2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Fast and Accurate Semantic Mapping through Geometric-based Incremental Segmentation

Yoshikatsu Nakajima[1], Keisuke Tateno[2], Federico Tombari[2] and Hideo Saito[1]

***Abstract*— We propose an efficient and scalable method for incrementally building a dense, semantically annotated 3D map in real-time. The proposed method assigns class probabilities to each region, not each element (e.g., surfel and voxel), of the 3D map which is built up through a robust SLAM framework and incrementally segmented with a geometric-based segmentation method. Differently from all other approaches, our method has a capability of running at over 30Hz while performing all processing components, including SLAM, segmentation, 2D recognition, and updating class probabilities of each segmentation label at every incoming frame, thanks to the high efficiency that characterizes the computationally intensive stages of our framework. By utilizing a specifically designed CNN to improve the frame-wise segmentation result, we can also achieve high accuracy. We validate our method on the NYUv2 dataset by comparing with the state of the art in terms of accuracy and computational efficiency, and by means of an analysis in terms of time and space complexity.**

I. INTRODUCTION

The task of incrementally building a semantically annotated 3D map is a challenging research topic for both the robotics and computer vision communities. It has a wide range of applications including autonomous grasping and manipulation of objects, scene understanding, robotics navigation and augmented reality. For this reason, a valuable research effort is currently undergoing in literature with the aim of developing efficient systems that can scale up to mobile/embedded architectures while being robust enough to generalize to unseen environments.

Motivated by the recent developments of deep learning and Convolutional Neural Networks (CNNs) for 3D data, recent methods have mostly focused on increasing the accuracy of the semantic segmentation map [1], [2], [3]. At the same time, they still face the critical issue of yielding real-time performance, since such systems are built on a set of computationally demanding processing stages, including 3D reconstruction, camera pose estimation and CNN-based semantic segmentation. This becomes even more relevant with regards to embedded and mobile architectures that are typically employed for the aforementioned applications of robotics navigation/grasping and augmented reality.

To achieve real-time performance, some of these methods suggested to only extract semantic information on a subset of the input frames. For example, the methods proposed by Hermans et al. [4] and McCormac et al. (SemanticFusion) [1] achieved, respectively, an output frame-rate of 4Hz and 25.3Hz, by running semantic segmentation, respectively, every 6 and every 10 frames. While such frame skipping strategy can improve run-time performance, it limits their range of application, since it tends to bring in inaccuracies under fast camera motions.

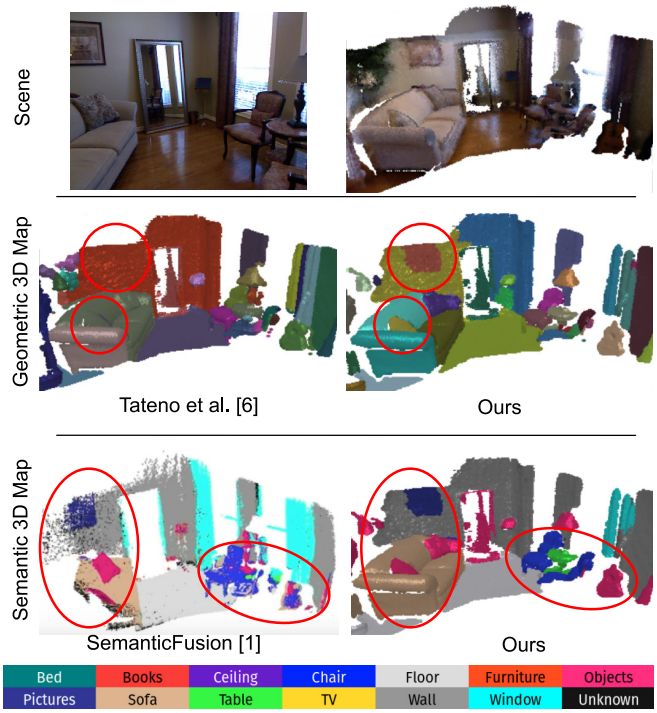

Fig. 1: Our method achieves accurate semantic mapping (comparable to the state of the art [1], bottom row) while being more efficient and scalable. It relies on the geometric segmentation that takes into account semantic information and thus providing meaningful segments than the method of Tateno et al. [6] (middle row).

In this paper, we propose a novel incremental semantic mapping approach that aims at overcoming such issues by yielding highly accurate semantic scene reconstruction (see bottom row of Fig. 1) in real-time. The framework relies on effectively combining a reliable camera pose tracking (InfiniTAM v3 [5]), an incremental segmentation approach [6], and an efficient CNN-based semantic segmentation method. In particular, the 3D map of the scene is built through the fast and robust surfel-based SLAM approach in [7], and geometric segmentation labels are assigned to each surfel

[1]Department of Science and Technology, Keio University, Kanagawa, (Japan) {nakajima, saito}@hvrl.ics.keio.ac.jp
[2]Chair for Computer Aided Medical Procedures (CAMP), TU Munich, Boltzmannstr. 3, 85748 Munich (Germany) {tateno, tombari}@in.tum.de

based on the approach of [6]. Class probabilities of each label are updated through a specifically designed CNN.

We introduce a new probabilistic strategy to deal with one of the most delicate stages, i.e. class probability assignment. According to this strategy, and in contrast to conventional semantic mapping methods which assign class probabilities to each surfel [4], [1], [2], we assign class probabilities to each segment. This reduces notably the time complexity since at each new frame probability distributions need to be updated for those segments which are visible on the image plane from the current camera pose, in contrast to conventional methods which need to update such probabilities for all surfels on the image plane. This strategy also reduces notably the space complexity since probability distributions need to be stored only at each segment rather than each surfel.

In return, the semantic information also improves the geometric-based segmentation from [6]. By taking into account semantic information, it provides additional edges that better represent the semantic structure of the scene, hence allowing to obtain accurate segment regions (see middle row of Fig. 1). Since smoothing of semantic labels is carried out at the geometric fusion stage, this allows us to utilize a CNN with a low resolution (i.e. 40×30) output, with a forward pass requiring only 19ms on an off-the-shelf GPU (i.e., a GeForce GTX 1080).

