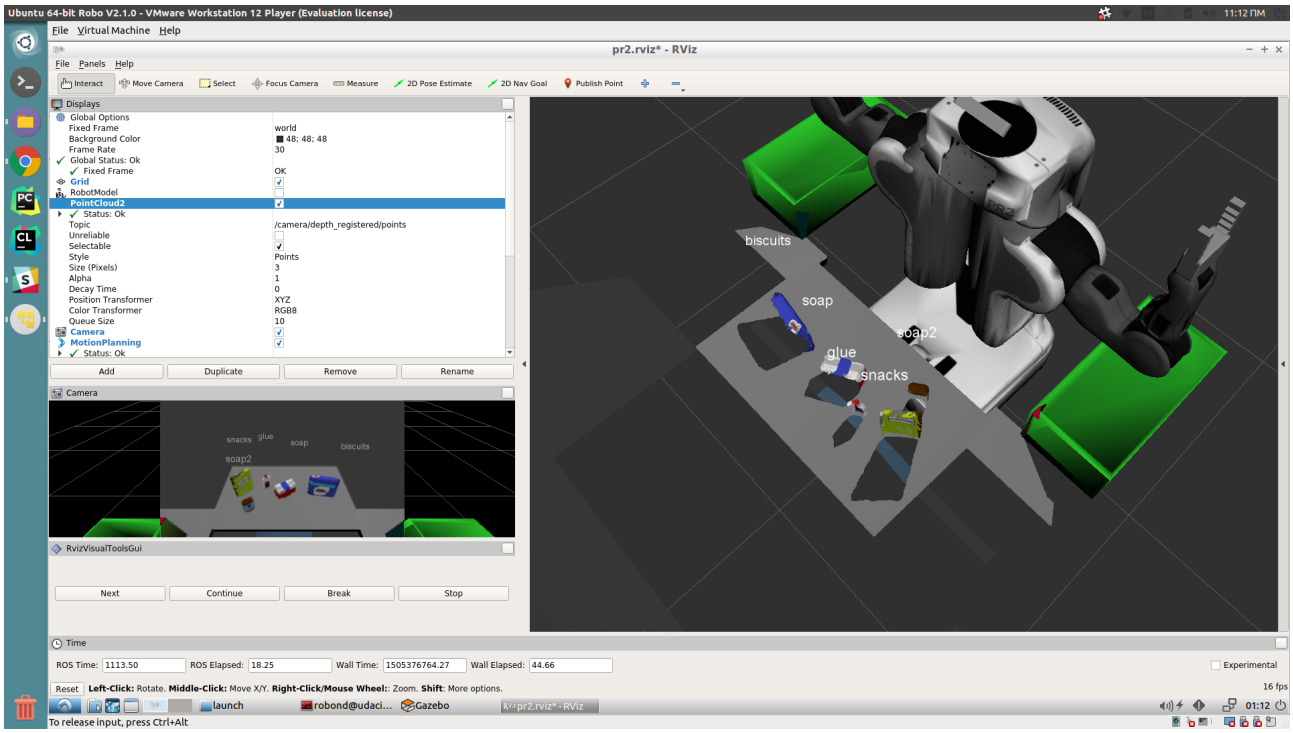
I got:

