IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 9, 2018, accepted December 5, 2018, date of publication December 17, 2018,
date of current version January 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2887022

Efficient Object-Oriented Semantic Mapping With Object Detector

YOSHIKATSU NAKAJIMA AND HIDEO SAITO
Department of Science and Technology, Keio University, Yokohama, Kanagawa 223-8522, Japan
Corresponding author: Yoshikatsu Nakajima (nakajima@hvrl.ics.keio.ac.jp)

This work was supported by JST CREST under Grant JPMJCR14E3 and Grant JPMJCR1683, Japan.

ABSTRACT Incrementally, building a 3D map in which object instances are semantically annotated has a wide range of applications, including scene understanding, human–robot interactions, and simultaneous localization and mapping extensions. Although researchers are developing efficient and accurate systems, these methods still face a critical issue: real-time processing, because the task requires a series of heavy processing components, e.g., camera pose estimation, 3D map reconstruction, and especially recognition. In this paper, we propose a novel object-oriented semantic mapping approach aiming at overcoming such issues by introducing highly accurate object-oriented semantic scene reconstruction in real time. For high efficiency, the proposed method employs a fast and scalable object detection algorithm for exploiting semantic information from the incoming frames. These outputs are integrated into geometric regions of the 3D map, which are carried by the geometric-based incremental segmentation method. The strategy of assigning class probabilities to each segmented region, not each element (e.g., surfels and voxels), notably reduces the computational cost, as well as the memory footprint. In addition to efficiency, by geometrically segmenting the 3D map first, clear boundaries between objects appear. We complementarily improve the geometric-based segmentation results beyond the geometric only to the semantic-aware representation. We validate the proposed method's accuracy and computational efficiency through experiments in a common office scene.

INDEX TERMS Semantic mapping, SLAM, segmentation, object detection.

I. INTRODUCTION

Incrementally building up a semantically annotated 3D map is a vital technology for the computer vision and robotics communities. As an annotated 3D map of indoor scenes has a wide range of applications, including human-robot interactions and mixed reality, much effort has been made to develop efficient and accurate systems for the goal of building each to its own map representation. Among them, an object-oriented map would be the most important and efficient representation considering it also could be applied to SLAM extensions that include object-level pose graph optimization and address dynamic objects in the scene.

Motivated by the recent developments of 2D semantic segmentation and object detection algorithms, many methods efficiently combine the algorithms with the state-of-the-art SLAM method for a highly accurate semantically segmented 3D map [1]–[7]. However, these conventional methods still face a critical issue, real-time performance, because such a system requires a series of processes with high computational costs, including 3D reconstruction, camera pose estimation, and especially, recognition. Basically, the recognition part can be divided into 2 steps, 2D recognition for incoming frames and updating the class probabilities of the reconstructed 3D map with the 2D recognition result. For instance, SemanticFusion [1] and MaskFusion [5] employ SegNet-based CNN [8] and Mask R-CNN [9] for the first step, respectively, and then update the class probabilities of each element (i.e., surfel and voxel) consisting the reconstructed 3D map. As these 2 steps have a huge time complexity, these methods suggested to only extract semantic information on a subset of the input frames. Although such a frame skipping strategy can improve the run-time performance, this method limits the range of application, since it tends to be inaccurate under fast camera motions. Furthermore, as these methods assign class probabilities to each element of the 3D map, the memory footprint for storing semantic information increases.

In this paper, we propose a novel semantic mapping framework for overcoming such issues by yielding highly accurate semantic scene reconstruction in real-time. In contrast to

conventional methods, which have a huge time complexity for 2D semantic segmentation for the incoming frames, we utilize a fast object detection method for understanding objects in the frame. To build up an accurately segmented 3D map, we incrementally build up a geometrically segmented 3D map first. Then, we annotate each segmented region with bounding boxes outputted by the fast object detector (i.e., YOLO v2 [10]). This strategy notably reduces the time complexity because at each new frame the probability distributions must be updated for the segments that are visible on the image plane from the current camera pose, in contrast to conventional methods that must update the probabilities for all surfels and voxels on the image plane. This strategy also notably reduces the space complexity because the probability distributions need to be stored only at each segment rather than at each element of the 3D map.

In addition to the computational efficiency, this approach carries high accuracy in terms of the shape reconstruction of the objects in the scene. In contrast to conventional methods [1]–[5], [7], which directly use the 2D semantic segmentation results for the segmentation of the reconstructed 3D map, by segmenting the 3D map geometrically, clear boundaries between objects appear. This geometric-based segmentation method does not require any priori models of the objects in the scene. Thus, considering the capability of the state-of-the-art object detection method (e.g., YOLO9000 [10]), the proposed framework can be carried out in any unseen indoor scenes.

