Fast and Robust Object Detection in Household Environments Using Vocabulary Trees with SIFT Descriptors

Dejan Pangercic, Vladimir Haltakov, Michael Beetz
{pangercic, haltakov, beetz}@cs.tum.edu
Technische Universität München, 85748 Munich, Germany

***Abstract*— In this paper we describe the *ODUfinder*, a novel perception system for autonomous service robots acting in human living environments. The perception system enables robots to detect and recognize large sets of textured objects of daily use. Efficiency, robustness, and a high detection rate are achieved through the combination of modern text retrieval methods that are successfully used for indexing huge sets of web pages and state-of-the-art robot vision methods for object recognition. The result is a robot object detection and recognition system that, with an accuracy rate of more than 80%, can recognize thousands of objects by learning and using vocabulary trees of SIFT descriptors.**

I. Introduction and Approach

A robot acting as an household assistant must be capable of recognizing the hundreds of objects of daily use that are present in its operating environment. It also has to be able to recognize new objects, for example, when emptying a shopping basket to put the purchased items where they belong. One way to equip robots with knowledge about the physical look of these various objects is to retrieve images of them from grocery webstores, such as `www.germandeli.com` (Germadeli), or image libraries, such as google images.

In this paper we report on the design and implementation of the Objects of Daily Use Finder (*ODUfinder*) perception system that can deal with some aspects of this challenge. The system can detect and recognize textured objects in typical kitchen scenes. The models for perceiving the objects to be detected and recognized can be acquired autonomously using the robot's camera as well as by loading large object catalogs such as the one by Germandeli into the system. In the system configuration described in this paper, the robot is equipped with an object model library containing approximately 3500 objects from Germandeli and more than 40 objects from the *Semantic3D* database[1]. The *ODUfinder* achieves an object detection rate of 10 FPS and recognizes objects reliably with an accuracy rate of over 80%. Object detection and recognition is fast enough so that it does not cause delays in the execution of the robot's tasks.

The *ODUfinder* system employs a state-of-the-art object perception technique Scale Invariant Feature (SIFT) [1] using a vocabulary tree [2], which we extend in two important ways. First, the comparison of object descriptions is done probabilistically instead of relying on the more error-prone original implementation with the accumulation of query sums. Second, the *ODUfinder* detects candidates for textured object parts by over-segmenting image regions and then combines the evidence of the detected candidate parts to infer the presence of the object. These extensions substantially increase the detection rate as well as the detection reliability, in particular in the case of partial obstruction and in certain lighting conditions like specular reflections on object parts. Another contribution is the mechanism realized to enable automatic acquisition of incomplete visual appearance templates, such as the ones from Germandeli. In a nutshell this paper provides the following main contributions:

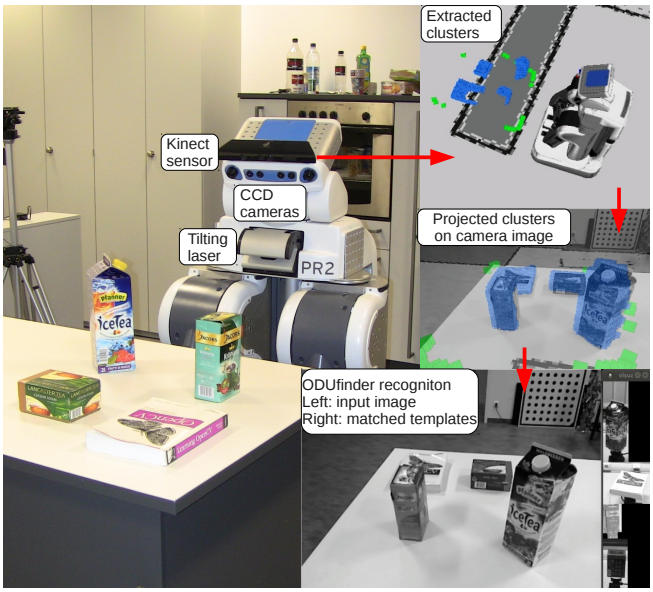

Fig. 1. TUM-PR2 robot recognizing objects lying on the tabletop using kinect sensor. Right column depicts extraction of clusters from point clouds (top), projection of clusters onto camera image and Region-Of-Interest extraction (middle) and, finally, *ODUfinder* recognizing objects (bottom).

