In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipeh, Taiwan, October 2010

Combining Depth and Color Cues for Scale- and Viewpoint-Invariant Object Segmentation and Recognition using Random Forests

Jörg Stückler and Sven Behnke

***Abstract*— In this paper we present an approach to object segmentation and recognition that combines depth and color cues. We fuse information from color images with depth from a Time-of-Flight (ToF) camera to improve recognition performance under scale and viewpoint changes. Firstly, we use depth and local surface orientation extracted from the ToF image to normalize color and depth image features with regard to scale and viewpoint. Secondly, we incorporate local 3D shape features into the classifier. The use of a Random Forest classifier facilitates the seamless combination of depth and texture features. It also provides image segmentation through pixel-wise classification. We demonstrate our approach on a labelled dataset of seven object categories in table-top scenes and compare it with a vision-only approach.**

I. Introduction

In unconstrained, daily life environments, the segmentation and recognition of objects is an important yet difficult to achieve capability for a service robot. Much effort in computer vision has been devoted to this task over the last decades with tremendous progress. In this paper, we present an approach to object segmentation and recognition that combines depth information from a Time-of-Flight (ToF) camera with images acquired with a color camera. The availability of dense depth measurements enables us to normalize texture and depth features for scale and viewpoint changes.

We base our approach on discriminative Random Forest classifiers which have been introduced to the computer vision community by Lepetit et al. [1]. This kind of classifier has many properties which can be useful in robotics applications: Both, training and classification can be performed with high computational efficiency which facilitates real-time operation. Random forests output a probability distribution over multiple class categories for each pixel and thus solve object segmentation and recognition concurrently. They also allow to seamlessly integrate a variety of features like color, texture, and depth from heterogeneous sensor modalities.

In our approach, we combine color and depth cues to improve classifier performance. From color images, we determine simple appearance features at pixels and by binary comparisons between pixels. Complementarily, we extract local shape features from the depth image.

We not only fuse color and ToF images, but also normalize features for scale and viewpoint changes of the object towards the camera. This can be achieved by scaling and rotating relative query points with respect to the local surface orientation on the object. We determine the local surface orientation efficiently from the dense depth image.

All authors are with the Autonomous Intelligent Systems Group, University of Bonn, Germany. Email: stueckler@ais.uni-bonn.de

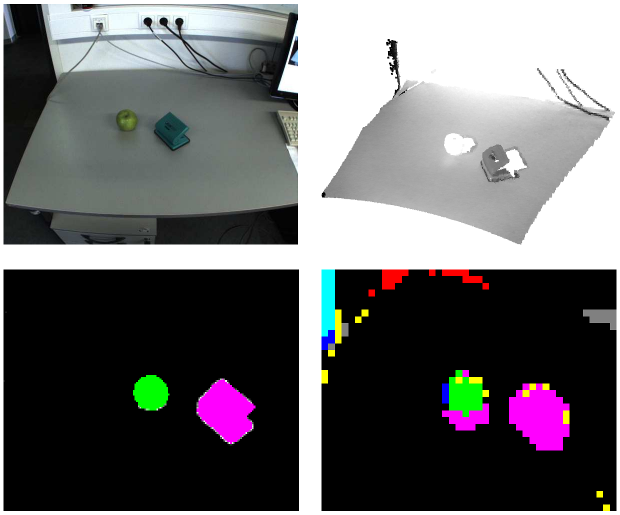

Fig. 1. Object segmentation and recognition with texture (top left) and depth (top right) cues. Depth is acquired with a Time-of-Flight camera. We apply Random Forest classification to segment the ToF image (176x144) pixel-wise with a subsampling factor of 2. The classifier outputs a probability distribution over class labels for each pixel. The segmentation shows the class label with maximum likelihood for each pixel (background: black, apple: green, puncher: magenta).