In return, the semantic information improves geometric-based segmentation. One of the common and critical issues of geometric-based segmentation approaches including the Tateno *et al.*'s [11] method is that these methods sometimes over-segment objects (e.g., dividing a chair into a backrest and a seating surface). We introduce a novel strategy for improving the incremental segmentation framework beyond the geometric-only to the semantic-aware representation by jointing the geometric labels recognized as parts of the same object.

The overall framework can work in real-time on off-the-shelf architectures, while requiring low computational complexity compared to the state of the art. In addition, differently from other methods, including [1]–[3], [12], the proposed approach does not require any post-processing based on, e.g., a conditional random field (CRF), to refine the reconstructed semantic map. We demonstrate the effectiveness of the proposed approach on a common office scene and report its high accuracy, run-time performance, and memory footprints.

II. RELATED WORKS

A. SEMANTIC MAPPING

SLAM++ [13] of Salas et al. is one of the first approaches for object-oriented mapping. They employed point pair features to detect objects to reconstruct a 3D map at the level of objects rather than points and planes. Although the scheme enabled the SLAM system to extend to object-level loop closure, detailed 3D models of objects are required beforehand for preprocessing to learn the 3D feature descriptors. Reference [14] also requires a priori known 3D object models. Tateno *et al.* [15] used the OUR-CVFH feature descriptor [16] to match pre-learned objects with geometrically segmented regions that incrementally built up in TSDF volumes [17]. Stückler and Behnke [18] also mapped learned objects in the point cloud reconstructed with the dense SLAM system by extracting multi-resolution 3D shapes and textures.

Motivated by the impressive performance of state-of-the-art CNN models for 2D semantic segmentations, recent works for object-orient mapping have mainly focused on effectively combining them with SLAM systems [1]–[7]. Co-fusion [7] employs SharpMask [19] to mask object regions, aiming at tracking and reconstructing detailed shape of dynamically moved objects. In addition, several methods employ Mask R-CNN [4], [5]. MaskFusion [5] carries out tracking and dense reconstruction for moving instances in real-time, as well as for the background static map. Fusion++ [4] utilizes the same prediction model but aims at extending the SLAM system by means of object-level pose graph optimizations and relocalizations. On the other hand, the huge processing time required for the forward pass of these very deep CNN models disrupt these systems to carry out in real-time. Thus, many methods [1]–[6] suggest extracting semantic information on only subset of the input frames.

Sünderhauf *et al.* [6]'s method is closely related to the proposed method. The method achieves object-oriented semantic mapping by projecting key depth frames, in which regions are segmented geometrically and annotated with Single Shot Multi-box Detector (SSD) [20], based on the camera pose estimated by ORB-SLAM2 [21]. Contrary to [6], the proposed method tracks each object instance in the scene with the geometric-based incremental segmentation method. Thus, semantic information from multiple views of the object region are integrated resulting in highly accurate object mapping.

In addition, several methods for recognizing an entire part of a 3D map have been proposed [1]–[3], [12]. SemanticFusion [1] employed the CNN model proposed by Noh *et al.* [8] for 2D semantic segmentation, a Bayesian framework for 2D-3D label transfer to fuse the 2D semantic segmentation labels to the 3D map, and a CRF for 3D map refinement as post-processing. Similar to the existing object-oriented mapping method, the forward pass of the CNN model is the crucial bottleneck, thus the method feeds an input frame to the CNN model once every 10 input frames to achieve real-time performance.

B. GEOMETRIC-BASED INCREMENTAL SEGMENTATION

A series of 3D geometric segmentation algorithms have been proposed to extract geometrically separated regions from incoming 3D information in an unsupervised fashion. Segmentation for a depth map in real-time has been proposed by Uckermann *et al.* [22], [23], Pieropan and Kjellstrom [24],

