Semantic Mapping Using Object-Class Segmentation of RGB-D Images

Jörg Stückler[1], Nenad Biresev[2], and Sven Behnke[1]

***Abstract*— For task planning and execution in unstructured environments, a robot needs the ability to recognize and localize relevant objects. When this information is made persistent in a semantic map, it can be used, e. g., to communicate with humans. In this paper, we propose a novel approach to learning such maps. Our approach registers measurements of RGB-D cameras by means of simultaneous localization and mapping. We employ random decision forests to segment object classes in images and exploit dense depth measurements to obtain scale-invariance. Our object recognition method integrates shape and texture seamlessly. The probabilistic segmentation from multiple views is filtered in a voxel-based 3D map using a Bayesian framework. We report on the quality of our object-class segmentation method and demonstrate the benefits in accuracy when fusing multiple views in a semantic map.**

I. INTRODUCTION

Autonomous robots require semantic knowledge about their surroundings in order to plan and execute complex tasks or to communicate with human users on a semantic level. In order to gain such world knowledge, a robot not only needs the capability to recognize and localize objects, but also to represent this information persistently.

In this paper, we propose a novel approach to learning semantic 3D maps containing object information. We combine object recognition in RGB-D images with simultaneous localization and mapping. For object recognition, we apply random decision forests to classify images pixel-wise. By exploiting depth information for the object-class segmentation algorithm, we obtain a scale-invariant classifier that incorporates shape and texture cues seamlessly. The classifier provides the probability over class labels for each pixel. Given the camera trajectory estimate of an RGB-D SLAM method, we filter this soft labeling in a voxel-based 3D map within a Bayesian framework (see Fig. 1). By this, we can fuse classification evidence from several views and improve the robustness of our method for classification errors. Our approach results in 3D maps augmented with voxel-wise object class information.

In experiments, we evaluate the performance of our object recognition method and demonstrate the benefits of fusing recognition information from multiple views in a 3D map.

II. RELATED WORK

Many mapping approaches build geometric representations of the environment. Different sensors have been used for this in the past, including 2D and 3D laser scanners, single cameras, and stereo systems. Recently, several methods have been proposed that acquire full 3D maps from RGB-D images. Henry et al. [1], for example, extract textured surface patches, register them using ICP [2] to the model, and apply graph-optimization to obtain an accurate map. Engelhard et al. [3] match SURF features between RGB-D frames and refine the registration estimate using ICP. In own work, we apply rapid registration of RGB-D images [4] and graph optimization to learn multi-resolution surfel maps. Such approaches do not incorporate valuable semantic information like place or object labels into the map.

Some systems have been proposed that map semantics. While most approaches utilize SLAM as a front-end to obtain a sensor trajectory estimate [5], [6], [7], [8], [9], [10], some methods also incorporate the spatial relation of objects into SLAM. Tomono et al. [11], for example, detect polyhedral object models in images and perform SLAM in 2D maps using the detected objects as landmarks. In contrast to our approach, this method is restricted to objects with clearly visible linear edges. Zender et al. [5] apply SLAM in 2D maps using laser scanners, recognize objects using SIFT features, and map their locations in the 2D map. In addition

