2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

EVSMap: An Efficient Volumetric-Semantic Mapping Approach for Embedded Systems

Jiyuan Qiu[1], Chen Jiang[1]*, Pengfei Zhang [1], Haowen Wang[1]

***Abstract*—Despite significant progress in perception tasks such as 3D scene mapping and semantic information extraction using SLAM and deep learning, applying these techniques within computationally constrained embedded systems remains a challenge. In this work, we introduce a novel end-to-end framework for efficient and real-time volumetric-semantic mapping. We have developed a lightweight and robust RGB-D segmentation network for extracting semantic information. Through the introduction of three distinct modules—CFIM, DAPPF, and LAD—our network significantly enhances real-time performance while achieving Mean Intersection over Union (MIoU) scores comparable to state-of-the-art (SOTA) models. Our model reduces the parameters by 8 to 26 times compared to similar networks and improves inference speed by 2 to 3 times. Additionally, we improved a multi-class bayesian updating strategy by refining penalty function to reduce the memory size of the semantic map and enhance the mapping speed. Compared with other volumetric-semantic mapping approaches, our work maintains the same level of detail in semantic information representation, while increasing mapping speed by 1.3 to 9.6 times and reducing memory size of the map by up to 2.6 times. Finally, we applied our work to real-world mobile robot exploration scenarios, demonstrating the efficiency of the proposed framework.**

I. Introduction

This paper primarily focuses on real-time environmental perception and scene understanding for mobile robots operating in indoor settings. As the operational environments and tasks of mobile robots become increasingly complex, utilizing multimodal sensor fusion to enhance their perceptual capabilities has emerged as a principal approach. With the widespread adoption of RGB-D cameras, the acquisition of depth data has become more accessible. Depth data, being less susceptible to changes in ambient lighting, can naturally describe 3D geometric information and represent the structural information of objects within the environment. Therefore, in indoor scenes characterized by dense objectives and significant environmental disturbances, depth data can be employed to complement RGB data, thereby augmenting the robot's ability to perceive its environment. One of the primary methods by which robots achieve environmental perception and scene understanding is through the construction of metric maps. Occupancy grid mapping, an effective technique for building metric maps, enables the differentiation between the navigable and occupied spaces surrounding mobile robots. As the requirements for metric maps escalate, efficiently integrating advanced semantic information from unknown environments into these maps has become increasingly vital. Currently, the mainstream methods can be divided into two types. The first involves conducting three-dimensional semantic segmentation directly on reconstructed large-scale 3D maps, a method that faces limitations due to the expansion of scene maps, which in turn increases data storage costs and computational resource demands. The second method maps the results of two-dimensional semantic segmentation onto three-dimensional maps to imbue them with semantic information. Our work is primarily based on this method, as it not only facilitates the convenient acquisition of a vast amount of image data but also offers higher computational efficiency, meeting the real-time requirements of mobile robot operations.

The authors are with the School of Aerospace Engineering, Tsinghua University, Beijing, 100084, China. (Chen Jiang* is the corresponding author, e-mail: jc2017@mail.tsinghua.edu.cn).

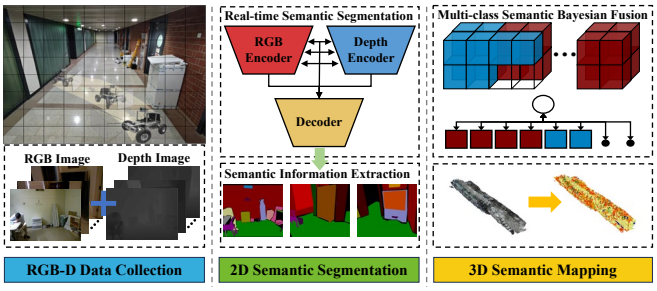


Fig. 1: Our semantic mapping framework EVSMap comprises three modules. The *RGB-D data collection module* which extracts information about the surrounding environment through an RGB-D camera mounted on the robot. The *2D semantic segmentation module* which utilizes a real-time network model to predict semantic information from the input data. Finally, The *3D semantic segmentation module* which maps the surrounding environment by integrating semantic information using a method based on an improved multi-class bayesian fusion.