In experiments on a labelled dataset of seven object categories acquired in a table-top scene, we demonstrate that our approach outperforms the standard approach to Random Forest classification with unnormalized texture features.

This paper is organized as follows: In Sec. II we will review related work on object segmentation and recognition from color/texture and depth. Fundamentals and properties of Random Forest classifiers are detailed in Sec. III. We describe our main contribution, the combination of depth and color cues for object segmentation and recognition, in Sec. IV. Finally, we report experimental results in Sec. V.

II. Related Work

Image segmentation is a well-studied topic in computer vision. Early methods segment images by subsuming regions with similar brightness, color, and texture or by separating regions at discontinuities of such features [2]. However, the basic assumption underlying these approaches is that objects appear uniform in such features, which is typically not the case for images of real-world scenes.

For class-based segmentation of color images, many approaches have been developed (e.g., [3], [4], [5], [6]). Semantic Texton Forests [4] use simple features of luminance and color at single pixels or comparisons between two pixels in a Random Forest classifier. Using image-level priors and a second stage of Random Forests, local and scene context is incorporated into the classification framework. In [5] the basic Random Forest classifier is enhanced by further features such as Histograms of Oriented Gradients [7] and filterbanks. Spatial smoothness of the resulting segmentations is achieved using Conditional Random Fields (CRFs). Both approaches demonstrate state-of-the-art results on the MSRC [8] and VOC2007 [9] datasets. We extend the basic Random Forest classification approach in [4] by incorporating depth features and by normalizing features for scale and viewpoint changes.

Object and shape recognition in 3D point clouds has also been studied for some time in computer graphics and robotics. Wahl et al. [10] propose to represent 3D shapes by histograms of surflet-pair-relations, i.e. distance and orientation between points and corresponding local surface normals. In [11], such surflet-pair-relation histograms are used to describe the local neighborhood of points. The authors demonstrate that the proposed Point Feature Histograms (PFH) yield a persistent description of geometric shape primitives useful for segmentation. Fast Point Feature Histograms (FPFH), a fast approximation of PFH features, have been proposed in [12]. We combine FPFHs with texture features in our object recognition and segmentation framework.

Gould et al. [13] integrate range and vision sensing modalities for object detection of household objects. Features extracted from range data are used to focus the attention of image-based object detectors and, in this way, reduce computation. They furthermore combine image and 3D features in binary logistic classifiers to improve detection accuracy. Our approach seamlessly integrates vision and range information in an object segmentation and recognition framework.

III. Random Forests

Random Forests extend decision trees [14] to mitigate their shortcomings. Decision tree classifiers typically suffer from over-fitting. To overcome this problem, Random Forests combine the output of an ensemble of randomized decision trees. The randomness is incorporated into the selection of decision criteria during training. By this, Random Forests achieve lower generalization error than decision trees and comparable performance to SVMs on multi-class problems [15].

One major advantage of decision tree-based classifiers is their high computational efficiency. The computational load is mainly governed by the typically small depth and count of trees, and the feature extraction method. This property makes Random Forests ideally suited for real-time applications of object segmentation and recognition as often required in the robotics context.

A. Structure of Random Forests

A Random Forest $\mathcal{F}$ consists of K randomized decision trees $\mathcal{T}_k$. Each node n in a tree classifies an example by a binary decision on a scalar node function over features. In addition, each node is associated with a distribution $P(c|n)$ over class labels $c \in C$.

To determine the posterior distribution over class labels for an example, it is evaluated on each decision tree $\mathcal{T}_k$ in the ensemble. In this process, the example is passed down the tree, branching at each node according to its binary decision criterium until a leaf node l is reached. The posterior distribution is averaged over the individual distributions at the leaf nodes l_k the example reaches, i.e.

$$P(c|\mathcal{F}) = \frac{1}{K}\sum_{k=1}^{K} p(c|l_k, \mathcal{T}_k).$$

For classification, this posterior distribution is evaluated for each pixel in an image. Without further processing, the class label with maximum likelihood can be chosen to obtain an image segmentation into classes.

