PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things

Gaku Narita, Takashi Seno, Tomoya Ishikawa, Yohsuke Kaji[1]

***Abstract*— We propose *PanopticFusion*, a novel online volumetric semantic mapping system at the level of *stuff* and *things*. In contrast to previous semantic mapping systems, PanopticFusion is able to densely predict class labels of a background region (*stuff*) and individually segment arbitrary foreground objects (*things*). In addition, our system has the capability to reconstruct a large-scale scene and extract a labeled mesh thanks to its use of a spatially hashed volumetric map representation. Our system first predicts pixel-wise panoptic labels (class labels for *stuff* regions and instance IDs for *thing* regions) for incoming RGB frames by fusing 2D semantic and instance segmentation outputs. The predicted panoptic labels are integrated into the volumetric map together with depth measurements while keeping the consistency of the instance IDs, which could vary frame to frame, by referring to the 3D map at that moment. In addition, we construct a fully connected conditional random field (CRF) model with respect to panoptic labels for map regularization. For online CRF inference, we propose a novel unary potential approximation and a map division strategy.**

We evaluated the performance of our system on the ScanNet (v2) dataset. PanopticFusion outperformed or compared with state-of-the-art offline 3D DNN methods in both semantic and instance segmentation benchmarks. Also, we demonstrate a promising augmented reality application using a 3D panoptic map generated by the proposed system.

I. INTRODUCTION

Geometric and semantic scene understanding in 3D environments has an important role in autonomous robotics and context-aware augmented reality (AR) applications. Geometric scene understanding such as visual simultaneous localization and mapping (SLAM) and 3D reconstruction has been widely discussed since the early days of both the robotics and computer vision communities. In recent years, semantic mapping, which not only reconstructs the 3D structure of a scene but also recognizes what exists in the environment, has attracted much attention because of the great progress of deep neural networks.

Semantic mapping systems could take a variety of approaches in terms of geometry and semantics. When we think about robotic and AR applications that deeply interact with the real world, what kind of properties are required for the ideal semantic mapping system? In terms of geometry, it needs to be able to reconstruct a large-scale scene, not sparsely but densely. Additionally, the 3D reconstruction desirably needs to be represented as a volumetric map, not just point clouds or surfels, because it is difficult to directly utilize point clouds and surfels for robot–object collision detection or robot navigation. In terms of semantics, which we mainly focus on in this paper, we believe that it is important for the mapping system to have a *holistic* scene understanding capability, that is to say, dense semantic labeling as well as individual object discrimination. This is because densely labeled semantics is a crucial cue for intelligent robot navigation, and also, discriminating individual objects is essential for robot–object interaction.

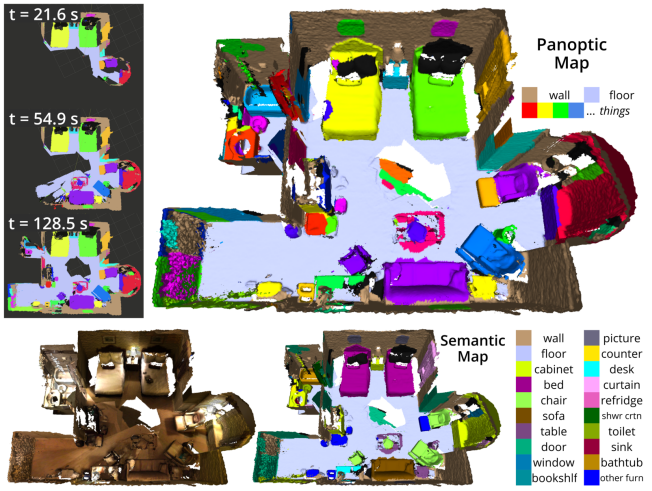

Fig. 1. PanopticFusion realizes an online volumetric semantic mapping at the level of *stuff* and *things*. The system performs large-scale 3D reconstruction, as well as dense semantic labeling on *stuff* regions and segmentation of individual *things* in an online manner, as shown in the top figure. It is also able to restore the class labels of *things* and yield a colored mesh, as shown in the bottom figures. The results obtained with scene0645_01 of ScanNet v2 are shown.

Turning our eyes to the field of 2D image recognition, an image understanding task called *panoptic* segmentation has been proposed recently [11]. In the panoptic segmentation task, semantic classes are defined as a set of *stuff* classes (amorphous regions, such as floors, walls, the sky and roads) and *thing* classes (countable objects, such as chairs, tables, people and vehicles) and one needs to predict class labels on *stuff* regions and both class labels and instance IDs on *thing* regions, where the predictions should be performed for each pixel. Extending this point of view to 3D mapping, in this paper we propose the *PanopticFusion* system. To the best of our knowledge, it is the first semantic mapping system that realizes scene understanding at the level of *stuff* and *things*. Our system incrementally performs large-scale 3D surface reconstruction online, as well as dense class label prediction on the background region and segmentation and recognition of individual foreground objects, as shown in Fig. 1.

Our approach first passes the incoming RGB frame to 2D

[1]The authors are with R&D Center, Sony Corporation. {gaku.narita, takashi.seno, tomoya.ishikawa, kaji.yohsuke}@sony.com

semantic and instance segmentation networks and obtains a panoptic label image in which class labels are assigned to *stuff* pixels and instance IDs to *thing* pixels. The predicted panoptic labels and depth measurements are integrated into the volumetric map. Before integration, we keep the consistency of instance IDs, which possibly change from frame to frame, by referring to the volumetric map at that moment. In addition, we regularize the map using a fully connected CRF model with respect to panoptic labels. For CRF inference, we propose a unary potential approximation using limited information stored in the map. We also present a map division strategy that achieves a significant reduction in computational time without a drop in accuracy.

We evaluated the performance of our system on the ScanNet v2 dataset [4], a richly annotated large-scale dataset for indoor scene understanding. The results revealed that PanopticFusion is superior or comparable to the state-of-the-art offline 3D DNN methods in the both 3D semantic and instance segmentation tasks. Note that our system is not limited to indoor scenes. Finally, we demonstrated a promising AR application using the 3D panoptic map generated by our system.

The main contributions of this paper are the following:

- The first reported semantic mapping system that realizes scene understanding at the level of *stuff* and *things*.
- Large-scale 3D reconstruction and labeled mesh extraction thanks to the use of a spatially hashed volumetric map representation.
- Map regularization using a fully connected CRF with a novel unary potential approximation and map division strategy.
- Superior or comparable results in both 3D semantic and instance segmentation tasks, in comparison with the state-of-the-art offline 3D DNN methods.