The overall framework is capable of working in real-time on off-the-shelf architectures, while the requiring a low computational complexity with respect to state of the art. In addition, differently from other methods such as [4], [1], [2], [3], [8], our approach does not require any post-processing based on, e.g., Conditional Random Field, to refine the output of the semantic mapping. We demonstrate the effectiveness and efficiency of our approach on a common benchmark, i.e. the NYUv2 dataset [9], reporting comparable accuracy than the state-of-the-art approaches while being notably faster and scaling better in terms of memory requirements. In addition, we also report an analysis in terms of time and space complexity of our method, demonstrating its advantages with respect to conventional approaches.

II. RELATED WORK

A. Semantic mapping

Related work aimed at incrementally computing a semantic 3D map of the environment are mostly build on top of the following three main stages: (i) frame-wise segmentation to estimate the per-pixel class probability of the input frame, (ii) 2D-3D label transfer to fuse the 2D semantic segmentation labels to the 3D map; and, (iii) 3D refinement to denoise the class probabilities of the 3D map [4], [1], [2], [3], [10], [8]. Notably, [4] employed Random Decision Forests (RDF), a Bayesian framework and Conditional Random Field (CRF) respectively to carry out the three above-mentioned stages.

Since the CRF works on each element of the 3D map reconstructed via SLAM, it is effective in refining the semantic model and obtain high accuracy. Nevertheless, it is computationally heavy, as it requires 400 to 1800ms just for the CRF stage, yielding a frame-rate of 3.9 to 4.6Hz even if the method computes the RDF once every 6 input frames and the CRF once every 30 frames. SemanticFusion [1] employs the CNN model proposed by Noh et al. [11] for 2D semantic segmentation, a Bayesian framework for 2D-3D label transfer, and a CRF for 3D refinement. By using a CNN to carry out semantic segmentation of each input frame, the method can achieve a better runtime performance. However, the CNN still requires 51.2ms and the Bayesian update scheme requires a further 41.1ms, eventually running at 25.3Hz by applying these stages once every 10 input frames.

Other related works include [12], [13], [14] that aim at building a semantic 3D map, although not incrementally. [12] firstly builds a 3D map of a scene through RGB-D SLAM framework, then assigns class probabilities to each point of the 3D map by means of a Dense CRF. [13] exploits relational information derived from the full-scene 3D map for object labeling relying on a Markov-Random-Field (MRF)-based model.

In addition, several methods for recognizing only a part of the 3D map without making a dense semantic 3D map have been proposed [15], [16], [17], [18]. SLAM++ [15] maps indoor scenes at the level of semantically defined objects. Bowman et al. [16] improved the RGB SLAM performance in terms of camera pose and scale estimation by utilizing not only low-level geometric features such as points, lines, and planes but also detected objects as landmarks.

B. 2D semantic segmentation

Several CNN models [19], [20], [11], [21] for semantic segmentation have been proposed, sometimes yielding impressive results. To achieve a highly precise semantic segmentation map, such methods aim at exploiting global information and context to improve the features extracted by the convolutional layers. In particular, Fully Convolutional Network (FCN) [19] proposed a skip architecture that combines semantic information from a deep layer with appearance information from a shallow layer to perform accurate and detailed segmentation.

C. 3D geometric segmentation

On the other hand, 3D geometric segmentation algorithms have been developed, to extract geometrically separated segments from 3D data by unsupervised fashion. Real-time segmentation for depth map has been investigated by the works of Uckermann et. al. [22], [23], Pieropan et al. [24] and Abramov et al. [25]. As a consequence, in addition to frame-wise segmentation, [26], [6] has addressed the problem of real-time geometric segmentation for 3D point cloud or 3D mesh reconstructed via dense SLAM by incremental approach.

III. METHOD

Fig. 2 shows the flow diagram of our framework. The input is represented by RGB and depth frames obtained from a moving RGB-D sensor, which are processed individually.

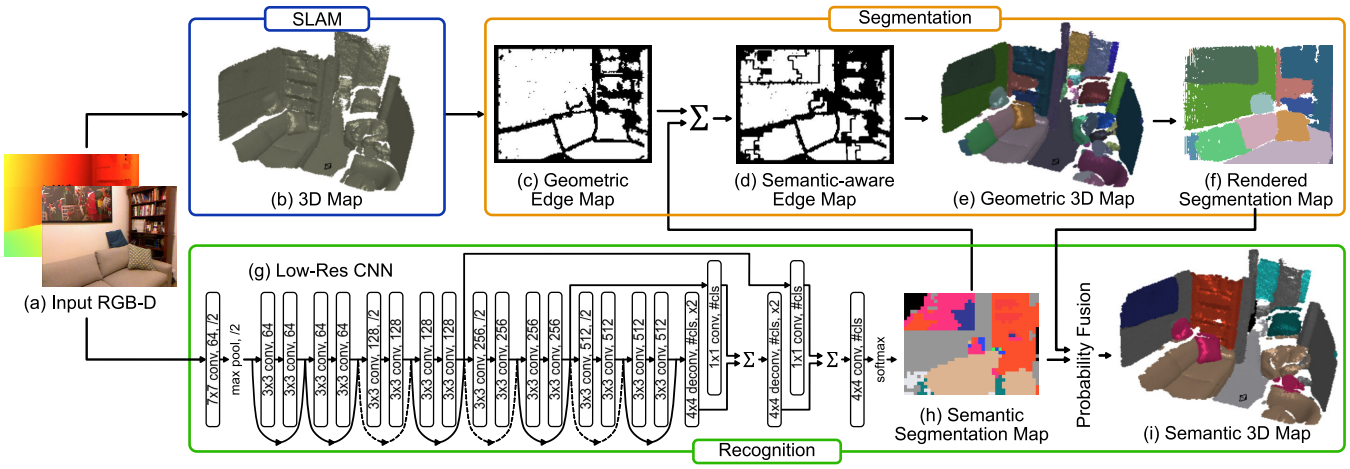

Fig. 2: Flow of the proposed framework. Efficient CNN-based semantic segmentation is exploited to refine the geometric edges on frame-wise segmentation, then it is efficiently fused in the SLAM-based model using the rendered viewpoint according to the estimated camera pose.