The main contributions of this work are as follows:

1) We propose a novel end-to-end framework EVSmap for efficient and real-time volumetric-semantic mapping, which balances the reliability and real-time performance of mapping. Fig. 1 provides an overview of the proposed system, which is mainly composed of three modules.

2) A new lightweight semantic segmentation network, named FEAGNet, is designed with two variants. The network structure incorporates three distinct feature fusion and extraction modules, enabling more effective feature extraction while maintaining low computational complexity.

3) We employ a multi-class bayesian approach for updating the voxel nodes of the semantic map and improve its node fusion strategy by refining penalty function to enhance the efficiency of semantic mapping.

4) An experimental platform based on the embedded processor Jetson Agx Xavier for unmanned systems was constructed and deployed. Experiments conducted on public datasets and in real-world environments demonstrate the efficiency of the proposed framework.

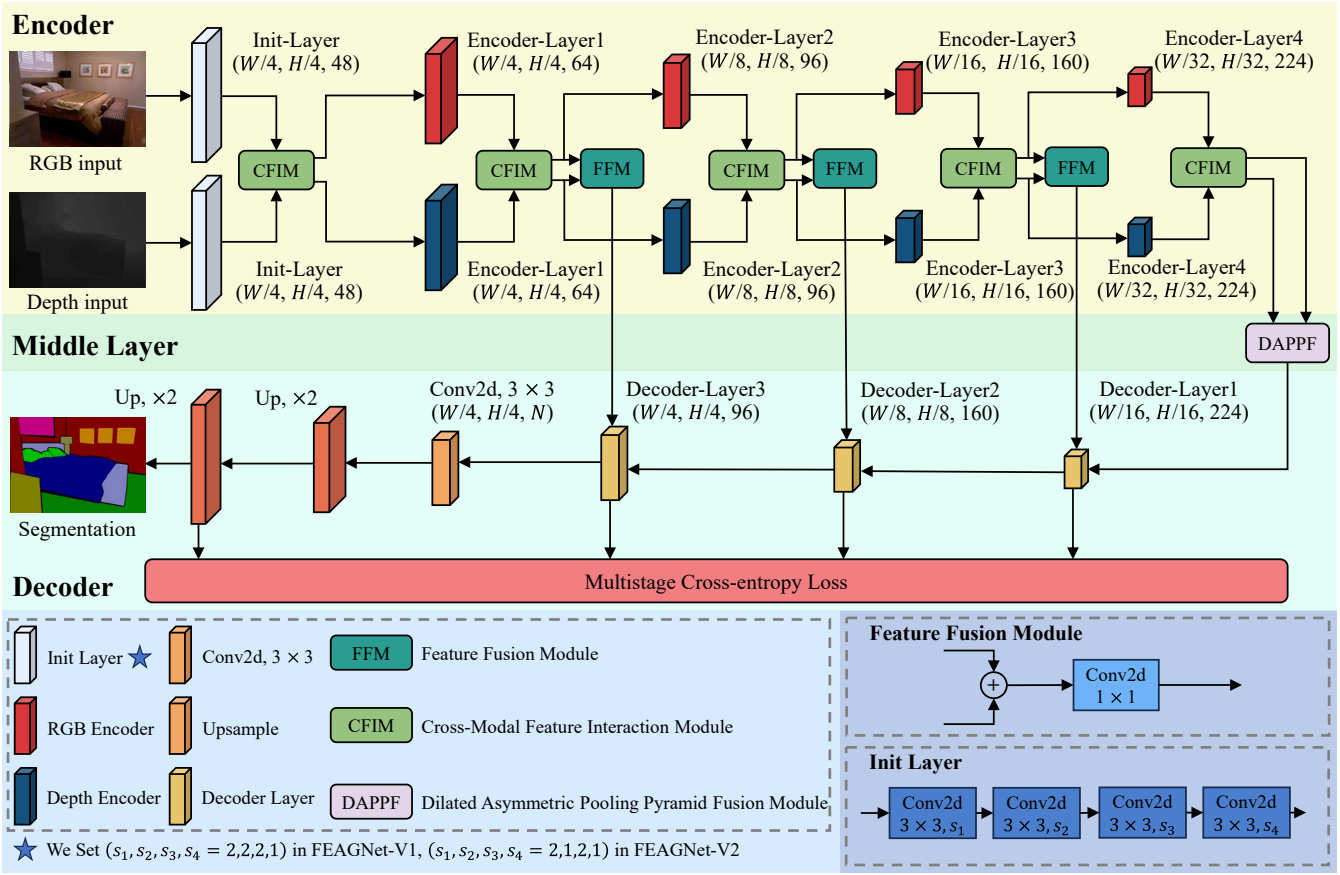


Fig. 2: Architecture of FEAGNet. From top to bottom are RGB Encoder, Depth Encoder, DAPPF Block and Decoder. The Encoder Block use FC-HarDNet-70 [11] as the feature extractor layer. The output of each layer is weighted by CFIM and fed into the next encoder block. The DAPPF Module aggregates global information and outputs it to the decoder. The Decoder is composed of the proposed LAD blocks.

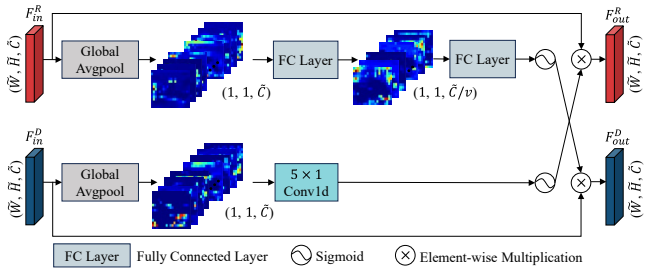


Fig. 3: Illustration of the CFIM.

II. Related Work

A. 2D Semantic Segmentation

The goal of semantic segmentation is to allocate category labels to each pixel within a scene. Semantic segmentation networks based on deep learning methods, such as FCN [1], PSPNet [2], and Deeplabv3+ [3], have been the focal point of attention due to their efficient feature extraction capabilities. With