ESE650 Final Project: Planar Surface Detection

Michael O'Meara

***Abstract*— In this project, I implemented the Mean Shift clustering algorithm aimed at detecting multiple planes in a RGBD image. The clustering was done in combination with the normal and euclidean feature spaces. Each normal vector was an estimate of a surface normal defined by the surrounding nearest neighbors, being a subset of the point cloud for all points and the euclidean feature space was defined as an estimate of the euclidean distance from the cluster center, together forming a six dimensional feature space. This feature space was the input to the mean shift algorithm and used to estimate the different planar surfaces within the given image.**

I. INTRODUCTION

The problem of multiple planar surface detection is a well known low-level problem in computer vision and requires a robust approach to ensure reliable and timely results in order to be useful in practical applications. The mean shift procedure is one possible solution that tries to solve the problem without needing to tune a large numbers of parameters while still returning accurate outputs. The algorithm itself accomplishes this task quite well, since, what it does is simply find the modes of the feature space, the denser regions corresponding to local maxima of the probability density function of that feature space. It should be noted that while the mean shift algorithm in it's general form does not require a lot of tuning, construction of the feature space and the actual implementation of the algorithm requires a lot of careful choices which I will describe later in this paper. With that said, mean shift is an iterative algorithm, clustering like surfaces in an EM fashion.

II. PROCEDURE

A. Constructing the feature space

The feature space can be constructed in many different ways and it's implementation can have varied results when mapping inputs to clusters. The first step in my implementation was to compute the surface normals of each point. However, one issue I encountered immediately was the depth data from the RGBD image was quite noisy so, merely taking the point cloud's nearest neighbor subset around each point yielded equally noisy results. In order to remediate this problem, I used a median filter with a 15×15 patch on just the depth information. This had the effect of smoothing the surface around each point so when I ran k-nearest neighbor search again on each point and then found its surface normal the normals were closer to my expectation for that region. The downside of this decision was that edges appeared less sharp since they too were also being smoothed out. I tried different sized patches for smoothing and found 15×15 worked best. Another important factor that required a lot of trial and error was choosing the number of surrounding points to calculate the surface normal. After experimentation with a wide range of values from 3 to 1000 nearest neighbors, I found 64 nearest neighbors worked relatively well. Some sort of visual cross validation of median filter patch size and size of nearest neighbors points might prove helpful to determine the best combination of parameters. Based on a paper from the Symposium of Geometry Processing 2012, "Fast and Robust Normal Estimation for Point Clouds with Sharp Features" by Alexandre Boulch and Renaud Marlet, the inputs and construction of the feature space can be vastly improved, but is beyond of the scope of this paper due to time constraints.

B. Calculating the normal vector

```
x = pts(:,1);
y = pts(:,2);
z = pts(:,3);
N = length(pts);
mu = [mean(x), mean(y), mean(z)];
dst = [x-mu(1), y-mu(2), z-mu(3)];
A = (1/N)*(dst'*dst);
[U,S,V] = svd(A);
normal = U(:,end);
```

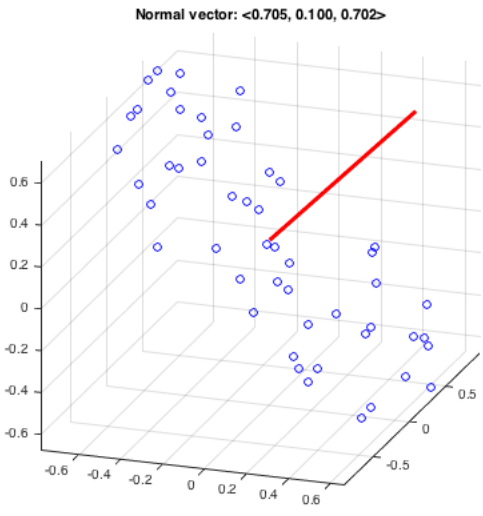

Fig. 1. Example normal to a set of points

Michael O'Meara is a student at the University of Pennsylvania, in the MSE Robotics Program, momeara@seas.upenn.edu

