Hierarchical Object Geometric Categorization and Appearance Classification for Mobile Manipulation

Zoltan-Csaba Marton, Dejan Pangercic, Radu Bogdan Rusu, Andreas Holzbach, Michael Beetz
Intelligent Autonomous Systems, Technische Universität München
{marton, pangercic, holzbach, beetz}@cs.tum.edu, rusu@willowgarage.com

***Abstract*— In this paper we present a comprehensive object categorization and classification system, of great importance for mobile manipulation applications in indoor environments. In detail, we tackle the problem of recognizing everyday objects that are useful for a personal robotic assistant in fulfilling its tasks, using a hierarchical multi-modal 3D-2D processing and classification system. The acquired 3D data is used to estimate geometric labels (plane, cylinder, edge, rim, sphere) at each voxel cell using the Radius-based Surface Descriptor (RSD). Then, we propose the use of a Global RSD feature (GRSD) to categorize point clusters that are geometrically identical into one of the object categories. Once a geometric category and a 3D position is obtained for each object cluster, we extract the region of interest in the camera image and compute a SURF-based feature vector for it. Thus we obtain the exact object instance and the orientation around the object's up-right axis from the appearance. The resultant system provides a hierarchical categorization of objects into basic classes from their geometry and identifies objects and their poses based on their appearance, with near real-time performance. We validate our approach on an extensive database of objects that we acquired using real sensing devices, and on both unseen views and unseen objects.**

I. Introduction

The use of accurate object models enables personal robotic agents doing everyday manipulation in indoor environments to perform their tasks more reliably, flexibly, and efficiently. As these robots get more sophisticated manipulation capabilities, they require more expressive and comprehensive object models, beyond their position and appearance, including information about their precise shape or additional semantic information that is useful with respect to the robot tasks.

Though the set of objects of daily use that a personal robot could encounter in its tasks is unlimited, there are certain regularities that can be exploited with respect to the object shapes, textures, or uses. Therefore, in some sense, the perception system can specialize itself to a specific set of objects that are usually present in the world, while at the same time retain a certain degree of flexibility with respect to the incorporation of novel objects in its internal models. For example, a new flavor of iced tea should be recognized as an instance of a tea box from its geometry (shape) or use, even though the robot has never seen it before, and therefore could not understand the semantics of the object from its visual appearance.

Radu Bogdan Rusu (now at Willow Garage, Inc) and Andreas Holzbach (now at the Institute for Cognitive Systems, TUM) were at the Intelligent Autonomous Systems group at TUM when carrying out this work.

This paper proposes a comprehensive multi-modal perception system comprised of hierarchical object geometric categorization and appearance classification for personal robots manipulating in indoor environments. Our focus is on robustly identifying objects of interest supported by planar surfaces such as tables that can be manipulated by the robot. Since the goal is to have personal robotic assistants working and operating in the same environments, in our case kitchen, for months or years without interruption, a clear need to learn accurate models of the objects that are to be manipulated again and again arises. This requires the creation of efficient object classifiers that can discriminate but also generalize between the objects present in the world over time.

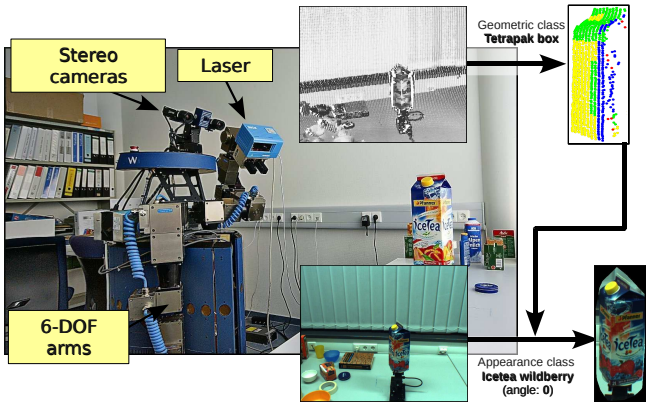


Fig. 1. The mobile manipulation platform used for obtaining the database and performing hierarchical object categorization and classification. The hardware setup consists of a B21 mobile base with two 6-DOF arms, stereo cameras, a laser sensor mounted on the end effector and a rotary table. The right area of the image shows the input as observed by the robot and the surface and geometric categorization and classification of an iced tea box.