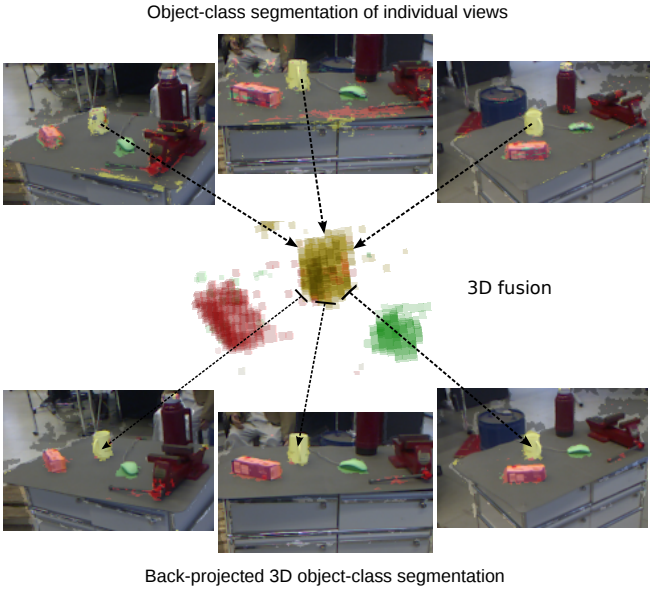

Fig. 1. We fuse learned object-class segmentations of various views in 3D in a Bayesian framework. We not only obtain 3D object-class maps: Filtering in 3D from multiple views also reduces false positives and improves segmentation quality significantly. This reflects in the crisp back-projection of the 3D object-class map into the images.

[1]Autonomous Intelligent Systems, Computer Science Institute VI, University of Bonn, 53113 Bonn, Germany stueckler at ais.uni-bonn.de, behnke at cs.uni-bonn.de

[2]Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS, 53757 Sankt Augustin, Germany biresev at cs.uni-bonn.de

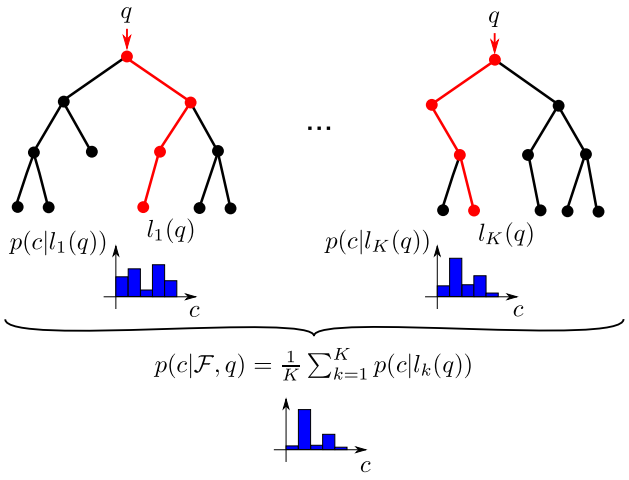

Fig. 2. Random decision forests consist of multiple decision trees. For each tree, a query pixel q passes through several decision nodes. The node functions make a binary decision on the pixel using scalar features and a threshold. Each query pixel arrives at a specific leaf node $l(q)$ in a tree and is assigned the class probability $p(c|l)$ of training pixels that arrive at the leaf. The final class probabilities are computed by averaging the individual trees.

to SIFT-based recognition, Vasudevan et al. [6] also detect doors by analyzing laser scans, since they are important topologic objects that connect rooms. Meger et al. [7] combine semantic 2D mapping of objects with attention mechanisms. We build 3D semantic maps containing dense object information. Ni¿½chter et al. [8] apply ICP, plane segmentation, and reasoning to label planar segments in 3D maps that they acquire using 3D laser scanners. They apply AdaBoost on Haar wavelets and SVM classifiers on contour descriptions to detect objects and persons in the 3D maps. In our approach, we segment the original image data and fuse segmentation evidence from multiple views. Castle et al. [9] and Civera et al. [10] propose purely vision-based means to acquire 3D maps with object labellings. In both approaches, SLAM is solved with feature-based monocular EKF-SLAM formulations. Objects are recognized using SIFT features and persistently maintained in the 3D feature map. The approach of Ranganathan and Dellaert [12] learns 3D constellation models of places composed of objects using SIFT features. In this approach, the map consists of a set of places with associated models. The aforementioned approaches, however, do not build 3D maps with dense object information.

We integrate image-based object-class segmentation with SLAM from RGB-D images into a semantic 3D mapping framework. Each image is segmented pixel-wise into object classes and irrelevant background. Based on the SLAM estimate, this information is then projected into 3D to fuse object recognition results from multiple views into a consistent 3D map. This not only provides 3D segmentations of objects, but also improves classification accuracy significantly.