the advent of RGB-D sensor, the incorporation of depth information has proven to enhance the accuracy of network segmentation. Current fusion techniques often rely on straightforward methods such as addition or concatenation, as demonstrated by networks like FuseNet [4], LDFNet [5] and RedNet [6]. However, networks that integrate context information and cross-modal complementary data, such as ESANet [7], and SGACNet [8], significantly improve the capability for feature extraction. Despite most networks concentrating on augmenting segmentation quality, there is a relative scarcity in studies on real-time RGB-D segmentation networks. Therefore, our work focuses on enhancing the inference speed and reducing the parameter size of RGB-D semantic segmentation networks to achieve real-time performance requirements.

B. 3D Semantic Mapping

3D semantic mapping can be divided into vision-based and LiDAR-based types, with the former offering advantages such as low sensor costs and abundant information. These initial attempts harnessed dense point clouds, exploiting depth measurements and machine learning algorithms to significantly elevate the fidelity of mapping outputs. Recent studies have shifted towards utilizing deep learning for semantic mapping, like PanopticFusion [9], achieving advanced mapping outcomes. However, these systems are generally tailored for small indoor scenes, and challenges persist in mapping large-scale environments. In response to these challenges, some studies have attempted to create 3D maps with semantic information or implement semantic maps in the form of SparseFusion, but real-time performance remains a bottleneck. To address this, new approach like ClusterVO, DSP-SLAM, NodeSLAM, Dyna-SLAM and SimVODIS++ have been introduced [10], Striving to enhance real-time capabilities by optimizing data association, reducing computational load, and simplifying the structure of the system.