We motivate the additional use of geometry along appearance (i.e., texture) in our approach for three equally important problems. On one hand, our experience is that texture alone can lead to false positive matches such as the example shown in Figure 2. Given a template picture of some object that we need to find in a scene, say a beer bottle, the system can mistakenly identify it on parts of the world which contain similar pictures of that object, a mug in this case. Simply put, without geometry, texture can be deceiving. Though the case presented here is slightly exaggerated, the point still stands.

On the other hand, there are several cases where for a

general purpose problem such as the one of identifying objects in indoor environments, the space of possible solutions becomes so big that the model that is to be learned has to support thousands of different classes or more. This leads to increased training or update times for the model, but can also result in decreased classification speed and accuracies. By considering geometry as well, we simplify the problem and create a more logical categorization and classification framework, by creating separate models for groups of objects that have the same or similar geometry (e.g., cereal boxes, mugs, etc). This results in an overall optimization (in terms of the computational complexity) for supervised learning.

Finally, there are objects which have no texture, in the sense that they are uniformly colored and do not exhibit texture properties that can be identified by regular keypoint descriptors in computer vision.

Without appearance however, it would be very hard to discover the orientation of an object, especially for objects with a near-symmetrical geometry. Though it might be argued that the orientation of a symmetrical object is not very important, in some cases (for example if the iced tea should be poured out of the tetrapak) this subtle disambiguation is mandatory for mobile manipulation. Additionally, we might want to discern between objects which have the same geometry but a different appearance and purpose, like for example a ketchup bottle versus a mustard bottle.

In detail, the approach presented herein combines our previously proposed 3D Radius-based Surface Descriptor (RSD) [1] and Global Point Feature Histograms (GFPFH) [2] with the 2D Speeded Up Robust Features (SURF) [3] to create a hierarchical geometric-appearance supervised classification scheme. Using the rough geometrical shape of the object we cluster objects in distinct categories and use that to influence the problem of classification based on appearance. The key contributions of the research reported in this paper thus include the following:

- the use of a fast 3D feature computation method to intuitively annotate surface units with geometric labels;
- the proposal of a powerful global descriptor (GRSD) that can generalize over objects with geometric similarities to limit the possibilities of which object instance could a cluster be;
- the synergy of depth and visual processing and learning techniques in a multi-modal hierarchical architecture for the problem of robustly identifying discriminative object classes and their rough orientations with respect to the robot camera view.

Fig. 2. An example of a good model match using SURF features extracted from 2D images (left), where a beer bottle template is successfully identified in an image. However, zooming out from the image, we observe that the bottle of beer is in fact another picture stitched to a completely different 3D object (in this case a mug).

The structure of this paper is organized as follows. Related work is described in Section II. Next, we give a brief description of our system architecture in Section III. The acquisition of a database of models is presented in Section IV. We present the geometric and appearance processing pipelines in Section V followed by a discussion of experimental results in Section VI. We conclude in Section VII.

II. Related Work

The are two principal mainstream lines in the area of the object recognition related research: one aiming at recognition of objects in camera images, and one using 3D depth data acquired through range scanning devices. Combining both of them leads to a hybrid approach and our work falls into this category. Depending on the type of perception data, various different 2D (e.g. [4]) and 3D (e.g. [5]) distinctive local features have been developed. Taken individually however, these are still insufficient to solve the full object recognition problem as both are prone to failure in situation where texture-less objects are present or depth data is too noisy or ambiguous. That is why different research initiatives have decided to combine sets of local features and cluster them together using different metrics (kernels), in order to be able to infer the global identifiers for objects.

