PAPER

Joint 2D and 3D Semantic Segmentation with Consistent Instance Semantic

Yingcai WAN[†], ***Student Member and*** **Lijin FANG**[†a)], ***Nonmember***

SUMMARY 2D and 3D semantic segmentation play important roles in robotic scene understanding. However, current 3D semantic segmentation heavily relies on 3D point clouds, which are susceptible to factors such as point cloud noise, sparsity, estimation and reconstruction errors, and data imbalance. In this paper, a novel approach is proposed to enhance 3D semantic segmentation by incorporating 2D semantic segmentation from RGB-D sequences. Firstly, the RGB-D pairs are consistently segmented into 2D semantic maps using the tracking pipeline of Simultaneous Localization and Mapping (SLAM). This process effectively propagates object labels from full scans to corresponding labels in partial views with high probability. Subsequently, a novel Semantic Projection (SP) block is introduced, which integrates features extracted from localized 2D fragments across different camera viewpoints into their corresponding 3D semantic features. Lastly, the 3D semantic segmentation network utilizes a combination of 2D-3D fusion features to facilitate a merged semantic segmentation process for both 2D and 3D. Extensive experiments conducted on public datasets demonstrate the effective performance of the proposed 2D-assisted 3D semantic segmentation method.

key words: *semantic segmentation, 3D reconstruction, SLAM, consistent segmentation*

1. Introduction

Scene understanding systems play a crucial role in enabling robots and smart devices to intelligently interact with unfamiliar environments by providing spatial and semantic information about the 3D scene. However, most existing 3D semantic segmentation methods [1]–[3] heavily rely on the availability of a complete and accurate 3D point cloud model as input, which is challenging to obtain in realistic scenarios. Therefore, leveraging 2D semantic labels that correspond to 3D scenes is a crucial approach to enhance the understanding of 3D scenes and improve the performance of 3D semantic segmentation.

The 2D semantic segmentation have achieved remarkable performance in terms of accuracy and speed [4], [5]. Compared to 3D semantic segmentation methods based on point clouds, such as PointNet++ [1], MCCNN [6], and MinkowskiNet [7], [8], 2D images provide detailed texture and color information that can assist 3D semantic segmentation, enhancing the robustness and generalization ability of 3D models. Recently, researchers have proposed end-to-end joint semantic segmentation methods that combine multi-view RGB-D data with 3D models to improve 3D semantic segmentation. For example, 3D-SIS [9] introduced a multi-modal instance semantic segmentation method that effectively fuses 2D context and 3D geometry information. Following the 2D-3D fusion strategy, BPNet [3] proposed a bidirectional projection module to improve the 2D and 3D semantic segmentation performance of RGB images and point clouds. However, these attempts were made without fully leveraging the additional 2D information that complements the existing 2D-3D fusion methods.

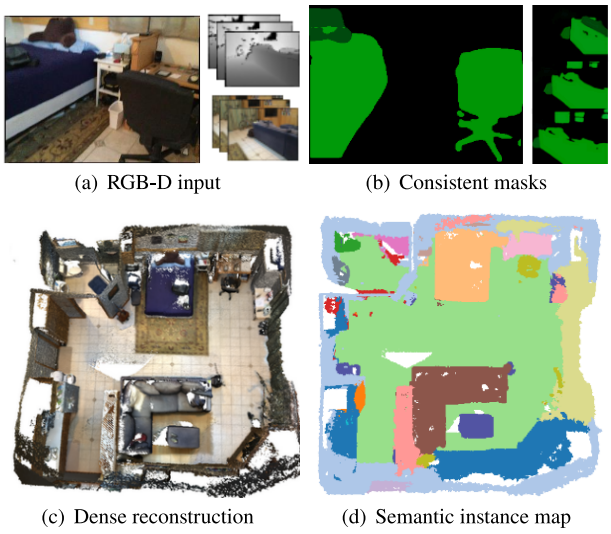
In addition, object instance semantic segmentation for a single RGB image is performance, but degenerates sharply in the continuous frame and partial scans, which affects 2D-3D joint semantic segmentation. Although video semantic segmentation considers temporal and spatial consistency for continuous frame segmentation [10], [11], motion blur, occlusion, viewpoint changes, lighting variations, and object incompleteness pose significant challenges in the field of video semantic segmentation. This leads to the fact that video segmentation methods cannot be directly applied to 2D-3D joint semantic segmentation. Compared to the existing 3D segmentation networks [3], [7], [9] and scene graph generation approach [12], our 2D-3D joint semantic segmentation framework utilizes temporal and spatial consistency of SLAM to achieve high probability labels, fusing 2D semantic feature into 3D features, and output dense 3D semantic model. The whole process is shown in Fig. 1. The input of our framework is RGB-D frames, and the output is consistent 2D semantic segmentation Fig. 1(b), dense 3D reconstruction model Fig. 1(c), and 3D semantic segmentation model Fig. 1(d), realizing the end-to-end process from RGB-D image to 3D semantic reconstruction.

Based on the 2D segmentation methods [13], [14] and SLAM [15], [16], we maintain a semantic sparse map that saves the probability semantic mask of each object. Since semantic predictions from partial views are not as reliable as global views, the semantic sparse map of SLAM is used to correct 2D segments that existed in bad cases. After obtaining camera poses, consistent 2D semantic masks, and a dense 3D reconstruction model, the proposed Semantic propagation Block (SP-Block) is responsible for extracting the multi-scale features from 2D consistent object segments and projecting channels of features into volumes that are fused with those extracted from the encoder of MinkowskiNet [7] after the Domain Transformation (DoT) operation. Compared with MinkowskiNet [7] and BPNet [3], our 2D semantic masks of different views provide accurate 2D ob-

Manuscript received August 9, 2023.
Manuscript revised October 31, 2023.
Manuscript publicized December 15, 2023.

[†]Faculty of Robot Science and Engineering, Northeastern University, Shen Yang, 110170, China.
a) E-mail: ljfang@mail.neu.edu.cn (Corresponding author)

DOI: 10.1587/transfun.2023EAP1095



(a) RGB-D input (b) Consistent masks

(c) Dense reconstruction (d) Semantic instance map

Fig. 1 System's Input (a) and Outputs (b, c, d). The consistent semantic masks (b) and a comprehensive mesh model (c) facilitate our network's prediction of semantic instances (d).

jects' regions that make traditional 3D semantic predictions more accurate.