II. RELATED WORK

Previously proposed representative semantic mapping systems related to our PanopticFusion system are shown in Table I. These systems can be divided into two categories from the perspective of semantics: the dense labeling approach and the object-oriented approach.

The dense labeling approach builds a single 3D map and assigns a class label or a probability distribution of class labels to each surfel or voxel to realize a dense 3D semantic segmentation. Hermans *et al.* [8] utilize random decision forests for 2D semantic segmentation and transfer the inferred probability distributions to point clouds with a Bayesian update scheme. Extending the approach of Hermans *et al.* [8], SemanticFusion [17] improves the recognition performance by using CNNs for 2D semantic segmentation and makes use of ElasticFusion [31] for a SLAM system to generate a globally consistent map. Xiang *et al.* [32] presented KinectFusion[19]-based volumetric mapping with novel data associated RNNs for improving the segmentation accuracy. While these methods realize dense scene understanding, they suffer from the drawback that they are not able to distinguish individual objects in the scene.

TABLE I
SEMANTIC MAPPING SYSTEMS RELATED TO PANOPTICFUSION.

Method	Speed	Geometry			Semantics			
	Online	TSDF Volume	Surfels	Large-scale	Model-free	Dense Labeling	Object-level	Map Reg.
SLAM++ [25]	✓						✓	
2.5D is not enough [29]	✓	✓					✓	
SemanticFusion [17]	✓		✓	✓	✓	✓		✓
DA-RNN [32]	✓	✓			✓	✓		
MaskFusion [24]	✓		✓		✓		✓	
Fusion++ [16]	✓	✓			✓		✓	
PanopticFusion (Ours)	✓	✓		✓	✓	✓	✓	✓

Methods adopted in the early days of the object-oriented approach leverage 3D model databases. SLAM++ [25] performs point pair feature-based object detection and feeds the detected objects into a pose graph. Tateno *et al.* [29] proposed a 3D object detection and pose estimation system that combines unsupervised geometric segmentation and global 3D descriptor matching. These methods, however, require the shapes of objects in the scene to be exactly the same as the 3D models in the database. Recently, several studies on the object-oriented approach using a CNN-based 2D object detector have been reported. Sünderhauf *et al.* [27] and Nakajima *et al.* [18] combine a 2D object detector and unsupervised geometric segmentation in order to detect objects in point clouds or a surfel map. MaskFusion [24], Fusion++ [16] and MID-Fusion [33] introduced an object-oriented map representation that individually builds 3D maps for each object based on 2D object detection. The object-oriented map representation enables tracking of individual objects [24], [33] and an object-level pose graph optimization [16]. However, the quantitative recognition performance of these methods is not clear because they mainly evaluate the camera trajectory accuracy. Furthermore, they focus on foreground objects, resulting in a lack of semantics and/or geometry of background regions.

In contrast to these related studies, PanopticFusion realizes holistic scene reconstruction and dense semantic labeling with the ability to discriminate individual objects. Our system builds a single volumetric map, similar to dense labeling approaches, yet each voxel stores neither class labels nor class probability distributions but DNN-predicted panoptic labels in order to seamlessly manage both *stuff* and *things* semantics. The class labels of foreground objects can be restored by a probability integration process. In addition, our 3D reconstruction leverages the truncated signed distance field (TSDF) volumteric map with the voxel hashing data structure [20], which allows us to reconstruct a large-scale scene as well as extract labeled meshes by using marching cubes [15], in contrast to the 3D maps of previous methods, which are based on point clouds [8], [27], surfels [17], [18], [24] and a fixed-sized voxel grid [32], [16]. It should be noted that, with 3D DNN methods that directly apply deep networks to 3D data such as point clouds or voxel grids, high recognition performance has been reported [22], [5], [35], [9]. Nevertheless, with those methods, it is basically necessary to reconstruct the whole scene in advance, requir-

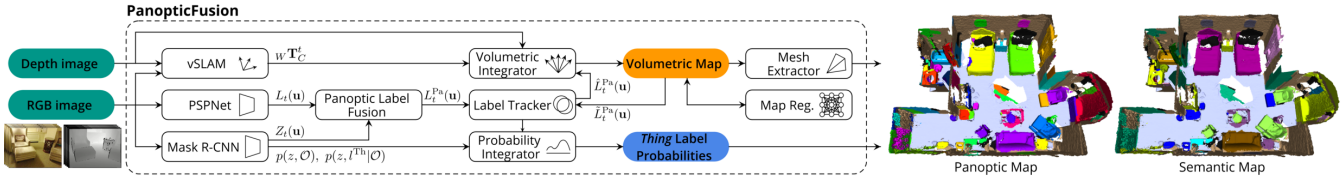

Fig. 2. System overview of PanopticFusion.

ing offline processing, which could limit their application to robotics and AR. On the contrary, PanopticFusion is an online and incremental framework.

III. METHOD

Fig. 2 shows the system overview of PanopticFusion. Our system first feeds an incoming RGB frame into 2D semantic and instance segmentation networks and obtains pixel-wise panoptic labels by fusing the two outputs (Section III-C). The panoptic labels are carefully tracked by referring to the volumetric map at that moment (Section III-D) and are integrated into the map with depth measurements (Section III-E). Probability distributions of class labels for foreground objects are also incrementally integrated (Section III-F). In addition, online map regularization with a fully-connected CRF model is performed for a further improvement of the recognition accuracy. Note that camera poses with respect to the volumetric map are given by an external visual SLAM.

A. Notations

We denote all class labels by $\mathcal{L}$, and they are divided into *stuff* labels $\mathcal{L}^{\mathrm{St}}$ and *thing* labels $\mathcal{L}^{\mathrm{Th}}$: such that $\mathcal{L} = \mathcal{L}^{\mathrm{St}} \cup \mathcal{L}^{\mathrm{Th}}$ and $\mathcal{L}^{\mathrm{St}} \cap \mathcal{L}^{\mathrm{Th}} = \emptyset$. A set of instance IDs for discriminating individual *things* is denoted by $\mathcal{Z}$. Here we define a set of panoptic labels $\mathcal{L}^{\mathrm{Pa}} = \mathcal{L}^{\mathrm{St}} \cup \mathcal{Z} \cup l_{\mathrm{unk}}$ in order to seamlessly manage *stuff* and *things* level semantics in the 3D map. l_{unk} denotes the *unknown* label.

B. Volumetric Map

We use the TSDF-based volumetric map representation with a voxel hashing approach [20], which manages spatially hashed small regular voxel grids called voxel blocks. This approach is memory efficient compared with a single voxel grid approach like the original KinectFusion [19] and enables us to reconstruct large-scale scenes. Our implementation is based on voxblox [21], which is a CPU-based TSDF mapping system, but we extend it to integrate the semantics.