In [6] objects are recognized and reconstructed using image databases. The overall approach is based on finding the consistent matches in the subsets of all images. Following a structure and motion of each object is solved using a Sparse Bundle Adjustment algorithm. Fergus et al. [7] is proposing an unsupervised scale-invariant learning scheme, in order to detect objects on a wide range of images. Objects therein are modeled as flexible constellations of parts using a probabilistic representation for all significant aspects of the object. The work exploits the expectation-maximization algorithm in a maximum-likelihood setting. The method in [8] estimates 6-DOF object poses in cluttered scenes by matching local descriptors to stored models. Since the objects present in household environments are most often texture-less, our approach constitutes an important advantage over the above proposed research initiatives, which fail to work in the absence of good textured objects.

The work in [9] uses an iterative matching procedure to merge similar models in an unsupervised manner, while a spectral clustering of similarity matrix is used to terminate the merging convergence. However, it is unclear how well the proposed algorithm would i) generalize to unknown, novel objects and ii) infer semantic properties of those. Lai et al. [10] perform outdoor laser scans classification combining manual labeling and data downloaded from the Internet in an effort coined domain adaption. While their presented recall curves outperform others, the number of objects is relatively low and household objects are less distinct. In [11], the authors investigate the extraction of GOODSAC point features and object recognition from range images, that are in turn computed from point cloud datasets. These object models are, as in our case, created from real 3D data but processed using the work in [9].

The combination of depth information with camera images is addressed in [12]. The authors calculate depth information for each pixel in the scene by applying laser-line triangulation with a rotating vertical laser and a camera. To obtain high resolution 3D images, each scanning requires 6 seconds with an additional 4 seconds spent on post-processing and triangulation. Thus a waiting period of 10 seconds has to be expected before object detection and robot manipulation could be performed.

In the work of [13] the grasping of objects modeled in the 3D object modeling center [14] was presented. The center employs a digitizer, a turntable and a pair of RGB cameras mounted to a rotating bracket which allows for views from above the scene. At the time being, there are around 40 high-detailed, high-precision objects available publicly. While working with such a system and data would yield high quality results, its downside is in that it is extremely expensive and cannot be used for the online detection phase, i.e. mounting and consequential mapping with a real robot. It is our belief that the characteristics of the training dataset (density, noise level) have to be comparable to the testing one. In another initiative, a database (Columbia Grasp Database [15]) has been built. The major difference between this work and ours lies in how the models were obtained. The authors created artificial 3D models whereas we acquired our models by scanning real world objects and surfaces, and are thus facing the problem of noisy and cluttered data.

III. System Architecture

The platform used for the acquisition of models is briefly described in Figure 1, and consists of a B21 mobile base with Amtec Powercube 6-DOF arms and sensors such as a SICK LMS400 laser device and Basler Scout stereo cameras.[1] Given that both the laser and the arm is very fast and accurate, dense scans of tables can be made in under a second. To facilitate the assembly of a large database of object models, we have created a rotating table using a DP PTU47 pan-tilt unit that is controlled by the robot over the network. Objects placed on this rotating table are scanned and geometric and appearance models are created for them automatically using supervised learning techniques. The resultant database of object models is then used to categorize and classify objects found in natural table setting scenes while performing manipulation tasks. Our approach is built as part of the Robot Operating System (ROS)[2] open source initiative, and makes use of modular and robust components that can be reused on other robotic systems different than ours.

The general architecture of our framework together with the geometric and processing pipelines is illustrated in Figure 3. For a better understanding of the underlying blocks and the connections between them, the overall system is divided into three major components, namely: acquisition, geometric processing, and appearance processing. The hardware devices that are used by our system are shown in orange boxes, the processing steps are depicted with rounded blue boxes, and their outputs are represented as yellow dotted boxes.