The contributions of this paper are summarized as follows:

- A 2D-3D joint semantic segmentation framework is proposed that integrates the SLAM-based 2D consistent semantic segmentation with 3D semantic segmentation enhancement.
- A novel strategy is proposed to obtain accurate and consistent 2D semantic masks by leveraging 2D segments and a sparse semantic map in a consistent object prediction framework.
- SP-Block is built to extract and project multi-scale deep features from a 2D semantic map, which are transformed to the feature domain from point clouds by the DoT operation.

2. Related Work

2.1 Object Instance Segmentation

Early 2D semantic segmentation methods, including Faster R-CNN [4], Mask R-CNN [5], make instance mask predictions before object semantic recognition. More recent networks [17]–[19] are anchor-based approaches that predict boxes' offsets relative to a collection of fixed boxes. Although semantic instance segmentation has achieved reliable results, more and more segmentation tasks have put forward requirements for efficiency. YOLACT [13] is the real-time (more than 30 fps) instance segmentation algorithm that is updated as YOLACT++ [14] by incorporating deformable convolutions into the backbone network. However, those approaches focus on single image processing topics, leading to inconsistent scene interpretation issues due to illumination changes, occlusions and other variations over time. To solve the problem, video-based instance segmentation (VIS) [20] tracks object instances interested in a video sequence, but these methods require the target information determined in the first frame. Different from them, each RGB-D pair is segmented in geometric and semantic manners to obtain correct boundaries in this paper. Moreover, we build a global sparse semantic map in real-time to maintain 2D consistent semantic segments.

2.2 2D-3D Segmentation

Point clouds are prevalent for representing 3D scenes due to their efficiency and superior geometric details over 2D imagery. 3D ShapeNet [21] pioneered this field, using a 3D convolutional deep belief network trained on a shape database. Subsequently, PointNet [22] and PointNet++ [1] introduced more effective 3D surface representations.

Research highlights the complementary nature of joint 2D and 3D features, leading to enhanced local performance. 3DMV [23] integrates spatial and RGB attributes in an end-to-end design. Building on this, 3D-SIS [9] merges 2D color images with 3D geometry by projecting 2D RGB view features into a voxel grid. BPNet [3] offers a bidirectional feature exchange between 2D and 3D CNNs across pyramid levels using a proposed module. In our work, after procuring 2D consistent semantic maps, the SP-Block captures 2D semantic segment features and transfers them to a domain encoded by MinkowshiNet [7].

2.3 Scene Understanding from RGB-D Sequences

Using RGB-D images, tracking and mapping methods like [24] construct global 3D maps essential for scene comprehension. While multi-feature trackers optimize indoor scenes' robustness, their emphasis is on minimizing camera pose drift with sparse features, not dense map reconstruction. In contrast, KinectFusion [25] and BundleFusion [26] prioritize GPU-driven dense 3D reconstruction.

SemanticFusion [27] introduced 3D semantic segmentation by incrementally integrating neural network-labeled semantic surfels. PanopticFusion [28] furthered this with a 2D-to-3D approach using pixel-wise predictions. While intuitive, these methods are bottlenecked by 2D segmentation accuracy. Our method ensures 2D label consistency and employs a 3D network for mesh segmentation, enhancing accuracy by jointly considering global and local features.

3. System Overview

In this section, we introduce the main modules of the pipeline, as shown in Fig. 2, the system is divided into three parts, 1) SLAM system; 2) 2D Consistent semantic segmentation; 3) 2D-3D semantic segmentation.

3.1 SLAM System

The 3D segmentation method takes a 3D point cloud model

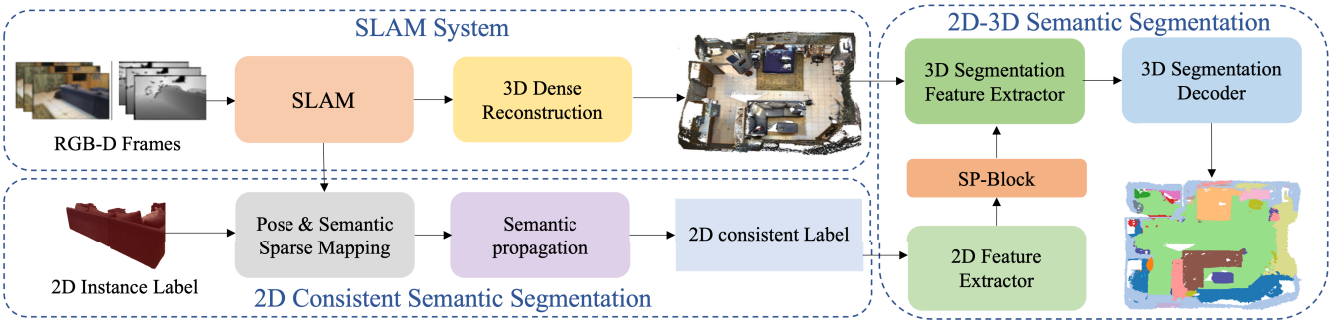
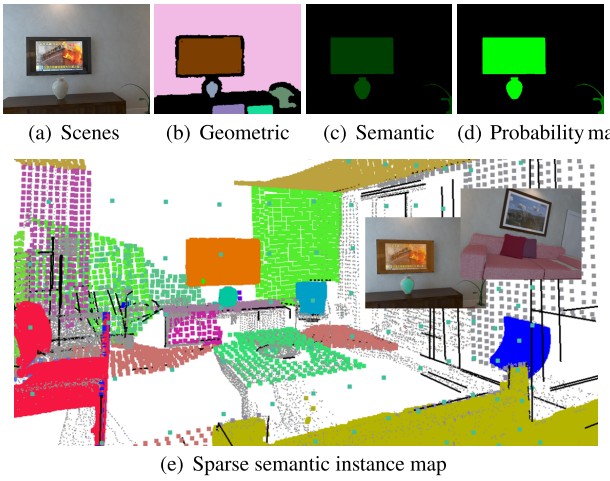


Fig. 2 Pipeline of the proposed system that is fed by sequential RGB-D pairs and generates 2D consistent semantic masks, a dense mesh model, and a 3D semantic segmentation result.



