Perception Pick and Place Simulation

Date: September 12th, 2018

Menghong Feng

1 Steps

1.1 Step 1: Voxel Grid Downsampling

RGB-D cameras can generate feature rich and dense point clouds. In many cases, there are more points packed in per unit volume than a Lidar point cloud. However, processing full resolution point cloud can be slow and may not be necessary, since less dense point cloud may generate very similar results compared to full resoluation point cloud does. Therefore, it is common to downsample the data and increases processing efficiency. In this case, we will use a VoxelGrid Downsampling Filter to derive a full resolution point cloud into sparsely sampled point cloud but still do a good job to represent the objects.

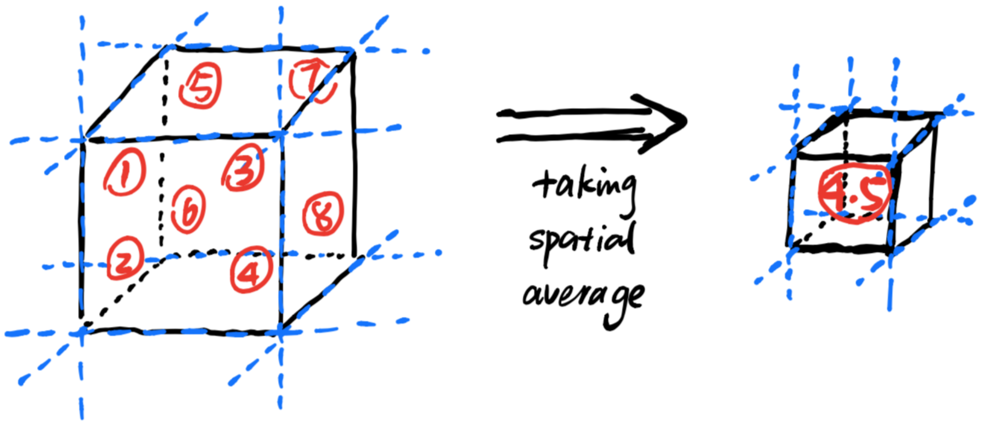


Figure 1: Voxel Grid Downsampling

Voxel stands for ”volume element”. VoxelGrid is a method to segment entire volume into small volume element. The principle of VoxelGrid Downsampling Filter is to convert large volume point cloud into small volume by taking spatial average of the points in the cloud confined by each voxel, as shown in figure xx. In this case, the voxel size is recommended to set as 0.01.

1.2 Step 2: Pass Through Filtering

The Pass Through Filter is used to crop the 3D region. The region allowed to pass through the filter is called region of interest. Filter axis and axis limits are defined to crop the region.

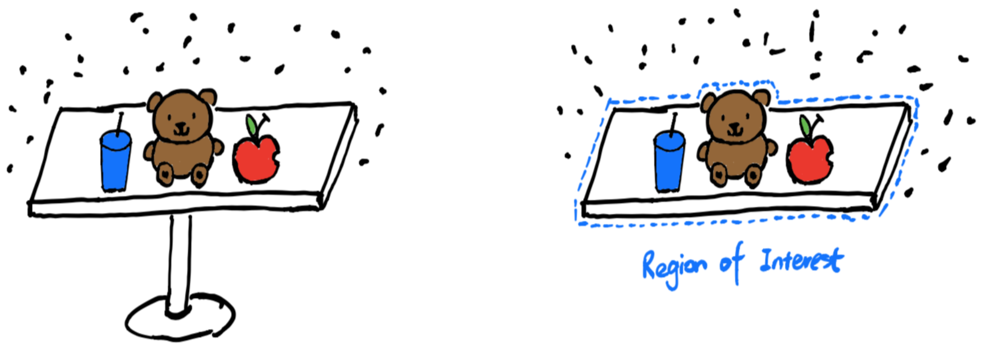


Figure 2: Pass Through Filtering

1.3 Step 3: Outlier Removal Filter

Outlier Removal Filter is used to remove noise (outlier). Noise is often caused by external factors such as dust, humility in air, presence of light sources, and so go. The principle of such filter is to perform statistical analysis in the neighborhood of each point, and remove those points which do not meet a certain criteria.