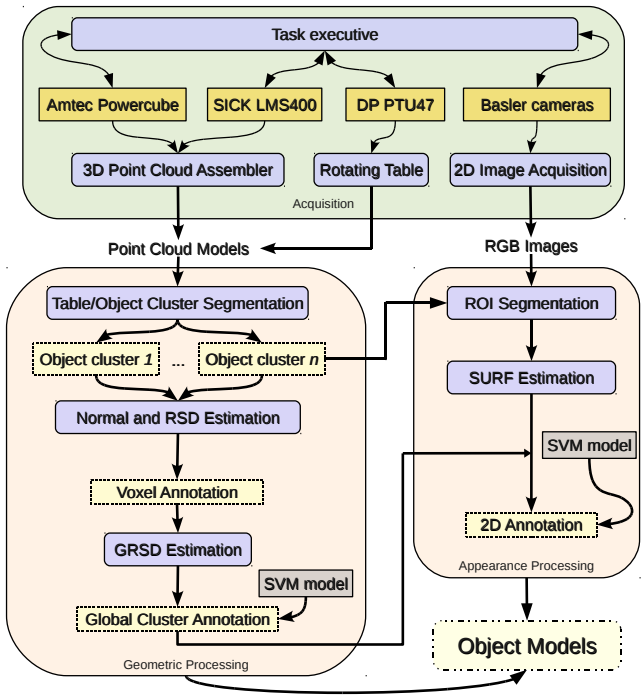


Fig. 3. The architecture of our system, with the major processing steps highlighted with blue color. The outputs are represented as yellow dotted boxes. The three main areas represent the acquisition (green), the geometric processing (yellow, left) and the appearance processing (yellow, right). Note that the *Rotating Table* was used only for acquiring the training data.

The acquisition component is responsible for acquiring the 3D depth data and the 2D RGB images that are used by the processing components. A task executive running on the robot controls the four hardware devices used, triggers the assembly of 3D point clouds, and takes image snapshots.

The point clouds are then processed through a series of geometric reasoning steps including the segmentation of the supporting table planes, the clustering of object candidates into independent data structures, gross outlier removal, and normal estimation [16]. Using the RSD descriptors [1], we describe the underlying surface geometry at every surface unit as plane, cylinder, edge, rim and sphere, as detailed in Section V-A. In a next step, for each cluster, a GRSD descriptor is computed using the previously acquired voxel labels, and an SVM model is used to categorize clusters into following categories: 1) bowl, 2-3) medium and small box, 4-6) tall, short and small cylinder, 7-8) big and small flat boxes, 9) pan, 10) plate and 11) tetrapak. The output of the geometric processing component is a set of annotated object clusters, categorized into classes which give hints with regards to their geometric structure.

For each object cluster obtained, a Region Of Interest (ROI) segmentation step is applied on the acquired camera images, in order to obtain a smaller, cropped image that represents that particular object. Note that for this step to be successful, we employ a precise camera to laser calibration step offline based on [17]. Then, for each image patch

[1]Please note that the TOF camera located on the robot head was not used for the experiments presented in this paper.

[2]http://www.ros.org

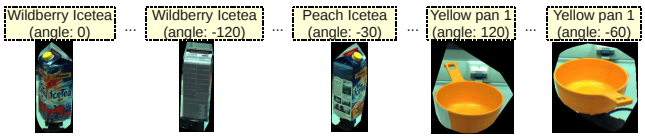


Fig. 4. Example object images for different objects from the database.

representing the object of interest we estimate SURF features for points of interest in the image, resulting in a vector of n features. Based on the results obtained from the geometric global cluster annotation, the SURF feature vector is tested against a subset of images in the database that could possibly represent appearance models for it.