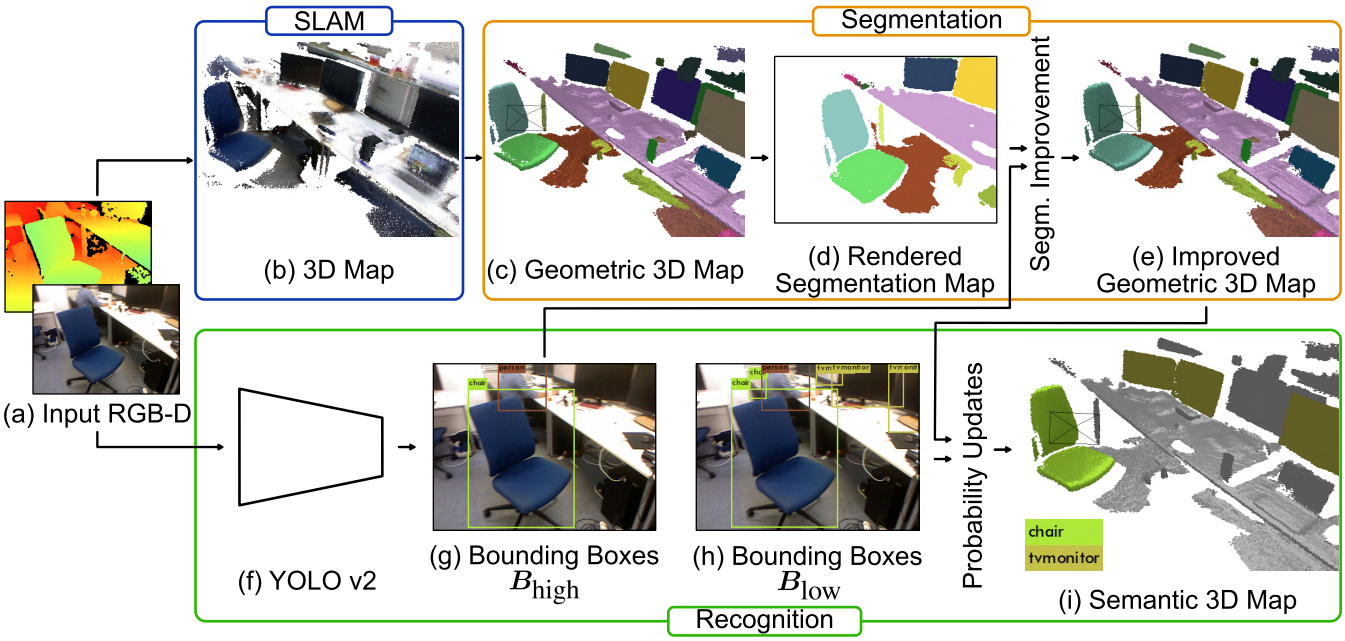

FIGURE 1. Flow of the proposed framework. Bounding boxes, outputted by the pre-trained object detector YOLO v2 [10], are used to update class probabilities assigned to each segmented region and improve the geometric-based segmentation results towards semantic-aware representations.

and Abramov *et al.* [25]. In addition to frame-wise segmentation, Tateno *et al.* [11] and Finman *et al.* [26] have developed real-time geometric-based segmentation methods for a 3D mesh and a 3D point cloud, respectively, reconstructed with dense SLAM in an incremental approach.

C. 2D RECOGNITION

Several CNN models [9], [10], [19], [20], [27], [28] for exploiting semantic information from an input image have been proposed, sometimes yielding impressive results. In particular, state-of-the-art object detectors, including YOLO v2 [10] and SSD [20], deliver an outstanding performance in terms of fast processing time for the forward pass, scalability, and accuracy.

III. METHOD

FIGURE 1 shows the flow diagram of the proposed framework. The input is represented by RGB and depth frames obtained from a free-moving RGB-D sensor, which are processed individually.

The proposed method has five components: the SLAM framework, 2D object detection, incremental building of a geometric 3D map, updating of the class probabilities assigned to each segment of the geometric 3D map, and improvement of the geometric 3D map with semantic information. In the following section, we describe these components in detail.

A. SLAM

The proposed system requires the camera pose in the target scene. In this work, we employ the dense approach of InfiniTAM v3 [29], relying on the efficient and scalable data representation proposed by Keller et al. [30], which uses a set of surfel s_k to build the 3D map. With 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 [31] and RGB alignment. The new surfels generated from the current depth map are fused into the 3D map with the estimated camera pose and are used to refine the 3D coordinates and normal associated with the existing surfels.

B. OBJECT DETECTION

In order to recognize object instances in the scene, first we apply the CNN-based object detector to the input image. Although there are many ways to recognize objects in the image, these methods are not sufficiently fast (e.g., Mask R-CNN takes 0.5 sec per image). However, recent works of the CNN-based object detection algorithm achieved real-time performance while showing high accuracy [10], [20].

In this work, we use the YOLO v2 [10], which has achieved impressive results on the established computer vision benchmarks, including MS COCO [32] and PASCAL VOC [33]. Given the input image $\mathcal{I}_t(\boldsymbol{u}), \boldsymbol{u} = (x, y) \subset \mathbb{Z}^2, 0 \leq x < W, 0 \leq y < H$, YOLO v2 [10] outputs a set of bounding boxes as $b_i, i \subset \mathbb{N}, 1 \leq i \leq M$, and class probabilities are assigned to each bounding box as $\mathcal{P}(c|\mathcal{I}_t) \subset \mathbb{R}$ by letting M be the number of bounding boxes and $c \subset \mathbb{Z}$ be the class category.

C. GEOMETRIC-BASED INCREMENTAL SEGMENTATION

Because the form of the bounding box is not enough to retrieve information on a 3D shape, we geometrically

segment the 3D map of the target scene toward the goal of precise object-oriented 3D map reconstruction. Then, we assign class probabilities of objects to each segment with the bounding boxes. We employ the unsupervised incremental segmentation approach proposed by Tateno *et al.* [11]. This method associates a segmentation label l_i with each surfel s_k, by properly propagating and merging segments extracted from the current depth map.