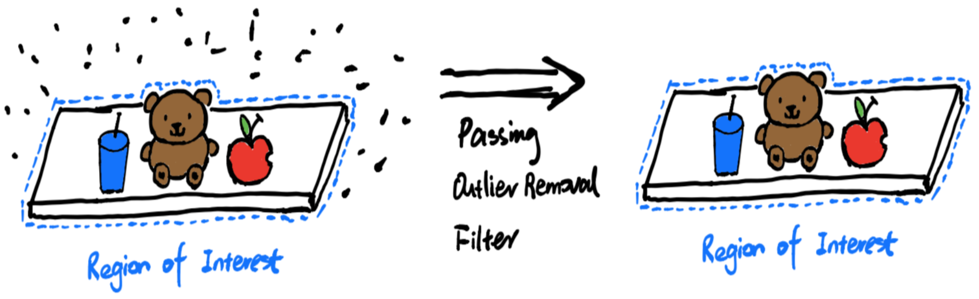


Figure 3: Outlier Removal Filter

1.4 Step 4: RANSAC Plane Segmentation

RANSAC technique is known as Random Sample Consensus. It is an algorithm to identify points in dataset that belong to a particular model. RANSAC algorithm assumes that all of the data in a dataset is composed of both inliers and outliers, where inliers can be defined by a particular model with a specific set of parameters, while outliers do not fit that model and hence can be discarded. In this case, we can model the table as a plane, you can remove it from the point cloud to obtain like shown in figure xx. The disadvantage of RANSAC algorithm is you have to trade-off between compute time versus model detection accuracy. The less iteration you pick the less accurate your model will be segmented, but there is no upper limit on the time it can take to compute the model parameters.

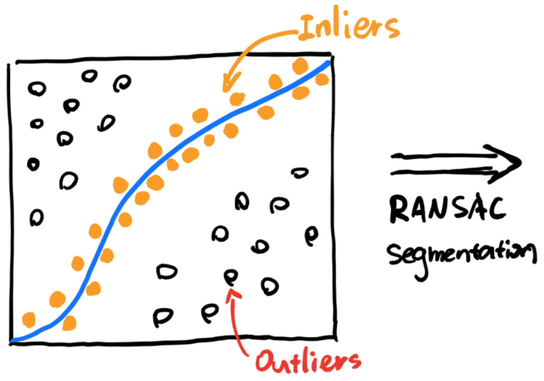


Figure 4: RANSAC Plane Segmentation

1.5 Step 5: Extracting Indices

With RANSAC we identified which indices in point cloud correspond to the table, and the indices not corresponding to table represent objects. To apply ExtractIndices Filter, we can extract points from a point cloud by providing lits of indices.

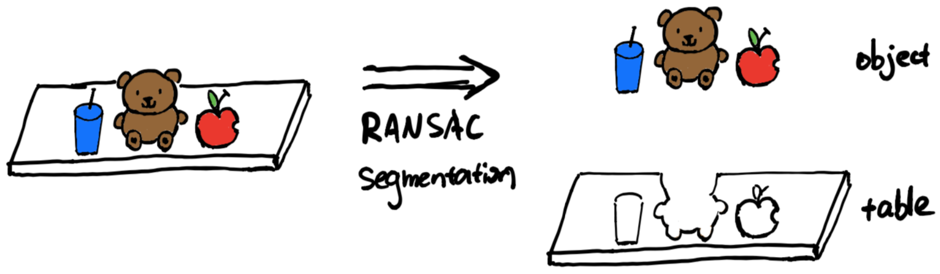


Figure 5: Extract Objects

1.6 Step 6: Euclidean Clustering

In order to train your system to recogninize the object, the first step is to segment out each objects from a point cloud. Clustering technique is often used to perform object segmentation. In this case, we use DBSCAN algorithm.

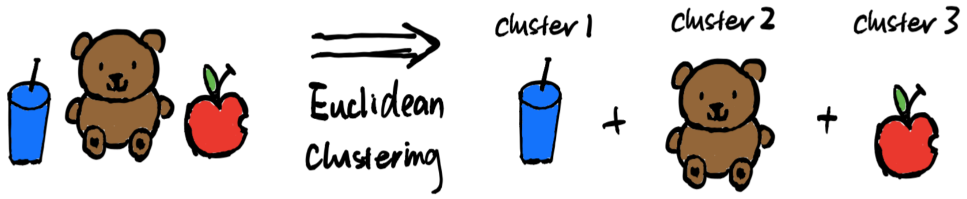


Figure 6: Clustering Individual Object

DBSCAB stands for Density-Based Spatial Clustering of Applications with Noise. This algorithm is a nice alternative to k-mean clustering method when we don't know how many exactly clusters to expect in dataset. The objects will be clustered in terms of density. The DBSCAN algorithm creates clusters by grouping data points that are within some threshold distance d_t from the nearest other point in the data. It is also called "Euclidean Clustering", because the decision of whether to place a point in a particular cluster is based upon the "Euclidean distance" between that point and other cluster members.