Our volumetric map stores the truncated signed distance $\mathtt{D}_t(\mathbf{v}) \in \mathbb{R}$, the RGB color $\mathtt{C}_t(\mathbf{v}) \in \mathbb{R}^3$ and the associated weight $\mathtt{W}_t^{\mathrm{D}}(\mathbf{v}) \in \mathbb{R}_{\geq 0}$ at each voxel location $\mathbf{v} \in \mathbb{R}^3$, as with [19]. Our system additionally stores the panoptic label $\mathtt{L}_t^{\mathrm{Pa}}(\mathbf{v}) \in \mathcal{L}^{\mathrm{Pa}}$ and its weight $\mathtt{W}_t^{\mathrm{L}}(\mathbf{v}) \in \mathbb{R}_{\geq 0}$. Here t denotes the time index.

C. 2D Panoptic Label Prediction

For the incoming RGB frame, we predict pixel-wise panoptic labels by fusing both 2D semantic and instance segmentation outputs. We utilize the state-of-the-art CNN architectures of PSPNet [36] and Mask R-CNN [7] for 2D semantic and instance segmentation, respectively. PSPNet infers pixel-wise class labels $L_t(\mathbf{u}) \in \mathcal{L}$, where $\mathbf{u} \in \mathbb{R}^2$ denotes the image coordinates. Mask R-CNN outputs instance IDs for each pixel $Z_t(\mathbf{u}) \in \mathcal{Z} \cup l_{\mathrm{unk}}$, where the regions without any foreground objects are filled with l_{unk}. The foreground object probability $p_t(z, \mathcal{O})$ and conditional probability distribution of *thing* labels $p_t(z, l^{\mathrm{Th}}|\mathcal{O})$ with respect to instance z are utilized in the probability integration step described in Section III-F. We obtain pixel-wise panoptic labels $L_t^{\mathrm{Pa}}(\mathbf{u})$ from $L_t(\mathbf{u})$ and $Z_t(\mathbf{u})$ preceding the instance IDs:

$$L_t^{\mathrm{Pa}}(\mathbf{u}) = \begin{cases} Z_t(\mathbf{u}) & Z_t(\mathbf{u}) \neq l_{\mathrm{unk}} \\ L_t(\mathbf{u}) & Z_t(\mathbf{u}) = l_{\mathrm{unk}} \wedge L_t(\mathbf{u}) \in \mathcal{L}^{\mathrm{St}} \\ l_{\mathrm{unk}} & \text{otherwise.} \end{cases} \tag{1}$$

D. Panoptic Label Tracking

Direct integration of raw panoptic labels $L_t^{\mathrm{Pa}}(\mathbf{u})$ into the volumetric map induces label inconsistency because Mask R-CNN does not necessarily output a consistent instance ID for the same object through multiple frames. To avoid this problem, we need to estimate consistency-resolved panoptic labels $\hat{L}_t^{\mathrm{Pa}}(\mathbf{u})$ before the integration. The simplest way is to track the foreground objects in the 2D image sequence using a visual object tracker. This approach unfortunately is not able to re-identify an object in the case of a loopy camera trajectory. Therefore, we take a map reference approach similar to [24], [16].

We first prepare the reference panoptic labels $\tilde{L}_{t-1}^{\mathrm{Pa}}(\mathbf{u})$ by accessing the map. Here, ${}_W\mathbf{T}_C^t$ denotes the live camera pose, $\mathbf{K}$ the camera intrinsic parameters, and $D_t(\mathbf{u})$ the live depth map:

$$\tilde{L}_{t-1}^{\mathrm{Pa}}(\mathbf{u}) = \mathtt{L}_{t-1}^{\mathrm{Pa}}({}_W\mathbf{T}_C^t \mathbf{K}^{-1} D_t(\mathbf{u})[\mathbf{u}, 1]^{\mathrm{T}}). \tag{2}$$

To track labels, we compute the intersection of union (IoU) $U(\tilde{z}, z)$ of instance ID z of raw panoptic labels $L_t^{\mathrm{Pa}}(\mathbf{u})$ and instance ID $\tilde{z}$ of reference panoptic labels $\tilde{L}_{t-1}^{\mathrm{Pa}}(\mathbf{u})$:

$$U(\tilde{z}, z) = \mathrm{IoU}\big(\{\mathbf{u}|\tilde{L}_{t-1}^{\mathrm{Pa}}(\mathbf{u}) = \tilde{z}\}, \{\mathbf{u}|L_t^{\mathrm{Pa}}(\mathbf{u}) = z\}\big) \tag{3}$$

Here, IoU is defined as $\mathrm{IoU}(A, B) = |A \cap B|/|A \cup B|$.

When the maximum value of IoU exceeds a threshold θ_U, $\tilde{z}$ giving the maximum value is associated with z. Otherwise a new instance ID is assigned to z:

$$\hat{z} = \begin{cases} \arg\max_{\tilde{z}} U(\tilde{z}, z) & \max_{\tilde{z}} U(\tilde{z}, z) > \theta_U \\ z_{\mathrm{new}} & \text{otherwise.} \end{cases} \tag{4}$$

The association is processed in descending order in the mask area $|\{\mathbf{u}|L_t^{\mathrm{Pa}}(\mathbf{u}) = z\}|$. Once a reference instance ID $\tilde{z}$ is

associated with z, that instance ID is not associated with any other z. The utilization of IoU instead of an overlap ratio, as used in [24], [16], and the exclusive label association is for avoiding under-segmentation of foreground objects in the map. From the associated instance IDs and raw *stuff* labels, we obtain the consistency-resolved panoptic labels $\hat{L}_t^{\mathrm{Pa}}(\mathbf{u})$ as follows, which are used in the integration step:

$$\hat{L}_t^{\mathrm{Pa}}(\mathbf{u}) = \begin{cases} L_t^{\mathrm{Pa}}(\mathbf{u}) & L_t^{\mathrm{Pa}}(\mathbf{u}) \in \mathcal{L}^{\mathrm{St}} \\ \hat{z} & L_t^{\mathrm{Pa}}(\mathbf{u}) \in \mathcal{Z} \\ l_{\mathrm{unk}} & \text{otherwise.} \end{cases} \tag{5}$$

