Perceiving Clutter and Surfaces for Object Placement in Indoor Environments

Martin J. Schuster Jason Okerman Hai Nguyen James M. Rehg Charles C. Kemp

Abstract— **Handheld manipulable objects can often be found on flat surfaces within human environments. Researchers have previously demonstrated that perceptually segmenting a flat surface from the objects resting on it can enable robots to pick and place objects. However, methods for performing this segmentation can fail when applied to scenes with natural clutter. For example, low-profile objects and dense clutter that obscures the underlying surface can complicate the interpretation of the scene. As a first step towards characterizing the statistics of real-world clutter in human environments, we have collected and hand labeled 104 scans of cluttered tables using a tilting laser range finder (LIDAR) and a camera. Within this paper, we describe our method of data collection, present notable statistics from the dataset, and introduce a perceptual algorithm that uses machine learning to discriminate surface from clutter. We also present a method that enables a humanoid robot to place objects on uncluttered parts of flat surfaces using this perceptual algorithm. In cross-validation tests, the perceptual algorithm achieved a correct classification rate of 78.70% for surface and 90.66% for clutter, and outperformed our previously published algorithm. Our humanoid robot succeeded in 16 out of 20 object placing trials on 9 different unaltered tables, and performed successfully in several high-clutter situations. 3 out of 4 failures resulted from placing objects too close to the edge of the table.**

I. INTRODUCTION

Flat surfaces are a prominent feature of indoor human environments. People often manipulate objects on flat surfaces elevated above the ground, such as desks, workbenches, counter tops, and tables. People also store objects on flat surfaces, such as exposed shelves and shelves found within cabinets, refrigerators, and closets. Humanoid robots capable of manipulating objects on these same surfaces would be advantageous for a variety of tasks, including providing assistance to people with motor impairments. Unfortunately, objects in real scenes tend to be grouped together in a disorganized fashion, resulting in the phenomenon known as clutter.

Researchers have previously demonstrated methods for perceiving and manipulating objects on flat surfaces, but these methods can fail when applied to scenes with real-world clutter. For example, our previous research on grasping objects requires that much of the surface is visible and that the target object is in relative isolation [1]. In addition, this method only used point clouds from a tilting laser range finder, which resulted in poor performance on low-profile objects and objects made out of transparent or reflective materials.

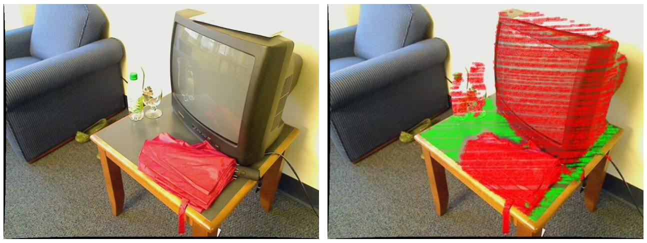


Fig. 1. Left: Example of a natural cluttered surface. Right: Labeling of surface (green) and clutter (red) produced by our algorithm.

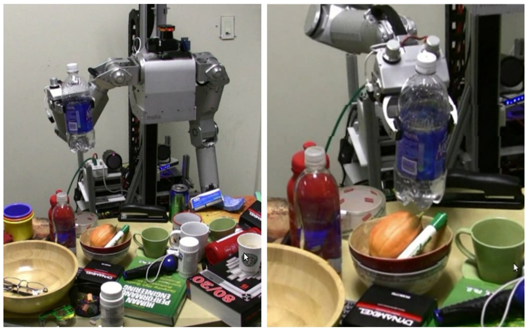


Fig. 2. Humanoid robot *Cody* placing an object in an informal test in which Cody successfully placed two objects on a highly cluttered table

In order for robots to robustly perform manipulation tasks in real-world environments, they will need methods that handle real-world clutter. For example, older adults tend to have more clutter in their homes due to downsizing their living environments. Clutter creates both perceptual and physical challenges for robot manipulation. Within this paper, we focus on perception of clutter and the placement of objects. Specifically, we consider perception of flat surfaces at close range from a vantage point comparable to our previous work [1], and similar to the perspective a human might take when preparing to manipulate objects on a surface. In [2] clutter is defined as everything except the object of interest for manipulation. For this paper, we use a similar definition of clutter as everything on top of the surface of interest. We also look at how a humanoid robot can place an object on a cluttered surface. For this task, we consider an object to be placed well if it is placed on a clear area of the underlying surface. In the future, we anticipate that this will be an important skill for robots, since they will be likely to place objects almost as frequently as they grasp objects.