B. Learning Random Forests

Each randomized decision tree in the forest is trained independently. Starting from the root node, the training of a tree either proceeds depth first or breadth first by successively choosing binary decision criteria in a randomized manner. The trees are limited to a maximum depth.

To select the decision criterium of a node, only a random subset of the training data and the available node functions on feature values is presented. The training algorithm needs to determine the node function and a threshold on its value that separates the training examples best. Commonly, information gain is maximized for this purpose. The class distributions of the nodes are estimated from the empirical distribution given by all training examples.

We follow the approach of [4] and sample a distinct number of threshold values. From these thresholds we select the one with highest information gain. We also weigh each training example for a class label with the inverse class label frequency in the training dataset. This prevents the preference of the classifier to better separate classes with larger portions of the training set.

IV. Object Segmentation and Recognition from Depth and Color Cues

In our approach to object segmentation and recognition we combine two complementary sensor modalities. While a color camera provides detailed texture information on the viewed scene, a Time-of-Flight camera measures depth of the scene densely. We use the perceptually uniform CIELab color space in our implementation. Fig. 2 shows the sensor setup on the head of our domestic service robot Dynamaid [16].

In our Random Forest classification framework, features are computed from both types of images. In the depth image, we extract features that describe 3D shape locally. In addition, for each depth image pixel we determine local texture features through projection of the pixel's 3D point coordinate into the color image. The available depth enables us to normalize the texture features for scale and viewpoint. We assume that the rotation between object and camera only changes in pitch and yaw.

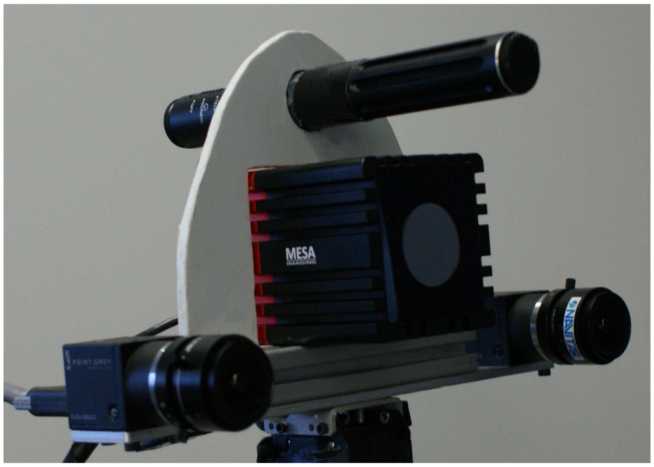

Fig. 2. Sensor setup used in our experiments. A MESA SR4000 camera and PointGrey Flea2 13S2C-C cameras acquire depth and color images. The sensor head is mounted on a pan-tilt unit.

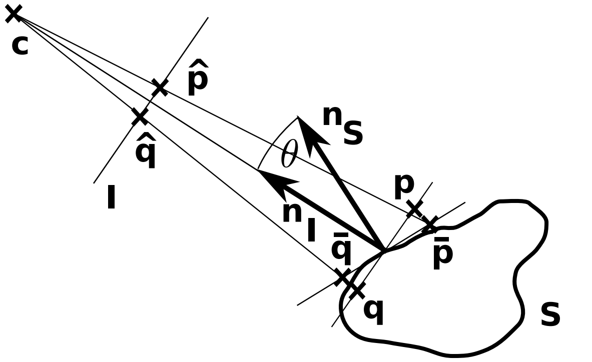

Fig. 3. Two-dimensional illustration of feature normalization. We normalize color and depth image features by rotating relative query positions p and q onto the local surface orientation. We use the shortest rotation from image plane normal n_I onto surface normal n_S. For depth features, we determine the nearest neighbors of the rotated query points $\overline{p}$ and $\overline{q}$. To determine texture features, the query points are projected onto the image plane I yielding pixel positions $\hat{p}$ and $\hat{q}$.