IV. Object Database Acquisition and Qualification

Our database of 3D objects, available at `http://semantic-3d.cs.tum.edu`, was obtained using the hardware devices mentioned in Section I. The images in the dataset have been acquired using Basler Scout scA1390 stereo cameras at a resolution of 1390x1038 pixels. The 3D depth data was recorded using a SICK LMS400 range scanner with $0.5°$ angular resolution, resulting in point clouds with roughly 200-1000 points per object after the gross statistical removal procedure. The range scanner was tilted with $30rad/s$ during one scanning cycle.

The set of objects encompasses the ones commonly used in a typical household environment (mugs, utensils, books, etc) and is envisioned for a larger expansion in the future. In a pursue to account for a wide variety of view angles, we rotated the objects on the rotating table with a given angle-step ($30°$ in the preliminary version) and acquired partial snapshots from a human-eye perspective, i.e. the ones that the best approximate the robot's view point during its working cycle. We consider this to be an important point as opposed to similar initiatives (e.g., [14]) where the datasets are acquired using high-precision but non-affordable, fixed sensors, and thus not usable for applications such as ours.

V. Geometric and Appearance Processing

The input of the geometric processing step is a set of partial 3D point clouds $\mathcal{P}$, acquired from the tilting LMS400 laser sensor installed on the Powercube arm.

The robot then proceeds at extracting supporting horizontal planes from $\mathcal{P}$, with our assumption being that the robot is already close or in the vicinity of a table. If this is not already the case, in a more general sense, we make use of global planar segmentation techniques such as the ones we proposed in [18], and drive the robot close to a table on which a search for objects needs to be performed. After a large horizontal plane has been identified, all point clusters supported by it are extracted and a set of possible object candidates $\mathcal{O} = \{\mathsf{o}_1 \cdots \mathsf{o}_n\}$ is created.[3]

Because the laser sensor used for the purpose of the experiments in this paper is a general purpose laser measurement system and not made especially for the precise acquisition of 3D object models, the resultant data contains sparse outliers and noise. Filtering these out thus becomes an important step which dictates the quality of the final models. However, since the point clusters o_i are located at different distances from the viewpoint origin v, the density of the data varies, from dense for clusters near v, to very sparse for clusters located far away from it. Applying statistical techniques such as the ones we previously proposed in [16] on the entire dataset $\mathcal{P}$ thus makes little sense, as the mean density of $\mathcal{P}$ will affect the point clusters located at large distances from v. Instead, we apply a statistical gross outlier removal procedure on each separate point cluster o_i in parallel, thus decoupling the relationship between the point density of o_i and v, or in other words enabling the filtering step to be viewpoint and density independent.

Additional smoothing is performed by displacing the points to lie on the regression plane of their local neighborhood, as this approach is much faster than a full-blown surface estimation (by for example using MLS), yet it reduces the "thickness" of scanned surfaces considerably.

A. Voxel Annotation

To speed up the local surface classification, we label the surface units needed for the global classification (see next subsection) directly, instead of taking the dominant point-based label for each voxel. This reduces the complexity proportionally to the average number of points in a voxel. Also, the neighborhood can be directly constructed using the points in the current and surrounding voxels.

Once the neighborhood is constructed, we compute the RSD features, i.e. the radius of the highest and lowest curvature in the local neighborhood, as described in [1]. As a short overview, from the distribution of normal angles by distance we take the minimum and maximum normal variations by distance, and solve the equation:

$$d_{(\alpha)} = \sqrt{2}r\sqrt{1 - cos(\alpha)} \tag{1}$$

We can assume $d = r\alpha$ as $\alpha \in [0, \pi/2]$ and the Taylor decomposition of equation 1 is:

$$d_{(\alpha)} = r\alpha + \frac{r\alpha^3}{24} + O(\alpha^5) \tag{2}$$

which greatly reduces the problem of finding r_{min} and r_{max}.