We make three main contributions. First, we begin to characterize the statistics of real-world clutter. We have collected

The authors are affiliated with the Georgia Institute of Technology, Atlanta, Georgia, USA. The first author is also affiliated with the Technische Universität München, Germany. Email: martin.schuster@gatech.edu, jokerman@gatech.edu, haidai@cc.gatech.edu, rehg@cc.gatech.edu, charlie.kemp@hsi.gatech.edu

and hand labeled 104 scans of real-world cluttered surfaces using a tilting laser range finder (LIDAR) and a camera. We describe our method of data collection and present notable statistics from the dataset, such as distributions of height, surface normals and colors. Second, we introduce a machine learning approach to discriminate surface from clutter, as shown in Figure 1. We first define a variety of features, including several range, intensity, color and texture features. We then use these features and the labeled dataset to perform supervised machine learning, resulting in a classifier that assigns a label of *surface* or *clutter* to an input feature vector. Given a new scene, we label the scene using this classifier and perform post processing. We evaluated this approach with leave-one-out cross-validation testing on the collected real-world data. We also show that using both camera and LIDAR features results in superior performance relative to single-modality features. Furthermore, we show that this new approach significantly outperforms our previous method [1]. Third, we present a method that enables a humanoid robot to place objects on clutter-free areas of cluttered surfaces, see Figure 2. This method uses our clutter perception algorithm that has been trained on real-world clutter. We also evaluated the performance of this method when used by the humanoid robot Cody in 20 trials.

II. RELATED WORK

Relative to grasping objects, placing objects has received little attention, for example [1] [3]. In this paper, we look at placing objects on indoor surfaces that have significant clutter on them. Specifically, we focus on the challenge of discriminating clutter from the underlying surface, so that the robot can select a clutter-free area at which to place the object. Previous research related to discriminating clutter from surface can be divided into three main groups: methods for the general segmentation of objects from their supporting surfaces without using trained classifiers; segmentation approaches which rely on learned data statistics; and methods that rely on active manipulation of the scene to obtain more effective perception of object properties.

A. Segmentation without trained classifiers

Several previous works consider the problem of segmenting surfaces and clutter without using learned data statistics. One advantage of this approach is its generality, as it does not require access to training data, and can therefore be more easily applied to a wide range of environments.

In our previous work [1] we segment tables and objects based on point cloud histograms and clustering via 3d connected components. [4] uses WRANSAC-based plane fitting for table segmentation, assuming that tables consist of at least 30% of the dataset's values and lie in a height range (60-100cm). These assumptions do not always hold true for real world data, as shown in section IV. In our dataset, 9 out of 26 tables are below 60cm, see Figure 6.

In [5], plane- and edge-fitting as well as histograms over surface curvatures and intensities are applied for door and door handle detection. In [6], curvature based features are used for the registration of multiple point clouds and as the basis for a region growing approach to identify planar regions for plane and cuboid fitting. These techniques can fail if the relevant features are not distinguishable in the LIDAR data alone due to material properties, sensor noise and geometric constraints like occlusions. Since occlusions are the normal case in cluttered environments, we focus on scenes with multiple overlapping objects that leave the underlying surface only partially visible.

B. Segmentation using trained classifiers

A different approach, also followed in the current paper, is to use classifiers trained on a large dataset of example situations.

A promising machine learning technique are Conditional Random Fields (CRFs) which incorporates spatial relationships. They are used in single image processing for segmentation [7], to recover occlusion boundaries and depth-ordering from images [8] and for holistic scene understanding by combining CRFs in a multi-tier graph structure [9]. In [10] Support Vector Machines (SVMs) and CRFs are compared for the classification of point clouds into geometric primitive surface types using Fast Point Feature Histograms (FPFH) to compactly represent local curvature features. Their focus is on separating objects resting on a table whereas our segmentation method deals with the distinction between surface and objects.

[11] uses supervised learning on synthetic images to find grasp points on objects. Although they present one experiment featuring a cluttered background, the classifier was specifically trained for that scenario.

[12] and [13] describe methods for sensor fusion at the feature level and train neural networks on the combined features for segmentation. Multiple specialized classifiers can be combined in different ways [14] [15] to lead to better results than a single classifier is able to achieve on the complete input. Although this approach is not pursued here, it can be considered for future work in the segmentation of clutter.

C. Manipulation driven perception

Segmenting objects in unstructured environments, especially clutter, is still a very hard problem to solve by vision alone. An option to overcome these difficulties is to use manipulation of the scene to gain a better understanding of the object boundaries, their rigid bodies, joints and even their dynamic behavior [16] [17] [18]. Interactive perception is an attractive option for segmenting objects in highly-cluttered scenes where passive perception may be very challenging.

III. DATA COLLECTION

A. Sensor setup

The hardware setup that we used to collect data of cluttered tables consists of two sensors:

- a *Hokuyo UTM-30LX* laser scanner, tilted by a servo to create a 3d point cloud and register intensity values

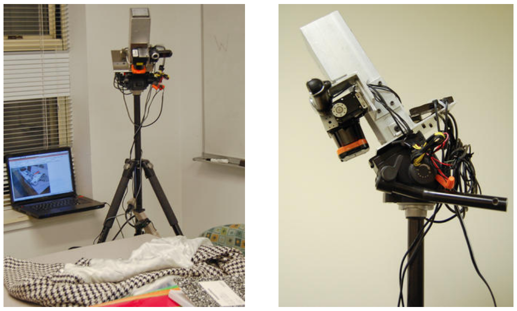


Fig. 3. Scanner: Tilting laser sensor and webcam mounted on tripod

- a *Logitech QuickCam Vision Pro* 2-megapixel webcam used to take single color images

We calibrated the range finder and camera in order to project 3d range measurements into the 2d image, associating each 3d range point with a corresponding pixel. We mounted the sensors on a tripod to support gathering data at a variety of locations, see Figure 3.