(a) Scenes (b) Geometric (c) Semantic (d) Probability map

(e) Sparse semantic instance map

Fig. 3 Sparse semantic instance map building and consistent 2D semantic image generation.

as input, which is generated by the Simultaneous Localization and Mapping (SLAM) system [15]. The RGB-D video stream captured by the SLAM system serves as the primary input of 3D reconstruction, thus SLAM plays a bridge role to RGB-D frames and 3D semantic segmentation. SLAM system consists of key components including frontend, backend, map representation, data association, loop closure detection, and pose estimation [15], [16], [24].

To correct the consistency of continuous 2D semantic segmentation labels and fuse them with 3D semantics segmentation, we add a sparse semantic map into the SLAM system and maintain and update the 2D semantic map during system operation, as shown in Fig. 3. In the mapping component, we extract 2D semantic information from each keyframe and combine them with geometry structure to obtain a 3D object semantic map. 3D objects' semantic labels are obtained from geometric and semantic segments of every keyframe, which are fused into a global sparse semantic map, as shown in Fig. 3. These local sparse semantic maps are then fused into a global sparse semantic map by using camera poses to achieve global semantic consistency. After that, the consistent object instance maps are fed to 3D segmentation pipeline to enhance 3D semantic segmentation, as shown in Fig. 2.

3.2 2D Consistent Semantic Segmentation

The consistent semantic segmentation strategy in this architecture is also an important module that is responsible to provides stable 2D semantic instance predictions of different views. In this module, two segmentation branches, learned [13] (Fig. 3(b)) and geometric [29] (Fig. 3(c&d)) methods, are used to deal with RGB and depth maps, respectively. As we all know, images that only capture part information of objects are useful for 3D semantic segmentation, since more details and boundary information can be obtained from there. Those partial scans, however, bring huge challenges to 2D semantic segmentation networks. To keep the consistency of segments, we take advantage of camera poses and the global semantic sparse map to correct those ill-posed results.

3.3 2D-3D Semantic Segmentation Network

In this module, an encoder-decoder network is implemented for the final 3D dense semantic segmentation task. In the encoder module, the proposed SP-Block is connected with the original encoder of MinkowskiNet [7] to build deep embedding features that are decoded in semantic predictions. Benefiting from the SP-Block, deep features from 2D and 3D domains can be fused to predict 3D semantic segments from the 3D dense reconstruction.

4. Odometry Based Consistent 2D Semantic Segmentation

Inconsistent semantic segmentation prediction between different RGB images of the same scene is a common issue in semantic segmentation methods [13], [14]. To solve this issue, an incremental joint 2D segmentation strategy is proposed to achieve sharp and consistent segments from each keyframe.

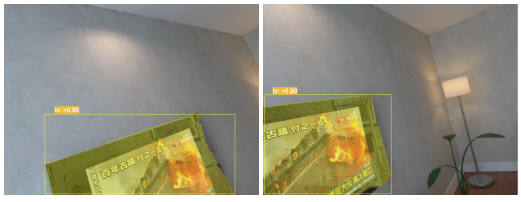
4.1 Segments from a Single RGB-D Pair

In this paper, each RGB image is fed to YOLACT [13] to segment instances and predict objects' labels, there are two

(a) *Couch* and *Chair* predicted from Yolact [13]

(b) *Couch* and *Couch* generated from our method

(c) *TV* and *Oven* predicted from Yolact [13]



(d) *TV* and *TV* generated from our method

Fig. 4 Consistent semantic segmentation performances in the ICL sequence.

types of outputs, label R_{rgb} and probability maps R_p, from the network, where the first one codes each index of detected objects as shown in Fig. 3(c) while the three channels of the second map (see Fig. 3(d)) is used to save the corresponding probabilities.

Since boundaries of semantic masks generated from an RGB image are commonly noisy, we extract areas with discontinuous depth information from the corresponding depth map, as shown in Fig. 3(b). Given depth maps, geometric-based shape segmentation methods [2], [29] are used to segment the scene into different instances according to the normal edge analysis. As shown in Fig. 4, the TV is segmented from a wall since the normal map detects disconnection regions between them.

Therefore, a filtered segmentation map R^* is obtained by

$$R^* = R_{rgb} \cdot R_d \tag{1}$$

here R_d is a binary map where instance-covered pixels are denoted as 1. We have to note that those segments extracted from the RGB image will be removed if they do not exist in the geometric map, which will be completed when the

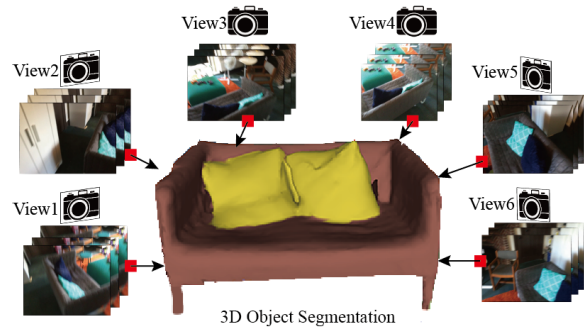


Fig. 5 The fusion of 2D image segmentation from different viewpoints.

information appears in both semantic and geometric images.

4.2 Semantic Propagation

According to the principle of multi-view geometry, objects in a 3D scene can be decomposed into RGB-D images with different viewpoints, which provide the basis for the 2D-assisted 3D scene understanding in this paper, as shown in Fig. 5. To ensure object consistency across varied views, we preserve a semantic sparse point cloud map composed of distinct geometric landmarks and semantic entities. This map aids in rectifying incorrect semantic labels deduced from partial scans and is constructed upon the foundational sparse map generated by visual odometry techniques.

To maintain consistent object representation across various views, a semantic sparse point cloud map, which embeds different geometric landmarks and semantic objects, is employed. This map serves as an anchor to refine the semantic labels derived from partial or erroneous scans, and is founded upon the preliminary sparse map provided by visual odometry methods.

4.2.1 Initialization of Semantic Labels

For each 3D object in the map, initialization is paramount to ensure coherent tracking throughout. The initial value of an object O_i is derived from the very first object segmentation.