Since these values have physical meaning, we can categorize surfaces using simple, intuitive rules, into: planes (large r_{min}), cylinders (medium r_{min}, large r_{max}), edges (small r_{min} and r_{max}), rims (small r_{min}, medium to large r_{max}), and spheres (similar r_{min} and r_{max}). Figure 5 and the top right part of Figure 1 show annotated the surface types.

B. Object Categorization

Once all voxels are annotated locally using a geometric class, our processing pipeline constructs a global feature space that can produce a unique signature for each object cluster. This space is based on the idea that, for a set of labeled voxels, a global feature can be constructed by observing the relationships between all these local labels

[3]The implementation details of these steps have already been described in [18] and fall outside the scope of this paper.

(and the encapsulated free space). Since the labels represent geometric classes obtained from the classification of RSD descriptors, we call this new feature the Global Radius-based Surface Descriptor (GRSD).

The computation of GRSD is similar to GFPFH [2], with the exception that we sum up the individual $\mathcal{H}_{f_{ij}}$ histograms instead of computing their distribution, to further reduce computational complexity. This way, the complete processing of a cluster (correcting, estimating normals, computing the voxelized RSD values, labeling voxels and constructing the GRSD) takes between 0.3 and 0.7 seconds (depending on object size) on a single core (using voxel size of $1.5cm$ as in the presented examples).

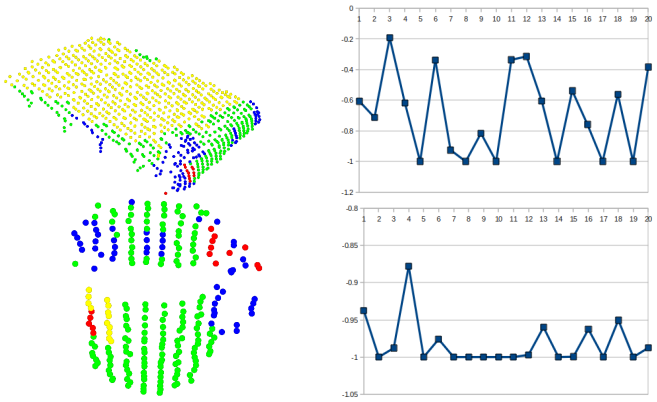


Fig. 5. Example of RSD classes and GRSD plots for a big flat box (i.e. book, upper row) and a short cylinder (i.e. mug, bottom row). The histogram bin values are scaled between -1 and 1 according to the training data, and the colors represent the following local surfaces: red - sharp edge (or noise), yellow - plane, green - cylinder, light blue - sphere (not present), and dark blue - rim (i.e. boundary, transition between surfaces). *Best viewed in color.*

Figure 5 shows two sets of histograms of different objects generated by the GRSD estimation. We then selected 11 overall categories, namely: $bowl$, box_medium, box_small, $cylinder\_big$, $cylinder\_short$, $cylinder\_small$, $flat\_big\_box$, $flat\_small\_box$, pan, $plate$ and $tetrapak$. Although the classes were picked by hand, they match general geometric categories of objects we encountered, and are intuitive as to guide the modeling pipeline [1] into picking the right reconstruction for successful grasping. An SVM model is then trained using the global histograms and the predefined classes.

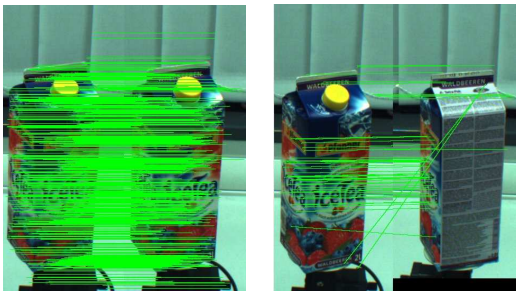
C. Visual Feature Classification

The visual feature detection is carried out only on the region of interest (ROI) in the image, in order to avoid false positive matches with the background. To obtain the ROI, we use the 3D boundary points [16] detected in the point cloud cluster, which we then project onto the image. The convex hull of these projected points is calculated and used as boundary to cut off the background.