III. OBJECT-CLASS SEGMENTATION USING RANDOM DECISION FORESTS

Object-class image segmentation is a challenging, actively researched problem in computer vision [13], [14], [15], [16]. One branch of research applies variants of random decision forests (RF, [17]). RFs are efficient classifiers for multi-class problems. They ensemble multiple random decision trees and achieve lower generalization error than single decision trees alone. RFs have been demonstrated to achieve comparable performance to SVMs [18]. Their major advantage is their high computational efficiency during recall. Implemented on GPU, training can be performed on massive datasets [19].

Semantic Texton Forests proposed by Shotton et al. [13] use simple features of luminance and color at single pixels or comparisons between two pixels in a RF classifier. Using image-level priors and a second stage of RFs, local and scene context is incorporated into the classification framework. Schroff et al. [20] enhance the basic RF classifier by further features such as image regions, Histograms of Oriented Gradients [21], and filterbanks. They demonstrate that post-processing of the RF segmentation with Conditional Random Fields further improves segmentation quality. Recently, the RF approach has been successfully applied for segmenting human body parts and tracking body pose in real-time using depth images [19]. Shotton et al. propose to normalize feature queries with the available depth to obtain scale-invariant recognition. We extend RF classification by incorporating both depth and color features. In contrast to previous own work [22], we use simple region features in color and depth and only normalize for scale changes to gain an efficient classifier for RGB-D images.

A. Structure of Random Decision Forests

A random decision forest $\mathcal{F}$ is an ensemble of K random 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 $c \in \mathcal{C}$.

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

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

For learning a forest, each tree is trained independently on a random subset of the training examples. At each node in a tree, we sample many features and thresholds randomly and select the one that seperates the training examples best according to the measure of information gain. This allows for mixing different kinds of features such as functions in color and depth cues.

B. RGB-D Image Features

For a pixel q, we determine region features in depth and color cues and utilize dense depth to normalize the region

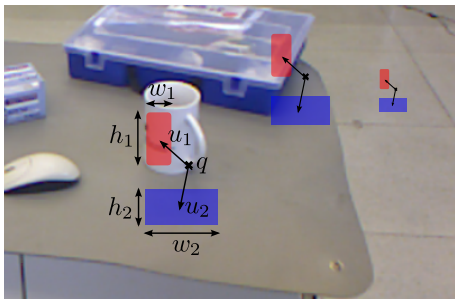

Fig. 3. Features in the random decision classifier compare the average values in two regions relative to the query pixel q. We normalize for perspective scale changes in the image by exploiting the dense depth available in RGB-D images. We scale relative offset locations u_i and region extents w_i, h_i by the inverse of the depth $d(q)$ measured at the query pixel.

queries for scale changes in the image (see Fig. 3). We parametrize features at pixel q by

$$f_\theta(q) := \frac{\sum_{p\in R_1(q)} \phi_1(p)}{|R_1(q)|} - \frac{\sum_{p\in R_2(q)} \phi_2(p)}{|R_2(q)|}, \quad (1)$$

where $R_j(q) := R\left(q + \frac{u_j}{d(q)}, \frac{w_j}{d(q)}, \frac{h_j}{d(q)}\right)$ is the rectangular image region at the offset u that is normalized in offset position and size by the depth $d(q)$ measured at the query pixel. The set of feature parameters θ comprises the unnormalized offset positions u_j, the region extents w_j, h_j, and the image channels ϕ_j. Note, that we restrict comparisons to either two depth regions or to any two color regions. We represent the color cues in the CIE Lab color space. In the depth image, the region size $|R_j(q)|$ counts the number of valid depth readings in the region. If an offset region contains no valid depth measurement or lies beyond the image, its feature value is set to a large positive constant. We efficiently implement region features using integral images.

Each node in the decision tree decides on the query pixels with a threshold τ to either pass it to its left or right child. Individually, each feature gives only small information about the object class at a pixel. Within the cascades in the decision trees, however, the tests are sufficient to accurately classify pixels.