D. CLASS PROBABILITY UPDATES

Conventional methods assign class probabilities to each element that composes the 3D map. Conversely, we propose to assign class probabilities to each segmentation label l_i associated with each region constituting the geometric 3D map. With this 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, by letting N be the number of class categories, 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 [1], [12] 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, as 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 object detector properly with the 3D map, we update the class probabilities assigned to each segmentation label l_i using a confidence-based approach. First, we render the updated geometric 3D map on 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 with 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. Through the label map, we associate the bounding boxes b_i, the output of the object detector, with the segmented object in the reconstructed 3D model.

First, we define geometric-based bounding boxes $b_{l_i}^{\text{geo}}$ through the rendered label map as follows:

$$\begin{aligned} b_{l_i}^{\text{geo}}(\text{left}) &= \min\{x|\mathcal{L}(\boldsymbol{u}) = l_i, \boldsymbol{u} = (x, y) \subset \mathbb{Z}^2\} \\ b_{l_i}^{\text{geo}}(\text{right}) &= \max\{x|\mathcal{L}(\boldsymbol{u}) = l_i, \boldsymbol{u} = (x, y) \subset \mathbb{Z}^2\} \\ b_{l_i}^{\text{geo}}(\text{top}) &= \min\{y|\mathcal{L}(\boldsymbol{u}) = l_i, \boldsymbol{u} = (x, y) \subset \mathbb{Z}^2\} \\ b_{l_i}^{\text{geo}}(\text{bottom}) &= \max\{y|\mathcal{L}(\boldsymbol{u}) = l_i, \boldsymbol{u} = (x, y) \subset \mathbb{Z}^2\} \end{aligned} \tag{1}$$

As for the bounding box b_i outputted by YOLO v2 [10], we define set of bounding boxes $\boldsymbol{B}_{\text{low}}$ as follows:

$$\boldsymbol{B}_{\text{low}} = \{b_i | \max_c \mathcal{P}(c|\mathcal{I}_t) > \sigma_{\text{low}}\}, \tag{2}$$

Next, we find the bounding box $b_i \subset \boldsymbol{B}_{\text{low}}$ that maximizes the following equation to each label l_i in the rendered label map $\mathcal{L}$:

$$\text{IoU}(b_{l_i}^{\text{geo}}, b_i) = \frac{b_{l_i}^{\text{geo}} \cap b_i}{b_{l_i}^{\text{geo}} \cup b_i}. \tag{3}$$

Similarly for each element $b_i \subset \boldsymbol{B}_{\text{low}}$, we find the label l_i that minimizes equation (3) to obtain a set $\mathcal{U}$ of pairs (l_i, i) that match each other.

The class probabilities $\mathcal{P}(c|\mathcal{I}_{1\dots t})$ and the probability confidence Γ of l_i of each element $(l_i, i) \subset \mathcal{U}$ are updated with class probabilities $\mathcal{P}(c|\mathcal{I}_{1\dots t})$ of its pair b_i

$$\mathcal{P}(c|\mathcal{I}_{1\dots t}) \leftarrow \frac{1}{Z} \cdot \frac{\Gamma\mathcal{P}(c|\mathcal{I}_{1\dots t-1}) + \mathcal{P}(c|\mathcal{I}_t)}{\Gamma + 1}, \quad \Gamma \leftarrow \Gamma + 1, \tag{4}$$

which is applied to all class probabilities. Here, the constant Z is for normalizing the class probabilities to the proper distribution.

E. SEGMENTATION IMPROVEMENT

Since the geometric-based segmentation [11] described in section III-C is processed only with geometric information, some of the objects in the target scene might be divided into pieces (e.g., dividing a chair into a backrest and a seating surface). Here, we merge these pieces with semantic information which is provided in the form of a bounding box. The goal of this stage is to identify and merge these corresponding segments together.

With this goal, all possible label pairs $(l_a, l_b), l_a \neq l_b$ are associated with a confidence $m(l_a, l_b) \subset \mathbb{Z}$. If a pair (l_a, l_b) is identified for the first time, its associated confidence is initialized as follows: $m(l_a, l_b) = 0$. Next, the set of bounding boxes with high class probability $\boldsymbol{B}_{\text{high}}$ is defined as follows:

$$\boldsymbol{B}_{\text{high}} = \{b_i | \max_c \mathcal{P}(c|\mathcal{I}_t) > \sigma_{\text{high}}\}, \tag{5}$$

Then, we define a set Ψ of pair (l_a, l_b) for each bounding box $b_i \subset \boldsymbol{B}_{\text{high}}$ with the following conditions:

$$\Psi = \{(l_a, l_b) | \frac{|\mathcal{U}_{l_a}^{b_i}|}{|\mathcal{U}_{l_a}|} > \sigma_{\text{same}} \wedge \frac{|\mathcal{U}_{l_b}^{b_i}|}{|\mathcal{U}_{l_b}|} > \sigma_{\text{same}}\}. \tag{6}$$