- An application of a vocabulary tree matcher to real perception problems;
- A probabilistic comparison of objects' descriptors;
- An over-segmentation-based recognition of textured objects;
- A mechanism for automatic acquisition of incomplete visual appearance templates.

ODUfinder system is out-of-the-box and open-source

[1]http://ias.cs.tum.edu/download/semantic-3d

available as a ROS package [2] and can be easily deployed in any kind of robot equipped with a 3D sensor and a camera that are calibrated with respect to each other.

The remainder of this paper will proceed as follows: in the next section we discuss similar approaches, which is followed by a brief description of the system's architecture. SIFT based object recognition is explained in Section V, followed by Section VI focusing on the *ODUfinder's* capability to learn new objects. In Section VII we present the results of experiments and, finally, in the end we conclude and give suggestions for future research.

II. Related Work

Nakayama et al. [3] present the AI Goggles system, which is a wearable system capable of describing generic objects in the environment and of retrieving the memories of these objects by using visual information in real time without any external computation resources. The system is also capable of learning new objects or scenes taught by users. As the core of the system, a high-accuracy and high-speed image annotation and retrieval method supporting online learning are considered. The authors use color higher-order local auto-correlation (Color-HLAC) features and the Canonical Correlation Analysis (CCA) algorithm in order to learn the latent variables.

Arbeiter et al. [4] implemented a framework for 3D perception and modeling. The proposed algorithm can be used to reconstruct a 3D environment or learn models for object recognition on a mobile robot. Both color and time-of-flight cameras are used, and 2D features are extracted from color images and linked to 3D coordinates. Those coordinates then serve as input for a modified fastSLAM algorithm that is capable of rendering environment maps or object models.

A self-referenced 3D modeler is presented in [5] by Strobl et al., where the authors demonstrate that an ego-motion algorithm can simultaneously track natural, distinctive features and provide 3-D modeling of the scene. The use of stereo vision, an inertial measurement unit and robust cost functions for pose estimation further increased system's performance.

Incremental learning and recognition of objects is done in an unsupervised manner in [6], but Triebel et al. focus mainly on chairs, and it is not clear how well multiple objects could be reliably detected without any prior information. Moreover, scalability is hard to assess since only one view is analyzed at a time.

III. System Overview

The *ODUfinder's* mode of operation is depicted in Figure 2. The robot simultaneously takes a 3D scan and captures an image of the scene in front of it. The robot generates object hypotheses by detecting candidate point clusters in the 3D point cloud acquired by the depth sensor. These object hypotheses are then back-projected into the captured image as regions of interest and searched for detecting and recognizing objects (See section IV-A).