4.2.2 Matching and Rectifying 2D Semantic Labels

When integrating a new keyframe, 3D objects denoted as $O_i, i \in (1,n)$ (where n symbolizes the count of 3D objects within the map) are re-projected to yield 2D re-projections, represented as o_i^{rp}. The alignment of these projections with detected semantic labels is ascertained through the Intersection over Union (IoU) metric:

$$\mathrm{IoU}(O_i^{rp}, o_j) = \frac{|O_i^{rp} \cap o_j|}{|O_i^{rp} \cup o_j|} \tag{2}$$

Where $|O_i^{rp} \cap o_j|$ represents the area (or pixel count) of the intersection between O_i^{rp} and o_j. $|O_i^{rp} \cup o_j|$ represents the area (or pixel count) of the union of the two regions.

To determine if O_i^{rp} and o_j are matched, two conditions must be satisfied: 1) they must share the same index, implying they belong to the same semantic class. 2) their associated probabilities must meet a predefined threshold to ensure a confident match. Let's assume the probability of o_j is represented as $P(o_j)$ and the probability associated with the re-projection O_i^{rp} is $P(O_i^{rp})$. A match is considered when:

$$\text{Index}(O_i^{rp}) = \text{Index}(o_j) \tag{3}$$

$$|P(O_i^{rp}) - P(o_j)| \leq \epsilon \tag{4}$$

where, ϵ is a small tolerance value to account for minor differences in probability estimates due to noise, inconsistencies, or other factors. If the difference between the probabilities is less than or equal to ϵ, they're deemed to be effectively the same, and thus, the objects are considered matched.

This calculation is undertaken for every $o_j, j \in (1, m)$, where m corresponds to the total number of identified objects in R^*. If the calculated IoU exceeds the predetermined threshold $t_{iou} = 0.4$, the next step involves evaluating the probability $P(o_j)$. A successful match between o_i^{rp} and o_j is ascertained when both the object index and $P(o_j)$ concur with those of o_i^{rp}. In instances where they don't align, an examination of the probabilities of other semantic labels in R^* is conducted. Objects showcasing probabilities that surpass the threshold $t_{p1} = 0.9$ are identified as new additions and consequently incorporated into the map.

4.2.3 Updating Object Probabilities

During the object fusion phase, when the re-projected data resonates with the current keyframe's semantic labels, there's a modification in the probability $P(O_i)$. Specifically, if the probabilities of the 2D segments exceed t_{p1}, an increment is applied to $P(O_i)$ as:

$$P(O_i) = P(O_i) + \alpha \tag{5}$$

Here, α denotes an increment factor, reflecting object reconfirmation across different perspectives. In the experiment, we set $\alpha = 0.1$. Conversely, in situations where these new segments achieve the required IoU but present varied semantic labels, the weights $W(O_i)$ of the corresponding 3D objects undergo a decrement:

$$W(O_i) = W(O_i) - \beta \tag{6}$$

In this context, β represents the decrement factor. In the experiment, we set $\alpha = 0.1$. 3D objects that plummet to a weight below the set threshold $t_{p2} = 0.7$ are subsequently excised from the map.

5. 2D-3D Semantic Segmentation

Given consistent 2D semantic images, camera poses, and a dense 3D model within the same coordinate, an encoder-decoder architecture as shown in Fig. 6 is introduced in this section to predict semantic segmentation.

5.1 Interested Regions Selection

Mapping 2D features onto their 3D counterparts is a fundamental process in many 3D semantic segmentation methods. A notable approach in this domain is presented by 3DMV [23] and 3D-SIS [9], where they utilize differentiable projection layers to achieve this mapping. While promising, there's a pertinent challenge: introducing undesired noise into the 3D branch, particularly when 2D features lack corresponding matches in the 3D models.

To mitigate this challenge and achieve a cleaner projection, we adopt a bidirectional projection strategy. In the Fig. 6, the bidirectional arrow signifies a reciprocal projection mechanism between 2D imagery and 3D models. Specifically, 2D images can be projected onto the 3D model leveraging viewpoint and depth cues. Conversely, the 3D

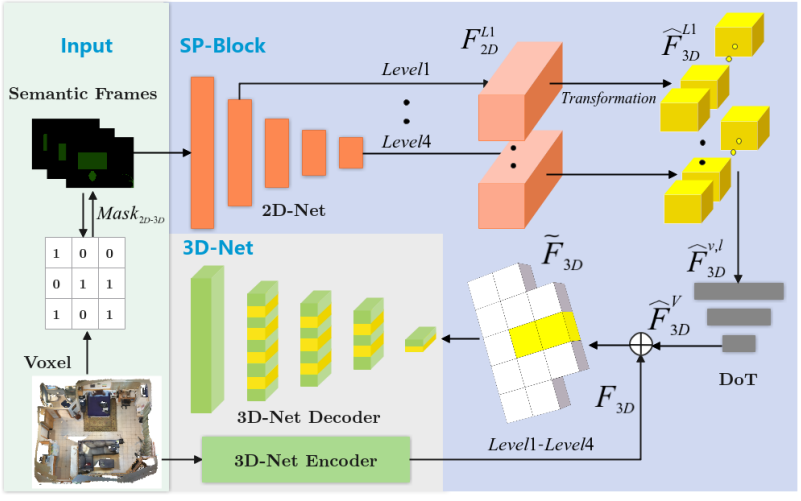


Fig. 6 2D-3D semantic segmentation network. The input including 2D semantic segmentation is associated with 3D voxels, and the corresponding relationship mask is established through back projection.

model can be back-projected onto 2D images. This reciprocity ensures that, within the 3D realm, only regions possessing definitive and meaningful correspondences in 2D imagery are contemplated. This strategy produces a binary mask B_k that indicates the intersection of the projections, inspired by the literature [3].

Formally, the mask can be defined as:

$$B_k = KT_{kw}P \bigcirc R_k^* \tag{7}$$

Here, $\bigcirc$ denotes the *and* operation, signifying an intersection between the k^{th} semantic input image, represented by R_k^*, and the re-projected image derived from the 3D model. The transformation matrix T_{kw}, capturing the 6 Degrees of Freedom (DoFs) pose, facilitates the conversion from the world coordinates to the k^{th} camera coordinate. Additionally, K signifies the intrinsic matrix, capturing the camera's internal characteristics, essential for the accurate re-projection process.

By employing this approach, we not only reduce noise in our projections but also ensure that our 2D-3D mapping remains consistent, meaningful, and directly relevant to the scene's semantics.