C. Training

Each of the K decision trees is trained with a subset $\mathcal{D}$ of images from the training set. We split the training set into K equally sized sets and extract $|\mathcal{D}| \cdot N$ random pixels from all images (using $N = 2000$ in our experiments). Since we also train explicitly on the background class and since the individual object classes may differ in the number of pixels, we balance the classes by random sampling of equally sized sets for each class. In this way, small objects are well sampled for training. We will, however, have to consider the actual distribution of class labels in the training images at later training stages in order to incorporate the prior probability of each class into the classifier.

We train the decision trees in a depth-first manner by choosing feature parameters θ and a threshold τ at each node and splitting the pixel set Q accordingly into left and right subsets Q_l and Q_r:

$$\begin{aligned} Q_l(\theta,\tau) &:= \{q \in Q | f_\theta(q) < \tau\} \text{ and} \\ Q_r(\theta,\tau) &:= \{q \in Q | f_\theta(q) \geq \tau\}. \end{aligned} \quad (2)$$

Since the parameter space cannot be evaluated analytically, we sample P random parameter sets and thresholds (e. g., $P = 2000$) and select feature and threshold that yield maximal information gain

$$I(\theta,\tau) := H(Q) - \sum_{s\in\{l,r\}} \frac{|Q_s(\theta,\tau)|}{|Q|} H\left(Q_s(\theta,\tau)\right), \quad (3)$$

where $H(Q) := -\sum_{c\in\mathcal{C}} p(c|Q) \log_2\left(p(c|Q)\right)$ is the Shannon entropy of the distribution of training class labels in pixel set Q. This splitting criterion finds feature parameters and threshold that most distinctively seperate the pixel set at a node. Each node is split until a maximum depth is reached in the tree, or the number of pixels lies below a minimum support threshold.

At each leaf node l, we want to maintain the distribution $p(c|l, D)$ of pixels of class c that arrive at the node from the original training set. Since we train the decision tree from pixels with equally distributed class labels, we actually measure the class distribution $p(c|l, Q)$ of training pixels Q at the leaf, i. e.,

$$p(c|l, Q) := p(c(q)|l, q \in Q) = p(c(q)|l, q \in Q, q \in \mathcal{D}). \quad (4)$$

The distribution of interest can be obtained by applying Bayes rule:

$$\begin{aligned} p(c|l, Q, \mathcal{D}) &= \frac{p(q \in Q|c(q), l, q \in \mathcal{D})\ p(c(q)|l, q \in \mathcal{D})}{p(q \in Q|l, q \in \mathcal{D})} \\ &= \frac{p(q \in Q|c(q), q \in \mathcal{D})\ p(c(q)|l, q \in \mathcal{D})}{p(q \in Q|q \in \mathcal{D})}. \end{aligned} \quad (5)$$

For the desired distribution we obtain

$$p(c(q)|l, q \in \mathcal{D}) = \frac{p(c(q)|l, q \in Q)\ p(q \in Q|q \in \mathcal{D})}{p(q \in Q|c(q), q \in \mathcal{D})} \quad (6)$$

We can further reformulate the probability of a pixel of class c to be included in the class-equalized training data Q to

$$p(q \in Q|c(q), q \in \mathcal{D}) = \frac{p(c(q)|q \in Q)\ p(q \in Q|q \in \mathcal{D})}{p(c(q)|q \in \mathcal{D})}, \quad (7)$$

and obtain

$$p(c(q)|l, q \in \mathcal{D}) = \frac{p(c(q)|l, q \in Q)\ p(c(q)|q \in \mathcal{D})}{p(c(q)|q \in Q)}. \quad (8)$$

By design, $p(c(q)|q \in Q)$ is uniform among class labels and, hence, we incorporate the distribution of classes in the complete training set into the leaf distributions through

$$p(c|l, D) = \eta\ p(c|l, Q)\ p(c|\mathcal{D}), \quad (9)$$

where $\eta^{-1} := p(c|Q) = 1/|\mathcal{C}|$.