E. Volumetric Integration

For integration, we take the raycasting approach, as with [21]. For each pixel $\mathbf{u}$, we cast a ray from the sensor origin $\mathbf{s}$ to the back-projected 3D point $\mathbf{p_u} = {}_W\mathbf{T}_C^t \mathbf{K}^{-1} D_t(\mathbf{u})[\mathbf{u}, 1]^{\mathrm{T}}$ and update the voxels along the ray within a truncated distance. Regarding TSDF values, we update them by weighted averaging, similar to [19]:

$$\mathtt{D}_t(\mathbf{v}) = \frac{\mathtt{W}_{t-1}^{\mathrm{D}}(\mathbf{v})\mathtt{D}_{t-1}(\mathbf{v}) + w_t(\mathbf{v}, \mathbf{p_u}) d_t(\mathbf{v}, \mathbf{p_u}, \mathbf{s})}{\mathtt{W}_{t-1}^{\mathrm{D}}(\mathbf{v}) + w_t(\mathbf{v}, \mathbf{p_u})}, \tag{6}$$

$$\mathtt{W}_t^{\mathrm{D}}(\mathbf{v}) = \mathtt{W}_{t-1}^{\mathrm{D}}(\mathbf{v}) + w_t(\mathbf{v}, \mathbf{p_u}). \tag{7}$$

Here, d_t denotes the distance between the voxel and the surface boundary, and w_t a quadric weight [21] that takes the reliability of depth measurements into account. Similar updating is applied to the voxel color $\mathtt{C}_t(\mathbf{v})$.

In contrast to TSDF and colors of continuous values, weighted averaging cannot be applied to panoptic labels of discrete values. The most reliable and simplest way to manage panoptic labels is to record all integrated labels. This, unfortunately, results in a significant increase in memory usage and frequent memory allocation. Instead we store a single label at each voxel and update its weight by the increment/decrement strategy. If the pixel-wise panoptic label $\hat{L}_t^{\mathrm{Pa}}(\mathbf{u})$ estimated in the previous section is the same as the current voxel panoptic label $\mathtt{L}_{t-1}^{\mathrm{Pa}}(\mathbf{v})$, we increment the weight $\mathtt{W}_t^{\mathrm{L}}(\mathbf{v})$ with the quadric weight:

$$\mathtt{L}_t^{\mathrm{Pa}}(\mathbf{v}) = \mathtt{L}_{t-1}^{\mathrm{Pa}}(\mathbf{v}),\ \mathtt{W}_t^{\mathrm{L}}(\mathbf{v}) = \mathtt{W}_{t-1}^{\mathrm{L}}(\mathbf{v}) + w_t(\mathbf{v}, \mathbf{p_u}). \tag{8}$$

In contrast, if those panoptic labels do not coincide, we decrement the weight:

$$\mathtt{L}_t^{\mathrm{Pa}}(\mathbf{v}) = \mathtt{L}_{t-1}^{\mathrm{Pa}}(\mathbf{v}),\ \mathtt{W}_t^{\mathrm{L}}(\mathbf{v}) = \mathtt{W}_{t-1}^{\mathrm{L}}(\mathbf{v}) - w_t(\mathbf{v}, \mathbf{p_u}). \tag{9}$$

Note that in the case where $w_t > \mathtt{W}_{t-1}^{\mathrm{L}}$, that is, when the weight considerably falls, we replace the voxel label with the newly estimated label:

$$\mathtt{L}_t^{\mathrm{Pa}}(\mathbf{v}) = \hat{L}_t^{\mathrm{Pa}}(\mathbf{u}),\ \mathtt{W}_t^{\mathrm{L}}(\mathbf{v}) = w_t(\mathbf{v}, \mathbf{p_u}) - \mathtt{W}_{t-1}^{\mathrm{L}}(\mathbf{v}). \tag{10}$$

F. Thing Label Probability Integration

The *thing* label predicted by Mask R-CNN is frequently uncertain even while the segmentation mask is accurate, especially in the case where a small part of the object is visible. Hence we probabilistically integrate *thing* labels instead of assigning a single label to each foreground object:

$$p_{1\cdots t}(z, l^{\mathrm{Th}}) = \frac{\sum_t p_t(z, \mathcal{O}) p_t(z, l^{\mathrm{Th}} | \mathcal{O})}{\sum_t p_t(z, \mathcal{O})}. \tag{11}$$

Weighting the probability distributions with the detection confidence $p_t(z, \mathcal{O})$ allows the final distribution to preferentially reflect reliable detections.

G. Online Map Regularization

While the integration scheme described above yields a reliable 3D panoptic map, it is possible to further improve the recognition accuracy by using a map regularization with a fully connected CRF model. A fully connected CRF with Gaussian edge potentials has been widely used in 2D image segmentation since an efficient inference method was proposed [12]. Recently, several studies that apply it to a 3D map, such as surfels or occupancy grids, have been reported [8], [17], [34]. In those approaches, CRF models are constructed with respect to class labels whose number is fixed, whereas we consider the CRF with respect to panoptic labels whose number depends on the scene and is theoretically not limited. Here we are faced with two problems: how to properly compute unary potentials for panoptic labels, and how to infer a CRF whose number of labels is potentially large within a practical time.

1) Problem Setting: We construct a fully connected graph whose nodes are individual voxels. We assign a label variable $x_v \in \mathcal{L}^{\mathrm{Pa}}$ to each node and infer the optimal labels $\mathbf{x} = \{x_v\}$ that minimize the Gibbs energy E by the mean-field approximation and a message passing scheme:

$$E(\mathbf{x}) = \sum_v \psi_u(x_v) + \sum_{v<v'} \psi_p(x_v, x_{v'}). \tag{12}$$

While it is non-trivial which unary potentials should be used for a panoptic label CRF, we use a negative logarithm of a probability distribution following a standard class label CRF:

$$\psi_u(x_v) = -\log p(x_v). \tag{13}$$

We utilize a linear combination of Gaussian kernels for pairwise potentials because the efficient inference method [12] can be applied:

$$\psi_p(x_v, x_{v'}) = \mu(x_v, x_{v'}) \sum_m w^{(m)} k^{(m)}(\mathbf{f}_v, \mathbf{f}_{v'}). \tag{14}$$

Here, $\mu(x_s, x_s') = 1_{[x_s \neq x_s']}$ is a simple Potts model. As in [12], we chose the following two kernels which regularize the map with respect to voxel colors and locations, respectively:

$$k^{(1)}(\mathbf{f}_v, \mathbf{f}_{v'}) = \exp\left(-\frac{|\mathbf{v} - \mathbf{v}'|^2}{2\theta_\alpha^2} - \frac{|\mathtt{C}(\mathbf{v}) - \mathtt{C}(\mathbf{v}')|^2}{2\theta_\beta^2} \right), \tag{15}$$

$$k^{(2)}(\mathbf{f}_v, \mathbf{f}_{v'}) = \exp\left(-\frac{|\mathbf{v} - \mathbf{v}'|^2}{2\theta_\alpha^2} \right). \tag{16}$$

2) Unary Potential Approximation: Previous approaches [8], [17], [34] assigned a probability distribution to each surfel or voxel, which can be used directly to compute unary potentials; in contrast, from the viewpoint of memory efficiency, we store only a single label in each voxel. Therefore, we approximate the unary potentials using only a single label, and weights stored in a voxel, based on a certain assumption described as follows.