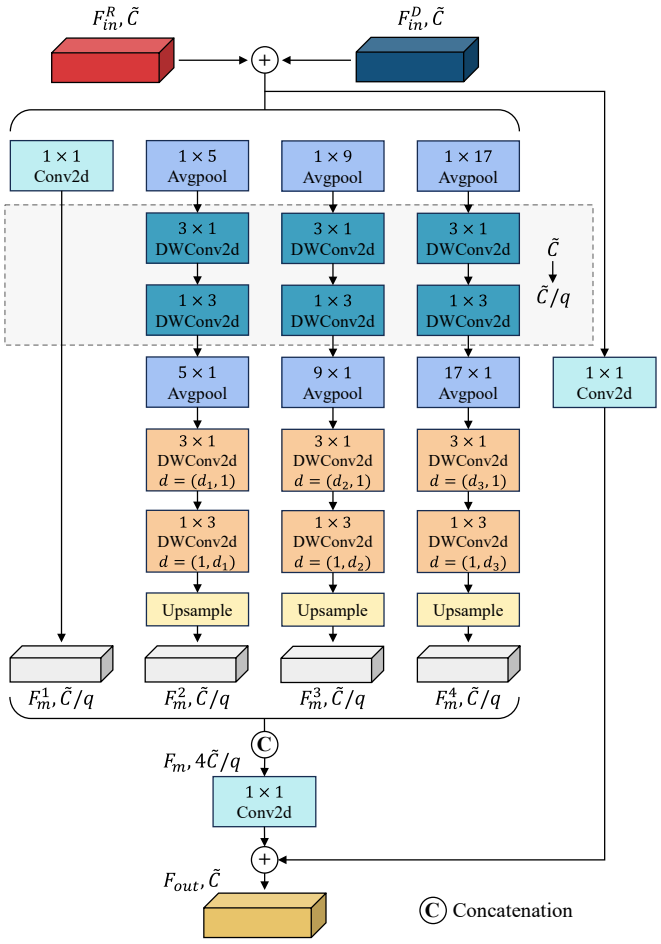


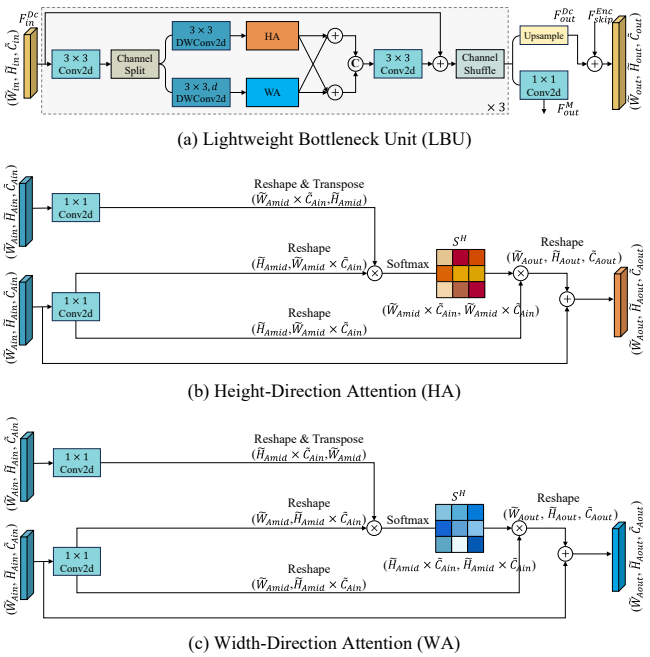
Fig. 4: Illustration of the DAPPF.