5.2 Semantic Projection Block

As shown in Fig. 6, the embedding $\widetilde{F}_{3D}$ is constructed by two branches, F_{3D} from the encoder of MinkowskiNet [7] and $\hat{F}_{3D}^V$ from our semantic projection block introduced in this section, where V is the number of views which chose 4. The view selection is based on the adjacency principle to ensure capturing the same object from different angles and to simplify the 2D-3D mapping computation.

First, we extract deep feature pyramids by using ResNet-18 [30] from multi-view semantic images. For each view, there are four levels of feature maps $F_{2D}^l, l \in [1 \dots 4]$ extracted from 2D image. To maintain compatibility with deep features from the encoder of 3D-Net [7], we project each feature channel of F_{2D}^l to the shape of $N \times 1$ based on the camera pose and intrinsic matrix, where N is the number of voxels in the 3D model. Therefore, each level's feature maps (with C channels) are transferred to a shape as $N \times C$. And then at the same level of V views, following [3] we concatenate those transferred shapes along the channel direction to obtain $\hat{F}_{3D}^{v,l}$ with the size of $N \times C \times V$.

Then the Domain Transformation (DoT) operation that is constructed by four 3D sparse convolutional layers and a sparse max-pooling layer is proposed to aggregate feature volumes $\hat{F}_{3D}^{v,l}$ from different views, as shown in Fig. 7.

$$\hat{F}_{3D}^V = \sum_{v=1}^{V} \sum_{l=1}^{L} DoT(\hat{F}_{3D}^{v,l}) \tag{8}$$

where L is the size of feature levels while V is the number of views. Via the DoT operation, the shape of $\hat{F}_{3D}^{v,l}$ is transferred with the same size of F_{3D}. Moreover, the spatial and semantic information from different views is fused.

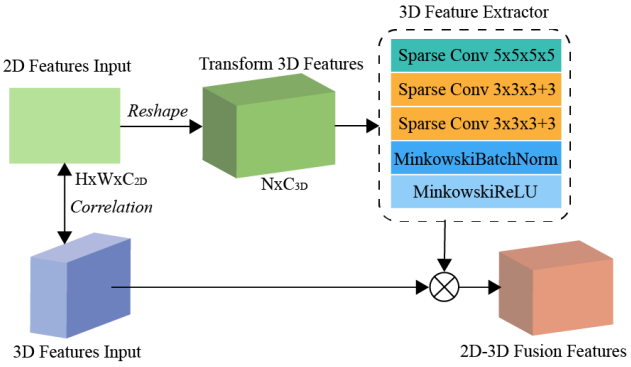


Fig. 7 Detail of DoT.

Finally, corresponding levels of encodes' features F_{3D} based on MinkowskiNet and $\hat{F}_{3D}^V$ based on our SP-Block are fused by a concatenation operation as

$$\widetilde{F}_{3D} = F_{3D} \oplus \hat{F}_{3D}^V \tag{9}$$

where $\widetilde{F}_{3D}$ is the fusion embedding that is fed to the decoder to obtain semantic prediction results.

5.3 Decoder and Training

In our work, we've integrated the decoder from MinkowskiNet [7] to facilitate the generation of semantic labels for individual 3D points. This choice was made due to MinkowskiNet's robust architecture and proven performance in handling 3D data. For those who wish to delve deeper into its mechanics, we direct them to the original MinkowskiNet paper [7].

To optimize our encoder-decoder structure, we've incorporated the conventional cross entropy method [31], known for its effectiveness in segmentation tasks. It's worth noting that during the training phase, we rigorously ensure that our network is fed with accurate ground truth data. This includes both 2D semantic labels and the respective 3D reconstruction models for each scenario, ensuring comprehensive training and improved accuracy in results.

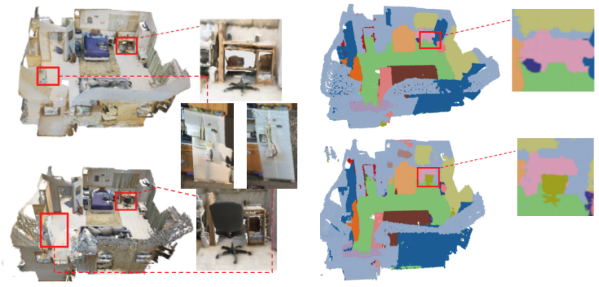
5.4 Implementation Details

The network is implemented on the PyTorch platform, which exploits the SGD optimizer with a learning rate of 0.01 and a momentum of 0.9. Furthermore, the network is trained on the machine with 4 NVIDIA GeForce 2080TI GPUs and 64 GB RAM, where the batch size is set to 12 in 100 training epochs.

The size of RGB-D images fed to our tracking system is 480×640, while the size of semantic images for SP-Block is downsampled to 240×320. The channels' size of four layers in the feature pyramid are 512, 256, 128 and 96, respectively. After the DoT operation, the channels' size of four layers in $\hat{F}_{3D}^V$ are 256, 128, 128 and 96, respectively.

Table 1 The quantitative accuracy comparison of the final semantic segmentation results on the ScanNetV2 validation dataset. We use bold and blue numbers to mark the best and second results per instance, respectively.