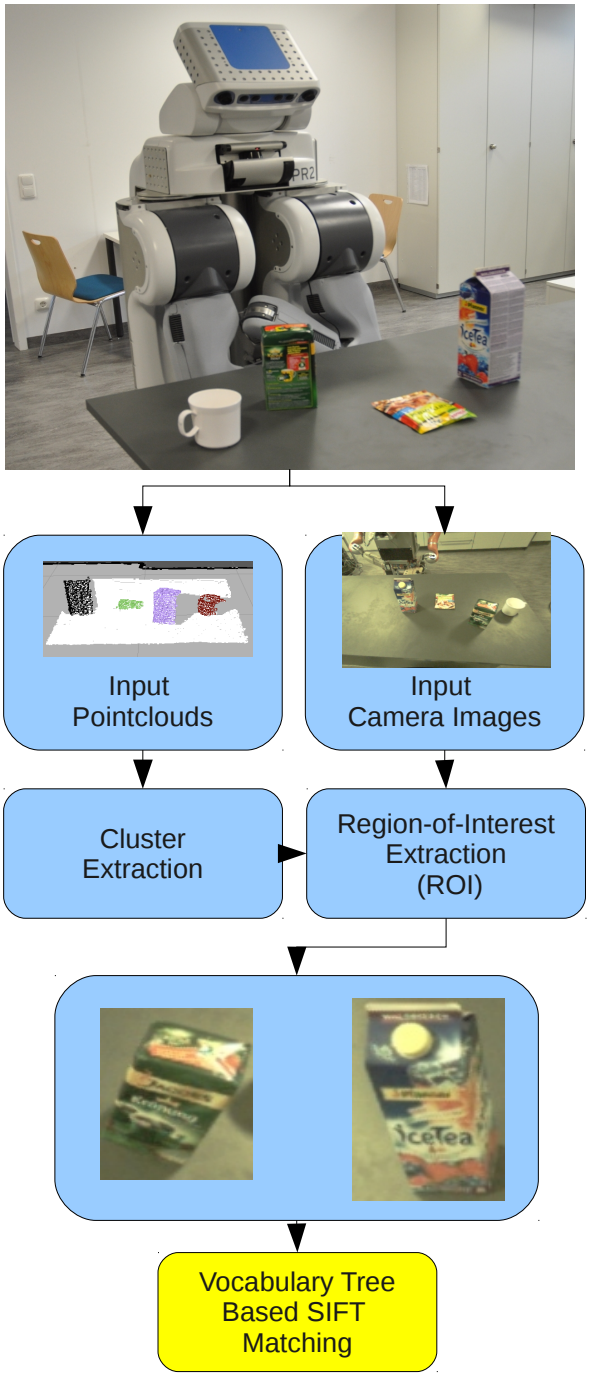

Fig. 2. System overview

For the SIFT-based detection we first determine segments by performing region growing on detected features in image space, which typically results in an over-segmentation of the region of interest (shown in Figure 3, right). Identifying the object and the image region it belongs to is then performed through methods transferred from document retrieval. In document retrieval tasks, for example in a web searches we

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

look for the documents that best match a given query term. To do so the search engines compute frequency statistics for discriminative words or better word stems as a pre-processing step performed on all documents. Given a search term, fast indexing mechanisms quickly search for the documents that are, with respect to frequency, particularly relevant for the search term. The application of text retrieval technology for object matching is promising because it is very mature and the techniques allow for rapid functioning with high recall and precision rates.

The computational idea of textual document retrieval can be mapped to object description matching in the following way: the descriptors computed from the regions of interest that belong presumably to (or are partial views of) the same object are considered to be the search term. The object descriptors of the different views of the relevant objects are the documents of the document retrieval model. Word frequencies are replaced by the frequency of visual object features. Given a large set of objects represented by their object descriptors and the feature descriptor of an image region, we can then index the objects where the particular features are particularly prominent using the respective methods of document retrieval.

In this paper we apply vocabulary trees for TF-IDF (Term Frequency Inverse Document Frequency [7]) indexing, a method used in document retrieval to find documents that best fit a given textual user query. In this reformulation of object identification, vocabulary trees speed up the retrieval of the matching objects.

The methods for object descriptor matching do not only match a given region descriptor to the large set of object descriptors, they also learn new object descriptors to be put into the visual object library.

The subsequent sections describe the individual computational steps performed by the *ODUfinder* in greater detail.

IV. Theory of Region of Interest Extraction

In human living environments the objects of daily use are typically standing on horizontal, planar surfaces or, as physics-based image interpretation states it, they must be in stable force-dynamic states. The scenes they are part of can be cluttered or the objects are more or less isolated. To account for these conditions, the *ODUfinder* employs two alternative methods for region extraction: first, the combined 2D-3D extraction for objects standing more or less isolated on planar surfaces, and, second, the region-growing based extraction for cluttered scenes, such as the objects standing in a cupboard. These two methods are described below.

A. Combined 2D-3D Object Candidate Detection

The combined 2D-3D object detection takes a 3D point cloud acquired through a tilting laser scanner or a kinect sensor and a camera image of the same scene. Figure 3 (left) shows how the *ODUfinder* detects major horizontal, planar surfaces within the point cloud and point clusters that are supported by the planes [3]. The identified point clusters in the point cloud are then back-projected into the captured image to form the region of interest that corresponds to the object candidate. An accurate back-projection is possible thanks to the accurate robot calibration, as described by Pradeep et al. [9]. The sensors are calibrated using a non-linear bundle-adjustment-like optimization to estimate various parameters of the TUM-PR2 Robot.