Our method has four components: SLAM framework, 2D semantic segmentation with a specifically designed CNN, incrementally building a geometric 3D map, and updating class probabilities assigned to each segment of the geometric 3D map. Firstly, SLAM and semantic segmentation with the CNN are performed simultaneously. In the segmentation stage, the geometric edge map is generated from the current depth frame and improved with edges extracted from the semantic segmentation result toward the semantic-aware representation. The geometric 3D map is updated through the edge map, and rendered to the current image plane. Finally, class probabilities assigned to each segmented region are updated with the rendered segmentation map. The following section describes these components in more detail.

A. SLAM

To carry out SLAM in terms of camera pose estimation and fusion we employ the dense approach of InfiniTAM v3 [5], relying on the efficient and scalable data representation proposed by Keller et al. [7], which uses a set *surfels* s_k to build the 3D map. As per this method, at the t-th incoming RGB-D frames, the current camera pose $\boldsymbol{T}_t \in \mathbb{SE}(3)$ is estimated through Iterative Closest Point [27] and RGB alignment. The new surfels generated from the current depth map are fused into the 3D map by means of the estimated camera pose, and are used to refine the 3D coordinates and normal associated to the existing surfels.

B. CNN architecture

The details of the CNN architecture proposed in our framework, Low-Res Net, are shown in Fig. 2 (g). The architecture combines concepts from state-of-the-art CNN models, i.e. Deep Residual Networks (ResNet) [28] and FCN [19]. Specifically, the original FCN architecture [19] utilizes the VGG model [29] to extract features and outputs a semantic segmentation result at the same resolution of the input image. On the other hand, Low-Res Net employs the ResNet architecture [28], which achieved higher accuracy than the VGG model [29] in ImageNet [30], and employs skip connections as done by FCN [19].

Towards the goal of achieving a fast forward pass, we do not incorporate multi-layered upsampling and design it only with two deconvolution layers with two strides. Therefore, given the input image $\mathcal{I}_t(\boldsymbol{u}), \boldsymbol{u} = (x, y) \subset \mathbb{Z}^2, 0 \leq x < W, 0 \leq y < H$, Low-Res Net outputs a semantic segmentation map in Fig. 2 (h) as a set of semantic class probabilities, i.e.

$$\tilde{\mathcal{S}}(\boldsymbol{v}) = \mathcal{P}(c|\mathcal{I}_t) \tag{1}$$

where $\boldsymbol{v} = (s, t) \subset \mathbb{Z}^2, 0 \leq s < W/8, 0 \leq t < H/8$. Here, $\mathcal{P}(c)$ denotes a class probability, where $\mathcal{P}(c) \subset \mathbb{R}, 0 \leq \mathcal{P}(c) \leq 1, c \subset \mathbb{Z}, 0 \leq c < N$ with N being the number of categories. The symbol $\tilde{}$ denotes instead hereinafter a map of size $H/8 \times W/8$. In our implementation, $H = 240$, $W = 320$, and the number of channels of the input image $\mathcal{I}$ is 3 as in ResNet [28].

C. Segmentation

Our geometrical segmentation scheme is based on the method proposed by Tateno et al. [6]. The method incrementally builds up a geometric 3D map, where a segmentation label l_i is associated with each surfel s_k, by properly propagating and merging segments extracted from the current depth map.

As a result, we obtain a binary geometric edge map $\mathcal{B}^g$ in Fig. 2 (c) from the input depth frame by comparing neighboring normal angles and vertex distances and by relying on a vertex and normal map as proposed in [6]. Here, $\mathcal{B}^g(\boldsymbol{u})$ takes 1 if $\boldsymbol{u}$ is on an edge and 0 for otherwise. It is important to point out that, while $\mathcal{B}^g$ is stable since those edges are extracted geometrically, edges between objects that do not present notable geometric characteristics (e.g., such as with two nearby flat objects) can not be extracted.

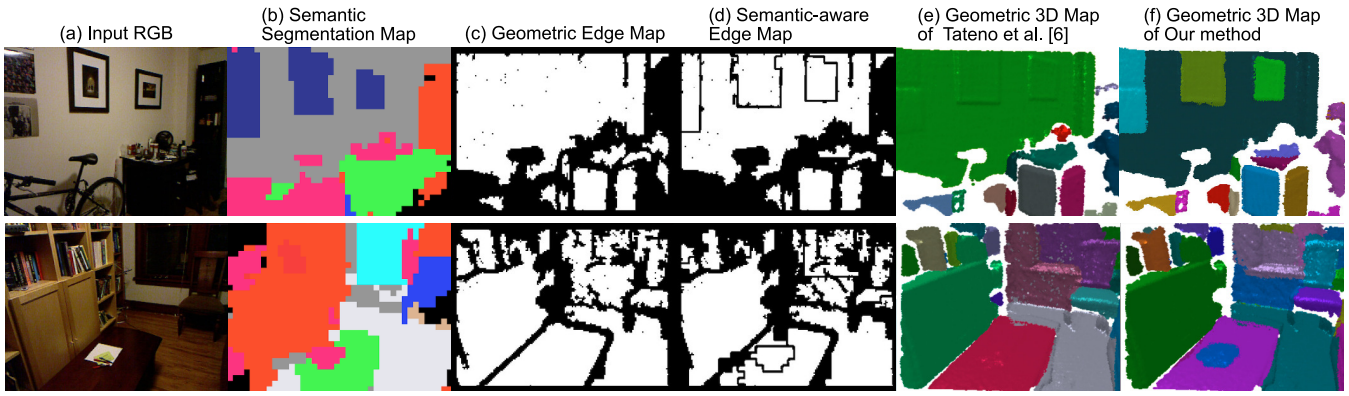

Fig. 3: Example results of our segmentation improvement scheme.

