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

Jörg Stückler1, Nenad Biresev2, and Sven Behnke1

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 laser scanners, recognize objects using SIFT features, and map their locations in the 2D map. In addition
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 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 $F$ is an ensemble of $K$ random decision trees $T_k$. 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 $T_k$ 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.,

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

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 separates 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.
The set of feature parameters \( \phi \) comprise the unnormalized offset positions \( u_i \), the region extents \( w_i, h_i \), and the image channels \( \phi_j \). 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 \( \tau \) 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 \( D \) of images from the training set. We split the training set into \( K \) equally sized sets and extract \( |D| \cdot N \) 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 \( \theta \) and a threshold \( \tau \) at each node and splitting the pixel set \( Q \) accordingly into left and right subsets \( Q_l \) and \( Q_r \):

\[
Q_l(\theta, \tau) := \{ q \in Q \mid f_\theta(q) < \tau \} \quad \text{and} \quad Q_r(\theta, \tau) := \{ q \in Q \mid f_\theta(q) \geq \tau \}.
\]

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

\[
I(\theta, \tau) := H(Q) - \sum_{s \in \{l,r\}} \frac{|Q_s(\theta, \tau)|}{|Q|} H(Q_s(\theta, \tau)),
\]

where \( H(Q) := -\sum_{c \in C} p(c|Q) \log_2 (p(c|Q)) \) 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 separate 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.,

\[
p(c|l, Q := p(c|q)|l, q \in Q) = p(c|l, q \in Q, q \in D).
\]

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

\[
p(c|l, Q, D) = \frac{p(q \in Q|c(q), l, q \in D) \cdot p(c|l, q \in D)}{p(q \in Q|l, q \in D)}
\]

\[
= \frac{p(q \in Q|c(q), q \in D) \cdot p(c|q)|l, q \in D)}{p(q \in Q|q \in D)}.
\]

For the desired distribution we obtain

\[
p(c|q)|l, q \in D = \frac{p(c|q)|l, q \in Q \cdot p(q \in Q|c(q), q \in D)}{p(q \in Q|q \in D)}
\]

We can further reformulate the probability of a pixel of class \( c \) to be included in the class=equalized training data \( Q \) to

\[
p(q \in Q|c(q), q \in D) = \frac{p(c|q)|q \in Q) \cdot p(q \in Q|q \in D)}{p(c(q)|q \in D)}
\]

and obtain

\[
p(c|q)|l, q \in D = \frac{p(c|q)|l, q \in Q) \cdot p(c(q)|q \in D)}{p(c(q)|q \in Q)}.
\]

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

\[
p(c|l, D) = \eta \cdot p(c|l, Q) \cdot p(c|D),
\]

where \( \eta^{-1} := p(c|Q) = 1/|C| \).
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 $640 \times 480$ 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

$$p(\hat{x}|x_i, x_j) = N(\hat{x}; \Delta(x_i, x_j), \Sigma(\hat{x})),$$

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

$$p(x_1, \ldots, x_N|\hat{x}_1, \ldots, \hat{x}_M) \propto \prod_k p(\hat{x}_k|x_{i(k)}, x_{j(k)}).$$

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

$$Bel(c(v)) = p(c(v)|Z, S),$$

where $Z$ is the set of RGB-D images with probabilistic labelling and $S$ 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 \( q_{1:N} := \{ q_1, q_2, \ldots, q_N \} \) that fall into the voxel:

\[
p(c(v)|q_{1:N}, S) = \sum_{c(q_1), \ldots, c(q_N)} p(c(v), c(q_1), \ldots, c(q_N)|q_{1:N}, S). \tag{13}
\]

Neglecting the known trajectory and applying Bayes rule yields

\[
p(c(v)|q_{1:N}) = \sum_{c(q_1), \ldots, c(q_N)} p(c(v)|c(q_1), \ldots, c(q_N)) p(c(q_1), \ldots, c(q_N)|q_{1:N}). \tag{14}
\]

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

\[
p(c(v)|q_{1:N}) = p(c(v)) \prod_i \eta_i p(c(q_i)|c(v)) p(c(q_i)|q_i), \tag{15}
\]

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 \( p(c(q_i)|c(v)) = 1_{\{c(v)\}}(c(q_i)) \) we assume a deterministic one-to-one mapping. It follows that

\[
p(c(v)|q_{1:N}, S) = Bel_0(c(v)) \prod_i \eta_i p(c(q_i) = c(v)|q_i, \mathcal{F}), \tag{16}
\]

which can also be applied recursively.

V. EXPERIMENTS

We evaluate our approach on a dataset containing RGB-D videos of three smaller table-top object classes and four larger object classes. The dataset contains 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 configurations. 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 III  
PER CLASS SEGMENTATION PERFORMANCE FOR SMALL OBJECTS.

<table>
<thead>
<tr>
<th>class</th>
<th>prec.</th>
<th>recall</th>
<th>acc.</th>
<th>prec.</th>
<th>recall</th>
<th>acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>cup</td>
<td>0.42</td>
<td>0.82</td>
<td>0.38</td>
<td>0.76</td>
<td>0.94</td>
<td>0.73</td>
</tr>
<tr>
<td>teabox</td>
<td>0.28</td>
<td>0.64</td>
<td>0.24</td>
<td>0.41</td>
<td>0.72</td>
<td>0.36</td>
</tr>
<tr>
<td>mouse</td>
<td>0.43</td>
<td>0.47</td>
<td>0.29</td>
<td>0.98</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>background</td>
<td>0.99</td>
<td>0.96</td>
<td>0.96</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
</tr>
</tbody>
</table>

TABLE IV  
PER CLASS SEGMENTATION PERFORMANCE FOR LARGE OBJECTS.

<table>
<thead>
<tr>
<th>class</th>
<th>prec.</th>
<th>recall</th>
<th>acc.</th>
<th>prec.</th>
<th>recall</th>
<th>acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>palette</td>
<td>0.93</td>
<td>0.84</td>
<td>0.78</td>
<td>0.98</td>
<td>0.90</td>
<td>0.88</td>
</tr>
<tr>
<td>barrel</td>
<td>0.92</td>
<td>0.73</td>
<td>0.68</td>
<td>0.95</td>
<td>0.85</td>
<td>0.81</td>
</tr>
<tr>
<td>canister</td>
<td>0.74</td>
<td>0.13</td>
<td>0.12</td>
<td>0.95</td>
<td>0.22</td>
<td>0.22</td>
</tr>
<tr>
<td>human</td>
<td>0.56</td>
<td>0.59</td>
<td>0.40</td>
<td>0.69</td>
<td>0.64</td>
<td>0.49</td>
</tr>
<tr>
<td>background</td>
<td>0.91</td>
<td>0.94</td>
<td>0.86</td>
<td>0.92</td>
<td>0.97</td>
<td>0.89</td>
</tr>
</tbody>
</table>

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