III. METHODOLOGY

In this section, we primarily introduce the proposed EVSMap system. Firstly, we describe the overall architecture of the proposed FEAGNet in III-A, including three key modules: Cross-Modal Feature Interaction Module (CFIM), Dilated Asymmetric Pooling Pyramid Fusion Module (DAPPF), and Lightweight Attention Decoder (LAD). And then in III-B, we detail the 3D semantic segmentation module of the system.

A. 2D Semantic Segmentation Module

1) Overall Architecture: As shown in Fig. 2, our FEAGNet is based on the encoder-decoder architecture. We take FCHarDNet as a baseline, which employs HarDBlock [11] to reduce network parameters and memory usage while maintaining high accuracy. Therefore, FEAGNet is designed with two symmetric encoders for extracting multi-scale features from RGB images and depth images respectively. The structure of both encoders remains consistent except for the number of input channels at the Init Layer. Additionally, we refine FEAGNet into two versions (V1 and V2), by adjusting the stride of the convolution in the Init Layer to further reduce the network's computational load. Through the proposed CFIM module, local information fusion of RGB and depth features is achieved. And unlike FCHarDNet, the decoder in FEAGNet is not a mirror of the encoder, we use our DAPPF module at the last encoder layer to fully leverage



(a) Lightweight Bottleneck Unit (LBU)

(b) Height-Direction Attention (HA)

(c) Width-Direction Attention (WA)

Fig. 5: Illustration of the LAD.

the global information extracted by the encoder. The proposed Lightweight Attention Decoder (LAD), based on our Height-Direction and Weight-Direction attention, merges features from encoder skip connections and upsample them by a factor of 2. Finally, a 1×1 convolution restores the resolution to the input size.

2) Cross-Modal Feature Interaction Module: Within the encoder's five stages, to facilitate the interaction and transfer between the two types of features, we designed the Cross-Modal Feature Interaction Module (CFIM) to generate unbiased weights for dynamically selecting important information from both RGB features $F^R \in \mathbb{R}^{C\times H\times W}$ and depth features $F^D \in \mathbb{R}^{C\times H\times W}$. CFIM is primarily composed of two fully connected layers, a 5×1 1-D convolution operation, and several other elements, as illustrated in Fig. 3. It first performs a 1×1 global average pooling operation on the input features, followed by the reintegration of F^R and F^D using fully connected layers and a 1-D convolution. The fully connected layer performs dimensionality operations on feature F^R, achieving a transformation from C to C/v and then back to C, where $v = 16$, this operation can reduce the computational cost of the network. Finally, we can obtain the unbiased weights using the sigmoid function, which are then cross-multiplied with the input features to produce output features of the same size.

3) Dilated Asymmetric Pooling Pyramid Fusion Module: To more effectively aggregate multi-scale information and extract global context information, pyramid pooling modules have been widely applied in recent years. APPPM [12] is a pyramid pooling module that employs asymmetric pooling layers, aimed at more easily aggregating multi-scale information. Inspired by APPPM, we propose the DAPPF designed to reduce computational load, as illustrated in Fig. 4. This module comprises three main components: (a) The use of three sets of asymmetric pooling layers for deeper

extraction of feature map information. (b) The employment of two sets of asymmetric depthwise separable convolutions, which are computationally efficient albeit with slightly reduced accuracy, including one set with an asymmetric dilation rate $(4,8,12)$. (c) The number of channels in the pyramid branches is compressed from C to C/q, where the compression rate Q is set to 2 in our module. After the aforementioned operations, the features outputted from DAPPF maintain consistency in both the number of channels and size with the input feature maps.