Here, the set $\mathcal{U}_{l_i}$ and $\mathcal{U}_{l_i}^{b_i}$ are defined as follows:

$$\begin{aligned} \mathcal{U}_{l_i} &= \{\boldsymbol{u} = (x, y) \subset \mathbb{Z}^2 | \mathcal{L}(\boldsymbol{u}) = l_i\}, \\ \mathcal{U}_{l_i}^{b_i} &= \{\boldsymbol{u} = (x, y) \subset \mathcal{U}_{l_i} | x < b_i(\text{right}) \wedge x > b_i(\text{left}) \wedge \\ &\quad y < b_i(\text{bottom}) \wedge y < b_i(\text{top})\}. \end{aligned} \tag{7}$$

In words, $\mathcal{U}_{l_i}$ denotes the set of pixels assigned to the label l_i in $\mathcal{L}$, and $\mathcal{U}_{l_i}^{b_i}$ denotes the set of pixels assigned to the label l_i in the bounding box b_i. Thus, equation (7) verifies whether the label l_i is a part of an object instance by considering the ratio of the target label l_i in the bounding box b_i. We increment the confidence $m(l_a, l_b)$, which indicates l_a and l_b consist of the same object, of a pair $(l_a, l_b) \subset \Psi$ as follows:

$m(l_a, l_b) \leftarrow m(l_a, l_b) + 1$. If the confidence $m(l_a, l_b)$ exceeds the threshold σ_{merge}, we merge the label l_a with the label l_b only when the pair (l_a, l_b) satisfies the following condition:

$$\min |\boldsymbol{x}(k) - \boldsymbol{x}(s)| < \sigma_{\text{dist}}, \quad l_a(k) \subset \mathcal{L}(\boldsymbol{u}), \ l_b(s) \subset \mathcal{L}(\boldsymbol{u}). \tag{8}$$

Here, as mentioned in sections III-A and III-C, $\boldsymbol{x}(k)$, $\boldsymbol{x}(s)$ and $l_a(k)$, $l_b(s)$ denote the 3D position and the label of the surfel s_k and s_s, respectively. Condition (8) is based on the hypothesis that the labels l_a and l_b constituting the same object are separated by at most σ_{dist} in Euclidean distance.

IV. EXPERIMENTS

In this section, we evaluate the performance and efficiency of the proposed system. Following are the details of the evaluation environment: CPU: Intel Core i7-5557U 3.1GHz, GPU: GeForce GTX 1080Ti, and RAM: 16GB. We used the weights distributed on the official webpage of YOLO v2[1] [10], which is trained on the MS COCO dataset [32]. The setting of the threshold values is as follows: $\sigma_{\text{low}} = 0.24$, $\sigma_{\text{high}} = 0.6$, $\sigma_{\text{same}} = 0.9$, $\sigma_{\text{merge}} = 10$, and $\sigma_{\text{dist}} = 2.0$cm.

In order to evaluate our system, we captured 586 RGB-D frames in a common office with Kinect v1. We rescaled each input frame to 320×240 resolution as other methods did [1].

A. RUN-TIME PERFORMANCE AND MEMORY FOOTPRINT

In this section, we demonstrate the efficiency of the proposed method, which is one of the main contributions of the proposed method, by means of analysis of the run-time performance and the memory footprint. We quantitatively compare the run-time performance with state-of-the-art approaches and show the results in TABLE 1. As shown in TABLE 1, we achieved 27.2 Hz while performing all processing components on every input frame, while MaskFusion [5] and Fusion++ [4] achieved 30 Hz and 4–8 Hz by extracting semantic information on every 12 and 30 frames, respectively.

TABLE 1. Comparison of run-time performance. FQ denotes the frequency recognition of the input frame is performed and the class probabilities of the 3D map are updated.

Method	Representation	FQ	FPS
Hermans et al. [12]	Dense	Every 6 frames	3.9–4.6 Hz
SemanticFusion [1]	Dense	Every 10 frames	25.3 Hz
MaskFusion [5]	Object-oriented	Every 12 frames	30 Hz
Fusion++ [4]	Object-oriented	Every 30 frames	4–8 Hz
Ours	Object-oriented	Every frame	**27.2 Hz**

TABLE 2 shows the average time spent on each processing stage, described in section III, through all frames of the office sequence. Although the bottleneck is the process for exploiting the semantic information from the input frame (i.e., object detection) as in conventional methods [1]–[7], [12], the proposed method has drastically reduced the bottleneck considering that these conventional methods spend about 500 ms on the processing. As for processing the class probability updates and the segmentation improvement, we shorten the computational time by limiting the processing target to B_{low} and B_{high} which are the subset of all the bounding boxes outputted by YOLO v2 [10].