Differently from the geometric segmentation from [6], we introduce semantic information into the segments. First, we generate a class map $\tilde{\mathcal{S}}^c$, where each component $\tilde{\mathcal{S}}^c(\boldsymbol{v})$ has a class category c, with

$$\tilde{\mathcal{S}}^c(\boldsymbol{v}) = \arg\max_c \tilde{\mathcal{S}}(\boldsymbol{v}) = \mathcal{P}(c|\mathcal{I}_t). \tag{2}$$

After applying a median filter to $\tilde{\mathcal{S}}^c$ to remove isolated points, we resize $\tilde{\mathcal{S}}^c$ to $\mathcal{S}^c$ with a nearest neighbor interpolation. We would like to point out that the choice of such an efficient interpolation approach over a higher quality resizing such as bilinear interpolation is motivated by the fact that contours of a CNN-based semantic segmentation map are often imprecise, hence a better interpolation method would not yield benefits in terms of accuracy. At the same time, noise in the segment contours is eventually averaged out by the employed confidence-based label fusion approach. Then, we generate a binary semantic edge map $\mathcal{B}^s$ with the following scheme:

$$\mathcal{B}^s(\boldsymbol{u}=(x,y)) = \begin{cases} 1 & \text{if } \begin{array}{l} \mathcal{S}^c(x,y) \neq \mathcal{S}^c(x+1,y) \vee \\ \mathcal{S}^c(x,y) \neq \mathcal{S}^c(x,y+1) \vee \\ \mathcal{S}^c(x,y) \neq \mathcal{S}^c(x+1,y+1) \end{array} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The final binary semantic-aware edge map $\mathcal{B}$, (d) in Fig. 2, is obtained by applying a binary OR operator between $\mathcal{B}^g$ and $\mathcal{B}^s$.

In Fig. 3, the geometric edge map in (c) and the semantic-aware edge map in (d) show the benefit of our segmentation improvement scheme. Edges between objects which have poor geometric characteristics (i.e., wall and picture in the upper row and desk and paper in the bottom row) are successfully merged to the edge map.

Similar to [6], segments of the semantic-aware edge map $\mathcal{B}$ are properly extracted by means of a connected component algorithm and utilized for incrementally propagating and merging into the geometric 3D map with the estimated camera pose $\boldsymbol{T}_t$.

D. Probability fusion

Conventional methods assign class probabilities to each element that composes the 3D map [4], [1], [2], [3], [10], [8]. Conversely, we propose to assign class probabilities to each segmentation label l_i associated to each region constituting the geometric 3D map. With our approach, each label l_i is assigned to a discrete probability distribution $\mathcal{P}(c|\mathcal{I}_{1\dots t})$ and to a probability confidence Γ. $\mathcal{P}(c|\mathcal{I}_{1\dots t})$ is initialized to 0 over all class probabilities and Γ is also initialized to 0. Therefore, the space complexity for storing class probabilities is $O(N \cdot N_l)$, where N_l denotes the number of segmentation labels, in contrast to conventional methods [4], [1] which require $O(N \cdot N_s)$, where N_s is the number of elements of the 3D map (e.g., the number of surfels). This is an important difference in terms of scalability since typically $N_s \gg N_l$. This also appears as a more natural approach, since it could be argued that humans recognize objects by assigning semantic labels in a region-wise manner rather than element-wise.

In order to fuse the output of the Low-Res CNN properly with the 3D map, we update class probabilities assigned to each segmentation label l_i using a confidence-based approach. Firstly, we render the updated geometric 3D map onto the current image plane using the estimated camera pose $\boldsymbol{T}_t$ and the 3D position $\boldsymbol{x}(k)$ associated with each surfel s_k. The rendered segmentation map $\mathcal{L}(\boldsymbol{u})$, where each component is associated to a segmentation label l_i, is generated with $\mathcal{L}(\pi({\boldsymbol{T}_t}^{-1}\boldsymbol{x}(k))) = l_i(k)$ by denoting the segmentation label l_i of a surfel s_k with $l_i(k)$. Here, $\mathcal{L}(\boldsymbol{u})$ takes ϕ on the pixel $\boldsymbol{u}$ which is not filled with a label l_i.

Although the CNN-based semantic segmentation used in our framework is fast, its output $\tilde{\mathcal{S}}$ has a low resolution. Using the rendered segmentation map $\mathcal{L}$ whose size is $H \times W$ (i.e. the size of input image), detailed information is introduced to $\tilde{\mathcal{S}}$ to update the class probabilities of each label l_i with the following update scheme.

First, a set $\mathcal{C}_{\boldsymbol{v}}$ and a set $\mathcal{C}_{\boldsymbol{v},l_i}$ are defined as

$$\begin{aligned} \mathcal{C}_{\boldsymbol{v}=(s,t)} = \big\{\boldsymbol{u}=(x,y) \subset \mathbb{Z}^2 | \mathcal{L}(\boldsymbol{u}) \neq \phi \wedge \\ 8s \leq x < 8(s+1) \wedge 8t \leq y < 8(t+1)\big\} \end{aligned} \tag{4}$$

and

$$\mathcal{C}_{\boldsymbol{v},l_i} = \big\{\boldsymbol{u} \subset \mathcal{C}_{\boldsymbol{v}} | \mathcal{L}(\boldsymbol{u}) = l_i\big\}. \tag{5}$$

TABLE I: Quantitative results for the NYUv2 dataset [9]. These results were captured immediately after processing the frame. All accuracy evaluations were performed at 320×240 resolution. We calculated these accuracies with the same strategies as [1]. Ours-Geometric-Only denotes the method of building the geometric 3D map without our segmentation improvement scheme.