4) Lightweight Attention Decoder: The decoder is tasked with upsample the resolution of feature maps to restore them to the size of the original input, where employing an effective decoder can enhance the accuracy of predictions. In our work, we introduce a Lightweight Attention Decoder (LAD), as illustrated in Fig. 5. The entire structure adopts a dual-branch cross-channel model, with the core module being the Lightweight Bottleneck Unit (LBU), Height-Direction Attention (HA) and Width-Direction Attention (WA). Leveraging these two attention modules, we can generate co-attention affinity matrices from both the height and width dimensions, respectively, to facilitate feature aggregation within the decoder. Through extensive experimentation, it has been found that utilizing three consecutive LBU modules can achieve superior performance. Features processed through three LBU operations are extracted for use in the next stage of the decoder.

5) Loss Function: This paper employs multi-scale (MS) supervised training, utilizing a hybrid loss function to train the network. The total loss function L_{total} is given as follows:

$$L_{\text{total}} = \alpha * L_{\text{Dice}}\left(y_{c,i}, p_{c,i}\right) + \beta L_{\text{SCE}}\left(y_{c,i}, p_{c,i}\right). \tag{1}$$

where $y_{c,i}$ represents the ground truth, $p_{c,i}$ represents the probability that the pixel i belongs to class c, we set $\alpha = \beta = 1$, without fine tuning. L_{Dice} represents DiceLoss function.

$$L_{\text{Dice}} = 1 - \frac{2\sum_{i=1}^{N} y_{c,i} p_{c,i}}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} y_i^2}. \tag{2}$$

L_{SCE} represents SoftCrossEntropy loss function.

$$L_{\text{SCE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \log\left(p_{i,c}\right). \tag{3}$$

where N denotes the total number of pixels, C denotes the total number of classes.

B. 3D Segmentation Mapping Module

1) Multi-class Bayesian Updating Method: Occupancy grid mapping distinguishes between occupied, free, and unknown spaces using a method that employs log-odds of occupancy probabilities. By leveraging sensor measurements to infer hit and miss rates for voxel nodes, it updates the occupancy rate of voxel nodes in a probabilistic manner. At time t, all point clouds detected in the space by sensor ray Λ possess the same number of semantic classes C, with the pose

Algorithm 1 Multi class Bayesian Semantic Fusion

Input: Global map $\mathbf{G}$, current local map $\mathbf{C}$, node n, node class c, semantic set $\mathbb{N}$, class confidence $\zeta_i\,(i=0,1,2)$, other class confidence ξ, punishment value θ

Output: Updated global map $\mathbf{U}$

For n in $\mathbf{C}$ **do**

  If n exists **in** $\mathbf{G}$ **then**

  Function $\mathbb{N}_{\mathbf{U}} =$ **SEMANTICFUSION** $(\mathbb{N}_{\mathbf{G}}, \mathbb{N}_{\mathbf{C}})$

  $\xi_{\mathbf{G}} = 1 - \text{SUM}\left(\zeta_{\mathbf{G}}\right)$

  $\xi_{\mathbf{C}} = 1 - \text{SUM}\left(\zeta_{\mathbf{C}}\right)$

  $\mathbb{Z} =$ **UNIQUECLASS** $(\mathbb{N}_{\mathbf{G}}, \mathbb{N}_{\mathbf{C}})$

  $\Delta_{\mathbf{G}} = \xi_{\mathbf{G}} - \theta \log\left(1 + \text{Size}\left(\mathbb{Z}\right)\right)$

  $\Delta_{\mathbf{C}} = \xi_{\mathbf{C}} - \theta \log\left(1 + \text{Size}\left(\mathbb{Z}\right)\right)$

  For $z \in \mathbb{Z}$ do

    If $z \in \mathbb{N}_{\mathbf{G}}$ **and** $z \notin \mathbb{N}_{\mathbf{C}}$ **then**

      Add z **into** $\mathbb{N}_{\mathbf{U}}$, $\zeta_{\mathbf{U}}(z) = \dfrac{\left(\Delta_{\mathbf{G}} + \zeta_{\mathbf{G}}(z)\right)}{2}$

    If $z \in \mathbb{N}_{\mathbf{C}}$ **and** $z \notin \mathbb{N}_{\mathbf{G}}$ **then**