A. Sensor Data Preprocessing and Fusion

Time-of-Flight (ToF) cameras are compact, lightweight, solid-state sensors which measure depth to surfaces densely at a high frame rate and are therefore ideally suited for robotic applications. They employ an array of LEDs that illuminate the environment with modulated near-infrared light. The reflected light is received by a CCD/CMOS chip. Depth information is acquired by measuring the phase shift of the reflected light for every pixel in parallel. The use of ToF cameras has been studied in various fields of robotics. Main limitations of this sensor are its limited measurement range, measurement inaccuracies, limited unambiguity range, and its restricted field-of-view (FoV).

Measurements of ToF cameras are subject to several error sources [17]. From the image, we filter out measurements with low amplitude, as these indicate either highly noisy measurements of poorly reflecting objects or measurements of objects beyond the unambiguity range of the camera. Furthermore, we remove measurements at so-called jump-edges at object boundaries. They can be determined by examining local pixel neighborhoods. We detect jump-edges when two points approximately lie along the line-of-sight of the camera [18]. Since this procedure is sensitive to noise, we apply a median filter to the depth values beforehand.

To be able to fuse information from both cameras, we calibrate the cameras extrinsically similar to a stereo camera rig.

B. Scale and Viewpoint Normalization through Depth

The dense depth information acquired by the ToF camera enables us to estimate local surface properties. We exploit this to normalize texture and depth features for affine transformations of the objects under view.

At each ToF image pixel, we estimate the local surface normal n_S from 3D points in a local neighborhood of the pixel's 3D coordinate. The neighborhood is defined by a sphere with radius r. We determine the surface normal by the eigenvector of the 3D covariance of the neighboring points corresponding to the smallest eigenvalue. If necessary, we flip the extracted surface normal to point towards the viewpoint. The range query has been efficiently implemented through kd-trees.

In our approach, features are unary functions at or comparisons between pixels. In standard image processing approaches, values are extracted at relative pixel coordinates in a local image patch around a pixel. To compute scale and viewpoint invariant features, we rotate 3D query positions that are relative to the pixel's 3D coordinate from the image plane onto the local surface (cf. Fig. 3). We then determine nearest neighbors in the depth image or project the rotated query points into the color image.

We determine the rotation between surface and image plane by the shortest rotation from the image plane normal $n_I := (-1, 0, 0)^T$ onto the surface normal n_S. This is achieved by rotating with an angle θ along the axis v perpendicular to the image plane normal and the surface normal,

$$v := \frac{n_I \times n_S}{\|n_I \times n_S\|}.$$

From

$$R_{I \to S}(v, \theta) \cdot \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -\cos(\theta) \\ -v_z \cdot \sin(\theta) \\ v_y \cdot \sin(\theta) \end{pmatrix} = n \qquad (1)$$

we obtain $\theta = \operatorname{arctan2}\left(-\frac{n_y}{v_z}, -n_x\right)$. In our formulation, we assume that the object is upright with respect to the camera, i.e. no roll rotation occurs between object and camera.

C. Feature Types

We extract four kinds of texture and surface describing features from color and ToF depth images:

1) Texture: Features in the color image are simply computed from pixel values and comparisons at projected query points.