Here let us focus on the integration scheme of panoptic labels shown in Eq. (8)-(10). We denote the set of times when the predicted panoptic label is the same as, and not the same as, the current voxel label by $\mathcal{T}_+ = \{\tau \mid \hat{L}_\tau^{\mathrm{Pa}}(\mathbf{u}) = \mathtt{L}_t^{\mathrm{Pa}}(\mathbf{v})\}$ and $\mathcal{T}_- = \{\tau \mid \hat{L}_\tau^{\mathrm{Pa}}(\mathbf{u}) \neq \mathtt{L}_t^{\mathrm{Pa}}(\mathbf{v})\}$, respectively. If $\mathtt{L}_\tau^{\mathrm{Pa}}(\mathbf{v}) = \mathtt{L}_t^{\mathrm{Pa}}(\mathbf{v})$ for all $\tau = 1, \cdots, t-1$, that is to say, the voxel label has not changed, Eq. (17) holds strictly. If $p(x_v = \mathtt{L}_t^{\mathrm{Pa}}(\mathbf{v})) > 0.5$ and the number of integrations is sufficiently large, Eq. (17) holds asymptotically:

$$\sum_{t \in \mathcal{T}_+} w_t(\mathbf{v}, \mathbf{p_u}) - \sum_{t \in \mathcal{T}_-} w_t(\mathbf{v}, \mathbf{p_u}) \simeq \mathtt{W}_t^{\mathrm{L}}(\mathbf{v}). \tag{17}$$

In addition, from the TSDF update scheme in Eq. (7) we have,

$$\sum_{t \in \mathcal{T}_+} w_t(\mathbf{v}, \mathbf{p_u}) + \sum_{t \in \mathcal{T}_-} w_t(\mathbf{v}, \mathbf{p_u}) = \mathtt{W}_t^{\mathrm{D}}(\mathbf{v}). \tag{18}$$

Consequently, the probability that the current panoptic label in the voxel is actually correct can be calculated as,

$$\begin{aligned} p(x_v = \mathtt{L}_t^{\mathrm{Pa}}(\mathbf{v})) &= \frac{\sum_{t \in \mathcal{T}_+} w_t(\mathbf{v}, \mathbf{p_u})}{\sum_{t \in \mathcal{T}_+} w_t(\mathbf{v}, \mathbf{p_u}) + \sum_{t \in \mathcal{T}_-} w_t(\mathbf{v}, \mathbf{p_u})} \\ &\simeq \frac{1}{2}\left(1 + \frac{\mathtt{W}_t^{\mathrm{L}}(\mathbf{v})}{\mathtt{W}_t^{\mathrm{D}}(\mathbf{v})}\right). \end{aligned} \tag{19}$$

It is unfortunately not possible to calculate the exact probability that the voxel takes a label other than the current label because the map does not record all the information about previously integrated labels. Therefore, we approximate the probability as follows, where M denotes the number of panoptic labels in the map:

$$p(x_v) = \frac{1}{M-1}\big(1 - p(x_v = \mathtt{L}_t^{\mathrm{Pa}}(\mathbf{v}))\big) \quad (x_v \neq \mathtt{L}_t^{\mathrm{Pa}}(\mathbf{v})). \tag{20}$$

Finally, we obtain the unary potential from Eq. (13), (19) and (20). In spite of the approximated approach, it realizes quantitative and qualitative improvements in recognition accuracy, as shown in Section IV-C.

3) Map Division for Online Inference: The computational complexity of the inference algorithm proposed by Krähenbühl *et al.* [12] is $\mathcal{O}(NM)$, where N and M are the numbers of voxels and panoptic labels, respectively. In our problem setting, however, M is theoretically limitless and could in practice be large, e.g. several hundreds, which would make online inference impracticable. To solve this problem, we present a map division strategy. When we divide the volumetric map into S spatially contiguous submaps, the number of panoptic labels in each submap can be expected to be $\mathcal{O}(M/S)$. Hence, the total computational complexity could be reduced to $S \times \mathcal{O}(N/S \times M/S) = \mathcal{O}(NM)/S$. The map is divided by the block-wise region growing approach based on the predefined maximum number of voxel blocks. The division process has little effect on computational time.

IV. EVALUATION

A. Experimental Setup

For evaluating the performance of our system, we used the ScanNet v2 dataset [4], a large-scale dataset for indoor scene understanding. It provides RGB-D images captured by hand-held consumer-grade depth sensors, camera trajectories, reconstructed 3D models, and 2D/3D semantic annotations. In the following experiments, we used RGB-D images of size 640×480 pixels and the provided camera trajectories for fair comparison. The dataset was composed of 1201 training scenes and 312 *open* test scenes. In addition, 100 *hidden* test scenes without publicly available semantic annotations are provided for the ScanNet Benchmark Challenge [2]. For quantitative evaluations, 20 class annotations are generally used. In this paper, we define the wall and floor as the *stuff* class $\mathcal{L}^{\mathrm{St}}$ and the other 18 classes, such as chairs and sofas, as the *thing* class $\mathcal{L}^{\mathrm{Th}}$. Note that our system is not limited to indoor scenes, and the numbers of *stuff* and *thing* classes can be arbitrarily defined.

We employed ResNet-50 for the backbone of PSPNet. The network was initialized with the ADE20K [37] pre-trained weights, and was then fine-tuned using a SGD optimizer for 30 epochs with a learning rate of 0.01 and a batch size of 2. We leveraged ResNet-101-FPN for the Mask R-CNN's backbone. After initialization with MS COCO [13] pre-trained weights, the network was fine-tuned by 4-step alternating learning [23] using an ADAM optimizer for 25 epochs with a learning rate of 0.001 and a batch size of 1[1].

