CRF Based Point Cloud Segmentation

Jonathan Nation
jsnation@stanford.edu

1. INTRODUCTION

The goal of the project is to use the recently proposed fully connected conditional random field (CRF) model to segment a point cloud scene [1]. The dense CRF model has previously been used with positive results to enhance the accuracy of labeling of images.

Since devices to capture point clouds easily are relatively recent (Kinect), there has not been much research into segmenting out objects from a point cloud. Previous work in the segmentation of 3d point cloud scenes has usually involved the extracting geometric primitives using features like normals and curvatures [2, 3]. Other research has focused on segmenting out a single object foreground from the background in a semi-supervised or unsupervised manner [4]. None of the prior work found covers the segmentation of a class of objects from a single instance of that object, which is the main focus of this current work.

Segmenting objects out of a point cloud scene is a difficult task with many steps. First is to collect a dataset of several objects to segment out and a background. Then the dense CRF model is initially used with position based point features (position / normals / color). Next, position independent features are used, like normals calculated at different scales, and more invariant features like point feature histograms. The unary potentials of the dense CRF are initially determined by one or more user specified points in the scene, providing a semi-supervised segmentation of the scene. The goal is to move towards fully unsupervised segmentation of the scene.

2. DATASET

The data is 3D point cloud scenes collected using the Microsoft Kinect. The first scene consists of 3 mugs of various shapes, colors, and sizes, sitting on top of a circular wooden table. The mugs serve as the foreground object and the table the background.

2.1 Collecting the Dataset

Over 50 point clouds snapshots were collected from all angles around the scene, and then reconstructed using different reconstruction techniques from point cloud library. First, individual point clouds had to be preprocessed to clean them up and reduce noise. For each point cloud captured, a depth filter was used to filter all points outside the range of the scene (table + mugs). Then, a radial outlier removal filter was applied to each point cloud, removing all points with less than a threshold of neighbors within a given search radius.

Several techniques were attempted to get a clean reconstruction, including using feature based reconstruction with 3d SIFT features and RANSAC for an initial pose estimate and ICP for refinement. Due to the large amount of variation in features between the images, due mostly to self occlusion in the scene from the mugs, the features based reconstruction failed to produce a high quality scene. Instead, a control point selection tool was devised to select 4-6 corresponding points between each image and estimate the pose to fit those selected points, and then use ICP to refine the pose. This resulted in a high quality reconstructed scene. The process was made more efficient without much loss of quality by using feature based pose estimation on pairs or small groupings of images, and then using manual control point selection to align those groups together. The resulting point cloud scenes are shown in Figure 1. Both scenes had a voxel grid sampling applied to them with a grid size of 2.5mm to reduce the abundance of overlapping points after registration. Scene 1 contained approximately 200,000 points and scene 2 had approximately 450,000 points.

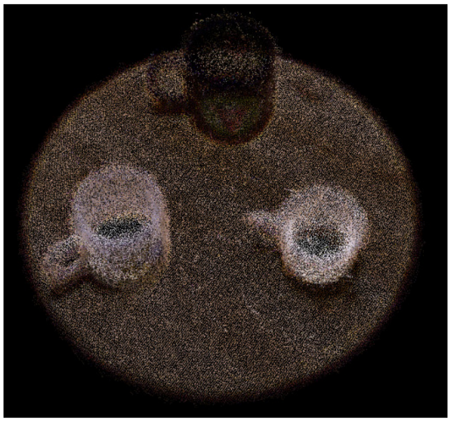


Figure 1. Point Cloud Scene 1 with 3 mugs on a wooden table

2.2 Labeling the Dataset

The dataset for scene 1 was labeled with 3 categories in order to use it to evaluate different segmentations. The categories were mug / table / and unknown. A tool for labeling a point cloud scene of this type efficiently was created, allowing the user to label all points on one side of a plane, or to label in different sized radial patches of points. Any ambiguous or spurious points in the scene, especially around the table/mug interface, were left as unknown points. The labeled scene is shown below in Figure 2, with green representing table, red representing mug, and natural pixel color representing unknown points.