1.7 Step 7: Extracted Feature Vectors

Objects can be recognized by identifying their unique features, such as color, shape, size, and so on. In this case, we will use object's color and shape to differentiate each other.

One of the common way to compare object's color and shape features is to analyze their color and normal histograms. There are two main color system used in computer vision problem: RGB(red, green, and blue) and HSV(hue, saturation, and value). The image is often read as RGB format, but we can converted it as HSV feature vector since HSV system is more accurate to differentiate unique color.

To analyze objects' shape, it is often refer to its surface normal. Surface normal is defined a group of vectors that is perpendicular to a given object's surface. Normal histogram is used to account normal vector's orientation on each surface. The distribution of surface's normal orientation can be used to differentiate each object's uniqueness in terms of its shape.

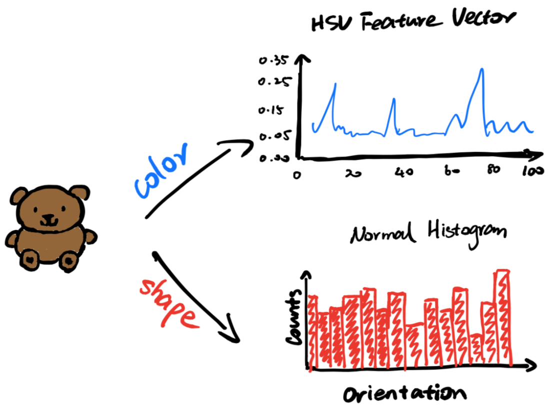


Figure 7: Extract Feature Vectors from Color and Normal Surface Histogram

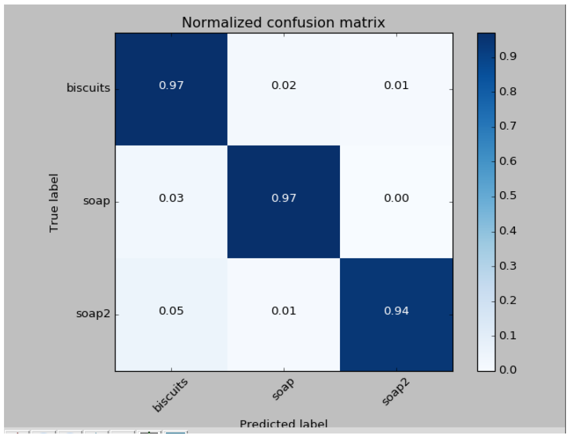
1.8 Step 8: Object Recognition

Object recognition task is often tackled by applying machine learning technique. Labeled training samples are fed into network. It will learn from samples and predict unlabeled test subject. In this case, we change the spatial orientation of objects in 3-dimensional environment and feed those images into network. The network will learn from object's color and normal histograms.

The machine learning technique we used in here is Support Vector Machine. Support Vector Machine, or SVM, is a very popular supervised machine learning algorithm that allows us to characterize the parameter space of your dataset into discrete classes. SVM works by applying an iterative method to a training dataset, where each item in the training set is characterized by a feature vector and a label. Each point in dataset is characterized by just two features, A and B. The color of each point corresponds to its label, or which class of object it represents in the dataset.

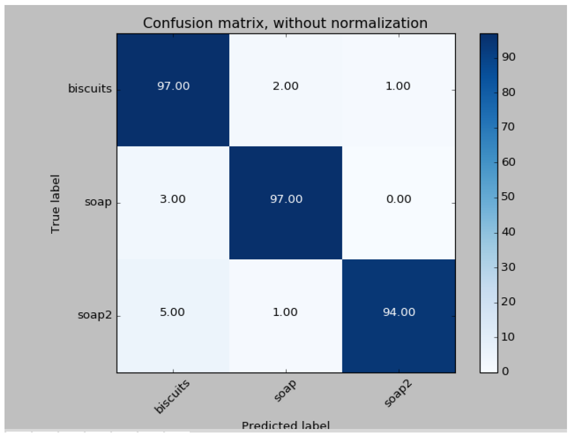
2 Result

The accuracy rate for testing dataset in test1, test2, and test3 are 96%, 95.8%, and 96% respectively. In Gazebo simulation enviornment, the pipeline correctly identify all of the objects (100% accuracy rate) in first two tests. However, only 7 out of 8 objects are recognized in test 3. The reason for this failure is because the object book blocks the object glue so that the camera cannot capture object glue's point cloud data. However, since the glue recognization accuracy in machine learning testing result is 98%, therefore it is reasonable to conclude that object glue should be correctly recognized if book doesn't block the glue. The screenshots of results are shown below.