      Add z **into** $\mathbb{N}_{\mathbf{U}}$, $\zeta_{\mathbf{U}}(z) = \dfrac{\left(\Delta_{\mathbf{C}} + \zeta_{\mathbf{C}}(z)\right)}{2}$

    Else

      Add z **into** $\mathbb{N}_{\mathbf{U}}$, $\zeta_{\mathbf{U}}(z) = \dfrac{\left(\zeta_{\mathbf{G}}(z) + \zeta_{\mathbf{C}}(z)\right)}{2}$

Descending sort on $\mathbb{N}_{\mathbf{U}}$ with respect to $\zeta_{\mathbf{U}}(z)$

$\Delta_{\mathbf{U}} = \exp\left(\dfrac{\Delta_{\mathbf{G}} + \Delta_{\mathbf{C}}}{2}\right)$

For $i > 3$ do

  $\Delta_{\mathbf{U}}$ += $\exp\left(\zeta_{\mathbf{U}}(z)[i].\text{logodds}\right)$

$\zeta_{\mathbf{U}}(z)[3:\text{end}].\text{Remove}$

$\xi_{\mathbf{U}}(z) = \log\left(\Delta_{\mathbf{U}}\right)$

Update $\mathbf{G}$ with $\mathbf{C}$ to get the updated global map $\mathbf{U}$

$\mathbf{X}$ of robot, and the observation for voxel node m_i located at position $\mathbf{p}$ and orientation $\mathbf{R}$ is $\mathbf{z} = (r, c)$, where r denotes the sensor's observation range and c denotes the class of the node. The state of the node can be represented by the probability density function $p_t\left(m_i \mid z; \mathbf{X}, \Lambda\right)$. For a newly detected point cloud, it primarily contains two states: occupied (hit) and free (missed), utilizing this information, a new probability density function $p_{t+1}\left(m_i \mid z; \mathbf{X}, \Lambda\right)$ can be updated. Therefore, for the probability density function $P_t(\mathbf{m})$ of a global map of size S, we can employ a multinomial logit model to represent it:

$$P_t(\mathbf{m}) = \prod_{i=1}^{S} p_t\left(m_i\right). \tag{4}$$

According to the description of the multi-class bayesian updating method in [13], here, a log-odds vector $\mathbf{h}_{t,i}$ is used to represent the probability density function $p_t(m_i)$ for a single node m_i with semantic class c:

$$\mathbf{h}_{t,i} \coloneqq \left[\log \frac{p_t(m_i=0)}{p_t(m_i=0)} \cdots \log \frac{p_t(m_i=C)}{p_t(m_i=0)} \right]^{\mathrm{T}} \in \mathbb{R}^{C+1}. \quad (5)$$

After using the log-odds vector $\mathbf{h}_{t,i}$, the probability density function of the single voxel node m_i for class c can be obtained using the Softmax function:

$$p_t(m_i=c) = \sigma_{c+1}(\mathbf{h}_{t,i}) \coloneqq \frac{\mathbf{e}_{c+1}^{\mathrm{T}} \exp(\mathbf{h}_{t,i})}{\mathbf{1}^{\mathrm{T}} \exp(\mathbf{h}_{t,i})}. \quad (6)$$

where $\mathbf{e}_c$ is the standard basis vector, where all elements are equal to 0 or 1, $\mathbf{1}$ is a vector where all elements are equal to 1. For a given set of observations Z_{t+1}, the prior $\mathbf{h}_{t+1,i}$ can be calculated from $\mathbf{h}_{t,i}$ and $\mathbf{h}_{0,i}$, and we can derive from (4) and (5):

$$\mathbf{h}_{t+1,i} = \mathbf{h}_{t,i} + \sum_{\mathbf{z} \in Z_{t+1}} (\mathbf{l}_i(\mathbf{z}) - \mathbf{h}_{0,i}). \quad (7)$$

where $\mathbf{l}_i(z)$ is the log-odds vector of the probability density function:

$$\mathbf{l}_i(\mathbf{z}) \coloneqq \left[\log \frac{p_t(m_i=0 \mid \mathbf{z})}{p_t(m_i=0 \mid \mathbf{z})} \cdots \log \frac{p_t(m_i=C \mid \mathbf{z})}{p_t(m_i=0 \mid \mathbf{z})} \right]^{\mathrm{T}}. \quad (8)$$