2) Local Surface Curvature: We use the second order measure of surface curvature as feature. From the eigenvalues $\Lambda = \{\lambda_i\}_{i=1}^{3}$ of the local 3D covariance at each pixel, we determine the curvature κ as

$$\kappa = \left| \frac{\min \Lambda}{\sum_{i=1}^{3} \lambda_i} \right| .$$

3) Moment Invariants: Three-dimensional moment invariants [19] are features of object surfaces that are invariant to rigid transformations. From the central moments in a local neighborhood $\mathcal{P}$

$$m_{ijk} = \sum_{p \in \mathcal{P}} \left(p_x - \mu_x\right)^i \left(p_y - \mu_y\right)^j \left(p_z - \mu_z\right)^k$$

the 3D moment invariants are determined by

$$\begin{aligned} I_1 =& m_{200} + m_{020} + m_{002} \\ I_2 =& m_{200}m_{020} + m_{020}m_{002} + m_{002}m_{200} \\ & - m_{011}^2 - m_{101}^2 - m_{110}^2 \\ I_3 =& m_{200}m_{020}m_{002} + 2m_{011}m_{101}m_{110} \\ & - m_{011}^2 m_{200} - m_{101}^2 m_{020} - m_{110}^2 m_{002}, \end{aligned}$$

where μ is the mean of the neighorhood $\mathcal{P}$.

4) Fast Point Feature Histograms: Recently, Fast Point Feature Histograms (FPFH) have been proposed as persistent 3D shape descriptors. The histograms are computed from 4-dimensional features determined from pairs of surflets, i.e. points p with associated local surface normals n.

For a pair of points p_i and p_j we extract surflet-pair-relation features in the following way: First we determine the source point p_s as the point with the smaller angle between its normal and the line between the points, i.e. if

$$\arccos\left(n_i \cdot (p_j - p_i)\right) \leq \arccos\left(n_j \cdot (p_i - p_j)\right),$$

the point p_i is chosen as source and p_j as target p_t. From the points and their normals we construct the Darboux frame with $u = n_s$, $v = (p_t - p_s) \times u$, and $w = u \times v$.

The four surflet-pair-relation features then describe the relative orientation and distance between the two surflets:

$$\begin{aligned} \alpha &= \operatorname{arctan2}\left(w \cdot n_2, u \cdot n_2\right), \\ \beta &= v \cdot n_2, \\ \gamma &= u \cdot \frac{(p_e - p_s)}{\|p_e - p_s\|}, \\ \delta &= \|p_e - p_s\| . \end{aligned}$$

We then compute so-called Simplified Point Feature Histograms (SPFH) over the angular surflet-pair-relation features between a point and its local neighbors in a specific range r. As proposed in [12], we neglect the distance feature δ. We bin each feature into K equally sized intervals of its value range.

The SPFHs are further compressed to Fast Point Feature Histograms (FPFH): At each point p, the FPFH is the weighted sum of the SPFHs in the point's local neighborhood $\mathcal{P}$

$$\mathrm{FPFH}(p) = \mathrm{SPFH}(p) + \frac{1}{|\mathcal{P}|} \sum_{q \in \mathcal{P}} \frac{1}{d(p,q)} \mathrm{SPFH}(q),$$

where $d(p,q)$ is a distance metric between points.

D. Node Functions

We use the above features in unary node functions or to compare shape or appearance between two local points at a pixel.

1) Unary Node Functions: As unary node functions we use value or absolute value of luminance, color, curvature, moment invariants, and the individual FPFH bin values at query positions relative to the pixel's 3D coordinate.

To determine feature values for the query positions in the depth image, we determine the nearest pixel in 3D coordinates. In the color images, depth is not available at every pixel. For this reason, we choose the relative query positions to reside in the local surface plane given by the pixel's normal. The query position is then projected into the color image to find its corresponding pixel.

It suffices to select relative positions and to rotate these positions onto the surface plane according to the image-plane-to-surface rotation $R_{I \rightarrow S}$ in (1). During training of the random forest, we randomly select relative query positions within a selection range r_{sel}.