Each dataset, which we refer to as a *scan*, consists of a 3d point cloud and a single color image. The point cloud has range and intensity values for each data point. We preprocessed this data by truncating it to the field of view of the camera and then rotating it to match the ground plane orientation. This rotation is assumed to be known as it should be easy to estimate for most mobile robot systems, assuming that both the robot and the table are on the same floor.

B. *Labeled scan database*

One of our goals is to create a database of scans of realistic clutter. Our motivation is three-fold: to obtain insight into the general statistical properties of clutter in the context of indoor environments, to provide datasets for training and testing of machine learning methods, like the Boosting classifier presented in section V-B, and to benchmark algorithms. Our dataset is available at [19].

Using the sensor described in section III-A, we scanned 26 tabletops at several different locations around the Georgia Tech campus, including student housing and the home of a person with amyotrophic lateral sclerosis (ALS), thus showing a variety of surfaces. Figure 4 gives examples of images in the dataset. All scans show natural clutter, we did not clean up or rearrange the scenes prior to scanning.

For each scan, we placed the scanner approximately 50cm from the front edge of the table and 50cm above the table surface, tilted down by approximately 45°. The fields of view of both camera and LIDAR include a small patch of floor, the surface front edge and large parts of the surface itself. The setup is shown in Figure 5.

We captured four scans of each tabletop from four different angles, in order to view objects from multiple sides and capture parts that are occluded in one view. We scanned 26 different tables, resulting in a total of 104 scans. Our current dataset size is consistent with previous works [20] [21] [10] which employed from 40 to 100 images or point clouds to train classifiers. We believe it is an effective starting point for clutter analysis.

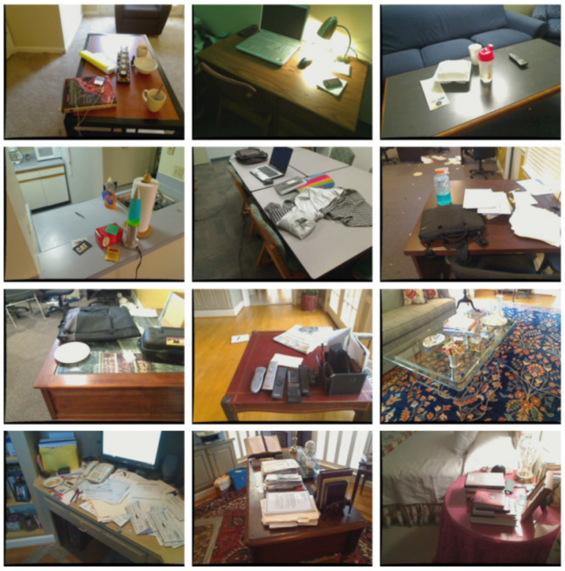


Fig. 4. Sample of 12 out of 104 images from the collected dataset

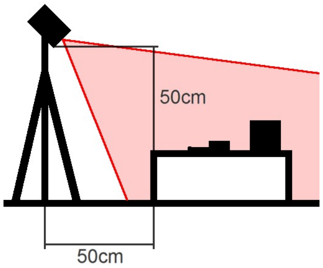


Fig. 5. Schematic of scanner setup (not drawn to scale)

Each scan is hand-labeled into three categories: *surface*, *clutter* and the *background*. We provide a rough estimation for the ground plane orientation by selecting three points on the floor by hand.

IV. CLUTTER STATISTICS

In this section, we present statistical properties of the collected dataset consisting of 104 scans from 26 different real world cluttered tables. We visualize the distribution of table heights as a histogram in Figure 6. The plot shows that many tables have a low height (40-50cm), such as a coffee table placed in front of a living room couch (upper right image in Figure 4). The peak is at approximately 75cm, a height often found for eating and office surfaces. A distribution over table heights could be useful as prior information for table detection. Another aspect for future research is the relationship between the height of a table and its function or the type of room in which it is located.

The height distribution of objects that are sitting on top of the surface can give insight into the type and amount of clutter. Figure 7 shows the height distribution over all 104 scans, while Figure 8 shows the distribution for two selected scans. The green bars show the measurements labeled as

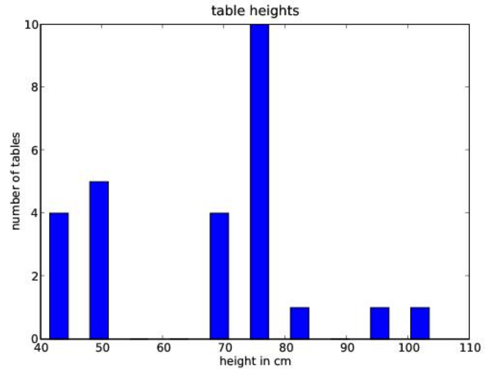


Fig. 6. Height distribution of all 26 tables from the dataset

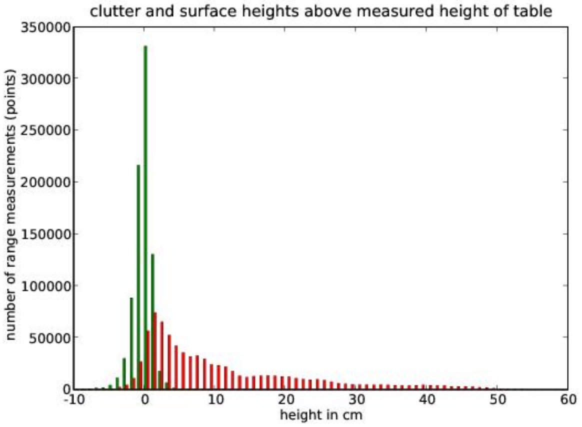


Fig. 7. Height of table (green) and clutter (red) measurements for all 104 scans. Classes are determined by hand labeling.