[1]https://pjreddie.com/darknet/yolov2/

TABLE 2. Average time spent on each processing stage. Note that the processing with * and the processing with ** can be processed simultaneously.

Component	Consumed time
SLAM *	9.54 ms
Geometric-based incremental segmentation *	5.22 ms
Object detection **	32.16 ms
Class probability updates	4.02 ms
Segmentation improvement	0.61 ms
Total	**36.78 ms**

FIGURE 2 shows the processing time spent on each frame. As shown, the proposed method demonstrates an almost constant complexity even if the size of the 3D map reconstructed with SLAM is increased. This is mainly caused by the proposed strategy of updating the class probabilities, where the processing target is limited to a subset of all bounding boxes, thus maintaining the complexity $\mathcal{O}(n^2)$ (i.e., the size of the input image) of the incremental geometric-based segmentation method [11]. However, several frames consumed huge processing time. On these frames, segmentation merging is performed, and processing for equation (8) had huge complexity.

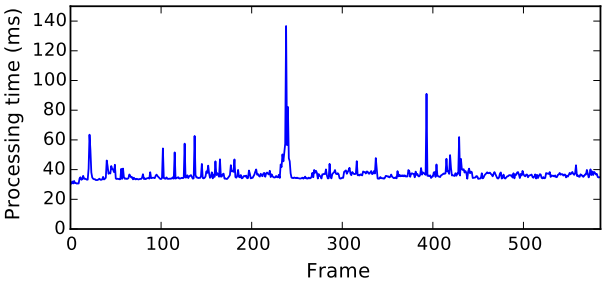

FIGURE 2. Processing time spent on each frame of the office sequence.

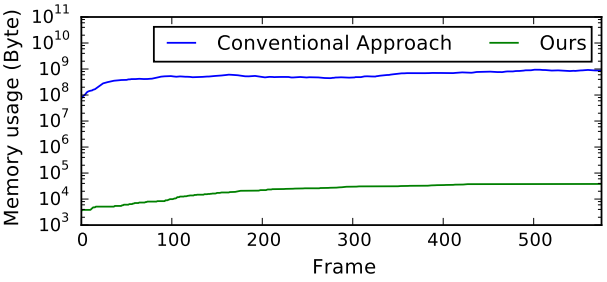

FIGURE 3. Comparison of memory usage for storing class probabilities with conventional approaches where class probabilities are assigned to each element of the 3D map [1], [3]–[5], [12], [34].

Last, we discuss the results of the memory footprint used for storing class probabilities as shown in FIGURE 3.