2) Binary Node Functions: Binary node functions compare features at two relative query positions. Analoguously to the unary node function case, we determine the query positions either on a plane for texture features or in 3D for depth features. We use a variety of node functions over different types of depth and texture features:

- **Texture:** We compare query positions in the color image by the value or absolute value of addition, subtraction, and multiplication. By this, each path through the decision tree is able to generate patch features similar to derivative filter kernels [4].
- **Point Statistics:** Similar to binary node functions on color, node functions are calculated on curvature and moment invariants as value or absolute value of addition, subtraction, and multiplication.
- **FPFH Matching:** We also use the chi-squared distance

  $$\chi^2(p,q) = \sum_k \frac{\left(\mathrm{FPFH}(p)_k - \mathrm{FPFH}(q)_k\right)^2}{\left(\mathrm{FPFH}(p)_k + \mathrm{FPFH}(q)_k\right)}$$

  between the FPFHs at the query points p and q as binary node functions.

V. Experiments

We evaluate our approach with a dataset of seven object categories which we acquired in a table-top scene. The object categories comprise cups, apples, bins, books, computer mice, punchers, and staplers (cf. Fig. 4). Each category consists of four example objects that add intra-class variety in shape and appearance. In the object-view dataset (221 images), the objects are placed at the same spot and are rotated in 45° angle intervals around their yaw axis. In an object-mix dataset (32 images), we place two or three objects in random positions and orientations on the table.

We train the Random Forest classifier with standard unnormalized texture features in the image-plane (patchsize $16 \times$

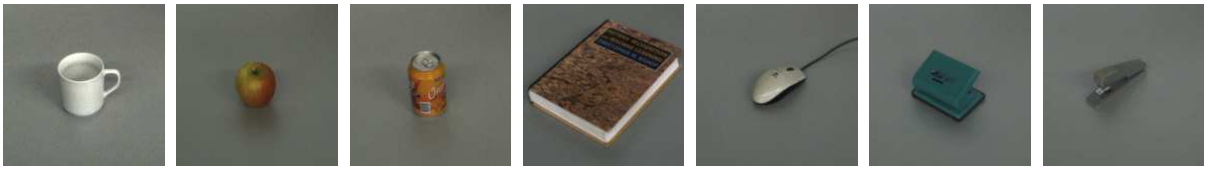

Fig. 4. Objects from the seven categories used in the experiments.

	with backgr.		w/o backgr.	
	glob.	avg.	glob.	avg.
standard texture	0.80	0.63	0.65	0.63
texture ($0.025m$)	0.81	0.59	0.62	0.57
texture ($0.1m$)	0.92	**0.75**	0.78	0.74
depth ($0.07m$)	0.90	0.51	**0.81**	**0.75**
texture, depth ($0.025m$, $0.07m$)	0.95	0.66	0.73	0.64
texture, depth ($0.1m$**,** $0.07m$**)**	**0.95**	**0.69**	**0.77**	**0.67**

TABLE I

Global and average accuracy of max-likelihood object segmentation on test images of the object-view dataset.

	with backgr.		w/o backgr.	
	glob.	avg.	glob.	avg.
standard texture	0.77	0.54	0.51	0.52
texture ($0.025m$)	0.80	0.50	0.46	0.46
texture ($0.1m$)	0.84	0.47	0.52	0.44
depth ($0.07m$)	0.81	0.45	**0.73**	**0.62**
texture, depth ($0.025m$**,** $0.07m$**)**	**0.89**	**0.57**	**0.64**	**0.60**
texture, depth ($0.1m$, $0.07m$)	0.86	0.42	0.52	0.37

TABLE II

Global and average accuracy of max-likelihood object segmentation on the object-mix dataset.

16) and several combinations and parameter settings for normalized texture and depth features. In experiments with the object-view dataset only, we split the dataset into test and training subsets. By this, the classifier has to generalize on unknown object views. Otherwise, we use the object-views dataset as training set and test segmentation on the object-mix dataset. The forest consists of 5 trees and we use a random 25% fraction of the training data for each tree. We set the maximum depth of the trees to 10 and select from 400 node functions and 5 thresholds drawn at random. Since it may be of interest to segment objects from background like the table-top, we add background as an additional class to the object categories in a second set of experiments.