Method	bath	bed	bkshf	cab	chair	cntr	curt	desk	door	floor	other	pic	fridge	shower	sink	sofa	table	toilet	wall	window
SF [27]	59.8	46.8	32.0	35.7	46.9	33.2	46.9	34.7	35.7	72.2	34.7	21.7	34.3	28.8	47.2	43.7	37.8	65.5	58.3	29.5
SR [34]	69.7	52.6	31.2	31.7	64.0	24.0	30.3	26.1	30.9	80.6	33.3	7.3	56.3	23.6	46.2	58.3	51.6	73.3	66.9	21.1
PF [28]	57.5	67.0	48.4	44.8	66.7	35.8	53.3	42.0	35.6	81.0	40.3	30.2	47.0	50.8	52.6	61.3	54.8	82.1	65.7	45.9
PsF [35]	65.6	61.2	65.7	48.6	68.4	41.7	54.9	48.9	47.5	87.1	43.7	25.7	41.8	34.5	53.4	59.8	54.0	78.9	70.6	47.0
FPC [36]	**85.4**	**82.3**	60.4	60.9	75.1	56.0	64.8	58.2	**64.8**	91.9	46.4	**40.6**	**64.2**	51.7	63.5	77.9	68.9	87.0	**83.8**	56.3
SPV [37]	73.4	78.5	79.1	60.5	80.6	59.3	**70.4**	59.9	60.5	91.1	57.8	35.0	57.5	**75.2**	61.3	72.6	64.4	86.4	80.5	61.7
BPNet [3]	83.0	80.1	78.2	61.8	89.0	61.9	58.5	65.7	57.1	93.8	53.3	23.7	44.2	61.7	65.2	79.4	72.7	89.5	81.7	59.3
Ours	83.4	79.2	76.9	61.0	88.9	57.9	58.3	64.2	58.3	93.9	53.2	24.3	44.4	65.1	63.4	80.2	71.4	89.0	81.4	58.5
Ours+	84.9	81.6	**80.8**	**65.9**	**89.9**	**62.4**	59.1	**70.5**	58.7	**94.7**	**58.8**	25.6	48.3	67.3	**65.7**	**83.4**	**75.5**	**89.9**	82.6	**62.0**



(a) Dense reconstruction (b) Semantic segmentation

Fig. 8 Dense reconstruction results. Up: ScanNet ground truth. Down: ours.

Table 2 3D semantic segmentation mIoU and mAccu results on the ScanNetV2 validation set. 'Y' means that a method has a dense reconstruction function. We use '-' to mark unsure situations and '*' means the result coming from [37].

Method	recon.	voxel	mIoU	mAcc
SF [27]*	Y	-	42.2	47.4
SR [34]*	Y	-	44.0	65.6
PF [28]*	Y	-	53.1	68.7
PsF [35]*	Y	-	55.0	70.3
FPC [36]*	Y	-	67.2	77.0
SPV [37]*	Y	1cm	68.3	79.6
BPNet [3]	N	5cm	67.1	88.3
Ours	Y	5cm	67.8	88.5
Ours+	Y	5cm	**68.7**	**89.3**

6. Experiments

In this section, the performances, including dense mapping and 3D segmentation, of the system are evaluated on public datasets and compared with state-of-the-art methods.

6.1 Dataset

6.1.1 ScanNetV2

The ScanNetV2 [32] dataset includes 1513 sequences (around 2.5 million RGB-D frames) from 70 unique indoor scenes, which provides ground truth annotations for training, validation, and testing directly on 3D reconstructions. Those sequences are split into training, validation, and testing datasets where the semantic labels are defined according to the rule of NYU40 [33].

6.2 Accuracy of Dense Reconstruction

We compare the qualitative reconstruction results between ours and the ground truth of ScanNetv2 that is built from BundleFusion [26]. As shown in Fig. 8(a), our method can reconstruct the chair completely. Benefiting from our accurate pose estimation module and the smooth dense reconstruction strategy, the refrigerator is reconstructed more accurately than the ground truth. Related semantic segmentation results are shown in Fig. 8(b).

6.3 3D Semantic Segmentation Results

6.3.1 Quantitativeanalysis

Following the common evaluation metrics in previous works, the standard mean Intersection over Union (mIoU) and mean Accuracy (mAcc) as used to evaluate the performance of our network. Our 3D semantic segmentation results are shown in Table 1 and Table 2, where we compare our network with state-of-the-art pipelines. Similar to our methods, SF [27], SR [34], SPV [37] and PF [28] are semantic reconstruction systems based RGB-D images, while BPNet [3] deals with point clouds. To demonstrate the efficacy of 2D-assisted 3D semantic segmentation and the efficiency of 2D consistent semantic segmentation, in the subsequent experiments, the label 'our' signifies outcomes from our 2D-3D fusion technique without 2D consistency correction. In contrast, 'ours+' denotes results from our method with the '2D consistent semantic segmentation' correction module integrated.

In Table 1, our approach achieves higher mIoU and mAcc scores across various semantic classes. Significantly, our method, denoted as "Ours," seamlessly integrates 2D semantic information with 3D semantics. Additionally, we introduce an enhanced version of our method, referred to as "Ours+," which incorporates a consistency correction step based on the 2D semantic results. Our method directly fuses 2D instance semantic, which demonstrates superior performance in accurately predicting tables and chairs, as indicated by the highest scores in these categories. Our method achieves even better results, with in-