surface and the red bars show the measurements labeled as clutter. We set a height of zero to be the mean height of the surface measurements. The large variance in table measurements could be caused by the surface-dependent sensor noise of 1-2cm as well as the coarse estimation of the rotation of the scanner with respect to the ground plane. As can be seen in Figure 7, the clutter height distributions show on average a descending slope as expected due to physical stability of objects stacked on top of each other. However, individual scans can contain multiple maxima, as illustrated in Figure 8. A thorough analysis of the types of clutter, the environments in which it accumulates, and the characteristics of the individuals who live or work there is an area for further research.

V. SEGMENTATION: SURFACE AND CLUTTER

This section is about the application of machine learning methods to classify the data into the two categories *surface* and *clutter* without knowledge about the height of the table.

A. Feature descriptions

As the definition of appropriate features is crucial for success in classification, we analyzed several geometry and appearance based features. We calculate a feature vector for every 3d point in the point cloud.

1) Range features: We use three different types of features based on the laser range measurements. The first features are the x, y and z components (nx, ny, nz) of the estimated surface normals on small patches of the scan. They are calculated using covariance analysis [22] of spherical neighborhoods, radius 3cm, around the current point. Normals pointing away from the scanner are flipped, as described in [6]. These features allow the distinction between tables and vertical or curved surfaces, but not between tables and other flat objects like paper or books because both have surface normals that point upwards. Figure 9 shows the surface normals of the 104 scans plotted as points on a unit sphere. The x-axis (red) is pointing away from the position of the scanner, the z-axis (blue) up, the y-axis (green) to the left. It can be seen that the table normals (green) are almost all pointing upwards whereas the clutter normals (red) are distributed over the sphere, their distribution having several peaks in horizontal direction which indicate vertical surfaces.

The other range features are measured relative to the point cloud of the current scan which we split into several horizontal layers of 2.5cm in the upward direction. We create a histogram with a bin for each layer, containing the number of range measurements. For each range measurement, we create a feature ($zhist$) which is the normalized value of the histogram bin in which the current range measurement falls into. We designed this feature to capture global properties of the distribution of range measurements such as the fact that the table is often associated with a high peak along the z-axis as can be seen from Figure 7 and Figure 8.

Another feature describes the spatial distribution of points in the same height slice as the current point, defined by the two largest eigenvalues (ev_1, ev_2) of a covariance analysis

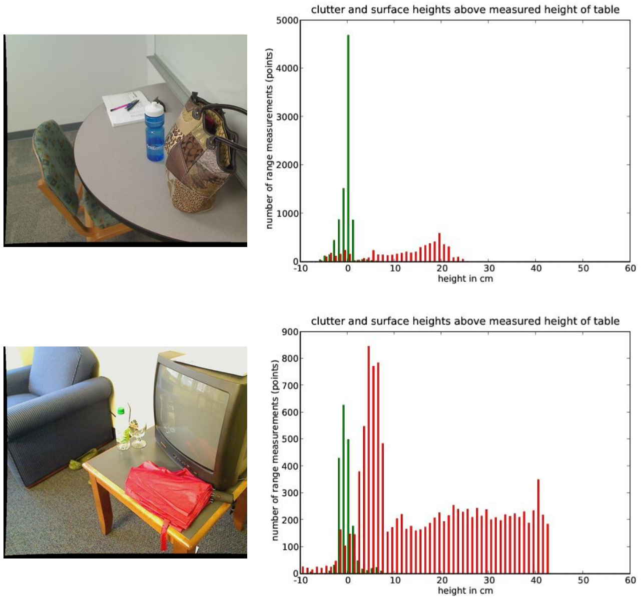


Fig. 8. Height of table (green) and clutter (red) measurements for two different cluttered tables

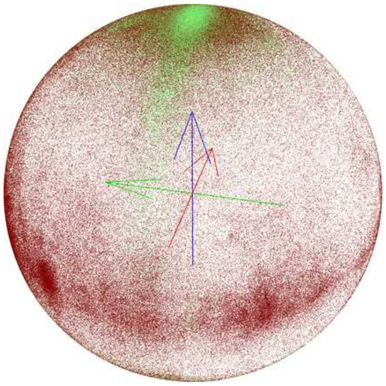


Fig. 9. Hemisphere showing the directions of 800000 surface normals that were randomly sampled from all 104 scans as dots on a unit sphere

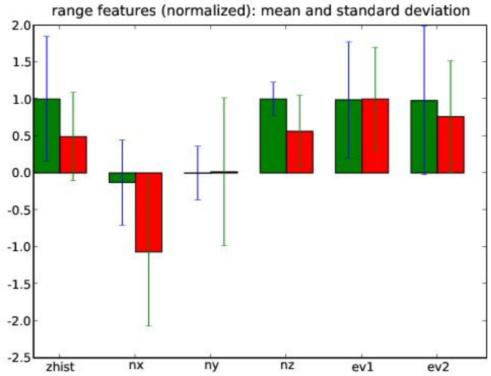


Fig. 10. Range features: mean and standard deviation. $zhist$ denotes the z-histogram feature, nx, ny, nz the surface normals, and $ev1, ev2$ the two largest eigenvalues of covariance analysis of the same height slice as the current point

of all points in the slice. As table borders are seldom occluded by objects on all sides, see Figure 4, we expect that tables in many cases have a wide spatial distribution in two dimensions whereas objects tend to be located close together with lower ev_1 and ev_2 values.