Table I shows global and average accuracy obtained on the object-view dataset. The use of normalized texture and depth features clearly outperforms the standard classifier with unnormalized texture features. It is remarkable that depth alone yields better segmentation accuracy than the combined approach when the background is neglected. However, the use of normalized texture improves the classification of background. The combination of depth and texture achieves highest overall accuracy in segmenting background and objects. Fig. 5 depicts examples for good and bad segmentation results obtained with this configuration.

On the object-mix dataset (Table II), our approach again achieves better accuracy than the standard approach. Again, the use of depth alone results in highest accuracy when the background is neglected. Considering background, highest overall and average accuracy is achieved by combining texture and depth (example segmentations can be found in Fig. 6).

VI. Conclusions

In this paper, we propose an approach to object segmentation and recognition that fuses information from color and Time-of-Flight cameras. From both types of images, we extract shape and appearance features that we use in a Random Forest classification framework. Furthermore, we use depth to estimate local surface orientation at each ToF pixel. By rotating features onto the surface orientation, we normalize texture and depth features for scale and viewpoint.

Our experiments demonstrate that the combination of depth and texture information yields superior classification results to the use of unnormalized texture alone. However, the small resolution of the Time-of-Flight camera and the inherent restrictions due to its measurement principle limit our approach. Depth from structured light or 3D laser range finders could further enhance the performance of our approach. Also, while the maximum likelihood segmentations obtained by our approach seem to be noisy and unsmooth at first glance, the probabilistic output of the Random Forest classifier could be used in a spatial smoothing stage using a CRF, for example. By this, larger spatial context could be incorporated into our recognition and segmentation approach.

References

[1] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[2] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[3] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[4] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[5] F. Schroff, A. Criminisi, and A. Zisserman. Object class segmentation using random forests. In *Proc. of the British Machine Vision Conference*, 2008.

[6] X. Wang and E. Grimson. Spatial latent dirichlet allocation. In *Proc. of the Int. Conf. on Neural Information Processing Systems (NIPS)*, 2007.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

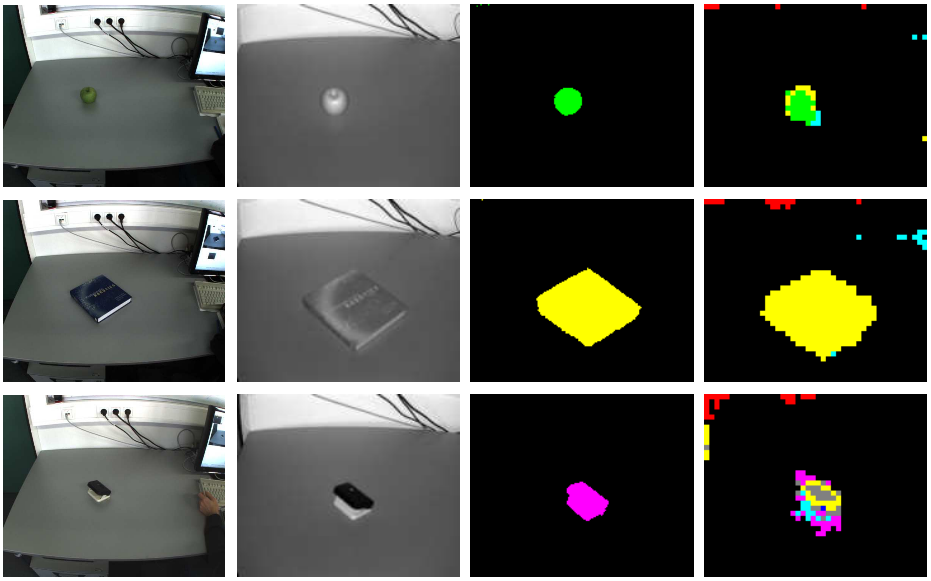

Fig. 5. Examples of segmentation results with depth ($r = 0.07m$) and texture ($r = 0.1m$) features on the object-views dataset. From left to right: Color image, Time-of-Flight amplitude image, ground truth segmentation, and segmentation result.