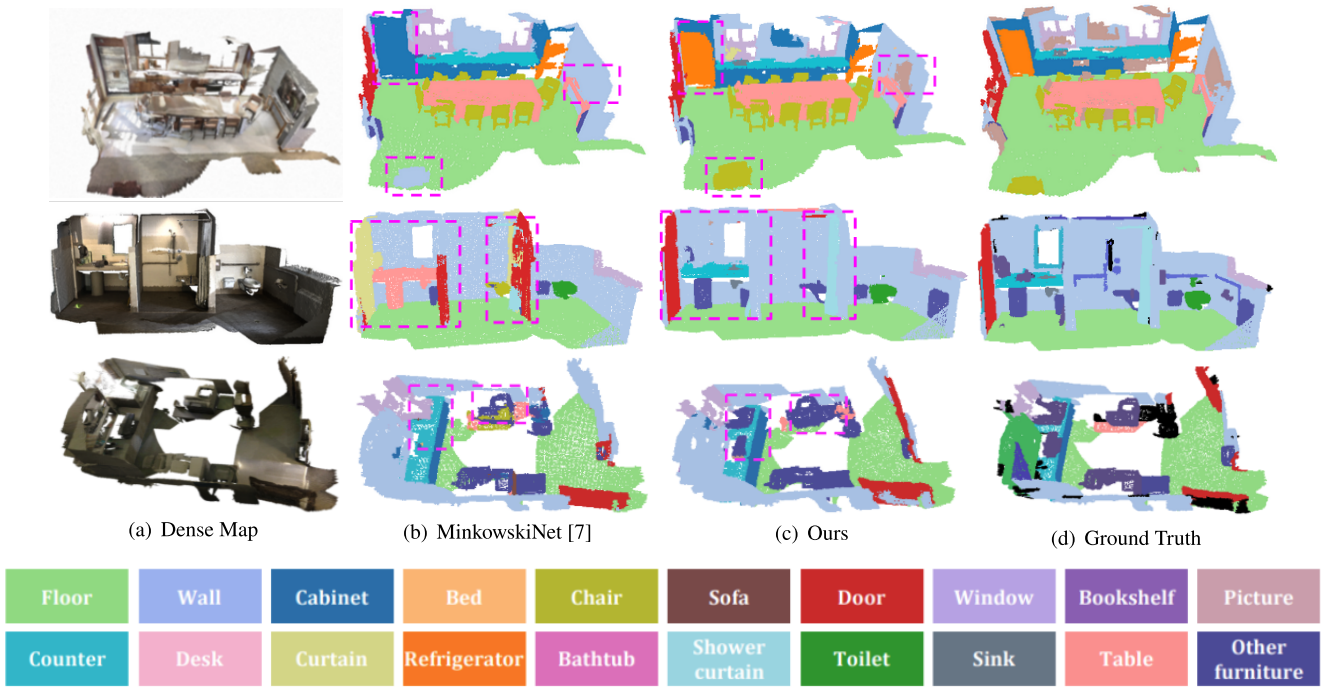


Fig. 9 A qualitative comparison of 3D semantic segmentation between MinkowskiNet and our method. We use pink boxes to highlight the difference between them. Different classification information is represented by different colors, and the classification information corresponds to the bottom color palette.

creased mIoU and mAcc scores in multiple classes, including "bath", "bed", "bkshf", "cab", "chair", "cntr", "curt", "desk", "door", "floor", "other", "pic", "fridge", "shower", "sink", "sofa", "table", "toilet", "wall", and "window". The results highlight the effectiveness and superiority of our proposed method in enhancing semantic segmentation accuracy compared to state-of-the-art approaches. The incorporation of 2D semantic information, along with the consistency correction step, contributes to improved performance across various semantic classes. The experimental results demonstrate the potential of our method for advancing robotic scene understanding tasks.

Furthermore, we conducted a comprehensive evaluation by comparing the mean Intersection over Union (mIoU) and mean Accuracy (mAcc) metrics among different methods, as presented in Table 2. It is important to note that for both BPNet [3] and our proposed method, the voxel size utilized was set to 5 cm. In contrast, SPV [37] employed a voxel size of 1 cm. The choice of voxel size plays a crucial role in balancing the trade-off between accuracy and computational complexity. When using a smaller voxel size, the models are able to capture more fine-grained and detailed information, leading to potentially more accurate predictions. However, it is worth noting that this finer voxel resolution also requires more intensive computational resources and imposes higher hardware requirements.

In our experiments, we evaluated the performance of various methods for semantic segmentation based on the reconstruction capability and voxel size used. Table 2 summarizes the results in terms of mIoU and mAcc metrics. Compared to these methods, our approach demonstrates promising results in terms of mIoU and mAcc scores. Specifically, our method achieves an mIoU of 67.8, outperforming SF, SR, and PF by 25.6%, 23.8%, and 14.7%, respectively. Similarly, our approach achieves an mAcc of 88.5, surpassing SF, SR, and PF by 41.1%, 22.9%, and 19.8%, respectively. Furthermore, we introduce a refined version of our method, denoted as "Ours+," which incorporates a consistency correction step based on 2D semantic results. Ours+ achieves even better results, with an mIoU of 68.7, representing an improvement of 0.9% over our base method, and an mAcc of 89.3, showing an improvement of 0.8%. These results highlight the effectiveness of our proposed method in achieving competitive performance in semantic segmentation. The incorporation of 2D semantic information, along with the consistency correction step, contributes to improved accuracy compared to existing methods. The experimental results demonstrate the potential of our method for enhancing the quality of semantic segmentation in robotic scene understanding tasks.

These results highlight the effectiveness of our proposed method in achieving competitive performance in semantic segmentation. The incorporation of 2D semantic information, along with the consistency correction step, contributes to improved accuracy compared to existing methods. The experimental results demonstrate the potential of our method for enhancing the quality of semantic segmentation in robotic scene understanding tasks.

Table 3 3D semantic segmentation results of different 2D view numbers on the validation set of ScanNetV2.

Method	1	2	3	4	5
IoU	65.32	66.25	67.54	68.3	66.65

Table 4 3D semantic segmentation results of different levels on the validation set of ScanNetV2.

Level-1	Level-2	Level-3	Level-4	**IoU**
✓				66.42
	✓			66.90
		✓		66.12
			✓	67.67
✓	✓			67.68
✓	✓	✓		68.14
✓	✓	✓	✓	68.76

6.3.2 Qualitative Analysis

As depicted in Fig. 9, we present a qualitative comparison of the 3D semantic segmentation results between our method and MinkowskiNet [9], along with the corresponding ground truth, in three different scenes. The experimental results demonstrate that our method is capable of accurately and comprehensively segmenting the scenes, especially for objects that are challenging to segment using traditional 3D point cloud methods, as indicated by the highlighted red regions in Fig. 9. These findings highlight the enhanced effectiveness of our method in achieving 3D semantic segmentation by incorporating 2D semantic consistency.

6.4 Ablation Experiment

6.4.1 Ablation for the Number of 2D Views

In Sect. 5.2, we use four distinct 2D views for projecting onto 3D to enhance the 3D features. We subsequently investigate the impact of the number of 2D views on 3D semantic segmentation, and present the results in Table 3. For 3D semantic segmentation, as the number of 2D views increases, the segmentation accuracy improves gradually. However, when the number of views reaches five, the accuracy slightly declines. This suggests that too few views fail to provide sufficient 2D information, while an excessive number of views hinders the network from effectively extracting useful information and discarding redundant data.

6.4.2 Ablation for 2D to 3D Projection Level

As shown in Fig. 6, the SP-Block integrates four 2D pyramid-level features into their corresponding 3D feature hierarchies. For ablation, we perform 2D-3D fusion at certain feature levels and compare the fusion results with the baseline method (MinkowskiUNet40) at each level. From the first four rows of Table 4, we observe that the 3D segmentation results are similar at each level. This indicates that individual levels of 2D features can enhance the 3D segmentation results, but the effect is comparable. However, as the feature fusion levels increase, as seen in rows 5 to 7 of Table 4, the 3D segmentation results significantly improve. This suggests that the combination of low-level and high-level features can better assist 3D semantic segmentation.