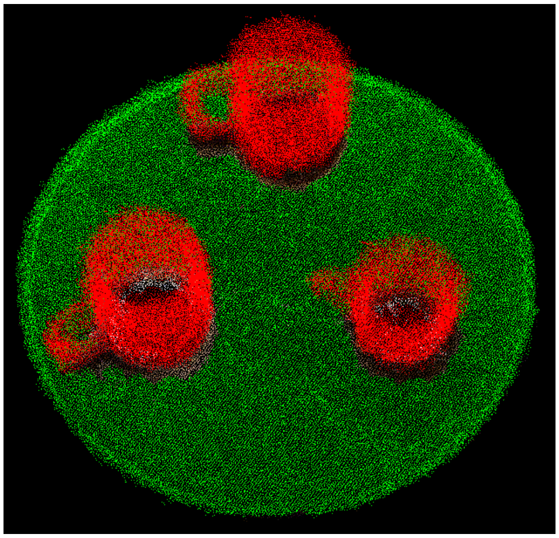


Figure 2. Point Cloud Scene 1 ground truth labeled as mug (red), table (green), or unknown (scene color).

3. CRF Segmentation

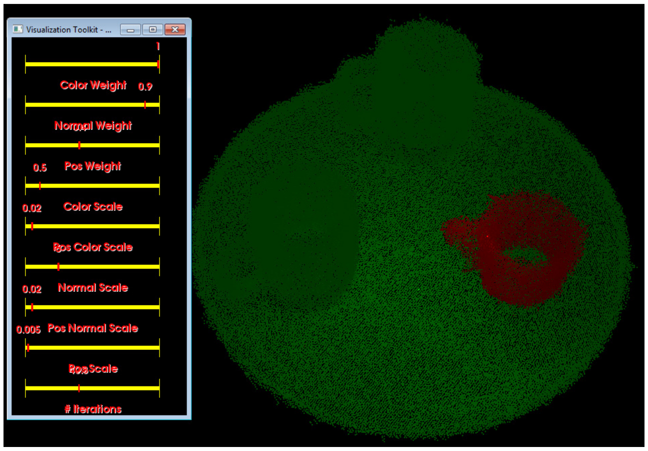
3.1 Segmentation with Position Features

The CRF model of segmentation was applied to the scene to segment out mug from table. The CRF model involves both unary and pairwise potentials defined for the scene. Unary potentials should ideally be computed using point features and define a prior probability over the label assignments for each point. At this point, a fixed unary potential was used for pre-selected mug or table points, using a fixed value of 50 for the non-assignment penalty. For all unspecified points, the unary potential for both assignments was set to 0.

Initially, pairwise potentials with combinations of Gaussian kernels of normals, positions, and colors were used. Pairwise potentials specified with Gaussian kernels of the form in equation 1 were used, with p representing the 3d position vector, n representing 3d normal vector, and I representing the 3 component point color.

$$k\left(f_i, f_j\right) = w^{(1)} \exp\left(-\frac{\left|p_i - p_j\right|^2}{2\theta_p^2}\right) + w^{(2)} \exp\left(-\frac{\left|p_i - p_j\right|^2}{2\theta_{pI}^2} - \frac{\left|I_i - I_j\right|^2}{2\theta_I^2}\right) + w^{(3)} \exp\left(-\frac{\left|p_i - p_j\right|^2}{2\theta_{pn}^2} - \frac{\left|n_i - n_j\right|^2}{2\theta_n^2}\right) \qquad [1]$$

There are many free variables to be optimized in these pairwise potentials, including the position, normal, and color Gaussian weights, the position scaling for each feature, and the normal and color weightings. Also, the number of iterations can be adjusted, but convergence was generally achieved within 10 iterations. In order to gain intuition on the effects of the different parameters of the pairwise potentials in Potts model used, an interface with sliders for each of the variables in the potentials was constructed and used to vary each value and see the effect immediately. Figure 3 shows the sliders used to adjust the values, and the results of a segmentation using a single preselected point on one mug, and a single point in the middle of the table.



1

Color Weight 0.9

Normal Weight

0.5 Pos Weight

0.02 Color Scale

Pos Color Scale

0.02 Normal Scale

0.005 Pos Normal Scale

Pos Scale

Iterations

Figure 3. CRF Segmented scene 1 using a single preselected point on the mug and table, with parameters adjusted by slider

In order cleanly segment all three mugs from the scene using this approach and set of features, at least one point on each mug needs to be specified. The results of specifying a point on each mug and 3 points on the table are shown below in Figure 4.