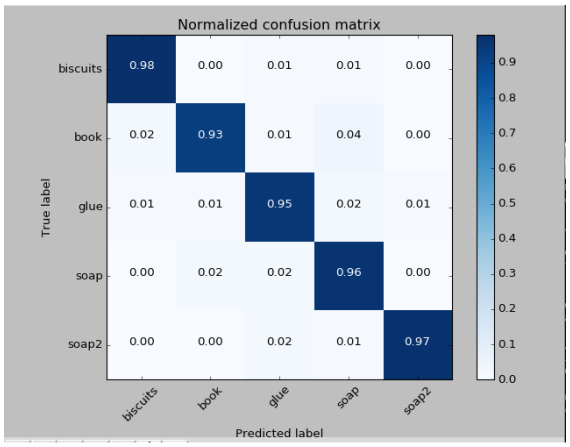
	biscuits	soap	soap2
biscuits	0.97	0.02	0.01
soap	0.03	0.97	0.00
soap2	0.05	0.01	0.94

Figure 8: Normalized Confusion Matrix in Test 1



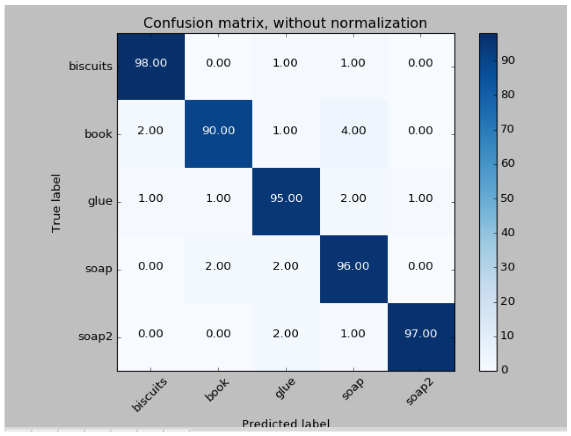
	biscuits	soap	soap2
biscuits	97.00	2.00	1.00
soap	3.00	97.00	0.00
soap2	5.00	1.00	94.00

Figure 9: Non-normalized Confusion Matrix in Test 1



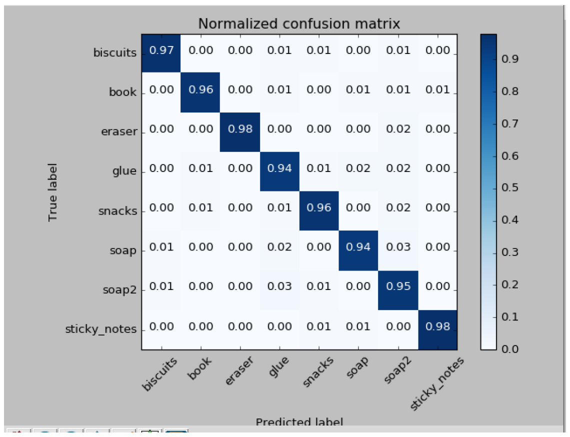
	biscuits	book	glue	soap	soap2
biscuits	0.98	0.00	0.01	0.01	0.00
book	0.02	0.93	0.01	0.04	0.00
glue	0.01	0.01	0.95	0.02	0.01
soap	0.00	0.02	0.02	0.96	0.00
soap2	0.00	0.00	0.02	0.01	0.97

Figure 10: Normalized Confusion Matrix in Test 2



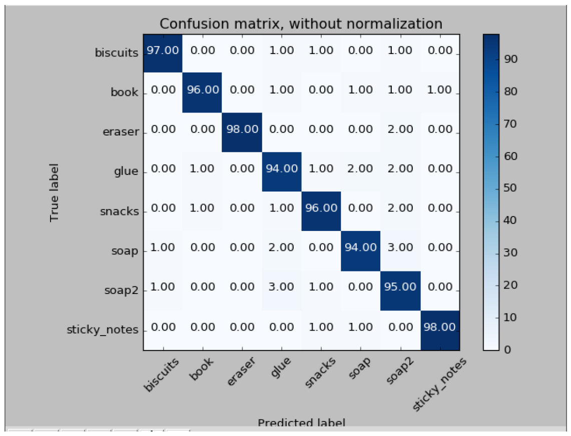
	biscuits	book	glue	soap	soap2
biscuits	98.00	0.00	1.00	1.00	0.00
book	2.00	90.00	1.00	4.00	0.00
glue	1.00	1.00	95.00	2.00	1.00
soap	0.00	2.00	2.00	96.00	0.00
soap2	0.00	0.00	2.00	1.00	97.00