We used the following parameters for the integration process: voxel size of 0.024 m, a truncation distance of 4×0.024 m, 16×16×16 voxels per voxel block, IoU threshold $\theta_U = 0.25$. In the map regularization, $w^{(1)} = 10$, $w^{(2)} = 15$, $\theta_\alpha = 0.05$ m, $\theta_\beta = 20$ were used with 5 iterations of CRF inference. The following experiments were performed on a computer equipped with an Intel Core i7-7800X CPU at 3.50 GHz and two NVIDIA GeForce GTX 1080Ti GPUs.

B. Quantitative and Qualitative Results

Fig. 3 shows examples of 3D panoptic maps generated by our system. Unfortunately, there are no semantic mapping systems or 3D DNNs that can recognize a 3D scene at the level of *stuff* and *things*. Therefore, we evaluated the performance on two sub-tasks, 3D semantic segmentation and instance segmentation, for a quantitative comparison. In this evaluation, we used the hidden test set of ScanNet v2. We show the results in Tables II and III. The bold and underlined numbers denote first and second ranks, respectively. In the table, the state-of-the-art methods that apply 3D DNNs to points or volumetric grids are listed. Note that the methods of [10], [5], [9] leverage RGB images with associated camera poses as well. Although they are offline methods that assume that the 3D scenes are reconstructed in advance, and are specially designed for individual tasks, our online system that uses only 2D-based recognition modules surprisingly achieves comparable or superior performance compared with those methods, thanks to the careful integration of multi-view predictions. In terms of the class-wise accuracy, the results revealed that our system has advantages especially in the

[1]We used a publicly available implementation of [1] and [3] for PSPNet and Mask R-CNN, respectively.

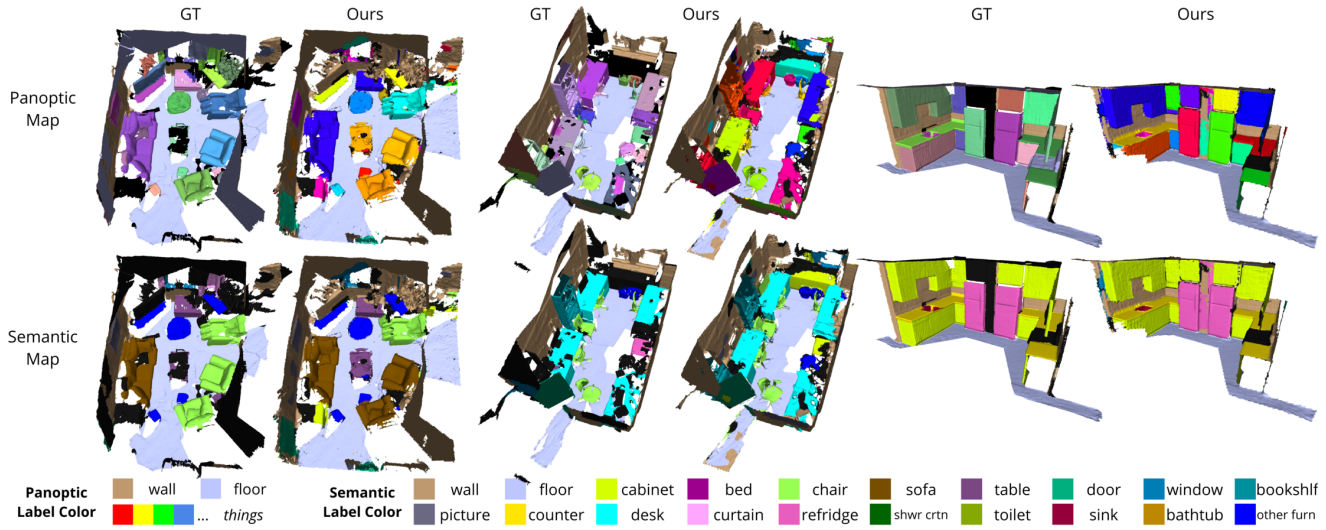

Fig. 3. Qualitative results obtained with PanopticFusion system. From left to right, typical scenes in ScanNet v2 of scene0608_00, scene0643_00 and scene0488_01 are displayed. Note that ground truth and our results leverage different reconstruction algorithms, and the colors of *things* in our results are not necessarily the same as the ground truth.

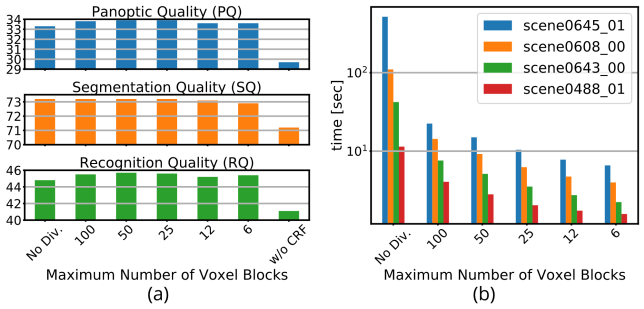

Fig. 4. Results of the map regularization with the map division strategy. The relationship between the maximum number of voxel blocks and (a) recognition accuracy and (b) computational time. Note the computational time is shown in a logarithmic scale.

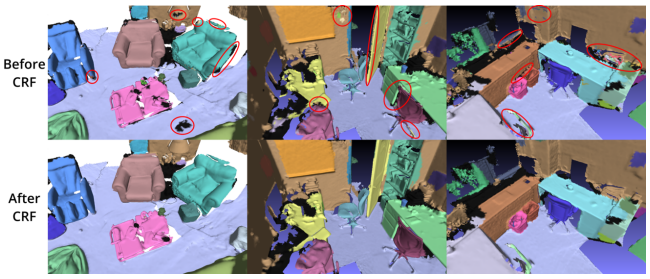

Fig. 5. Qualitative results of map regularization. The noisy predictions within red circles are appropriately regularized, taking a spatial context into account.

case of small objects such as sinks and pictures, and objects that are confusing to recognize only by their geometry, such as beds, bookshelves, and curtains.

Additionally, we evaluated 3D panoptic quality on the open test set of ScanNet v2, although there are no quantitatively comparable methods. We employed the evaluation criteria originally proposed in [11]. Note that the quality was evaluated with respect to each vertex instead of each pixel, and, as with the ScanNet 3D semantic instance benchmark, we ignored the predicted *things* with less than 100 vertices. We show the panoptic quality (PQ) as well as the segmentation quality (SQ) and recognition quality (RQ) in Table IV. We hope that the results of our evaluation will invigorate research in this field.

C. Evaluation of Map Regularization

In this section, we evaluate the map regularization proposed in Section III-G. First, we evaluated the effects of the map division on the recognition accuracy and computational time. We used the open test set for the recognition accuracy and typical scenes in ScanNet v2 for the computational time. The result is shown in Fig. 4. Note that, in this experiment, we applied regularization to the pre-generated map as a post process to evalute solely the effects of CRF.