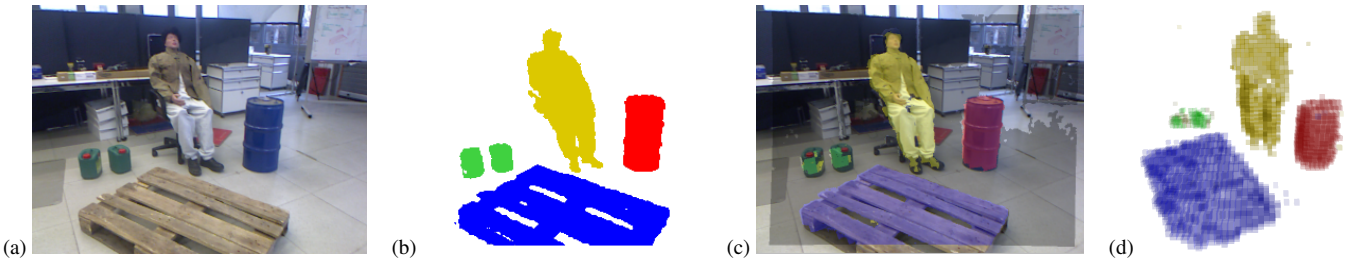

Fig. 4. Semantic mapping. a) RGB image of a scene. b) Ground truth object-class segmentation. c) Back-projected 3D object-class segmentation overlayed on RGB image. d) 3D object-class map obtained by fusing multiple views from a SLAM trajectory.

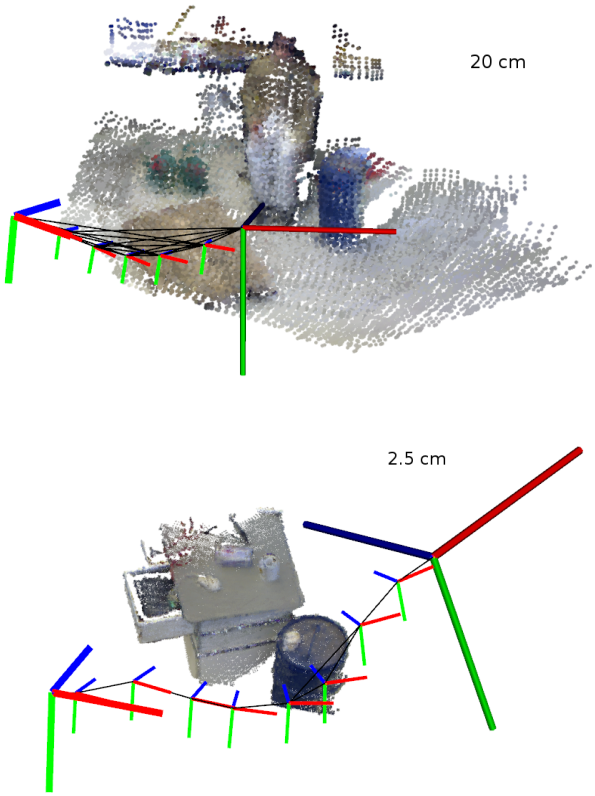

Fig. 5. We perform simultaneous localization and mapping by registering multi-resolution surfel maps of RGB-D images and optimizing spatial relations in a key view graph. The example maps are visualized by samples from the surfel distributions at 2.5 cm (bottom) and 20 cm (top) resolution.

IV. SEMANTIC MAPPING

We integrate our object-class segmentation method with SLAM to fuse the segmentations of individual images in a dense 3D map.

A. Simultaneous Localization and Mapping Front-End

We base our SLAM method on fast and accurate RGB-D image registration using multi-resolution surfel maps [4]. Our registration approach aligns 640×480 images at a framerate of about 10 Hz.

Since small registration errors may accumulate in significant pose drift over time, we establish and optimize a graph of probabilistic spatial relations between similar view poses (see Fig. 5). We denote a view pose in the graph as key view and register the current camera frame to the most similar key view in order to keep track of the camera pose. Similarity is measured by distance in translation and rotation between view poses. New key views are added to the graph, if the similarity measure indicates a significant motion of the camera. This also establishes a spatial relation between the new key view and the reference key view. In addition, we establish relations between further similar key views.