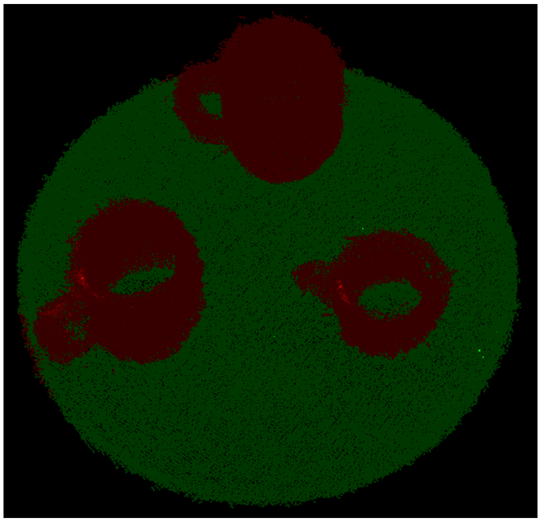


Figure 4. CRF Segmented scene 1 using a single preselected point on each mug and 3 points on the table. Brighter colors represent higher probabilities in

3.2 Segmentation in Feature Space

In order to have the unary potentials selected at a point on one object propagate the labeling to the other similar objects in the scene, non-position dependent features must be added to the pair-wise potentials of the CRF. These features should be descriptive of the object, i.e. by modeling its relation to the points around it in some meaningful way. Also, these features should be pose-invariant so they are consistent on objects of the same class in different orientations. Point normals calculated at three scales were attempted initially, and while these did provide a good measure of the geometry around a point, they were not rotationally invariant. Rotationally invariant non-position based features will propagate the unary potentials initially to object points of similar geometry, while the position-based features used earlier will allow those initial propagations to continue to label each object correctly. For example in scene 1, an initial point selected on the top of one cup will propagate through feature space to the top of the other cups, and then position-based features will continue the propagation down the rest of the cup.

3.2.1 Point Feature Histograms

Point Feature Histograms (PFH), and specifically their fast variant (FPFH) were used as a measure of the geometry in the k-neighborhood about each point [5, 6]. FPFH have been shown in prior work to be a consistent and descriptive feature used in 3D scene registration, so they are well suited to this application as well. For a given normal radius and feature radius, for each point the FPFH looks at the interaction of that points normal with every other point in its feature-radius neighborhood. The difference of each pair of point's normals and positions are encoded into 3 angular variations, and these variations are binned (because they have a finite range) for all pairs of points in the k-neighborhood. FPFH have 11 bins for each of the 3 independent angular features, resulting in 33 values total. While these features were seen to work well at describing the geometry of a each point, using 33 or more features in the pair-wise potentials is too computationally costly for the CRF model.

3.2.2 PCA on the FPFH

Principle components analysis was used on each scale of FPFH separately to reduce the number of features. PCA was also tried on all FPFH scales together, but this provided less benefit as the ability to give a different weighting to different scales of FPFH was lost when they were all combined together. Figure 5 shows a sample of the PCA variance curve for the FPFH features at any given scale. From the curve we can see that choosing 4 components represents ~65% of the variance, and choosing 8 components is ~80%. Four components were found to work well, considering that since 3 scales of FPFH were used there would be some overlap in data.

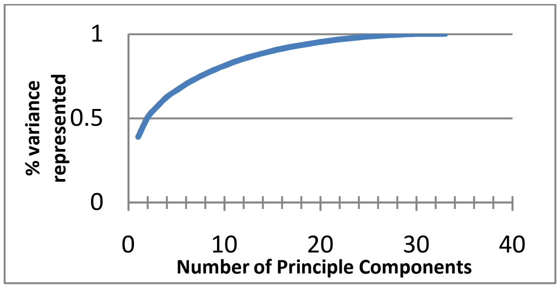


Figure 5. Percentage variance covered per components of the PCA on FPFH features – similar for each scale.

3.2.3 Results of Feature Space Segmentation

Feature Space segmentation using 3 scales of PCA calculated FPFH features was found to provide very good results on segmenting out multiple objects of a class given prior knowledge of one of those objects. Figure 6 shows the results of a segmentation using the same initial two points as in Figure 3, but now all 3 mugs are segmented out correctly. A measure of correctness of the labeling was computed by finding the number of points in each label that are within the truth label, and also the number of points mislabeled as the wrong truth label. The labeling of the two non-selected mugs in the scene were approximately 99.6% correct with ~0.4% mislabeled points.