As can been seen, the recognition performance was improved by the map regularization with the proposed unary potential approximation regardless of whether or not map division was used. The results also show that the map division strategy drastically reduced the computational time without a decrease in recognition performance, compared with the case of building a CRF model for a whole map.

Based on the above results, our online system employed map regularization with the map division strategy. We chose a maximum number of voxel blocks of 25 because of the better recognition accuracy and acceptable computational time. Table IV shows the difference in recognition performance due to whether or not map regularization was used in online processing. This result shows that the map regularization improved the recognition performance even when the system ran online. Note that the scores of almost all the classes were boosted by the proposed regularization. See Fig. 5 for qualitative effects of the map regularization.

D. Run-time Analysis

Table V shows computational times for each component of our system, which are measured on scene0645_01, a typical large-scale scene in ScanNet v2 (shown in Fig. 1).

TABLE II

SEMANTIC SEGMENTATION RESULTS ON SCANNET (V2) 3D SEMANTIC LABEL BENCHMARK (HIDDEN TEST SET) [2]. IOU IS REPORTED.

	avg.	wall	floor	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn
ScanNet [4]	30.6	43.7	78.6	31.1	36.6	52.4	34.8	30.0	18.9	18.2	50.1	10.2	21.1	34.2	0.2	24.5	15.2	46.0	31.8	20.3	14.5
PointNet++ [22]	33.9	52.3	67.7	25.6	47.8	36.0	34.6	23.2	26.1	25.2	45.8	11.7	25.0	27.8	24.7	21.2	14.5	54.8	36.4	58.4	18.3
SPLATNet [26]	39.3	69.9	92.7	31.1	51.1	65.6	51.0	38.3	19.7	26.7	60.6	0.0	24.5	32.8	40.5	0.1	24.9	59.3	27.1	47.2	22.7
Tangent Conv. [28]	43.8	63.3	91.8	36.9	64.6	64.5	56.2	42.7	27.9	35.2	47.4	14.7	35.3	28.2	25.8	28.3	29.4	61.9	48.7	43.7	29.8
3DMV [5]	48.4	60.2	79.6	42.4	53.8	60.6	50.7	41.3	37.8	53.9	64.3	21.4	31.0	43.3	57.4	53.7	20.8	69.3	47.2	48.4	30.1
TextureNet [10]	56.6	68.0	93.5	49.4	66.4	71.9	63.6	46.4	39.6	56.8	67.1	22.5	44.5	41.1	67.8	41.2	53.5	79.4	56.5	**67.2**	35.6
SparseConvNet [6]	**72.5**	**86.5**	**95.5**	**72.1**	**82.1**	**86.9**	**82.3**	**62.8**	**61.4**	**68.3**	**84.6**	**32.5**	**53.3**	**60.3**	**75.4**	**71.0**	**87.0**	**93.4**	**72.4**	64.7	**57.2**
PanopticFusion (Ours)	52.9	60.2	81.5	38.6	68.8	63.2	64.9	44.2	29.3	56.1	60.4	24.1	22.5	43.4	70.5	49.9	66.9	79.6	50.7	49.1	34.8

TABLE III

INSTANCE SEGMENTATION RESULTS ON SCANNET (V2) 3D SEMANTIC INSTANCE BENCHMARK (HIDDEN TEST SET) [2]. $AP_{0.5}$ IS REPORTED.

	avg.	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn
SGPN [30]	14.3	6.5	39.0	27.5	35.1	16.8	8.7	13.8	16.9	1.4	2.9	0.0	6.9	2.7	0.0	43.8	11.2	20.8	4.3
GSPN [35]	30.6	34.8	40.5	58.9	39.6	27.5	28.3	24.5	31.1	2.8	**5.4**	12.6	6.8	21.9	21.4	82.1	33.1	50.0	29.0
3D-SIS [9]	38.2	19.0	43.2	57.7	**69.9**	27.1	32.0	23.5	24.5	7.5	1.3	3.3	26.3	**42.2**	**85.7**	88.3	11.7	**100.0**	24.0
MASC [14]	44.7	**38.2**	55.5	**63.3**	63.9	**38.6**	**36.1**	27.6	38.1	32.7	0.2	**26.0**	50.9	45.1	57.1	**98.0**	36.7	52.8	43.2
PanopticFusion (Ours)	**47.8**	25.9	**71.2**	55.0	59.1	26.7	25.0	**35.9**	**59.5**	**43.7**	0.0	17.5	**61.3**	41.1	**85.7**	94.4	**48.5**	66.7	**43.4**

TABLE IV

3D PANOPTIC QUALITY ON SCANNET (V2) OPEN TEST SET.

method	metric	all	*things*	*stuff*	wall	floor	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn
PanopticFusion w/o CRF	PQ	29.7	26.7	56.7	37.5	76.0	18.6	29.1	37.8	38.2	29.5	13.8	14.1	13.0	26.5	8.3	14.9	11.6	38.0	28.8	72.4	33.3	28.0	24.3
	SQ	71.2	71.4	69.5	62.3	76.7	69.4	68.5	69.3	72.3	70.1	74.6	69.9	70.7	72.9	65.0	60.6	70.5	75.3	75.8	79.2	71.9	74.0	75.3
	RQ	41.1	36.8	79.6	60.2	99.0	26.8	42.5	54.6	52.8	42.1	18.5	20.1	18.4	36.3	12.8	24.6	16.4	50.4	37.9	91.3	46.4	37.8	32.2
PanopticFusion with CRF	PQ	33.5	30.8	58.4	40.4	76.4	23.8	35.8	46.7	42.1	34.8	18.0	19.3	16.4	26.4	10.4	16.1	16.6	39.5	36.3	76.1	36.7	31.0	27.7
	SQ	73.0	73.3	70.7	64.0	77.4	71.1	70.1	74.3	74.6	74.3	76.0	72.5	73.9	71.2	65.1	61.7	72.3	77.7	79.5	81.4	72.7	75.3	75.8
	RQ	45.3	41.3	80.9	63.1	98.7	33.5	51.1	62.8	56.3	46.9	23.6	26.7	22.2	37.1	16.0	26.0	23.0	50.8	45.7	93.5	50.5	41.2	36.5

TABLE V

RUN-TIME ANALYSIS.