Method	bed	books	ceiling	chair	floor	furniture	objects	painting	sofa	table	tv	wall	window	**class avg.**	**pixel avg.**
Hermans et al. [4]	68.4	45.4	**83.4**	41.9	91.5	37.1	8.6	35.8	28.5	27.7	38.4	71.8	46.1	48.0	54.3
RGBD-SF [1]	61.7	**58.5**	43.4	58.4	92.6	63.7	**59.1**	66.4	47.3	34.0	33.9	86.0	60.5	58.9	67.5
RGBD-SF-CRF [1]	62.0	58.4	43.3	59.5	**92.7**	64.4	58.3	65.8	48.7	34.3	34.3	86.3	62.3	59.2	67.9
Eigen-SF [1]	47.8	50.8	79.0	73.3	90.5	62.8	46.7	64.5	45.8	46.0	70.7	88.5	55.2	63.2	69.3
Eigen-SF-CRF [1]	48.3	51.5	79.0	**74.7**	90.8	63.5	46.9	63.6	46.5	45.9	**71.5**	**89.4**	55.6	**63.6**	69.9
Li et al. [2]	64.9	34.6	72.0	67.5	90.5	65.0	17.2	**67.3**	59.3	41.3	60.0	85.1	57.0	60.3	70.3
Ours-Geometric-Only	**83.7**	6.4	32.0	52.8	83.1	73.5	40.0	4.3	75.3	**56.6**	53.1	75.0	50.2	52.8	66.9
Ours	**83.7**	15.6	24.4	56.7	83.3	**76.1**	52.5	40.8	**77.7**	53.0	57.3	75.3	**64.4**	58.5	**70.7**

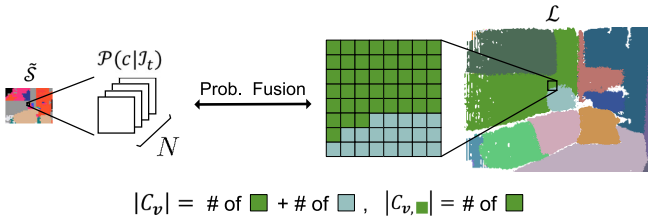

Fig. 4: Example definition of set $\mathcal{C}_{\boldsymbol{v}}$ and $\mathcal{C}_{\boldsymbol{v},l_i}$.

In words: $\mathcal{C}_{\boldsymbol{v}}$ is a set of coordinates to which the labels are assigned in the region of $\mathcal{L}(\boldsymbol{u})$ corresponding to $\tilde{\mathcal{S}}(\boldsymbol{v})$, while $\mathcal{C}_{\boldsymbol{v},l_i}$ is a set of coordinates to which the label l_i is assigned (See Fig. 4).

When the set $\mathcal{U}_{\boldsymbol{v}}$ of labels l_i which is included in the region of $\mathcal{L}(\boldsymbol{u})$ corresponding to $\tilde{\mathcal{S}}(\boldsymbol{v})$ is defined as

$$\mathcal{U}_{\boldsymbol{v}=(s,t)} = \big\{ l_i = \mathcal{L}(x,y) \subset \mathbb{Z} | 8s \leq x < 8(s+1) \wedge 8t \leq y < 8(t+1) \big\}, \tag{6}$$

the class probabilities $\mathcal{P}(c|\mathcal{I}_{1\ldots t})$ and the probability confidence Γ of each element $l \subset \mathcal{U}_{\boldsymbol{v}}$ are updated through

$$\mathcal{P}(c|\mathcal{I}_{1\ldots t}) \leftarrow \frac{1}{Z} \cdot \frac{\Gamma \mathcal{P}(c|\mathcal{I}_{1\ldots t\ \ 1}) + \gamma \mathcal{P}(c|\mathcal{I}_t)}{\Gamma + \gamma}$$
$$\Gamma \leftarrow \Gamma + \gamma, \ \ \gamma = \frac{|\mathcal{C}_{\boldsymbol{v},l}|}{|\mathcal{C}_{\boldsymbol{v}}|} \tag{7}$$

which is applied to all class probabilities. Here, the constant Z is for normalizing class probabilities to a proper distribution. With this scheme, the weight of the probability which cross over two or more segment regions (e.g., wall and object in Fig. 4) is reduced. By applying the same strategy to all $\boldsymbol{v}$ constituting $\tilde{\mathcal{S}}(\boldsymbol{v})$, we update class probabilities of all labels included in the rendered segmentation map $\mathcal{L}(\boldsymbol{u})$.

Therefore, letting the size of $\tilde{\mathcal{S}}(\boldsymbol{v}), H/8 \times W/8$ be $\tilde{H} \times \tilde{W}$, the time complexity for updating class probabilities is $\mathcal{O}(\tilde{H}\tilde{W}(8 \times 8 + |\mathcal{U}_{\boldsymbol{v}}|N))$, which means calculating set $\mathcal{C}_{\boldsymbol{v}}$, $\mathcal{C}_{\boldsymbol{v},l_i}$, and $\mathcal{U}_{\boldsymbol{v}}$ takes 8×8 and updating all class probabilities N assigned to each label in $\mathcal{U}_{\boldsymbol{v}}$ takes $|\mathcal{U}_{\boldsymbol{v}}|N$. Note that conventional methods [4], [1], [3], [10] take $\mathcal{O}(HWN)$ for updating class probabilities of the 3D map with a frame-wise recognition.

IV. EXPERIMENTS

A. Dataset and implementation

We evaluate our system on the common NYUv2 dataset [9]. The dataset contains 206 test set video sequences, however, for a fair comparison, we picked up 140 test sequences having a frame-rate over 2Hz which is the same as [1]. Since our Low-Res CNN outputs semantic segmentation with the size of $W/8 \times H/8$, we resized the ground truth $\mathcal{S}_{gt}$ to $\tilde{\mathcal{S}}_{gt}$ by filling $\tilde{\mathcal{S}}_{gt}(\boldsymbol{v})$ with the label which mostly occupies the area of $\mathcal{S}_{gt}(\boldsymbol{u})$ corresponding to $\tilde{\mathcal{S}}_{gt}(\boldsymbol{v})$. After training our Low-Res Net with the MS COCO dataset [31] for 10 epochs, we fine-tuned the network with the training dataset of the NYUv2 dataset [9] for 50 epochs. These evaluations are conducted on an Intel Core i7-5557U 3.1GHz CPU, GeForce GTX 1080 GPU, and 16GB RAM.