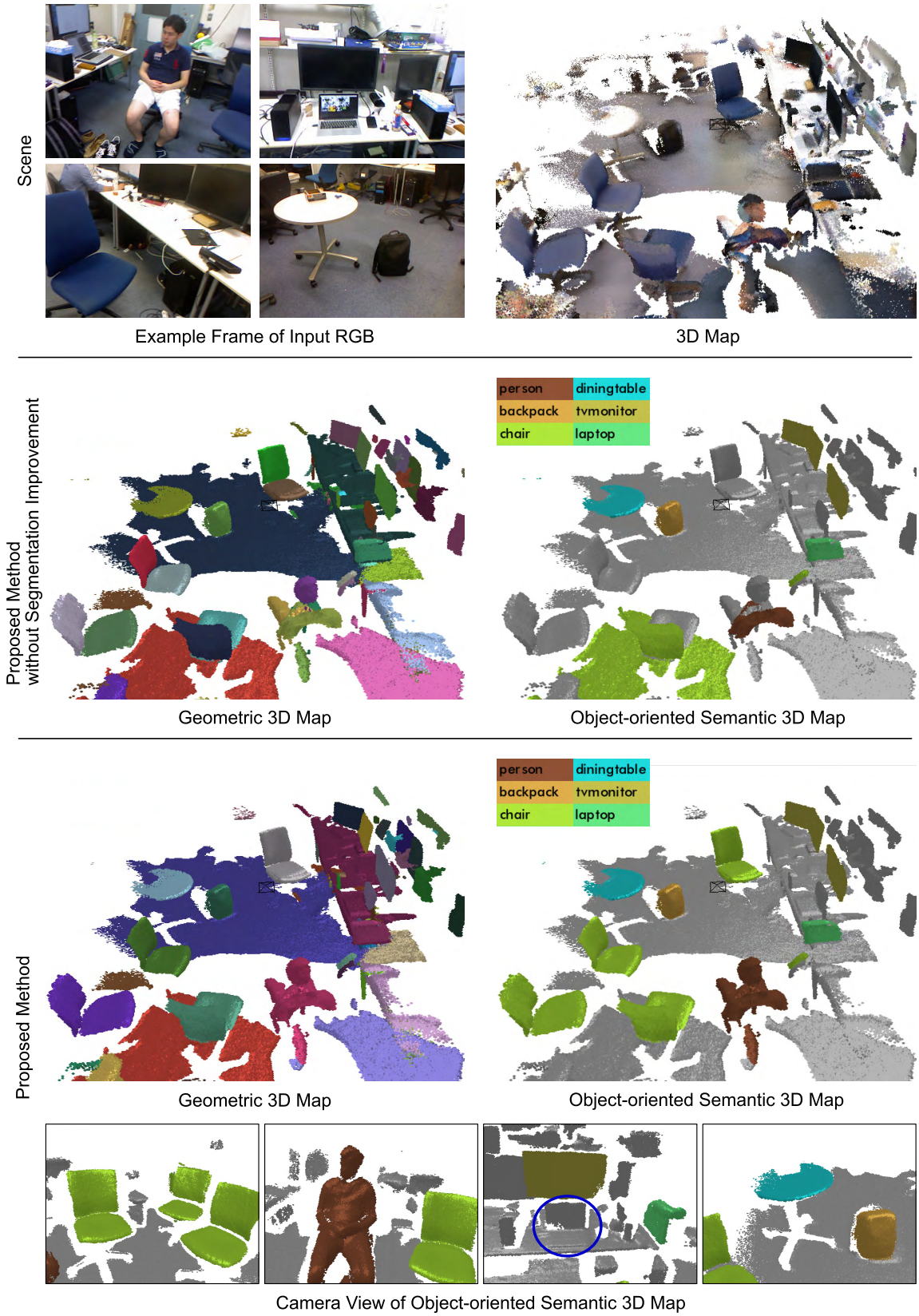

FIGURE 4. Qualitative results of the proposed object-oriented 3D semantic mapping method in an office scenario. In the geometric 3D map, different colors show different objects. Zoomed-in snapshots from the semantic 3D map of the proposed method are shown at the bottom.

We compared the proposed method with conventional approaches [1], [3]–[5], [12], [34], including MaskFusion [5] and Fusion++ [4], which assigns class probabilities to each element (e.g., surfels and voxels) of the 3D map rather than to each segment. As shown there, the memory usage of the proposed method is significantly reduced compared to the conventional approach over all frames. The average memory usage of the proposed method is 0.004% of those conventional approaches. The reason for this improvement is the significant reduction in the space complexity for storing class probabilities. As the proposed method stores class probabilities to each segmented region, the space complexity is $\mathcal{O}(N \cdot N_l)$, where N_l denotes the number of segmentation labels and N denotes the number of class categories, in contrast to conventional approaches which require $\mathcal{O}(N \cdot N_s)$, where N_s is the number of elements of the 3D map (e.g., the number of surfels). Here, N_l and N_s were 119 and 2748249 in the end of the scene, respectively. Thus, the memory usages for storing class probabilities of the proposed method and conventional approaches are 38KB and 879,440KB.

B. ACCURACY

FIGURE 4 shows the qualitative result of the object-oriented semantic 3D map reconstructed with the proposed method. As shown in FIGURE 4, we reconstructed the semantic map in which the 3D shapes of the object instances clearly appear by effectively merging geometric-based incremental segmentation and the object detector. In particular, by segmenting the reconstructed 3D model geometrically, clear boundaries between objects appear (e.g., the chair and floor in FIGURE 4), as witnessed by the close-up snapshot shown at the bottom of the figure.

In the geometric 3D map, the chairs and person were not divided into small pieces, and the 3D map was segmented in a semantic context compared with those without a segmentation improvement scheme. This scheme also contributed to the accuracy of the semantic 3D map compared to the semantic 3D map without segmentation improvement.

However, this method has the limitation. The accuracy of the semantic 3D map depends on the accuracy of the geometric segmentation because of the nature of this method: assigning class probabilities to each region which is mainly carried by the geometric-based segmentation method. As shown in FIGURE 4, the laptop on the desk was not segmented geometrically, thus resulting laptop was not mapped in the semantic 3D map (See the blue ellipse in the camera view of object-oriented semantic 3D map in FIGURE 4). We leave the exploration of improving this limitation to future work.

V. CONCLUSION

In this paper, we proposed an efficient method for object-oriented semantic mapping. The proposed method assigns class probabilities to each segment of the 3D map, which carried by incremental geometric-based segmentation method, with bounding boxes outputted by a fast object detector. In return, the segmented 3D map is improved with semantic information beyond the geometric-only to semantic-aware representations. With experiments, we confirmed that the proposed method has a capability of achieving highly accurate semantic mapping without priori 3D shapes of the objects. The proposed method also achieved 27.2 Hz while performing recognition for every input frame and notably reduced the memory footprints, thanks to the high efficiency that characterizes the computationally intensive stages of the proposed framework.