Our probabilistic registration method provides a mean and covariance estimate for each spatial relation. We obtain the likelihood of the relative pose observation $z = (\hat{x}, \Sigma(\hat{x}))$ of the key view j from view i by

$$p(\hat{x}|x_i, x_j) = \mathcal{N}\left(\hat{x}; \Delta(x_i, x_j), \Sigma(\hat{x})\right), \tag{10}$$

where $\Delta(x_i, x_j)$ denotes the relative pose between the key views under their current estimates x_i and x_j.

From the graph of spatial relations we infer the probability of the trajectory estimate given the relative pose observations

$$p(x_{1,\dots,N}|\hat{x}_1, \dots, \hat{x}_M) \propto \prod_k p(\hat{x}_k|x_{i(k)}, x_{j(k)}). \tag{11}$$

We solve this graph optimization problem by sparse Cholesky decomposition using the g^{2}o library [23]. Finally, our mapping framework supports the fusion of the RGB-D images in a single multi-resolution surfel map using the optimized trajectory estimate.

B. Probabilistic 3D Mapping of Object-Class Image Segmentations

Given the trajectory estimate from our SLAM approach and the depth information in the images, we project the probabilistic object-class segmentations into 3D and filter this information in a probabilistic octree map. Each voxel v of the octree stores a belief $Bel(c(v))$ that the object class $c(v)$ is present in its volume

$$Bel(c(v)) = p(c(v)|\mathcal{Z}, \mathcal{S}), \tag{12}$$

where $\mathcal{Z}$ is the set of RGB-D images with probabilistic labelling and $\mathcal{S}$ is the trajectory estimate. Our goal is to integrate segmentation evidence from multiple views in a 3D map and to improve segmentation quality.

We successively project the image pixels into 3D and determine corresponding octree voxels. The belief in the

voxel is then updated in a Bayesian framework with the pixel observations $q_{1:N} := \{q_1, q_2, ..., q_N\}$ that fall into the voxel:

$$p(c(v)|q_{1:N}, \mathcal{S}) = \sum_{c(q_1),\ldots,c(q_N)} p(c(v), c(q_1), \ldots, c(q_N)|q_{1:N}, \mathcal{S}). \quad (13)$$

Neglecting the known trajectory and applying Bayes rule yields

$$p(c(v)|q_{1:N}) = \sum_{\ldots} p(c(v)|c(q_1), \ldots, c(q_N), q_{1:N}) \; p(c(q_1), \ldots, c(q_N)|q_{1:N}). \quad (14)$$

The left term can be further factored using Bayes rule, while for the right term we impose independence between pixel observation. We arrive at

$$p(c(v)|q_{1:N}) = p(c(v)) \sum_{\ldots} \prod_i \eta_i \; p(c(q_i)|c(v)) \; p(c(q_i)|q_i), \quad (15)$$

where $\eta_i := 1/p(c(q_i)|c(q_{i+1}), \ldots, c(q_N))$. We approximate $p(c(q_i)|q_i)$ with the output of the RF classifier $p(c(q_i)|q_i, \mathcal{F})$. The probability $p(c(v)) =: Bel_0(c(v))$ incorporates prior knowledge on the belief. For the distribution $p(c(q_i)|c(v)) = \mathbf{1}_{\{c(v)\}}(c(q_i))$ we assume a deterministic one-to-one mapping. It follows that

$$p(c(v)|q_{1:N}, \mathcal{S}) = Bel_0(c(v)) \prod_i \eta_i \; p(c(q_i) = c(v)|q_i, \mathcal{F}), \quad (16)$$

which can also be applied recursively.

V. EXPERIMENTS

We evaluate our approach on a datasets containing RGB-D videos of three smaller table-top object classes and four larger object classes. The datasets contain 617 and 500 training images and 500 test images each from 47 and 40 scenes, respectively, with several instances of the object classes in varying configuration. We use precision, recall, and accuracy [24] measures to quantify segmentation quality. We assess the overall accuracy on each test set by counting over the pixel decisions of all classes. Since the background class is semantically different from the object classes, we also measure the segmentation quality of the object classes without background. To assess the quality of the fused semantic maps, we back-project the octree belief over object-classes into the test images.