B. Accuracy

In this section, we experimentally demonstrate the accuracy of our method by quantitatively comparing the accuracy with other state-of-the-art methods through Table I. Additionally, Fig. 5 and Fig. 6 show qualitative results of our dense semantic mapping.

As shown in Table I, our method achieves 0.8% higher average pixel accuracy compared to SemanticFusion [1] and 0.4% higher average pixel accuracy compared to Li et al. [2]. As it can be noted, our method is particularly capable of outperforming other semantic mapping methods for object categories characterized by a big size. For the class *bed*, there is a significant accuracy increase of 15.3% over the state of the art; while, for the class *furniture* and *sofa*, we achieve 11.1% and 18.4% improvement, respectively. The reason why we achieve high accuracy especially on such categories is that our segmentation strongly relies on geometric information, and geometric boundaries associated to these categories (e.g., *bed* and *wall* and *floor* and *furniture*) are often quite clear.

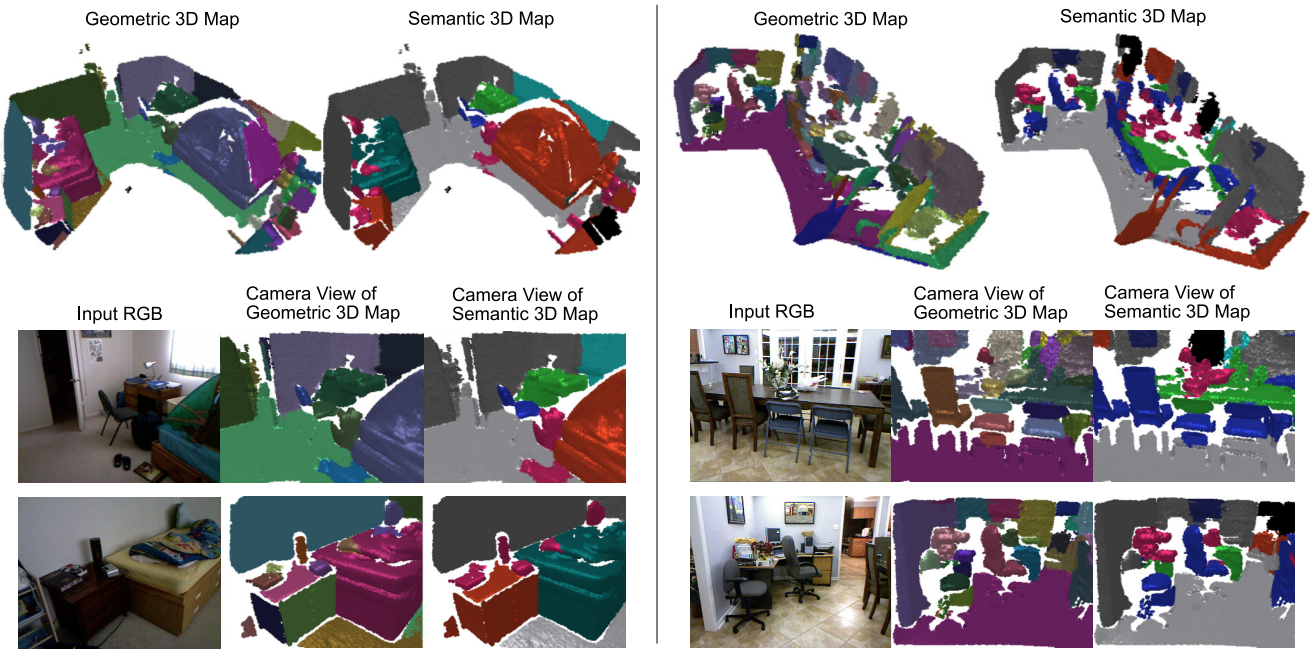

Fig. 5: Qualitative results of our dense 3D semantic mapping on two scenes (left: *bedroom_0112*, right: *dining_room_0017*). See Table I for class colors.

Fig. 6 shows the benefit of the segmentation improvement from the viewpoint of accuracy compared with "Ours-Geometric-Only", where we build the geometric 3D map without our segmentation improvement scheme. Particularly in the upper three rows, the *paintings* and the *window* on the *wall*, which are difficult to distinguish only with the geometric-based segmentation, are also segmented and annotated correctly. The geometric 3D map in Fig. 5 also shows the validity of the segmentation improvement especially on the above-mentioned regions. The example results of building a geometric 3D map with/without segmentation improvement are in Fig. 3 (e) geometric 3D map of Tateno et al. [6] and (f) geometric 3D map of our method. We achieved semantic-aware representation rather than the geometric-only incremental segmentation method [6]. This improved segmentation scheme allows achieving higher accuracy in terms of pixel average than state-of-the-art methods. As shown in Table I, the accuracies of the class *painting* and *window* are significantly improved for 36.5% and 14.2%, respectively, and 3.8% for overall categories between "Ours" and "Ours-Geometric-Only".

The lower two rows of Fig. 6 show failure cases. Since our method mainly extracts edges from the vertex and normal map obtained from the incoming depth image, it is difficult to successfully segment distant objects where depth values tend to be unstable (i.e., the third row of Fig. 6) and manage scenes where many small objects are lined up where vertices and normals are cluttered (i.e., the fourth row of Fig. 6). In Table I, this is the same reason why the categories of small objects such as *book* and *objects* score low accuracies. We leave the exploration of improving these limitation to future work.

TABLE II: Comparison of run-time performance. FQ denotes the frequency to perform a recognition of the input frame and update class probabilities of the 3D map.