For each image, we extracted the ROIs representing the objects of interest and computed a vector of SURF features. The next step is to quantize these features of each object view, that is, cluster them into a Bag of Features using standard K-Means techniques. This step is needed in order to obtain the constant bag size necessary for an SVM classifier and can be at best thought of as a histogram with the number of features in each cluster represented as tabulated frequencies. The classification was again performed using an SVM classifier, with an RBF Laplacian kernel:

$$K_{RBF}(x,y) = e^{-\gamma \cdot dist(x,y)}, dist = \sum_i |x_i - y_i| \qquad (3)$$

and the model is used to identify a test object and its visible side. Figure 6 presents two simple examples of matching SURF features for a scene containing an iced tea box with the same orientation (left), and different orientation (right).



(a) identical rotation (b) rotation of 180° to left

Fig. 6. Example of matching SURF features for an iced tea object.

VI. Discussions and Experimental Results

To validate our proposed framework, we have performed several experiments on geometric, as well as appearance-based data, using the processing pipelines presented in Section V. Overall, we have gathered around 400 datasets, i.e. partial views of 29 objects from the database (shown in Figure 7). We used the data taken using the turntable for training, and additional views of table scenes for testing (examples in Figure 8).

Fig. 7. Left: the database's objects. Right: some novel objects for test iii.

To evaluate the overall performance of our approach we carried out three types of object recognition test: i) test with the training dataset (around 300 views), ii) test with 76 views of the objects in regular table settings, and iii) test with 15 previously unseen objects/views on the table (geometric classification alone). While the best accuracy (95.45%, see bottom of Figure 8) was obtained in the first case, the accuracies for the remaining tests (85.53% and 80% respectively) were still encouraging. As the results of test ii suggest, the geometric category is informative even if texture based methods fail, but improved results could be obtained if the categories would be learned automatically, and the results evaluated together with visual matching and other features (e.g. color) in a probabilistic framework.

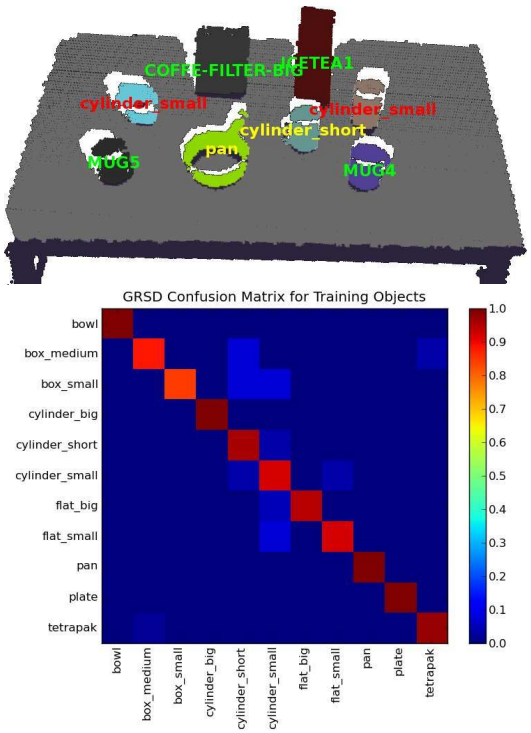


Fig. 8. The table scene at the top shows classification results of test ii. Red labels denote geometric miss-classification, yellow ones correct geometric category but incorrect visual classification, while green labels signal the correct object identification. The confusion matrix in he bottom provides the true and false positive normalized statistics for the test with training objects. *Best viewed in color.*

Since we are acting in a real world scenario, a fast overall performance is of a major importance for us. Our whole processing pipeline, including point cloud and image acquisition takes less than two seconds ($\approx 1s$ for acquisition) until an object gets classified.