A. Annotation Tool

In order to acquire large amounts of annotated training data in reasonable time, we developed an interactive semi-automatic annotation tool. In addition to directly annotating pixels with a pen tool or applying grab cut, our tool makes use of depth in several ways: Since typically objects are located on planar surfaces, the user can select image pixels on background planes and let points on the plane automatically be labelled as background. The user can also crop out foreground objects using depth continuity as segmentation hint.

TABLE I

OBJECT-CLASS SEGMENTATION PERFORMANCE FOR SMALL OBJECTS.

method	precision w (w/o) bg	recall w (w/o) bg	accuracy w (w/o) bg
unnorm. color	0.95 (0.14)	0.95 (0.14)	0.91 (0.07)
norm. color	0.95 (0.14)	0.95 (0.15)	0.91 (0.08)
norm. depth	0.96 (0.40)	0.96 (0.62)	0.93 (0.32)
norm. color + depth	0.96 (0.35)	0.96 (0.66)	0.91 (0.30)
norm. color + depth + 3D	**0.97 (0.64)**	**0.98 (0.97)**	**0.95 (0.42)**

TABLE II

OBJECT-CLASS SEGMENTATION PERFORMANCE FOR LARGE OBJECTS.

method	precision w (w/o) bg	recall w (w/o) bg	accuracy w (w/o) bg
unnorm. color	0.80 (0.61)	0.80 (0.48)	0.67 (0.37)
norm. color	0.85 (0.74)	0.85 (0.61)	0.74 (0.51)
norm. depth	0.80 (0.71)	0.80 (0.38)	0.67 (0.33)
norm. color + depth	0.87 (0.78)	0.87 (0.69)	0.78 (0.58)
norm. color + depth + 3D	**0.91 (0.87)**	**0.91 (0.76)**	**0.83 (0.68)**

We extract multiple views on a scene from image sequences in which the camera is swept through the scene. The user can then select one of the images in the sequence, segment the image using the aforementioned convenient tools, and project the segmentation to further images. It only requires little effort to refine the projected segmentations. We integrate our Bayesian filtering approach to fuse image annotations from multiple views in a 3D map. For this purpose, we preprocess the image sequence to obtain a trajectory estimate using our SLAM method.

B. Results

Table I and Table II show average results for different kinds of RF classifiers on both datasets. The 3D fusion of image segmentations using color and depth features clearly outperforms the other approaches. It improves in accuracy on purely image-based segmentations by about 10% for big objects without background and ca. 12% for smaller objects (w/o background). Remarkably, it achieves almost perfect recall (97%) from only 66% per image for smaller objects.

We also see that, in contrast to the small objects, for big objects depth-normalized color is a prominent feature and yields higher accuracy than normalized depth queries alone. Here, the scale normalization of the color features using depth enhances segmentation quality significantly.

In Tables III and IV we show results for the individual object classes. While for the big objects again the fusion in 3D is dominantly superior to image-based segmentation alone, we can see that for the small objects recall and accuracy of the computer mice is reduced. The computer mice are very flat objects and are easily confused with the table plane (background class). Nevertheless, the increase in precision shows that most false positive detections could be successfully removed by filtering image segmentations in 3D.

TABLE III

PER CLASS SEGMENTATION PERFORMANCE FOR SMALL OBJECTS.

	norm. color + depth			norm. color + depth + 3D		
class	prec.	recall	acc.	prec.	recall	acc.
cup	0.42	0.82	0.38	**0.76**	**0.94**	**0.73**
teabox	0.28	0.64	0.24	**0.41**	**0.72**	**0.36**
mouse	0.43	**0.47**	**0.29**	**0.98**	0.20	0.20
background	0.99	0.96	0.96	**0.99**	**0.98**	**0.98**

TABLE IV

PER CLASS SEGMENTATION PERFORMANCE FOR LARGE OBJECTS.