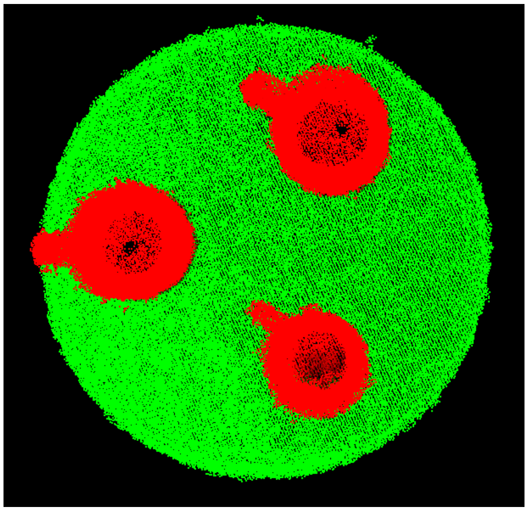


Figure 6. CRF segmented scene 1 using the same two points as unary potentials as in Figure 3.

4. More Complexity and Unsupervised

4.1 Greater Scene Complexity

Since the parameters of the dense CRF model were found to support the segmentation of scene 1, it is possible and very likely that this model is over-fit to the specific objects of scene 1. Creating new scenes is a very time-consuming process, so it was not possible to create a large number of scenes and learn the best parameters for the CRF model under the time constraints. Two other scenes was created using a larger number of object types (3 or 4), and more complicated orientations of objects. Figure 7 shows one of the new scenes collected, Scene 2, after its reconstruction. This scene includes 4 cups (one on its side), two small orange pumpkins, and 2 larger white pumpkins. The pumpkins can either be separated into two different classes, or combined into a single pumpkin class, as their underlying shape is the same, just their scale is different.

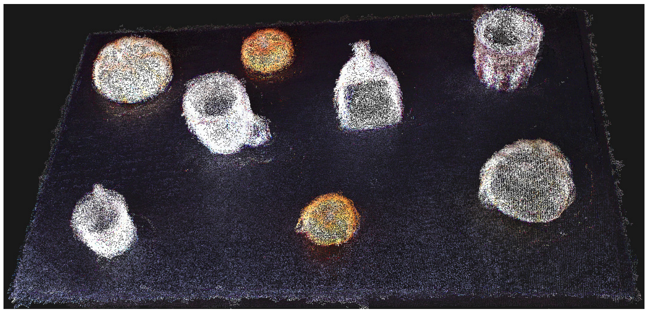


Figure 7. Scene 2 containing 4 mugs of varying orientations, and 4 pumpkins (which could be divided into 2 size subclasses).

The parameters used for the CRF model of this more complicated scene were nearly identical as those for those of scene 1. The only difference was an increased weight for the PCA FPFH features at the largest scale, probably due to the inclusion of objects at a larger scale such as the large pumpkins. This suggests a limitation of the FPFH features in that they are not scale invariant measures of shape, and because of that different scales of FPFH features will be necessary depending on the scale of objects you are attempting to segment out.

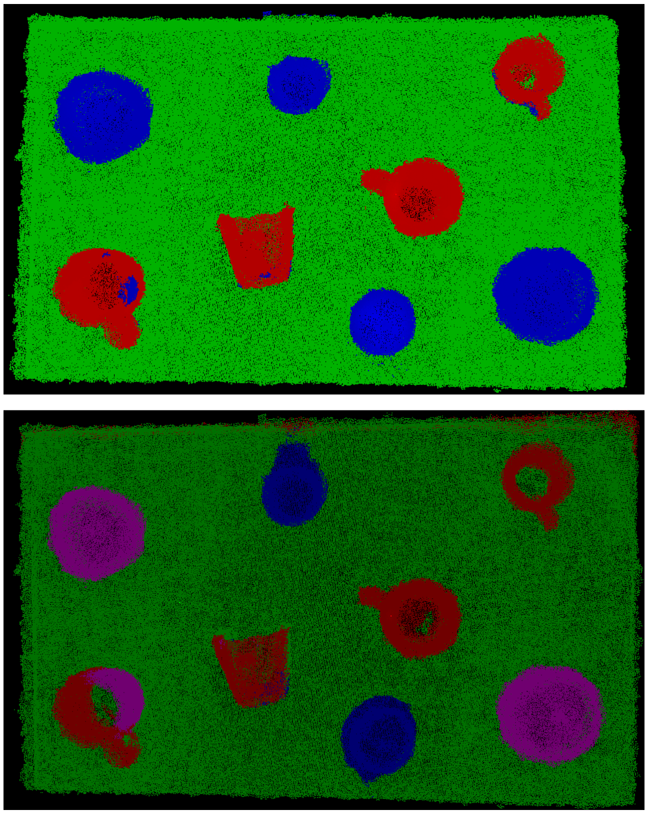


Figure 8. CRF segmentation of scene 2 with 3 labels (top) and 4 labels (bottom).