C. Adjusting Normals

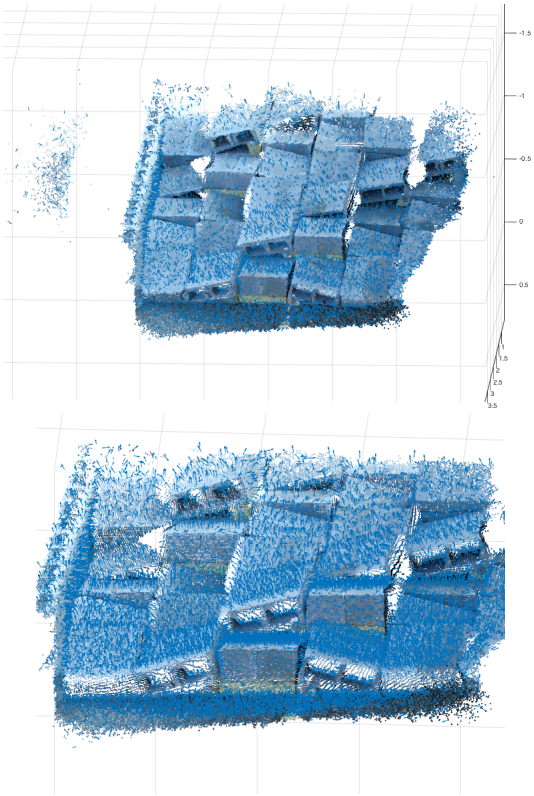

Fig. 2. Adjusted normals before and after

Since, normals pointing in opposite directions can represent the same surface and because not all the normals from the same surface point in the same direction relative to the camera, I adjusted their direction in order to get better results, as shown above. Notice, the adjusted normals pictured have greater quivers densities where more of them point up relative to the camera position.

D. General Mean Shift Algorithm

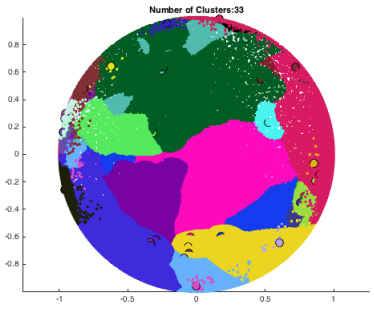

Fig. 3. Clusters in normal space

The implementation I followed was based on the paper [1], where an iterative procedure was followed starting with the selection of a random point from the point cloud and

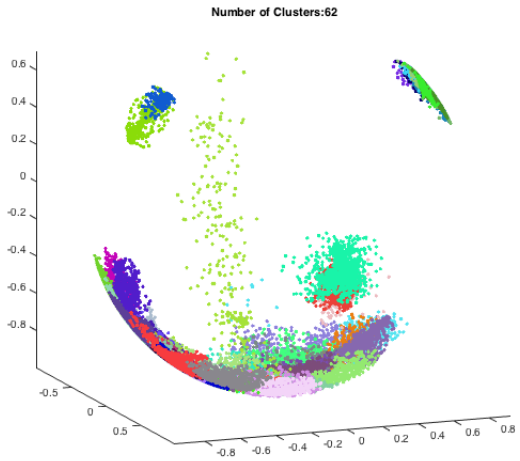

Fig. 4. Clusters in combine feature space

with the surrounding closest neighbors a mean is calculated. This mean is compared to the prior mean where the distance between them is minimized until convergence and each point gets associated with a cluster based on the clusters with the most votes. A new cluster is added unless the distance to one of the prior clusters is below a chosen threshold, in which case, the clusters are merged into a single cluster and the procedure continues until all the points in the point cloud are assigned to one of the existing clusters. For each point, the mean shifts until its location matches the mode of that region and all the surrounding closest points belong to that cluster center based on the feature space.

III. MATH

A. Equations

Given n data points points x_i, i 1,...,n in the d-dimensional space R^d, multivariate kernel density estimator with different kernels K depending on the feature space.

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K_{normal}(x - \bar{x}) \tag{1}$$

and

$$\hat{g}(x) = \frac{1}{n}\sum_{i=1}^{n} K_{euclidean}(x - \bar{x}) \tag{2}$$

$$C = \hat{f}(x) + \alpha\hat{g}(x) \tag{3}$$

Such that C is the cluster mean in the combine feature space. α is a scaling factor, weighting the influence of the euclidean distance, $\hat{g}(x)$, with that of the normal mean, $\hat{f}(x)$.

B. Improvements

In addition to clustering in the combined normal and euclidean feature space, I rotated each collection of nearest neighbor points based on the estimated normal to the xy-plane and then removed points in the z direction above and

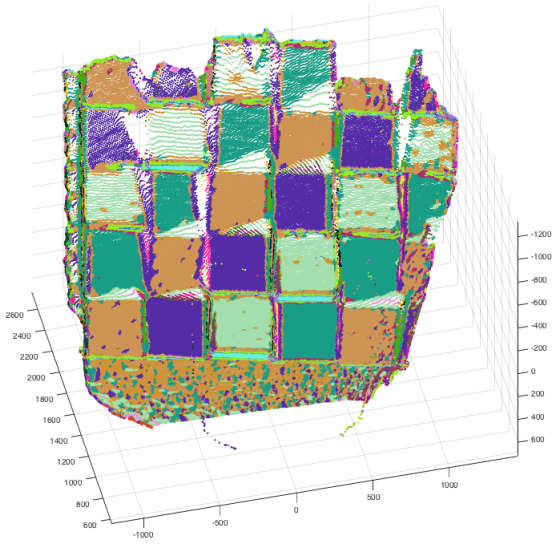

Fig. 5. Mean shift in normal space

below a threshold. The reason for this procedure was to remove noise from the calculation of the mean. That way, the mean is based on the expected plane instead of outliers from the desired plane. Then it should be possible to relax the euclidean distance constraint and still capture planes pointing in the same direction but potentially separated by co-planar distance or holes in the plane while differentiating between surfaces that are not co-planar. The implementation still needs improving since the normals contain a lot of noise and the collections of nearest neighbor points have means that are edge points and could have multiple valid planar surfaces in the same collection of points.