Method	3D map	FQ	FPS
Hermans et al. [4]	Dense	every 6 frames	3.9 - 4.6 Hz
SemanticFusion [1]	Dense	every 10 frames	25.3 Hz
Yang et al. [3]	Dense	every frame	2 Hz
Li et al. [2]	Semi-Dense	every key-frame	10 Hz
Ours	Dense	every frame	**30.9 Hz**

TABLE III: Average time spent on each processing stage. processing for segmentation are in line 2-4 and processing for recognition are in 5-7. Note that the processing with * and the processing with ** can be processed simultaneously.

Component	Consumed time
SLAM *	8.13 ms
Generate a binary geometric edge map $\mathcal{B}^g$ *	1.04 ms
Segmentation improvement	0.39 ms
Update the geometric 3D map	8.74 ms
Low-Res CNN **	19.32 ms
Generate a rendered segmentation map $\mathcal{L}$	2.52 ms
Probability fusion	1.37 ms
Total	32.34 ms

C. Computational cost

In this section, we demonstrate the advantage of reducing the computational complexity, i.e. one of the main contributions of this method. We quantitatively compare the run-time

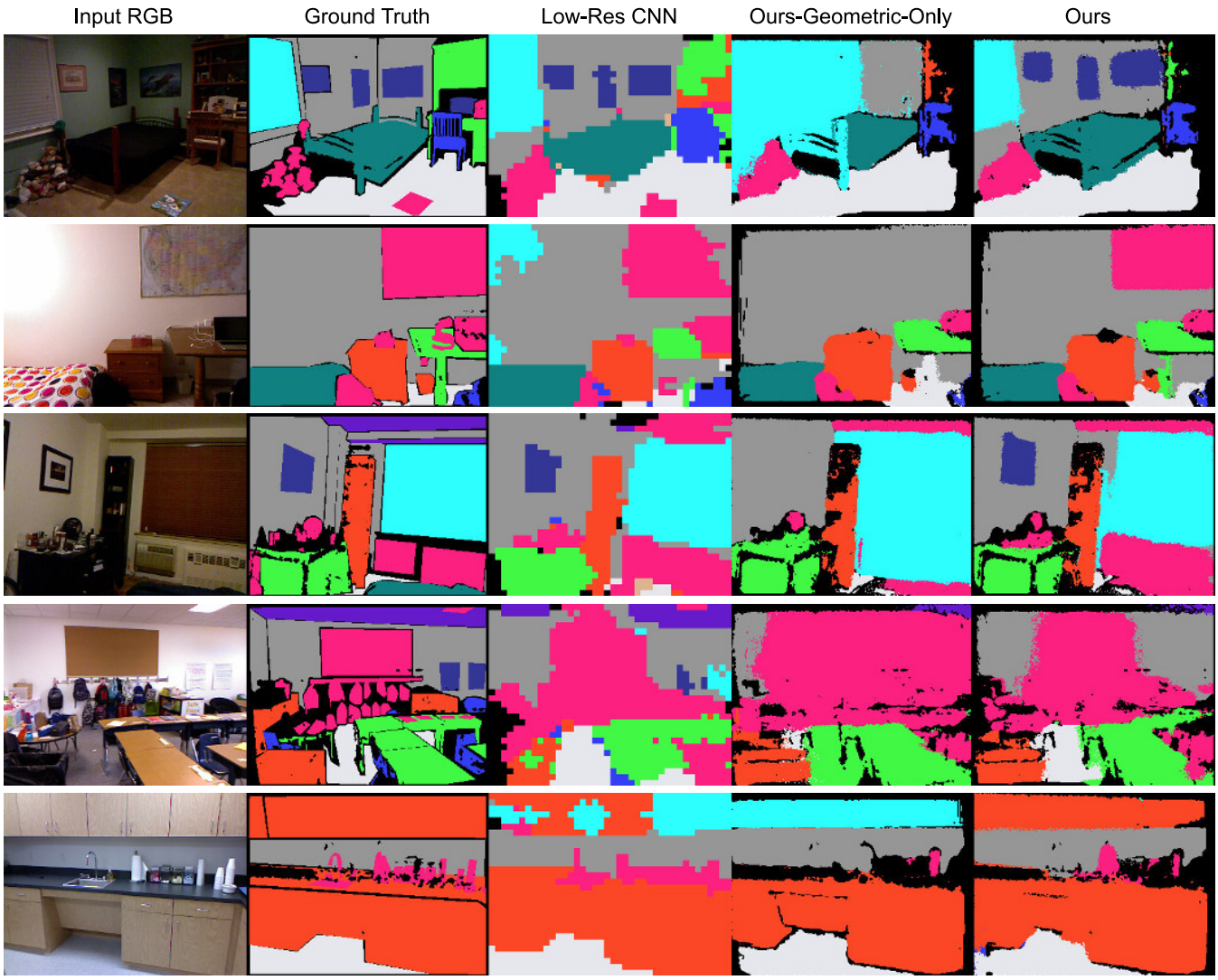

Fig. 6: Qualitative results for the NYUv2 dataset [9]. As with Table I, Ours-Geometric-Only denotes the method of building the geometric 3D map without our segmentation improvement scheme. See Table I for class colors.

performance with state-of-the-art approaches through Table II.

As shown in Table II, we achieved real-time performance (i.e., over 30Hz) while performing all processing components on every input frame. As analyzed in the last paragraph of Sec. III-D, the time complexity for updating class probabilities of the 3D map (i.e., Probability fusion) is $\mathcal{O}(\tilde{H}\tilde{W}(8 \times 8 + |\mathcal{U}_v|N))$. Considering the average number of $|\mathcal{U}_v|$ was 1.28 through the experiments, the average time complexity $\mathcal{O}(\tilde{H}\tilde{W}(8 \times 8 + |\mathcal{U}_v|N))$ turns into $\mathcal{O}(\tilde{H}\tilde{W}(8 \times 8 + N)) = \mathcal{O}(HW + \tilde{H}\tilde{W}N)$ in contrast to the one of conventional methods $\mathcal{O}(HWN)$ [4], [1], [3], [10]. Therefore, as shown in Table III, updating class probabilities of the 3D map only took 1.37ms on average, whereas SemanticFusion [1] spent 41.1ms for the processing. Furthermore, the processing for 2D recognition (i.e., Low-Res CNN) only took 19.32ms while maintaining high accuracy in the end, as mentioned in Section. IV-B.