Figure 8 shows the results of the CRF segmentation on scene 2 when using both 3 and 4 labels. For both, the same 4 points are selected for the unary potentials, one on a single object of each class – mug, small pumpkin, large pumpkin, and table. The mug in the bottom left corner of the scene images exhibits the largest amount of mislabeling. This could be due to its scale and shape of the mislabeled portion most closely resembling that of the large pumpkin. The mislabeling of the bottom left mug can be remedied completely by selecting a single point on that mug as an additional unary potential. This suggests that an iterative segmentation scheme allowing user input will help to correct any errors obtained in an initial segmentation. As the number of preselected points increases, the reliability of the segmentation increases dramatically. Also, it is worth noting that the mug to the left of the center, which is tipped over so its handle is pointing upwards, is segmented out well in both cases, despite it being in a vastly different orientation then the mug with the preselected point to the right. This suggests that the FPFH features rotation-invariance will work well for discriminating between similar object geometries in different poses.

4.2 Unsupervised Segmentation

The goal of this research was to move towards fully unsupervised segmentation of the point cloud scenes. So far, the unary potentials for the dense CRF model were specified in advance for a single point in each object category, and left as zero for all other points in the scene. Though it is not difficult to select one point for each object category, this involves some user supervision of the segmentation. To move to an unsupervised segmentation, the unary potentials in the scene would have to be determined without any user input other than the number of possible classes.

One was this can be done is by using an unsupervised clustering algorithm to cluster the scenes points, and then using this initial clustering to generate the unary potentials for the dense CRF model of the scene. A modified version of k-means clustering was used on the FPFH features in the scene to create the unary potentials. The normal k-means algorithm was applied, using the standard Euclidean distance between the PCA FPFH features of each point. The convergence of k-means was dependent on the number of clusters specified. For 2 clusters, convergence was usually scene in ~14 iterations for both scenes, while for 3 clusters, convergence would take about 30 iterations. After k-means had converged, soft assignments for all points to all cluster were computed, and then a ratio of distance to any one cluster over the distance to all clusters combined gave a measure of how likely a point was in any given cluster. One minus this value, normalized by the number of clusters minus 1, gave a score from 0 to 1 of how likely each point belonged to each cluster (summing to one over all clusters). This value was used to determine if a unary potential should be set for the point, and what value it should be set at. Several schemes for determining the unary potentials from the k-means soft assignments were attempted. It was generally better to only define unary potentials on points that were clearly belonging or not belonging to a certain label. This results in using the 5%

points with the most confident labeling from the k-means to seed the unary potentials. The actual level of this confidence varies between scenes, but choosing a percentage of the points in the scene to use seemed to work well on the scenes used. For scene 1, this consisted of taking all the points with clustering scores greater than ~50% higher than random (0.5).

The results for performing k-means segmentation on scene 1 before and after applying the dense CRF refinement are shown below in Figure 9. The k-means only segmented image over-labels the mug points (21% mislabeled), selecting some features that have similar geometric properties, like the edges of the table, to be mugs as well. Performing the CRF on using unitary potentials generated from the k-means reduces the percentage of mislabeled mug pixels asymptotically to near 0.5% after many iterations.