The mean value and standard deviation of all range features over the 104 scans is shown in Figure 10. This gives a six-dimensional range feature vector:

$$f_r = (zhist, nx, ny, nz, ev_1, ev_2) \tag{1}$$

2) Color features: For color and intensity, we use a variety of features as it is hard to find a clear decision boundary in HSV space because the colors and textures for tables and objects vary greatly throughout the collected dataset. Still, at least for some parts of the HSV color space, table and objects can be separated. Figure 11(a) shows the distribution of colors for both classes in hue and saturation space. Figure 11(b) marks the h- and s-values in a 256×256 grid for which one class is assigned to over 80% of the samples. We excluded value-pairs with less than 5 samples and did not apply any interpolation.

In our dataset we observed that tables have a high probability to either have some selected hue values that correspond to wood color or low saturation. All other high saturation values are good indicators of clutter. For example, bright blue tables are uncommon, but consumer packaging often includes highly saturated colors. Consequently, we use the raw color values in HSV space and the measured intensity from the laser range finder as features (h, s, v, i).

Table-surfaces often have only small variance in color and a low texture level whereas these properties vary for objects. To capture this property, we use normalized one-dimensional five-bin histograms for HSV and intensity in a spherical local neighborhood with a 3cm radius ($hist_h$,$hist_s$,$hist_v$,$hist_i$) as well as two texture-based features (tex_1, tex_2). We use the two eigenvalues returned by the OpenCV function CornerEigenValsAndVecs() as the texture features, based on 8x8 gray scale blocks from the camera image.

Additionally, for each 3d point we compute three values, (sh_h, sh_s, sh_i), that indicate the percentage of 3d points at the same height that have similar hue, saturation, and intensity. We expect the color and intensity across the surface to be relatively uniform, so these three feature values should tend to be higher at the surface's height. To accomplish this, we first compute normalized one-dimensional hue, saturation, and intensity histograms for each height slice. Given a 3d point, we then use the three histograms associated with its height to compute the percentage of points with similar hue, saturation, and intensity.

In total, the color feature vector has 29 dimensions:

$$\begin{aligned} f_c \quad = \quad & (h, s, v, i, hist_h, hist_s, hist_v, hist_i, \\ & tex_1, tex_2, sh_h, sh_s, sh_i) \end{aligned} \tag{2}$$

with

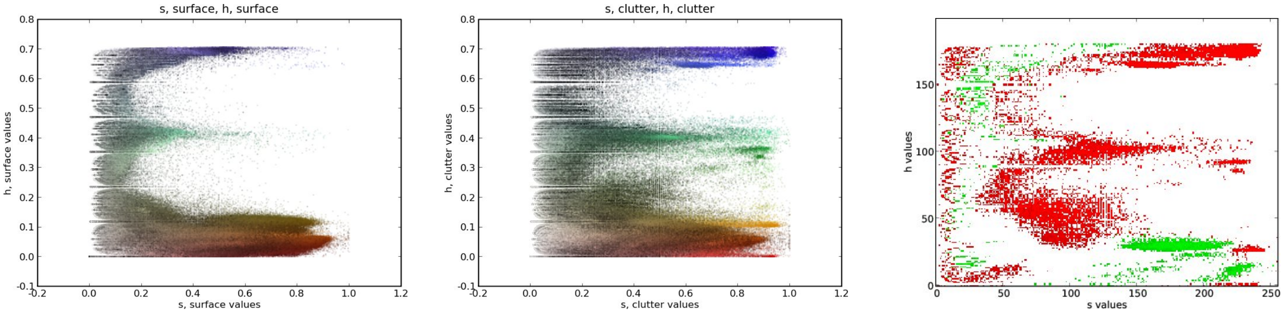
$$\begin{aligned} hist_x = (hist_{x1}, hist_{x2}, hist_{x3}, hist_{x4}, hist_{x5}) \\ for\ x \in \{h, s, v, i\} \end{aligned} \tag{3}$$

3) Combined features: The complete feature vector is the concatenation of the range and color features and has a dimension of $|f_v| = |f_r| + |f_c| = 6 + 29 = 35$.

B. Segmentation using Boosting

Our main goal is the segmentation of the 3d point cloud into the two classes *surface* and *clutter* using the features described in section V-A. To perform this classification task we apply a Decision Tree Boosting classifier. We use the AdaBoost implementation from the OpenCV library v1.1 ($CvBoost$) with default settings.