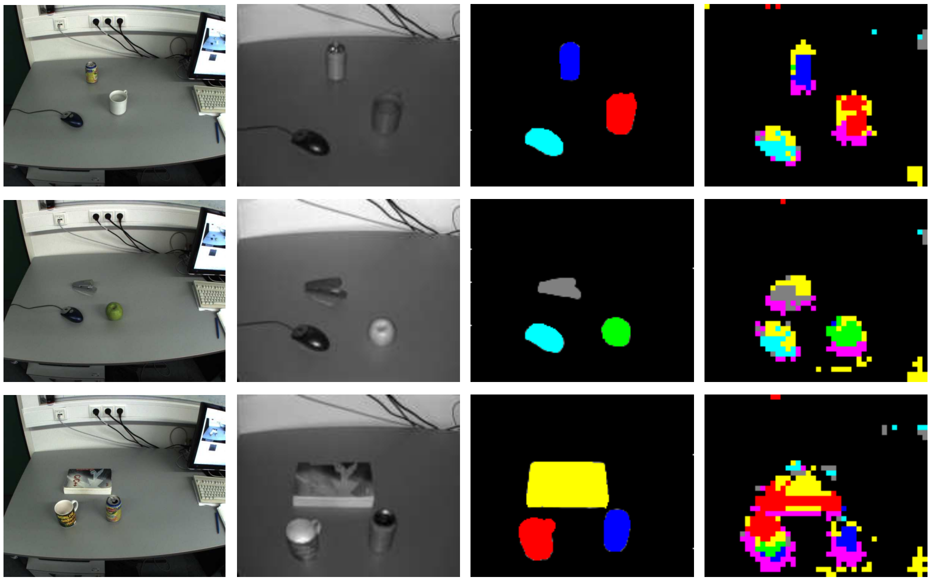

Fig. 6. Examples of segmentation results with depth ($r = 0.07m$) and texture ($r = 0.025m$) features on the object-mix dataset. From left to right: Color image, Time-of-Flight amplitude image, ground truth segmentation, and segmentation result.

[8] A. Criminisi. Microsoft research cambridge object recognition dataset, 2004. version 1.0.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

[10] E. Wahl, G. Hillenbrand, and G. Hirzinger. Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification. In *Proc. of the Int. Conf. on 3-D Digital Imaging and Modeling*, 2003.

[11] R. B. Rusu, Z. Csaba Marton, N. Blodow, and M. Beetz. Persistent Point Feature Histograms for 3D Point Clouds. In *Proc. of the 10th Int. Conf. on Intelligent Autonomous Systems (IAS-10)*, 2008.

[12] R. B. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Kobe, Japan, May 12-17 2009.

[13] S. Gould, P. Baumstarck, M. Quigley, A. Y. Ng, and D. Koller. Integrating visual and range data for robotic object detection. In *Proceedings of the ECCV workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2)*, 2008.

[14] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees*. Chapman & Hall/CRC, 1984.

[15] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV)*, 2007.

[16] J. Stückler and S. Behnke. Integrating Indoor Mobility, Object Manipulation, and Intuitive Interaction for Domestic Service Tasks. In *Proc. of the IEEE Int. Conf. on Humanoid Robots*, 2009.

[17] R. Lange. *3D time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology*. PhD thesis, University Siegen, 2000.

[18] S. May, D. Droeschel, D. Holz, S. Fuchs, and A. Nüchter. Robust 3D-Mapping with Time-of-Flight Cameras. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[19] X. Guo. Three-dimensional moment invariants under rigid transformation. In *Proc. of the 5th Int. Conf. on Computer Analysis of Images and Patterns (CAIP)*. Springer-Verlag, 1993.