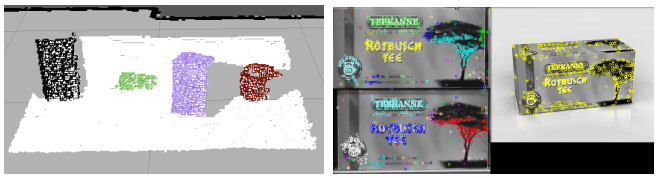

Fig. 3. Left: Region of Interest Extraction using back projection of 3D points, Right: Over-segmentation using a region-growing based approach.

B. Over-Segmentation-Based Object Candidate Detection

The second method for identifying image regions that might correspond to objects is the computation of clusters of visually distinctive pixels in the image space. This method exploits the fact that many objects of daily use have distinct textures.

In our case we determine the visually distinctive pixels using SIFT features and apply region growing algorithms to determine the clusters. Region growing starts from a point that does not belong to any clusters and incrementally adds points that are in a predefined radius *r* around the original point. The process is repeated for all newly added points. This results in clusters that represent the strongest texture "islands" in the image.

For our application, the quality of the segmentation results heavily depends on the appropriate setting of the radius parameter *r*. In order to improve performance, we adaptively chose the radius length in relation to the level of texturedness of the camera image using a scaled and shifted logistic sigmoid function:

$$r^2(x) = (r^2_{max} - r^2_{min})(K(1 - logsig(x - A))) + r^2_{min} \quad (1)$$

where logsig is defined as:

$$logsig(x) = \frac{1}{1 + e^{-x}}, \quad (2)$$

which tends to work well for input images of the same size.

In the equations above the argument x is the number of keypoints in the image. The parameters r_{min} and r_{max} denote the maximum and the minimum values of the radius. The parameter A denotes the value of x, where the value of the function is the average of the minimum and maximum value of the radius. The constant K denotes the speed at which the function approaches its minimum and maximum values. These 4 parameters are determined empirically and

[3]The implementation details of these steps have already been described in [8] and fall outside the scope of this paper.

are valid for images of roughly similar sizes. In the experiments below we use the following values: A = 800, K = 0.02, r_{min} = 200, r_{max} = 600.

This approach allows for bigger distances in images containing fewer features, thereby forming better shaped clusters and, respectively, it allows for small radiuses for images with a lot of features, thereby avoiding the use of extreme radius values.

V. Recognition of Textured Objects - Implementation

The *ODUfinder* performs object recognition of textured objects by computing the set of SIFT descriptors for all distinctive pixels in any given region of interest and then determines the object model in the library that best explains the set of SIFT descriptors of the region of interest. Each object view contains the set of SIFT descriptors of the distinctive pixels.

Unfortunately, comparing a region of interest with every object view in the object model library is prohibitively expensive. To this end, as proposed by Sivic and Zisserman [10], we consider object recognition as a document retrieval problem, which enables us to use fast data structures and retrieval algorithms and apply them to object recognition problems for large libraries of object models.

In this paper we employ vocabulary trees that were developed by Nister and Stewenius [2] for retrieving similar images in very large image libraries. In this section we show how we have specialized this technique for the purpose of object recognition in the context of robot perception. Our principal aim was to improve the capability of the proposed method for identifying objects in real scenes, which required taking different lighting conditions, obstruction and clutter, and the uncertainty and noise associated with physical sensors acting in the real world, into consideration.

A. Vocabulary Tree

The vocabulary tree of branching factor K and depth L is a tree data structure where the nodes in the tree represent a set of SIFT descriptors. The root node of the vocabulary tree represents the SIFT descriptors of all views of all object models in the library. If a node *n* in the vocabulary tree represents the set of SIFT descriptors $\mathcal{N}$ then its children nodes represent the partitioning of $\mathcal{N}$ into *k* subsets represented by the children nodes $cn_1 \ldots cn_k$, where the SIFT descriptors within a children nodes are similar and the ones of different children nodes dissimilar.