IV. Results

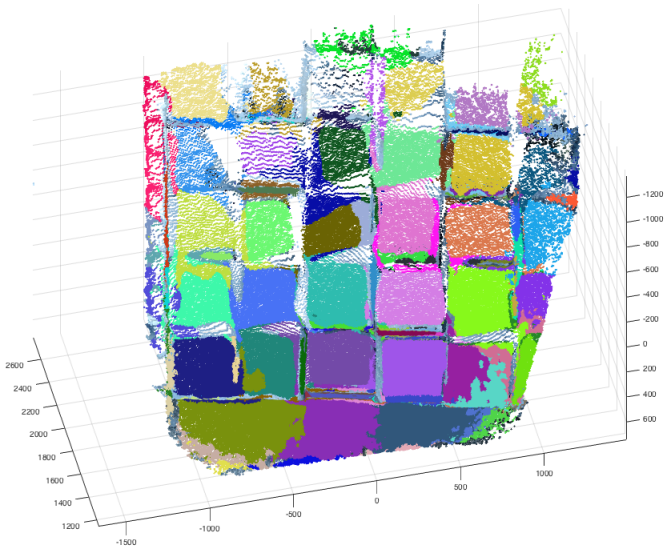

Fig. 6. Before improvement

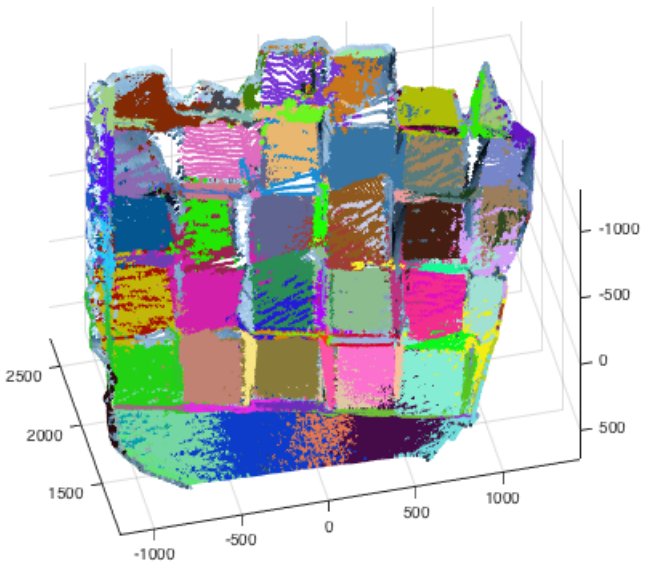

Fig. 7. After improvement, 160 inn

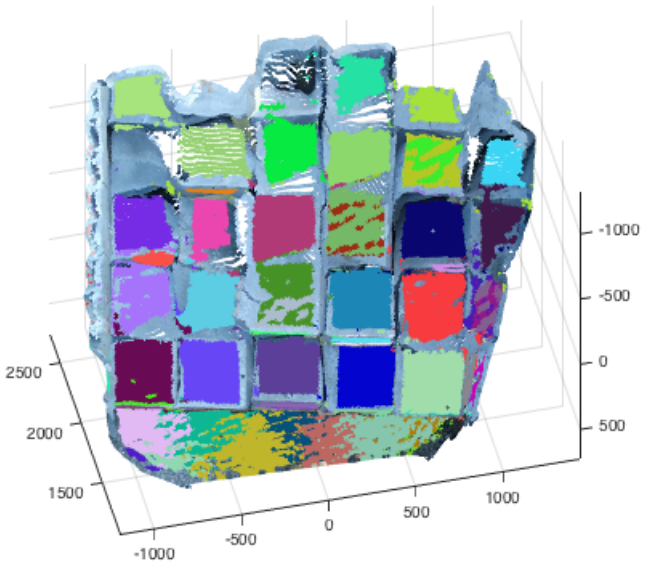

Fig. 8. After improvement, 64 knn

V. CONCLUSION

The use of the mean shift algorithm has clear advantages for image segmentation and with different kernels and different input feature spaces many possibilities exist to extract different types of features whether they are surfaces, edges or objects. Mean shift has also been shown to work in applications for tracking moving objects in a scene. According to the literature, implementations in C++ have shown to have realtime or near realtime results. And lastly, since mean shift doesn't require a lot of parameter tuning and isn't limited to a fixed number of clusters it has many advantages over other clustering algorithms like k-means.

APPENDIX

ACKNOWLEDGMENT

I would like to acknowledge the help of Dr. Daniel Lee, Bhoram Lee and Jinwook Huh for their suggestions and guidance on this project and throughout the semester. Thank you.

REFERENCES

[1] Comaniciu, D. and Meer, P., Mean Shift: A Robust Approach Toward Feature Space Analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2002

[2] Alexandre Boulch and Renaud Marlet, "Fast and Robust Normal Estimation for Point Clouds with Sharp Features", Symposium of Geometry Processing, 2012