7. Conclusion

In conclusion, this paper introduces a novel approach to enhance 3D semantic segmentation by utilizing 2D semantic segmentation from RGB-D sequences. This approach incorporates SLAM's tracking pipeline to generate consistent 2D semantic maps and integrates features from localized 2D fragments into their corresponding 3D semantic features using the SP block. The 3D semantic segmentation network employs a combination of 2D-3D fusion features to achieve a merged semantic segmentation process for both 2D and 3D data. Extensive experiments on public datasets demonstrate the state-of-the-art performance of the proposed 2D-assisted 3D semantic segmentation method. By leveraging the complementary capabilities of 2D and 3D segmentation, our approach effectively addresses the limitations of 3D semantic segmentation.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (62273081), Liaoning Provincial Foundation (2022JH2/101300202).

References

[1] C.R. Qi, L. Yi, H. Su, and L.J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," arXiv preprint arXiv:1706.02413, 2017.
[2] F. Furrer, T. Novkovic, M. Fehr, A. Gawel, M. Grinvald, T. Sattler, R. Siegwart, and J. Nieto, "Incremental object database: Building 3D models from multiple partial observations," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018.
[3] W. Hu, H. Zhao, L. Jiang, J. Jia, and T.T. Wong, "Bidirectional projection network for cross dimension scene understanding," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.
[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Trans. Pattern Anal. Mach. Intell., vol.39, no.6, pp.1137–1149, 2017.
[5] A.O. Vuola, S.U. Akram, and J. Kannala, "Mask-RCNN and U-Net ensembled for nuclei segmentation," 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019) 2019.
[6] P. Hermosilla, T. Ritschel, P.P. Vázquez, À. Vinacua, and T. Ropinski, "Monte carlo convolution for learning on non-uniformly sampled point clouds," ACM Trans. Graphics (TOG), vol.37, no.6, pp.1–12, 2018.
[7] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal ConvNets: Minkowski convolutional neural networks," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[8] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching efficient 3D architectures with sparse point-voxel convolution," European Conference on Computer Vision (ECCV), pp.685–702, 2020.
[9] J. Hou, A. Dai, and M. Nießner, "3D-SIS: 3D semantic instance segmentation of RGB-D scans," Proc. IEEE/CVF Conference on

Computer Vision and Pattern Recognition, 2019.
[10] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," Proc. IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2019.
[11] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, "End-to-end video instance segmentation with transformers," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.8741–8750, June 2021.
[12] S.C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari, "SceneGraphFusion: Incremental 3D scene graph prediction from RGB-d sequences," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
[13] D. Bolya, C. Zhou, F. Xiao, and Y.J. Lee, "YOLACT: Real-time instance segmentation," ICCV, 2019.
[14] D. Bolya, C. Zhou, F. Xiao, and Y.J. Lee, "YOLACT++: Better real-time instance segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol.44, no.2, pp.1108–1121, 2022.
[15] R. Mur-Artal, J.M.M. Montiel, and J.D. Tardos, "ORB-SLAM: A versatile and accurate monocular slam system," IEEE Trans. Robot., vol.31, no.5, pp.1147–1163, 2015.
[16] C. Campos, R. Elvira, J.J.G. Rodríguez, J.M. Montiel, and J.D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," IEEE Trans. Robot., vol.37, no.6, pp.1874–1890, 2021.
[17] Z. Tian, C. Shen, X. Wang, and H. Chen, "BoxInst: High-performance instance segmentation with box annotations," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.
[18] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," European Conference on Computer Vision, pp.213–229, 2020.
[19] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[20] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, "End-to-end video instance segmentation with transformers," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.
[21] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.
[22] C.R. Qi, H. Su, K. Mo, and L.J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[23] A. Dai and M. Nießner, "3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation," Proc. European Conference on Computer Vision (ECCV), pp.458–474, 2018.
[24] 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, 2017.
[25] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, and A.W. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," IEEE International Symposium on Mixed and Augmented Reality, 2012.
[26] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration," ACM Trans. Graph. (ToG), vol.36, no.4, pp.1–18, 2017.
[27] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," 2017 IEEE International Conference on Robotics and Automation (ICRA), 2016.
[28] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, "PanopticFusion: Online volumetric semantic mapping at the level of stuff and things," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020.
[29] K. Tateno, F. Tombari, and N. Navab, "Real-time and scalable incremental segmentation on dense slam," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[31] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol.521, no.7553, pp.436–444, 2015.
[32] A. Dai, A.X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[33] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," ECCV, pp.746–760, 2012.
[34] J. Jeon, J. Jung, J. Kim, and S. Lee, "Semantic reconstruction: Reconstruction of semantically segmented 3D meshes via volumetric semantic fusion," Computer Graphics Forum, vol.37, no.7, pp.25–35, 2018.
[35] Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Real-time progressive 3D semantic segmentation for indoor scenes," 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019.
[36] J. Zhang, C. Zhu, L. Zheng, and K. Xu, "Fusion-aware point convolution for online semantic 3D scene segmentation," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[37] S.S. Huang, Z.Y. Ma, T.J. Mu, H. Fu, and S.M. Hu, "Supervoxel convolution for online 3D semantic segmentation," ACM Trans. Graph. (TOG), vol.40, no.3, pp.1–15, 2021.

Yingcai Wan received his B.S. and M.S. degrees from Liaoning Technical University, China, in 2014 and 2017. He is currently a Ph.D. student in robotics science and engineering at Northeastern University. His main research interests include depth estimation, dynamic motion estimation, and 3D scene reconstruction.

Lijin Fang graduated from industrial electrical automation in Xi'an Jiaotong University, China, in July 1988. From November 1992 to November 1996, he was sent to Russia to study and obtained a Ph.D. degree. From August 1988 to June 2009, he worked at the Shenyang Institute of Automation, Chinese Academy of Sciences, China. From June 2009 to December 2016, he was a professor and doctoral supervisor at the School of Mechanical Engineering and Automation, Northeastern University, China. Since January 2017, he has been a professor and doctoral supervisor in the Faculty of Robot Science and Engineering, at Northeastern University, China. His current research interests include the design and control of the bionic climbing mobile robot, humanoid bionic control of robots with variable stiffness, and high-precision robot control.