Thus, by taking a SIFT descriptor *sd* and classifying it hierarchically through the vocabulary tree using the defined distance measure on the SIFT descriptors we quickly find the set of SIFT descriptors that are most similar in the object model database as the leaf nodes, whose representative SIFT descriptors have the smallest distances to *sd*. For efficiency, *sd* is not compared to all features in a given node, but to the centroid of its features.

The SIFT descriptors in the vocabulary tree also have a reference to the object model in which they occur. Thus, when *sd* matches a leaf node it votes for the object models that the SIFT descriptors of the identified leaf belong to.

The children nodes $cn_1 \ldots cn_k$ of $\mathcal{N}$ are computed by applying k-means clustering to the SIFT descriptors of node *n*. Since the TF-IDF algorithm works on words (the equivalent of leaf nodes), we use a vocabulary tree to convert the keypoint descriptors into words, where each word is an integer value corresponding to the number of the leaf node.

B. Building the database

In our approach we use a similar database (object model library) as described in [2]. In order to be able to detect objects the database only stores the quantized SIFT features of the images, but not the images themselves.

1) Extracting SIFT features: In order to extract the visual SIFT features from the images we use an open-source implementation [11] of the standard SIFT algorithm as initially described by [1]. Each SIFT feature is characterized by a 128 dimensional descriptor vector, 2 image coordinates, a scale and an orientation value. In the current implementation we only use the descriptor vectors for the detection process and the image coordinates for visualization.

2) Generating database documents: After we have the vocabulary tree, we quantize feature descriptors to single words. For every image, we take all SIFT features, we quantize them with the vocabulary tree and we group the resulting words into one document for every image. In this way each document is composed of a list of all quantized features corresponding to a single image.

3) Populating and training the database: After generating all image documents, we insert them into a specialized database as proposed in [2]. The database is then trained with the TF-IDF [7] algorithm. After this training the database can be queried with documents generated from input camera images in order to find the best database matches between objects in the image and objects in the database. The database documents, along with specific database information, can be stored in a binary format in order to allow for fast loading of the database. Additional information, like image file names, textures and feature coordinates, is also saved for visualization purposes.

The whole detection process is implemented as a single ROS node, which receives an image coming from the camera and outputs the most probable N matches from the database.

C. Retrieving Object Models

In order to find an object in the received image we have to generate a database document in the same way as described above. We first extract the SIFT features from the received image and we quantize the descriptor vectors to words with the vocabulary tree. A single document is formed from all words of the input image and we can query the database with it. The database returns the best N matches with their respective scores (between 0 and 2, where 0 is best and 2 is worst).

This approach performs well so long as there is only one object in the image, i.e., if we were able to nicely segment

out clusters as described in Section IV-A. If two or more objects are visible in the input image, and especially if more than one of them is also loaded in the database, the performance decreases. This happens because the database retrieval mechanism tries to find an image containing all of the objects together and, although the objects can still be detected, their scores are low and very similar. This makes it very difficult to tell which match truly corresponds to the object in the image.

In order to improve recognition performance in such cases we thus present a power-horse idea of this paper, namely, a clustering of features of the input image in 2D space (the position of the feature in the image). In this way we can find rich-textured sub-regions in the object candidate image. It is difficult to make the clustering algorithms find the exact regions of the objects, but our experiments show, that this is indeed not necessary. If we adjust the clustering algorithm to over-segment, we get several clusters per object. These clusters correspond to the strongest textures of the objects and are, in most cases, enough to identify the whole object (see Figure 4).

Fig. 4. Detection of objects by partial textures. Left part shows that only a "Jacobs" sign is sufficient, while the right part implies the same for a "Kronung" sign.