Visual appearance (SURF) classification was tested on a set of arbitrarily rotated objects from the training set and we were able to re-detect and thus infer the orientation for rich-textured objects (e.g. iced tea, books), with an accuracy of 97.6%. As expected, the classification failed on texture-less, single-color objects.

The overall advantage of our scheme is that in situations where the class of the object cannot be correctly inferred from visual features, the geometrical categorization still provides a rough description of the object.

VII. Conclusions

In this paper we presented a comprehensive object categorization and classification architecture for objects of everyday use. Our system uses a two layer classification scheme, by first annotating point clusters with categories based on their geometry such as: boxes, pans, plates, etc, and then using appearance based visual features to obtain the exact object and its orientation with respect to the object's upright axis. The two layers create a generative to discriminative perception system, useful for personal robots operating in indoor environments.

The angular resolution of 30° in the database images and the $1.5cm$ voxel width for the 3D data proved to be enough for dealing with the regular-sized objects we tested and the robot could manipulate. As the classification doesn't provide a 6DOF pose and geometric model, these will have to be created as presented in [1] for grasping applications. This process can also be aided by the geometric classification.

Acknowledgments: This work was supported by the DFG cluster of excellence CoTeSys (Cognition for Technical Systems) and by Willow Garage, Menlo Park, CA.

References

[1] Z.-C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz, "General 3D Modelling of Novel Objects from a Single View," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 18-22 2010, accepted for publication.

[2] R. B. Rusu, A. Holzbach, G. Bradski, and M. Beetz, "Detecting and segmenting objects for mobile manipulation," in *Proceedings of IEEE Workshop on Search in 3D and Video (S3DV), held in conjunction with the 12th IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, September 27 2009.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[5] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Learning Informative Point Classes for the Acquisition of Object Model Maps," in *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Hanoi, Vietnam, 2008.

[6] M. Brown and D. G. Lowe, "Unsupervised 3d object recognition and reconstruction in unordered datasets," Ottawa, Ontario, Canada, 2005, pp. 56–63.

[7] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *In CVPR*, Wisconsin, USA, 2003, pp. 264–271.

[8] A. C. Romea, D. Berenson, S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *IEEE International Conference on Robotics and Automation (ICRA '09)*, Kobe, Japan, May 2009.

[9] M. Ruhnke, B. Steder, G. Grisetti, and W. Burgard, "Unsupervised learning of 3d object models from partial views," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Kobe, Japan, 2009.

[10] K. Lai and D. Fox, "3D laser scan classification using web data and domain adaptation," in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[11] B. Steder, G. Grisetti, M. Van Loock, and W. Burgard, "Robust online model-based object detection from range images," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, 2009.

[12] M. Quigley, S. Batra, S. Gould, E. Klingbeil, Q. V. Le, A. Wellman, and A. Y. Ng, "High-accuracy 3D sensing for mobile manipulation: Improving object detection and door opening," in *ICRA*, Kobe, Japan, 2009.

[13] Z. Xue, A. Kasper, J. Zoellner, and R. Dillmann, "An automatic grasp planning system for service robots," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, Munich, Germany, June 2009.

[14] "Kit object models web database." [Online]. Available: http://i61p109.ira.uka.de/ObjectModelsWebUI/

[15] C. Goldfeder, M. Ciocarlie, H. Dang, and P. Allen, "The columbia grasp database," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, Kobe, Japan, May 2009, pp. 1710–1716.

[16] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3D Point Cloud Based Object Maps for Household Environments," *Robotics and Autonomous Systems Journal (Special Issue on Semantic Knowledge)*, 2008.

[17] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 5, pp. 698–700, 1987.

[18] R. B. Rusu, I. A. Sucan, B. Gerkey, S. Chitta, M. Beetz, and L. E. Kavraki, "Real-time Perception-Guided Motion Planning for a Personal Robot," in *Proceedings of the 22nd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, October 11-15 2009.