For testing, we used all labeled points from the 3d point clouds. For training there are two exceptions: We excluded points near the edges of the polygonal labels to reduce the influence of the calibration error between the two sensors. To avoid biasing the classifier towards one of the two classes we drew random samples from the class with the lower number of data points and added them to the training set on a per scan basis. This leads to an equal number of training samples for both classes, thus removing any prior probability in favor of one of the classes.



(a) Hue and saturation distributions for tables (left) and clutter (right)

(b) Probability of class membership over 80% for tables (green) and clutter (red)

Fig. 11. Statistics over hue and saturation space

C. Post processing

For some experiments, we perform an additional post processing step. We assume that all points correctly classified as *surface* lie on a planar surface. We therefore fit a plane model to all points classified as *surface* by the AdaBoost classifier. This is done by running a RANSAC plane fitting algorithm on them and ensuring that the found plane is approximately horizontal. We then change the labels of all *surface* points that are not in the consensus set to *clutter*, as these outliers are assumed to be misclassifications.

VI. EXPERIMENTS

A. Setup

We performed leave-one-out cross-validation on the 104 scans from 26 different tables. For each split, we divided the labeled scans into a test set, consisting of the four scans from one table, and a training set containing all other scans. We repeated this for each table in our database. Ground truth for training and testing was provided by the manually assigned labels described in section III-B.

As evaluation criteria we use the percentage of correctly classified laser range measurement points for each of the classes $C \in \{surface, clutter\}$ that is:

$$\%\ C \text{ correct} = \frac{\#\text{points classified and labeled as } C}{\#\text{points labeled as } C} \times 100$$

For the evaluation of our cross-validation testing, the measurement points are summed up over all folds. In the ground truth labeling, the total numbers of points labeled are 849944 for *surface*, and 815784 for *clutter*.

We conducted experiments using range features only (*range*), color features only (*color*), the combination of both (*all*) and this combination plus the post processing step (*all + post proc.*), as described in section V-C. For comparison, we also tested an algorithm for surface segmentation (*baseline algo*) that is currently used for the assistive robot *EL-E* at the Healthcare Robotics Lab (HRL) [1], which uses height histograms and connected components.

B. Results and comparison

We present the results as the percentage of correctly classified points for each class in the following table:

Dataset	Features / Algorithm	% surface correct	% clutter correct
total	range	72.01%	79.91%
	color	59.00%	73.64%
	all	79.36%	87.31%
	all + post proc.	**78.70%**	**90.66%**
	baseline algo	59.23%	78.64%

The rows labeled with *baseline algo* show the results from the algorithm used in [1] on the same data.

Figure 12 gives visualized test results for two scans from our dataset. The colors green and red represent surface and clutter, respectively.

The results show that a combination of range and color features improves the classification quality in almost all test cases over the usage of only one method. There are several cases where range or color features alone fail completely on one of the two classes. Considering the often unfavorable lighting conditions for the images, e.g. heavy reflections causing areas in the image to appear plain white as shown in Figure 4, and the noise of the laser measurements, the failure of single modality features is not surprising. But using range and color together, our algorithm is often able to achieve high quality results under these real world conditions.

The additional post processing step succeeds in pruning surface outliers and exhibits only a small negative impact on the number of correctly classified surface points due to removing correct surface labels. It therefore further improves the overall results. In comparison, our new learning based algorithm gives a higher percentage of correct class labels than the baseline algorithm.

Compared to the baseline algorithm, our method performs well for scans where large parts of the surface are occluded, as in Figure 1 - a case often found in cluttered environments and therefore explicitly addressed by this research. It also allows a better classification of flat objects like sheets of paper, that fall below the threshold of the baseline algorithm, as shown in Figure 12. In this case, the range features alone are not sufficient to detect the paper. On the other hand, the color features alone detect the paper correctly but lead to a misclassification of the shadows on the surface. The combination of both and a removal of surface-outliers (*all +*