The next step is to generate a document for every cluster size greater than the predefined size $S_{cluster}$ and query the database with those documents. Typical values for the $S_{cluster}$ are between 20 and 30, because smaller clusters are unlikely to produce meaningful results. Thus, every cluster has its own ranking of the most probable matches and we need to merge the results. In order to combine the results from every cluster into one final list of matches, we sum the scores ($clusters_{scores}$) which result from matching of every cluster against every image in the database. In this way, if several clusters vote with a high score for a specific image in the database, we understand that it is very likely that we have found the right object in the image. Note that if we had two objects in one input image, which also have respective entries in the database, then we will get more than two clusters from the input image (thanks to the over-segmentation) and the database retrieval mechanism will not search for the documents containing both objects, but rather only for parts of the objects, which will result in far more distinctive scores.

The final consensus is that, as our segmentation method tends to over-segment, the *ODUfinder* considers the image regions that could spatially lie on the same objects as multiple evidence for the respective objects and combines the evidences provided by the individual regions. Obviously the visual region-based object model appearance is particularly appropriate to handle partly obstructed objects and those which might have parts that cause reflections.

VI. Incremental Build-Up of Incomplete Models - Implementation

The *ODUfinder's* primary mode of operation provides basic functionality for online learning of new appearances of objects and mechanisms for storing and reloading them. We consider this feature to be very important for the continuous operation of service robots.

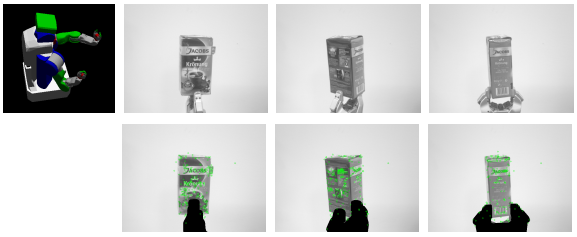

Fig. 5. **Top row:** Robot (left-most image) is manipulating an object in front of the camera **Bottom Row:** Extraction of keypoints and masking of robot's parts.

In this *ODUfinder's* mode of operation two cases may emerge: i) either the objects' appearances have been learned a priori and they only have to be located in the perceived scene, or ii) the robot encounters unknown objects (or unknown views of objects) and the new views have to be learned incrementally. While in the first case just a direct query for each appearance of the object candidate in the database is performed, in the second case we have to a) verify whether we have a partial template model of the object in question and, if so, b) the missing templates have to be acquired, features extracted and quantized with an existing vocabulary tree, and added to the existing database. In order to acquire missing object templates we implemented an in-hand object articulation and modelling process which is best explained through the following steps:

- classify object as unknown if the sum of clusters' scores $clusters_{scores} < 0.5$,
- calculate object grasp points on object's cluster [12],
- grasp the object, bring it in the frustum of the camera and set it upright,
- rotate the object around the up-right (z) axis to a viewpoint where you verify that it matches a template (from e.g. Germandeli),
- mask out parts of the robot and extract keypoints and region of interest,
- build documents from the keypoints, quantize them with the existing vocabulary tree and add them to the database,
- repeat above three steps until object has been rotated for $2\pi rad$ (note that our TUM-PR2 robot is equipped with the continuous revolute wrist joint). Also see Figure 5.

An important aspect of the learning of new models is that the vocabulary tree does not need to be computed again. The addition of a new document in a large database does not change the distribution of the words in the database substantially and therefore the existing quantization provided by the vocabulary tree is still adequate. Regeneration of the tree (and consequently of the database) is only needed if lots of new documents are added to a relatively small database, but this could be done later in an offline phase. A demonstration of the incremental build-up of incomplete models are available in the accompanying video submission [4] and Figure 5.

VII. Evaluation

A. Database Training

To evaluate our approach we have trained two vocabulary trees and built two databases with textured objects. In the first case we parsed the Germandeli website, downloaded product descriptions (semantic data, such classes of objects as well as appearances) and in the second case we generated a database out of the *Semantic3D* initiative which consists of over 40 household objects (see Figure 6) as described in [13]. The latter database was enriched with 10 more objects from the Germandeli website in order to demonstrate incremental build-up of additional models. While K,L parameters for the structure of vocabulary trees were 6,6 and 5,5 respectively, the rest of the properties of the databases are given in Table I.

	nr. images	**nr. features**	**training time**	**cluster query time**
Germandeli	3500	2500000	1h	90ms
Semantic3D	170	65000	1min	50ms

TABLE I

Technical data for the generated databases of objects.