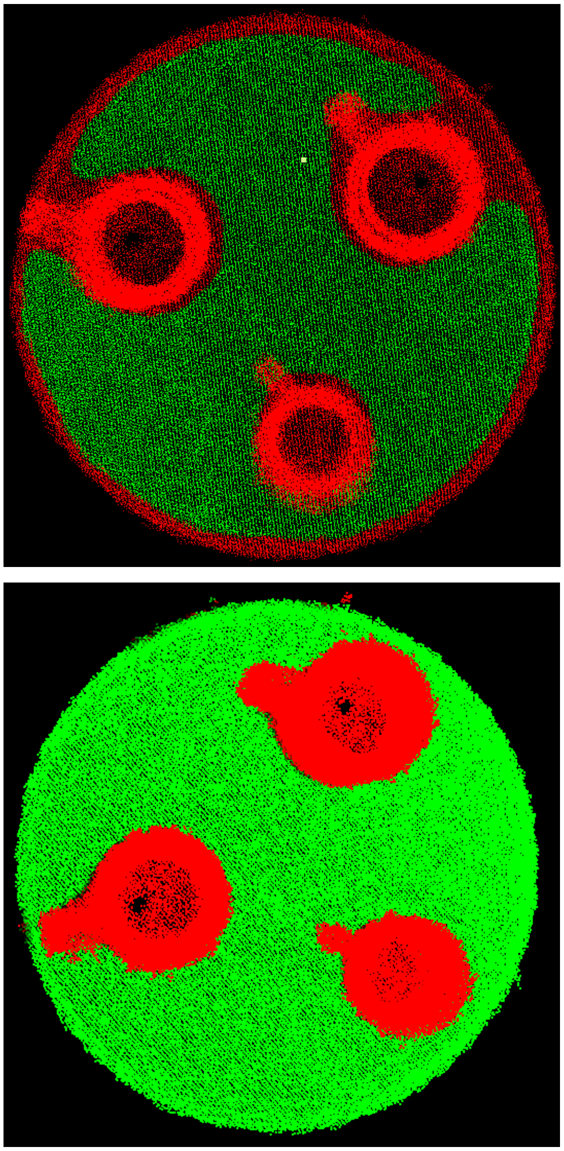


Figure 10. K-means only labeling (top), and k-means used to find the unary potentials for CRF segmentation (bottom)

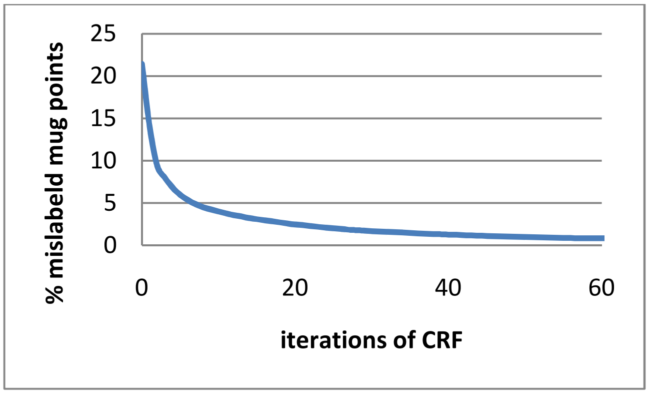


Figure 11. The percentage of points mislabeled as mug (red) for the example in Figure 10 by the number of iterations of the CRF. 0 CRF iterations is the k-means segmentation only.

Figure 11 shows the error as % of mislabeled mug points vs. the number of iterations used in the CRF. Just two iterations reduces the error by over half to 9%. The dense CRF model is able to quickly improve on the clusters produced by k-means alone, producing a cleaner clustering of the input scene.

5. Conclusions

The dense CRF model has been shown to produce good results in segmenting point cloud scenes. A large collection of datasets need to be examined in the future to see how well the model generalizes a set of objects. The similarity in parameters found for the two scenes suggests the model should generalize well. Future work include finding a dataset and using it to learn specific parameters to the CRF model to test the models generality. Preliminary tests with unsupervised segmenting with the CRF model show promise, but more methods of initial unary potential generation should be examined. Further work also includes finding which features are most distinctive for the data.

6. Acknowledgements

The author was advised by Philipp Krahenbuhl in the generation of ideas and extending his earlier work in [1] to this new domain.

REFERENCES

[1] Krahenbuhl, P., Koltun, V. “Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials”. http://graphics.stanford.edu/projects/densecrf

[2] T. Rabbani, F. van den Heuvel, and G. Vosselmann. Segmentation of point clouds using smoothness constraint. In *IEVM06*, 2006.

[3] R. Unnikrishnan and M. Hebert. Robust extraction of multiple structures from non-uniformly sampled data. In *IROS*, volume 2, pages 1322–29, October 2003.

[4] Golovinskiy, A. and Funkhouser, T. “Min-Cut Based Segmentation of Point Clouds.” In IEEE 12th ICCV Workshop, 2009.

[5] Rusu, R.B., et.al. “Persistent Point Feature Histograms for 3D Point Clouds.” In IAS, 2008.

[6] Rusu, R.B., Blodow, N. and Beetz, M. “Fast Point Feature Histograms (FPFH) for 3D Registration.” In ICRA, 2