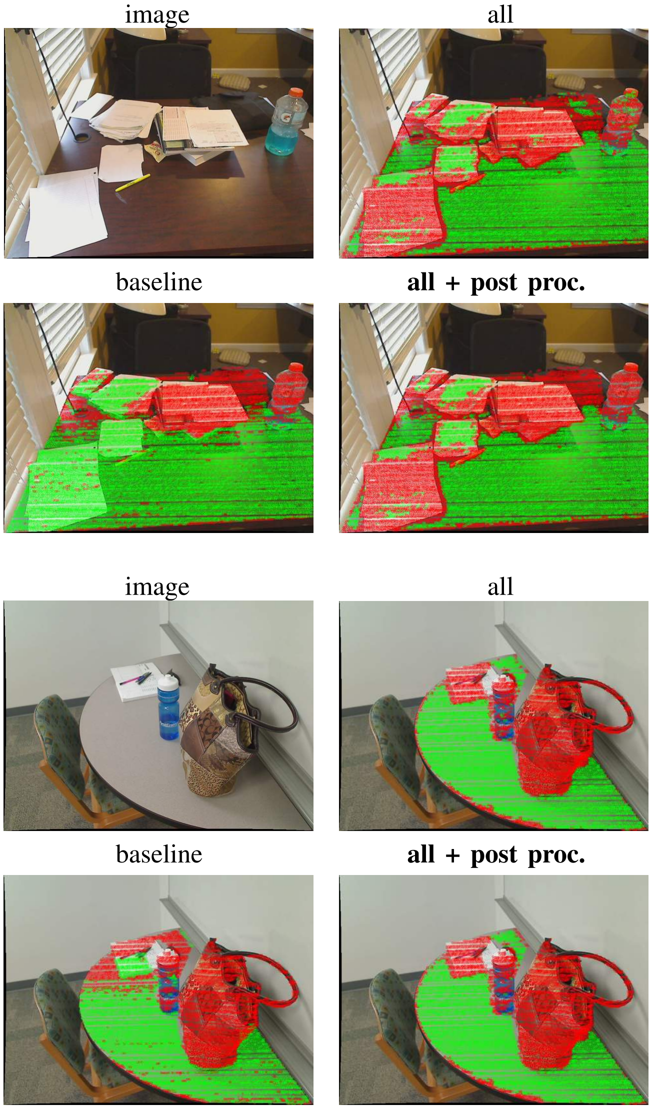


Fig. 12. Test result visualizations for two out of 104 tested scans

post proc.) yields the best results, leaving only small parts of the paper-stacks misclassified.

As our algorithm depends on its training data, it still has problems in some situations, misclassifying large flat objects because their color and texture-values are similar to tables in the training data. Although the post processing step can correct some of these errors if they occur far enough away from the estimated surface plane, we expect that further global features would help address this problem. The algorithm can also fail on transparent glass tables that lead to noisy range finder measurements located below the real surface and visual features based on the objects below, therefore rendering our feature-set almost useless.

VII. OBJECT PLACEMENT

In order to evaluate the segmentation algorithm in real environments, we used the HRL humanoid robot *Cody* to place objects on office desks. The task for the robot was to place an object on a clutter free space on a cluttered table located in front of the robot within the workspace of its arm.

A. Methods

To select a placement location on the table, the algorithm divides the detected plane of the table into square grid cells with 1cm resolution. It then calculates a score for each cell. A higher score indicates a better location and a score of 0 denotes an invalid location. To calculate this score, the algorithm sequentially applies the following evaluations:

- *Flatness*: First, the algorithm creates a height map from the 3d point cloud and computes the maximum height difference between the points found within the footprint of the object centered at the cell. If this difference is greater than 4cm the cell is given a score of 0, while between 2cm and 4cm results in a score of 1, and less than 2cm gives a score of 2.
- *Segmentation result*: Next, the algorithm randomly selects 1000 points in a volume of interest corresponding to the arm's workspace. It then classifies these points into surface and clutter using a modified version of the algorithm we described in section V. The robot's version uses a single decision tree classifier (CvDTree()) and removes isolated clutter points. It then sets the score of a cell to 0 if the footprint of the object centered on the cell contains any clutter points.
- *Free approach path*: Then, for collision avoidance during placement, the algorithm defines a region by extending the object's footprint towards the robot. If the difference between the height at the current cell and any height within this region is greater 8cm, the cell is given a score of 0.

After scoring, any cell with a score above zero becomes a candidate for object placement. In the code that we tested with the robot, these candidates were further weighted, albeit in a manner that does not appear to have influenced the results and is beyond the scope of this paper. The algorithm selects the cell with the highest score, which corresponds to the goal location for the center of the object's footprint. The robot uses inverse kinematics (IK) to find joint angles that place the object above the selected cell. If an IK solution is not found, then the next best point is selected until a candidate with an IK solution is found or the robot gives up. If found, the arm is first positioned above the 2D location of the selected cell and then lowered to place the object. The robot lowers the object until the robot detects a collision using its wrist-mounted force-torque sensor, at which point it releases the object.

B. Experiments

1) Setup: We performed preliminary placement tests to demonstrate an application for our segmentation algorithm. We tested the placement algorithm with the robot Cody using nine office desks that were part of the local lab environment. The items and orientation of clutter was intentionally left unchanged from how these desks were left at the end of a day in order to have a realistic sample set. We manually

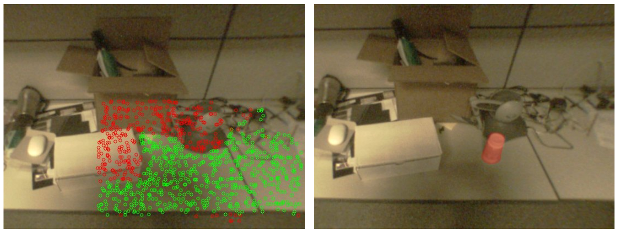


Fig. 13. Left: Segmentation of cluttered table, right: Placement of object

placed one of two pill bottles into the robot's gripper and passed its height and footprint size to the algorithm. Each pill bottle was placed once per table, with one table repeated, for a total of 20 trials.

Cody's highly compliant arm allows the robot to place objects in constrained spaces even with positioning inaccuracies. The robot's sensors included a web camera and tilting laser scanner, similar to the setup described in section III-A. For each test, the robot was positioned 24 to 29 cm away from the table edge.