Fig. 6. A subset of the collection of objects from Semantic3D database.

B. Recognition Results with Over-Segmentation

Used test images were taken with the hand-held camera in a German grocery store and encompass a wide variety of grocery products. We carried out recognition tests against the Germandeli database and present and discuss the results in Figure 7. We show the segmentation in feature space, where the circles denote SIFT keypoints and adjacent points with the same color belonging to the same cluster. The left part of every box in the Figure is the image received from the camera and the right image is the first match from the database.

In the first three rows of Figure 7 we see examples of the detection of 3 different objects. In only one case is the correct image not the first match found (row 3, column 3) and we attribute this to the test image's lack of the resolution.

The fourth row presents an interesting case. The camera image contains a strawberry juice, but the best match in the database is a juice with similar packaging, but of a different flavor. If we take a look in the top 10 matches for the test images in this row, we see that the first 6 matches are juices of the same make with very similar packaging, which differ only in respect to the small text in the middle of the package and in the flavor drawing on the bottom. This case is especially difficult because the strong texture from the Rauch and Happy Day logos and the upper part of the packaging are identical in all templates. This is why the correct flavor is not always first place, but in the top 5 matches. Thus, our system can find the right class of an object, in this case Rauch juice, but fails to find the right instance (in this case, flavor).

Fig. 7. Evaluation of recognition of objects found in German supermarkets with over-segmentation

C. Recognition Results with Combined 2D-3D Object Candidate Detection

We ran this test in our kitchen laboratory (see left column of Figure 8). The test was carried out against the SemanticDB database on a total number of 12 objects located at 4 different scenes (denoted with Scene 1 ... Scene 4 and depicted in top-down order in the right column of Figure 8). The robot was programmed to navigate to each of the scenes and capture

[4] http://youtu.be/Hjwj0YN2z5w

point clouds and images from several different views by traversing along the free paths around the scenes. The partial and total results of the evaluation are given in Table II.

Scene	#Views	#Failures	Success Rate [%]
Scene 1	52	10	80.7
Scene 2	11	5	54.5
Scene 3	24	2	91.6
Scene 4	12	0	100
Total	99	17	82.8

TABLE II

RECOGNITION OF OBJECTS USING SIFT WITH VOCABULARY TREES FROM COMBINED 2D-3D OBJECT CANDIDATE DETECTION.

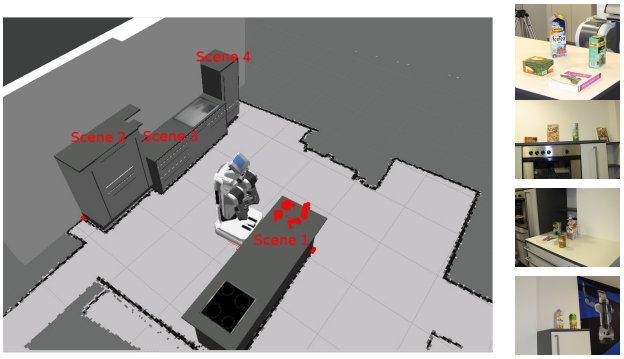

Fig. 8. Left column: We performed the final evaluation test on a total number of 12 objects located at 4 different scenes in our kitchen lab (denoted with Scene 1 ... Scene 4 and depicted in top-down order in the right column). Left column: The robot was programmed to navigate to each of the scenes and capture point clouds and images from several different views by traversing along the free paths around the scenes. Results of this test are presented in the Table II.

1) Novel Object Case: To demonstrate the capability of our system to acquire new object models on the fly we set up Scene 1 with 1 unknown object (green coffee box), which generated all 10 false positive measurements reported in the first row of Table II. Since setting the score value of the database retrieval mechanism to the experimentally determined value of *0.5* enables us to classify all measurements that exceed this value as unknown, we can introduce image templates generating this score as new object models. The assumption we are making here is that the scene remains static and that the image templates have consistent association with the cluster cloud with fixed 3D position in the world coordinate frame.