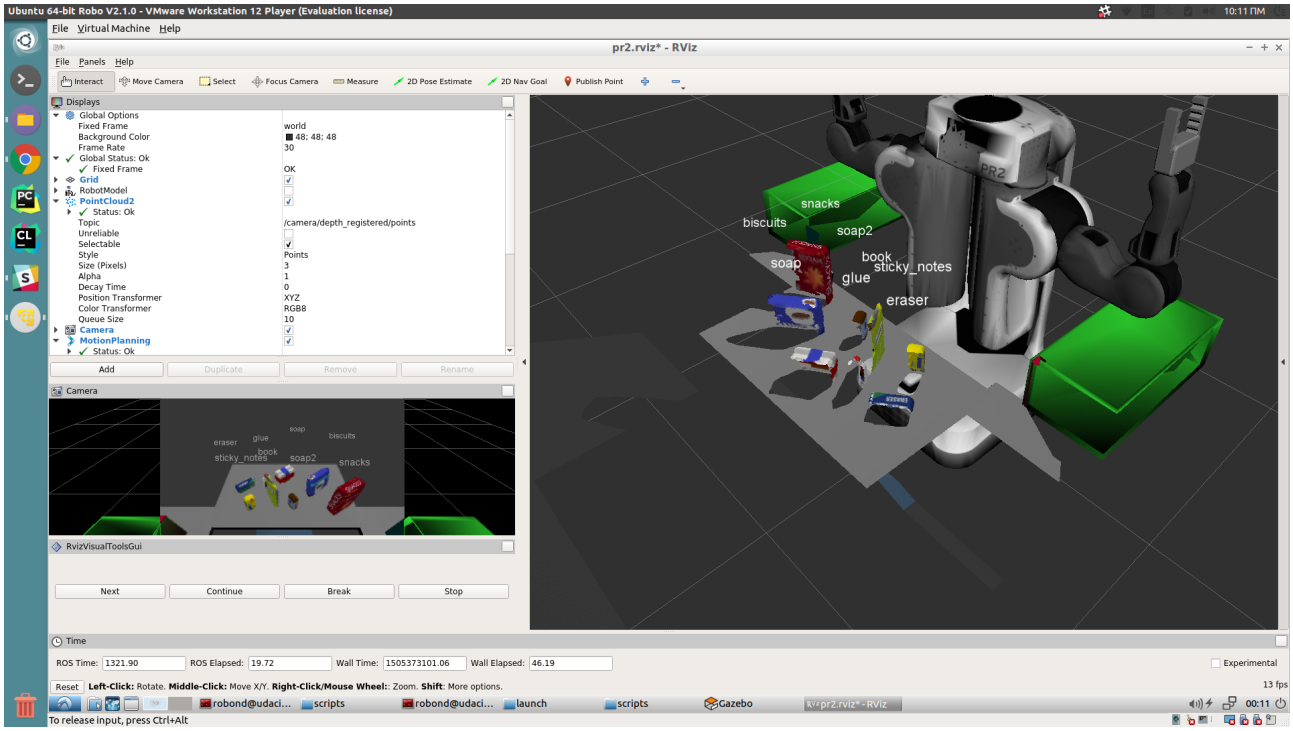
Picking list 1 accuracy 3/3
Picking list 2 accuracy 4/5
Picking list 3 accuracy 8/8

I found strange that the pipeline identified all objects in the third list and missclassified an object (book for snacks) from the second list where the objects were fewer.

Probably it was because one of them was behind and was covered partially by another object.






Ubuntu 64-bit Robo V2.1.0 - VMware Workstation 12 Player (Evaluation license)
File Virtual Machine Help
pr2.rviz* - RViz
File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame world
Background Color 48; 48; 48
Frame Rate 30
Global Status: Ok
Fixed Frame OK
Grid
RobotModel
PointCloud2
Status: Ok
Topic /camera/depth_registered/points
Unreliable
Selectable
Style Points
Size (Pixels) 3
Alpha 1
Decay Time 0
Position Transformer XYZ
Color Transformer RGB8
Queue Size 10
Camera
MotionPlanning
Status: Ok
Add
Duplicate
Remove
Rename
Camera
RvizVisualToolsGui
Next
Continue
Break
Stop
snacks
biscuits
soap2
soap
book
sticky_notes
glue
eraser
Time
ROS Time: 1321.90
ROS Elapsed: 19.72
Wall Time: 1505373101.06
Wall Elapsed: 46.19
Experimental
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
13 fps
robond@udaci...
scripts
robond@udaci...
launch
scripts
Gazebo
pr2.rviz* - RViz
To release input, press Ctrl+Alt