Figure 11: Non-normalized Confusion Matrix in Test 2



	biscuits	book	eraser	glue	snacks	soap	soap2	sticky_notes
biscuits	0.97	0.00	0.00	0.01	0.01	0.00	0.01	0.00
book	0.00	0.96	0.00	0.01	0.00	0.01	0.01	0.01
eraser	0.00	0.00	0.98	0.00	0.00	0.00	0.02	0.00
glue	0.00	0.01	0.00	0.94	0.01	0.02	0.02	0.00
snacks	0.00	0.01	0.00	0.01	0.96	0.00	0.02	0.00
soap	0.01	0.00	0.00	0.02	0.00	0.94	0.03	0.00
soap2	0.01	0.00	0.00	0.03	0.01	0.00	0.95	0.00
sticky_notes	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.98

Figure 12: Normalized Confusion Matrix in Test 3



	biscuits	book	eraser	glue	snacks	soap	soap2	sticky_notes
biscuits	97.00	0.00	0.00	1.00	1.00	0.00	1.00	0.00
book	0.00	96.00	0.00	1.00	0.00	1.00	1.00	1.00
eraser	0.00	0.00	98.00	0.00	0.00	0.00	2.00	0.00
glue	0.00	1.00	0.00	94.00	1.00	2.00	2.00	0.00
snacks	0.00	1.00	0.00	1.00	96.00	0.00	2.00	0.00
soap	1.00	0.00	0.00	2.00	0.00	94.00	3.00	0.00
soap2	1.00	0.00	0.00	3.00	1.00	0.00	95.00	0.00
sticky_notes	0.00	0.00	0.00	0.00	1.00	1.00	0.00	98.00

Figure 13: Non-normalized Confusion Matrix in Test 3

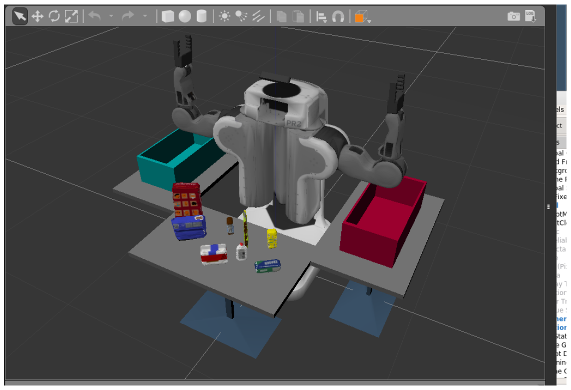


Figure 14: Pick and Place Environment

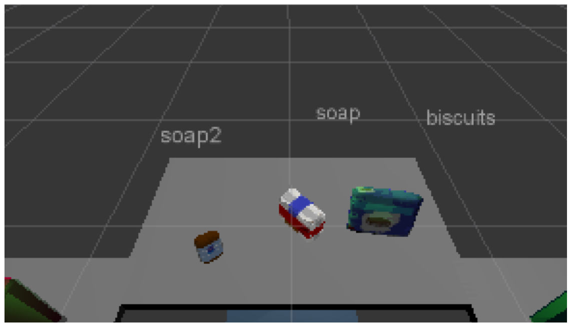


Figure 15: Object Recognization in Test 1

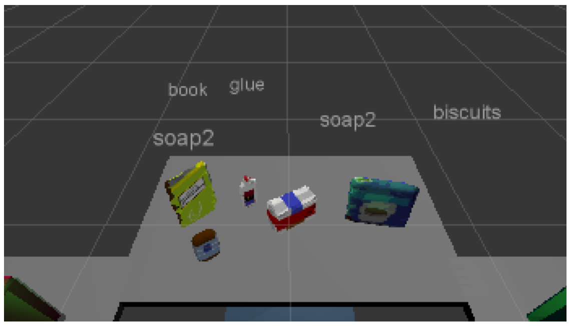


Figure 16: Object Recognization in Test 2

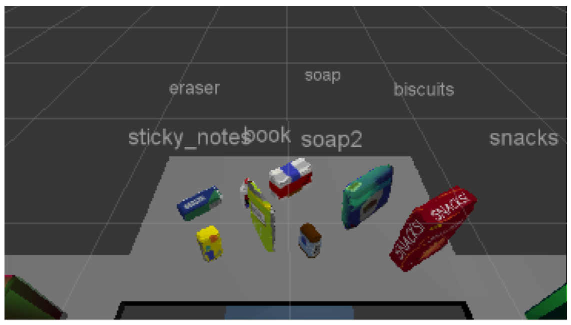


Figure 17: Object Recognization in Test 3