Scene	#Views	#Failures	Success Rate [%]
Scene 1	52	2	96.0

TABLE III

IMPROVED RECOGNITION RATE FOR SCENE 1 FROM FIGURE 8 AFTER THE FEATURES FOR GREEN COFFEE BOX WERE ADDED TO THE DATABASE.

After this we performed another test run on Scene 1 with the the updated database of SIFT descriptors and were able to reduce the number of false positives down to 2, as shown in Table III. Please also refer to the accompanying video submission.

VIII. CONCLUSIONS AND FUTURE WORK

This paper has presented a perception system for autonomous service robots acting in human living environments, coined the *ODUfinder*. The perception system enables robots to detect and recognize textured objects of daily use, it ensures real-time and robust operation and is modular with respect to the integration of new components (e.g. detection of texture-less or translucent objects). On the theoretic part, we consider an over-segmentation of image regions and the combination of the evidences of the detected candidate parts to infer the presence of the object, to be a major contribution herein.

In the future we plan to improve the segmentation of cluttered scenes using graph-based methods [14] and interactive perception approaches [15]. Furthermore, we plan to include more recognition routines (e.g. Dominant Orientation Templates [16], Transparent Object Detection [17]) and thus convert the *ODUfinder* into a bag-of-experts system. En route to ensure autonomous, continuous operation of the robot over large spans of time, we plan to look into i) sharing of model libraries between different robots and ii) inferring of semantic types of objects using barcodes.

IX. ACKNOWLEDGMENTS

We would like to thank our colleagues Thomas Rühr and Monica Simona Opris for their valuable support with our experiments. This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems – CoTeSys*, see also `www.cotesys.org`.

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2161–2168.

[3] H. Nakayama, T. Harada, and Y. Kuniyoshi, "Ai goggles: Real-time description and retrieval in the real world with online learning," *Computer and Robot Vision, Canadian Conference*, vol. 0, pp. 184–191, 2009.

[4] J. F. Georg Arbeiter and A. Verl, "3d perception and modeling for manipulation on care-o-bot 3," in *In Proceedings of the ICRA 2010 Workshop: Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, USA, 2010.

[5] K. H. Strobl, E. Mair, T. Bodenmüller, S. Kielhöfer, W. Sepp, M. Suppa, D. Burschka, and G. Hirzinger, "The Self-Referenced DLR 3D-Modeler," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, October 2009, pp. 21–28, best paper finalist.

[6] R. Triebel, J. Shin, and R. Siegwart, "Segmentation and unsupervised part-based discovery of repetitive objects," in *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[7] S. Robertson, "Understanding inverse document frequency: On theoretical arguments for idf," *Journal of Documentation*, vol. 60, p. 2004, 2004.

[8] 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 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, October 11-15 2009, pp. 4245–4252.

[9] V. Pradeep, K. Konolige, and E. Berger, "Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach," in *International Symposium on Experimental Robotics (ISER)*, New Delhi, India, 12/2010 2010.

[10] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the International Conference on Computer Vision*, vol. 2, Oct. 2003, pp. 1470–1477. [Online]. Available: http://www.robots.ox.ac.uk/~vgg

[11] "Fast sift image features library." [Online]. Available: http://sourceforge.net/projects/libsift/

[12] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Sucan, "Towards reliable grasping and manipulation in household environments," in *Proceedings of RSS 2010 Workshop on Strategies and Evaluation for Mobile Manipulation in Household Environments*, 2010.

[13] Z.-C. Marton, D. Pangercic, R. B. Rusu, A. Holzbach, and M. Beetz, "Hierarchical object geometric categorization and appearance classification for mobile manipulation," in *Proceedings of 2010 IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, USA, December 6-8 2010.

[14] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[15] D. Katz and O. Brock, "Manipulating articulated objects with interactive perception," in *IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena,USA, may 2008.

[16] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[17] U. Klank, D. Carton, and M. Beetz, "Transparent object detection and reconstruction on a mobile platform," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May, 9–13 2011.