Frequency	**Component**	**time**
Every Mask R-CNN frames	PSPNet	80 ms
	Mask R-CNN	235 ms
	Panoptic label fusion	2 ms
	Reference panoptic label gen.	19 ms
	Panoptic label tracking	9 ms
	Volumetric integration	139 ms
	Probability integration	$\sim$ 1 ms
Every 10 sec.	Map regularization	4.5 s
Every 1 sec.	Mesh extraction	14 ms
Throughput		4.3 Hz

PSPNet and Mask R-CNN each run on GPUs, and the other components are processed on a CPU. All components are basically processed in parallel. The throughput of our system is around 4.3 Hz, which is determined by Mask R-CNN, the bottleneck process of our system. Although our current implementation is not highly optimized, our system is able to run at a rate allowing interaction. Note that the computational time except for the map regularization does not depend on the scale of scenes nor the number of *things* because we utilize the raycasting approach for the integrations. The processing time of the map regularization increases to about 10 seconds at the end of the sequence, but it could be reduced by processing only the voxel blocks near the camera frustum.

V. APPLICATIONS

In this section, we demonstrate a promising AR application utilizing a 3D panoptic map generated online by the proposed system. A 3D panoptic map reconstructed as 3D meshes allows us to realize the following visualizations according to the context of the scene:

- Path planning on *stuff* regions such as floors and walls.
- Interaction with individual objects, or the *thing* regions.
- Interaction appropriate for the semantics of each region.

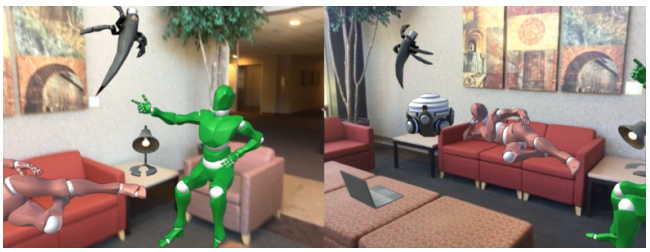

Fig. 6. An example of AR application using a 3D panoptic map generated by PanopticFusion system.

- Natural occlusion and collision visualization.

We show an example of an AR application utilizing the above visualizations in Fig. 6. Humanoids and insect-type robots are able to locomote on the floor and wall meshes, respectively, according to the automatic path planning. Additionally, the semantics of each object realizes context-aware interactions such that humanoids sit and lie on chairs and sofas, respectively, and CG objects appear on tables. Moreover, we can naturally visualize the occlusion effects, which are important for AR, because the 3D meshes of the scene are extracted. Note that, taking advantage of the accurately recognized 3D panoptic map, we can easily estimate the poses of seats of chairs and sofas, and top panels of tables by using simple normal- and curvature-based segmentation and plane detection.

We believe that our system is useful not only for AR scenarios but also for autonomous robots that explore scenes and manipulate objects.

VI. CONCLUSIONS

In this paper, we have introduced a novel online volumetric semantic mapping system at the level of *stuff* and *things*. It performs dense semantic labeling while discriminating individual objects, as well as large-scale 3D reconstruction

and labeled mesh extraction thanks to the use of a spatially hashed volumetric map representation. This was realized by pixel-wise panoptic label prediction and its volumetric integration with careful label tracking. In addition, we constructed a fully connected CRF model with respect to panoptic labels and inferred it online with a novel unary potential approximation and a map division strategy, which further improved the recognition performance. The experimental results showed that our system outperformed or compared well with state-of-the-art offline 3D DNN methods in terms of both 3D semantic and instance segmentation. In future work, we plan to extend our system to ensure global consistency against long-term pose drift, to perform high-throughput mapping by network reduction, and to support dynamic environments.

We believe that the *stuff* and *things*-level semantic mapping will open the way to new applications of intelligent autonomous robotics and context-aware augmented reality that deeply interact with the real world.

REFERENCES

[1] Pspnet-keras-tensorflow. `https://github.com/Vladkryvoruchko/PSPNet-Keras-tensorflow`.

[2] Scannet benchmark challenge. `http://kaldir.vc.in.tum.de/scannet_benchmark/`, accessed 2019-02-27.

[3] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. `https://github.com/matterport/Mask_RCNN`, 2017.

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. *arXiv preprint arXiv:1803.10409*, 2018.

[6] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2017.

[8] Alexander Hermans, Georgios Floros, and Bastian Leibe. Dense 3d semantic mapping of indoor scenes from rgb-d images. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

[9] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. *arXiv preprint arXiv:1812.07003*, 2018.

[10] Jingwei Huang, Haotian Zhang, Li Yi, Thomas Funkhouser, Matthias Nießner, and Leonidas Guibas. Texturenet: Consistent local parametrizations for learning from high-resolution signals on meshes. *arXiv preprint arXiv:1812.00020*, 2018.

[11] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. *arXiv preprint arXiv:1801.00868*, 2018.

[12] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, 2011.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conf. on Computer Vision (ECCV)*, 2014.

[14] Chen Liu and Yasutaka Furukawa. Masc: Multi-scale affinity with sparse convolution for 3d instance segmentation. *arXiv preprint arXiv:1902.04478*, 2019.

[15] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM, 1987.

[16] John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level slam. In *IEEE Int. Conf. on 3D Vision (3DV)*, 2018.

[17] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017.

[18] Yoshikatsu Nakajima and Hideo Saito. Efficient object-oriented semantic mapping with object detector. *IEEE Access*, 7:3206–3213, 2019.

[19] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[20] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6):169, 2013.

[21] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[22] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, 2017.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[24] M. Runz, M. Buffier, and L. Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *IEEE Int. Symposium on Mixed and Augmented Reality (ISMAR)*, 2018.

[25] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[26] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[27] Niko Sünderhauf, Trung T Pham, Yasir Latif, Michael Milford, and Ian Reid. Meaningful maps with object-oriented semantic mapping. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2017.

[28] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[29] Keisuke Tateno, Federico Tombari, and Nassir Navab. When 2.5 d is not enough: Simultaneous reconstruction, segmentation and recognition on dense slam. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016.

[30] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[31] Thomas Whelan, Stefan Leutenegger, Renato F. Salas-Moreno, Ben Glocker, and Andrew J. Davison. Elasticfusion: Dense slam without a pose graph. In *Robotics: Science and Systems (RSS)*, 2015.

[32] Yu Xiang and Dieter Fox. Da-rnn: Semantic mapping with data associated recurrent neural networks. In *Robotics: Science and Systems (RSS)*, 2017.

[33] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Mid-fusion: Octree-based object-level multi-instance dynamic slam. *arXiv preprint arXiv:1812.07976*, 2018.

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

[35] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. *arXiv preprint arXiv:1812.03320*, 2018.

[36] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[37] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, pages 1–20, 2016.