Lastly, we discuss about the results of reducing the space complexity through Fig. 7. As shown there, the memory usage of our method is significantly reduced compared to the one of SemanticFusion [1] over all frames. The average memory usage of our method is 0.08% of the one of SemanticFusion [1]. The reason for this significant improvement is that, as mentioned in Sec. III-D, the space complexity of our method is $O(N \cdot N_l)$ whereas SemanticFusion takes $O(N \cdot N_s)$, where N_l and N_s were 1032 and 844260 in the end of the scene respectively.

V. CONCLUSION

In this paper, we proposed an efficient semantic mapping approach by assigning class probabilities to each region of the geometric 3D map which is incrementally built up through a robust SLAM framework and a geometric-based incremental segmentation. Through our experiments, we demonstrated that our approach notably compressed the

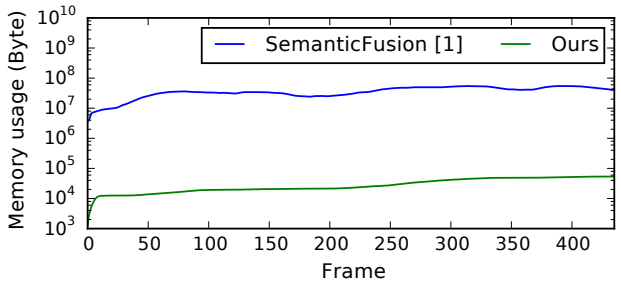

Fig. 7: Comparison of memory usage for storing class probabilities with SemanticFusion [1] on the scene *bedroom_0112* of the NYUv2 dataset [9].

computational complexity in terms of both of time and space while achieving comparable accuracy against state-of-the-art approaches without any post-processing to the semantic 3D map. Furthermore, we confirmed that our strategy improved the incremental segmentation framework beyond the geometric only to the semantic-aware representation.

ACKNOWLEDGMENT

This work was partially supported by JST CREST "Intelligent Information Processing Systems Creating Co-Experience Knowledge and Wisdom with Human-Machine Harmonious Collaboration."

REFERENCES

[1] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4628–4635, IEEE, 2017.

[2] X. Li and R. Belaroussi, "Semi-dense 3d semantic mapping from monocular slam," *arXiv preprint arXiv:1611.04144*, 2016.

[3] S. Yang, Y. Huang, and S. Scherer, "Semantic 3d occupancy mapping through efficient high order crfs," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[4] A. Hermans, G. Floros, and B. Leibe, "Dense 3d semantic mapping of indoor scenes from rgb-d images," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2631–2638, IEEE, 2014.

[5] V. A. Prisacariu, O. Kähler, S. Golodetz, M. Sapienza, T. Cavallari, P. H. Torr, and D. W. Murray, "InfiniTAM v3: A Framework for Large-Scale 3D Reconstruction with Loop Closure," *ArXiv e-prints*, 2017.

[6] K. Tateno, F. Tombari, and N. Navab, "Real-time and scalable incremental segmentation on dense slam," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4465–4472, IEEE, 2015.

[7] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb, "Real-time 3d reconstruction in dynamic scenes using point-based fusion," in *International Conference on 3DTV-Conference*, pp. 1–8, IEEE, 2013.

[8] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg, "Joint semantic segmentation and 3d reconstruction from monocular video," in *European Conference on Computer Vision (ECCV)*, pp. 703–718, Springer, 2014.

[9] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision (ECCV)*, pp. 746–760, Springer, 2012.

[10] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kähler, D. W. Murray, S. Izadi, P. Pérez, *et al.*, "Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 75–82, IEEE, 2015.

[11] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1520–1528, 2015.

[12] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. Torr, "Urban 3d semantic modelling using stereo vision," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 580–585, IEEE, 2013.

[13] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," in *Advances in neural information processing systems*, pp. 244–252, 2011.

[14] Z. Zhao and X. Chen, "Building 3d semantic maps for mobile robots using rgb-d camera," *Intelligent Service Robotics*, vol. 9, no. 4, pp. 297–309, 2016.

[15] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1352–1359, IEEE, 2013.

[16] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1722–1729, IEEE, 2017.

[17] D. Gálvez-López, M. Salas, J. D. Tardós, and J. Montiel, "Real-time monocular object slam," *Robotics and Autonomous Systems*, vol. 75, pp. 435–449, 2016.

[18] N. Fioraio and L. Di Stefano, "Joint detection, tracking and mapping by semantic bundle adjustment," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1538–1545, IEEE, 2013.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015.

[20] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, vol. 39, no. 12, pp. 2481–2495, 2017.

[21] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[22] A. Uckermann, R. Haschke, and H. Ritter, "Realtime 3D segmentation for human-robot interaction," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[23] A. Uckermann, C. Elbrechter, R. Haschke, and H. Ritter, "3D scene segmentation for autonomous robot grasping," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, oct 2012.

[24] A. Pieropan and H. Kjellstrom, "Unsupervised object exploration using context,," in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2014.

[25] A. Abramov, K. Pauwels, J. Papon, F. Worgotter, and B. Dellen, "Depth-supported real-time video segmentation with the kinect," in *IEEE Workshop on Applications of Computer Vision (WACV)*, 2012.

[26] R. Finman, T. Whelan, M. Kaess, and J. J. Leonard, "Toward lifelong object segmentation from change detection in dense RGB-D maps," in *2013 European Conference on Mobile Robots, ECMR 2013 - Conference Proceedings*, pp. 178–185, 2013.

[27] K.-L. Low, "Linear least-squares optimization for point-to-plane icp surface registration," *Chapel Hill, University of North Carolina*, vol. 4, 2004.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 770–778, 2016.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems (NIPS)*, pp. 1097–1105, 2012.

[31] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollar, "Microsoft coco: Common objects in context," *arXiv preprint arXiv:1405.0312*, 2014.