2) Results and Discussion: We performed 20 tests on 9 different tables with two object placements per table. One of the tables was used for two tests (4 placements).

result	**comment**	**count**
success	placed upright	11
	object tipped over	5
failure	placed on table edge	3
	segmentation failure	1

The overall success rate was 80%. Figure 13 shows the segmentation for one of the trials and the following successful placement of the object in the free area. In three of the failure cases, the object was placed too close to the edge of the table and fell off as the hand moved away. This might be the result of inaccuracies in the compliant arm control. The one other failure case resulted from an incorrect classification of a keyboard as table surface. Cody currently uses a different camera model with a different color calibration than we used for our training data, which may have degraded the performance.

VIII. CONCLUSIONS

We have presented statistics for real-world clutter, a method to perceptually discriminate clutter from surface, and an initial implementation of robotic object placement in clutter. Our results show that our presented perceptual method improves upon our prior algorithm and handles common challenges associated with natural clutter. The statistics we present indicate that real-world clutter has significant structure that can be exploited in robot algorithms. Further improvements in clutter perception, clutter manipulation, and the characterization of real-world clutter would likely benefit autonomous robots in human environments and might be necessary for some applications. We believe that data-driven approaches and data-driven evaluation will play an important role in future research.

IX. ACKNOWLEDGMENTS

We thank the members of the Healthcare Robotics Lab at Georgia Tech, especially Travis Deyle for support with the scanner hardware, Advait Jain for providing his implementation of the baseline algorithm and Tiffany Chen for supervising students who collected data. We thank Kristina Falkenstrom, Aakanksha Gupta, Mrinal Rath and Christopher Romano for collecting scans and labeling them. We thank Matthew Mason and Siddhartha Srinivasa for several useful discussions. This project was supported in part by a seed grant from the Center for Robotics and Intelligent Machines at Georgia Tech (RIM@GT), and by the following NSF awards: RI award 0916687, CBET-0932592, and CNS-0958545. The first author was supported by an ATLANTIS scholarship from the European Union.

REFERENCES

[1] A. Jain and C. Kemp, "EL-E: An assistive mobile manipulator that autonomously fetches objects from flat surfaces," in *Autonomous Robots*, 2010.

[2] M. T. Mason, S. Srinivasa, and A. S. Vazquez, "Generality and simple hands," in *International Symposium of Robotics Research*, July 2009.

[3] A. Edsinger, "Robot manipulation in human environments," Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[4] R. B. Rusu, N. Blodow, Z. Márton, A. Soos, and M. Beetz, "Towards 3D object maps for autonomous household robots," in *IROS*, 2007.

[5] R. B. Rusu, W. Meeussen, S. Chitta, and M. Beetz, "Laser-based perception for door and handle identification," in *International Conference on Advanced Robotics (ICAR)*, Munich, Germany, 2009.

[6] R. B. Rusu, Z. C. Marton, N. Blodow, M. E. Dolha, and M. Beetz, "Towards 3D point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[7] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *Int. J. Comput. Vision*, 2008.

[8] D. Hoiem, A. N. Stein, A. A. Efros, and M. Hebert, "Recovering occlusion boundaries from a single image," in *ICCV*, 2007, pp. 1–8.

[9] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, February 2008.

[10] R. B. Rusu, A. Holzbach, N. Blodow, and M. Beetz, "Fast geometric point labeling using conditional random fields," in *IROS*, 2009.

[11] A. Saxena, "Monocular depth perception and robotic grasping of novel objects," Ph.D. Thesis, 2009.

[12] U. Handmann, G. Lorenz, T. Schnitger, and W. V. Seelen, "Fusion of different sensors and algorithms for segmentation," in *IEEE Conference on Intelligent Vehicles*, 1998.

[13] C. Rasmussen, "Combining laser range, color, and texture cues for autonomous road following," in *ICRA*. IEEE, 2002, pp. 4320–4325.

[14] C. Dima, N. Vandapel, and M. Hebert, "Classifier fusion for outdoor obstacle detection," in *ICRA*. IEEE, 2004, pp. 665–671.

[15] G. Monteiro, C. Premebida, P. Peixoto, and U. Nunes, "Tracking and classification of dynamic obstacles using laser range finder and vision," in *IROS Workshop*, 2006.

[16] P. M. Fitzpatrick and G. Metta, "Towards manipulation-driven vision," in *IEEE/RSJ Conference on Intelligent Robots and Systems*, 2002.

[17] P. Fitzpatrick, G. Metta, L. Natale, A. Rao, and G. Sandini, "Learning about objects through action -initial steps towards artificial cognition," in *ICRA*. IEEE, 2003, pp. 3140–3145.

[18] D. Katz and O. Brock, "Manipulating articulated objects with interactive perception," in *ICRA*. IEEE, 2008, pp. 272–277.

[19] M. J. Schuster, J. Okerman, H. Nguyen, J. M. Rehg, and C. C. Kemp. (2010) Clutter dataset: scans of cluttered tables. [Online]. Available: http://www.hrl.gatech.edu/data/clutter

[20] S. Kumar and M. Hebert, "Discriminative fields for modeling spatial dependencies in natural images," in *NIPS*, 2003.

[21] S. V. N. Vishwanathan, N. N. Schraudolph, M. W. Schmidt, and K. P. Murphy, "Accelerated training of conditional random fields with stochastic gradient methods," in *ICML*. ACM, 2006.

[22] M. Pauly, M. H. Gross, and L. Kobbelt, "Efficient simplification of point-sampled surfaces," in *IEEE Visualization*, 2002.