REFERENCES

[1] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Jun. 2017, pp. 4628–4635.
[2] X. Li and R. Belaroussi. (2016). "Semi-dense 3D semantic mapping from monocular SLAM." [Online]. Available: https://arxiv.org/abs/1611.04144
[3] S. Yang, Y. Huang, and S. Scherer, "Semantic 3D occupancy mapping through efficient high order CRFs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 590–597.
[4] J. McCormac, R. Clark, M. Bloesch, A. J. Davison, and S. Leutenegger. (2018). "Fusion++: Volumetric object-level SLAM." [Online]. Available: https://arxiv.org/abs/1808.08378
[5] M. Rünz and L. Agapito, "MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects." [Online]. Available: https://arxiv.org/abs/1804.09194
[6] N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5079–5085.
[7] M. Rünz and L. Agapito, "Co-fusion: Real-time segmentation, tracking and fusion of multiple objects," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Jun. 2017, pp. 4471–4478.
[8] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1520–1528.
[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[10] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.
[11] K. Tateno, F. Tombari, and N. Navab, "Real-time and scalable incremental segmentation on dense SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2015, pp. 4465–4472.
[12] A. Hermans, G. Floros, and B. Leibe, "Dense 3D semantic mapping of indoor scenes from RGB-D images," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Jun. 2014, pp. 2631–2638.
[13] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1352–1359.
[14] E. Herbst, X. Ren, and D. Fox, "RGB-D object discovery via multi-scene analysis," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2011, pp. 4850–4856.
[15] K. Tateno, F. Tombari, and N. Navab, "When 2.5 D is not enough: Simultaneous reconstruction, segmentation and recognition on dense SLAM," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 2295–2302.
[16] A. Aldoma, F. Tombari, R. B. Rusu, and M. Vincze, "OUR-CVFH—Oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6DOF pose estimation," in *Proc. Joint DAGM (German Assoc. Pattern Recognit.) OAGM Symp.* Graz, Austria: Springer, 2012, pp. 113–122.
[17] S. Izadi *et al.*, "KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. 24th Annu. ACM Symp. User Interface Softw. Technol.*, 2011, pp. 559–568.
[18] J. Stückler and S. Behnke, "Model learning and real-time tracking using multi-resolution surfel maps," in *Proc. AAAI*, 2012, pp. 2081–2087.
[19] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 75–91.

[20] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 21–37.
[21] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[22] A. Ückermann, R. Haschke, and H. Ritter, "Realtime 3D segmentation for human-robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2013, pp. 2136–2143.
[23] A. Ückermann, C. Elbrechter, R. Haschke, and H. Ritter, "3D scene segmentation for autonomous robot grasping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2012, pp. 1734–1740.
[24] A. Pieropan and H. Kjellström, "Unsupervised object exploration using context," in *Proc. 23rd IEEE Int. Symp. Robot Hum. Interact. Commun. (RO-MAN)*, Aug. 2014, pp. 499–506.
[25] A. Abramov, K. Pauwels, J. Papon, F. Worgotter, and B. Dellen, "Depth-supported real-time video segmentation with the Kinect," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2012, pp. 457–464.
[26] R. Finman, T. Whelan, M. Kaess, and J. J. Leonard, "Toward lifelong object segmentation from change detection in dense RGB-D maps," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, 2013, pp. 178–185.
[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[29] V. A. Prisacariu *et al.* (2017). "InfiniTAM v3: A framework for large-scale 3D reconstruction with loop closure." [Online]. Available: https://arxiv.org/abs/1708.00783
[30] 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 *Proc. Int. Conf. 3DTV-Conf.*, Jul. 2013, pp. 1–8.
[31] K.-L. Low, "Linear least-squares optimization for point-to-plane ICP surface registration," Univ. North Carolina, Chapel Hill, NC, USA, Tech. Rep., 2004, vol. 4.
[32] T.-Y. Lin *et al.* (2014). "Microsoft COCO: Common objects in context." [Online]. Available: https://arxiv.org/abs/1405.0312
[33] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.
[34] V. Vineet *et al.*, "Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 75–82.

YOSHIKATSU NAKAJIMA received the B.E. and M.Sc.Eng. degrees in information and computer science from Keio University, Japan, in 2016 and 2017, respectively. He is currently pursuing the Ph.D. degree in science and technology with Keio University, Japan. His research interests include computer vision, robotics, and simultaneous localization and mapping.

HIDEO SAITO received the Ph.D. degree in electrical engineering from Keio University, Japan, in 1992. Since 1992, he has been on the Faculty of Science and Technology, Keio University. From 1997 to 1999, he joined the Virtualized Reality Project at the Robotics Institute, Carnegie Mellon University, as a Visiting Researcher. Since 2006, he has been a Full Professor with the Department of Information and Computer Science, Keio University. His research interests include computer vision and pattern recognition, and their applications to augmented reality, virtual reality, and human–robotic interaction. His recent activities in academic conferences include being the Program Chair of ACCV 2014, the General Chair of ISMAR 2015, and the Program Chair of ISMAR 2016.