	norm. color + depth			norm. color + depth + 3D		
class	prec.	recall	acc.	prec.	recall	acc.
palette	0.93	0.84	0.78	**0.98**	**0.90**	**0.88**
barrel	0.92	0.73	0.68	**0.95**	**0.85**	**0.81**
canister	0.74	0.13	0.12	**0.95**	**0.22**	**0.22**
human	0.56	0.59	0.40	**0.69**	**0.64**	**0.49**
background	0.91	0.94	0.86	**0.92**	**0.97**	**0.89**

VI. CONCLUSIONS

In this paper, we proposed a novel approach to semantic mapping. We apply object-class image segmentation to recognize objects pixel-wise in RGB-D images. We incorporate depth and color cues into a random decision forest classifier and normalize the features for scale using depth measurements. Based on trajectory estimates obtained with a SLAM method, we propose to fuse the image segmentations into a probabilistic 3D object-class map. In experiments on two datasets, we demonstrate that our approach not only provides a 3D segmentation of the object classes, but also improves 2D segmentation quality significantly.

Our approach directly operates on the original image measurements. While fusing RGB-D measurements in a 3D map and classifying the 3D volumes would also be possible, the aggregation into 3D typically involves some sort of compressive aggregation and, hence, loss of information to cope with the large amount of data. We note that the segmentation quality of our approach depends on the properties of the underlying object-class image segmentation method. While many other methods exist that demonstrate good segmentation results, the recall efficiency of the segmentation approach is of equal importance for online processing and application in a robotics setting.

In future work, we plan to integrate further descriptive image features like Histograms of Oriented Gradients or Fast Point Feature Histograms. In order to scale our approach to larger sets of objects, we will consider the combination of multiple random decision forests. Finally, we will implement interactive training tools using GPUs to enable online training on massive datasets.

REFERENCES

[1] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments," in *Proc. of International Symposium on Experimental Robotics (ISER)*, 2010.

[2] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1992.

[3] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard, "Real-time 3D visual SLAM with a hand-held camera," in *Proc. of RGB-D Workshop on 3D Perception in Robotics at European Robotics Forum*, 2011.

[4] J. Stückler and S. Behnke, "Robust real-time registration of RGB-D images using multi-resolution surfel representations," in *Proc. of the 7th German Conference on Robotics (ROBOTIK)*, 2012, to appear, for review: http://www.ais.uni-bonn.de/papers/robotik2012_mrsreg.pdf.

[5] H. Zender, O. M. Mozos, P. Jensfelt, G.-J. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 493 – 502, 2008.

[6] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots-an object based approach," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 359–371, 2007.

[7] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, and D. G. Lowe, "Curious george: An attentive semantic robot," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 503–511, 2008.

[8] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 915–926, 2008.

[9] R. O. Castle, G. Klein, and D. W. Murray, "Combining monoSLAM with object recognition for scene augmentation using a wearable camera," *Image Vision Computing*, vol. 28, no. 11, pp. 1548 – 1556, 2010.

[10] J. Civera, D. Galvez-Lopez, L. Riazuelo, D. Tardos, and J. M. M. Montiel, "Towards semantic SLAM using a monocular camera," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[11] M. Tomono and Y. Shin'ichi, "Object-based localization and mapping using loop constraints and geometric prior knowledge," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2003.

[12] A. Ranganathan and F. Dellaert, "Semantic modeling of places using objects," in *Proc. of Robotics: Science and Systems*, 2007.

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

[14] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical crfs for object class image segmentation." in *Proc. of the Int. Conf. on Computer Vision (ICCV)*, 2009.

[15] A. Ion, J. Carreira, and C. Sminchisescu, "Image segmentation by figure-ground composition into maximal cliques," *Proc. of the Int. Conf. on Computer Vision (ICCV)*, 2011.

[16] H. Schulz and S. Behnke, "Learning object-class segmentation with convolutional neural networks," in *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, 2012, to appear.

[17] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[18] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *Proc. of the Int. Conf. on Computer Vision (ICCV)*, 2007.

[19] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

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

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[22] J. Stückler and S. Behnke, "Combining depth and color cues for scale- and viewpoint-invariant object segmentation and recognition using random forests." in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[23] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[24] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.