Based on the occupancy rules of voxel nodes, we parameterize $\mathbf{l}_i(z)$ for modeling, implementing the Bayesian multi-category mapping described in [13]:

$$\mathbf{l}_i(\mathbf{z}) = \mathbf{l}_i((r,c)) \coloneqq \begin{cases} \phi^+ + \mathbf{E}_{c+1}\varphi^+, & m_i \text{ is occupied} \\ \phi^-, & m_i \text{ is free} \\ \mathbf{h}_{0,i}, & \text{otherwise} \end{cases}. \quad (9)$$

where $\mathbf{E}_c = \mathbf{e}_k \mathbf{e}_k^{\mathrm{T}}$, ϕ^+, ϕ^-, φ^+ are parameter vectors. Hence, we can update the map's probability density function through (8):

$$p_t(m_i=c \mid \mathbf{z}) = \sigma_{c+1}(\mathbf{l}_i(\mathbf{z})). \quad (10)$$

By introducing the bayesian multi-class update method, which utilizes log-odds piecewise constants, we reduce the uncertainty of voxel node occupancy rates, thereby decreasing the memory consumption for mobile robots mapping unknown spaces. To further enhance the performance of mapping, our experiments maintain only the probabilities of the three most likely semantic classes for each voxel node, while the probabilities of all other classes are consolidated into an others class.

2) Multi-class Bayesian Fusion Method: To compress and prune the occupancy map, we fuse information from two voxel nodes, which includes semantic information predicted by FEAGNet and the corresponding probabilities, namely confidence levels. By introducing a punishment value θ, we adjust the confidence levels of the others class for both nodes, ensuring the three highest probability semantic labels retained in the merged node represent classes shared by the two nodes.

TABLE I: FEAGNET COMPARED TO SOTA NETWORKS ON NYUV2 AND SUN RGBD. NOTE: R DENOTES RESNET, △ DENOTES THE MODIFIED RESNET. THE BEST RESULT IN EACH METRIC IS IN BOLD.

Method	MIoU		Params (M)	FLOPs (G)	FPS
	NYUv2	SUN RGBD			
3DGNN-R101$_{17}$	43.1	45.9	47.3	-	4.8
RDFNet-R101$_{17}$	49.1	-	443.8	101.8	7.9†
RefineNet-R101$_{17}$	44.7	45.7	118.1	309.2	13.6†
DCNN-R152$_{18}$	48.4	-	92.0	-	12.5
ACNet-R50$_{19}$	48.3	48.1	116.6	126.4	18.9†
SAGate-R101$_{20}$	49.1	-	110.9	176.8	11.9†
ESANet-R34△$_{21}$	50.3	48.2	46.9	45.1	13.4†
CANet-R50$_{22}$	50.0	48.1	87.1	122.4	18.1
SGACNet-R34△$_{23}$	49.4	47.8	35.7	37.6	15.9†
PGDENet-R34$_{23}$	**53.7**	**51.0**	100.7	178.8	12.2†
FEAGNet-V1	46.9	45.2	4.5	1.9	39.2
FEAGNet-V1*	46.9	45.2	4.5	**1.9**	**54.8**
FEAGNet-V2	47.6	46.4	4.5	7.8	37.7
FEAGNet-V2*	47.6	46.4	**4.5**	7.8	52.5

* Speed measured on Jetson Agx Xavier with TensorRT acceleration.

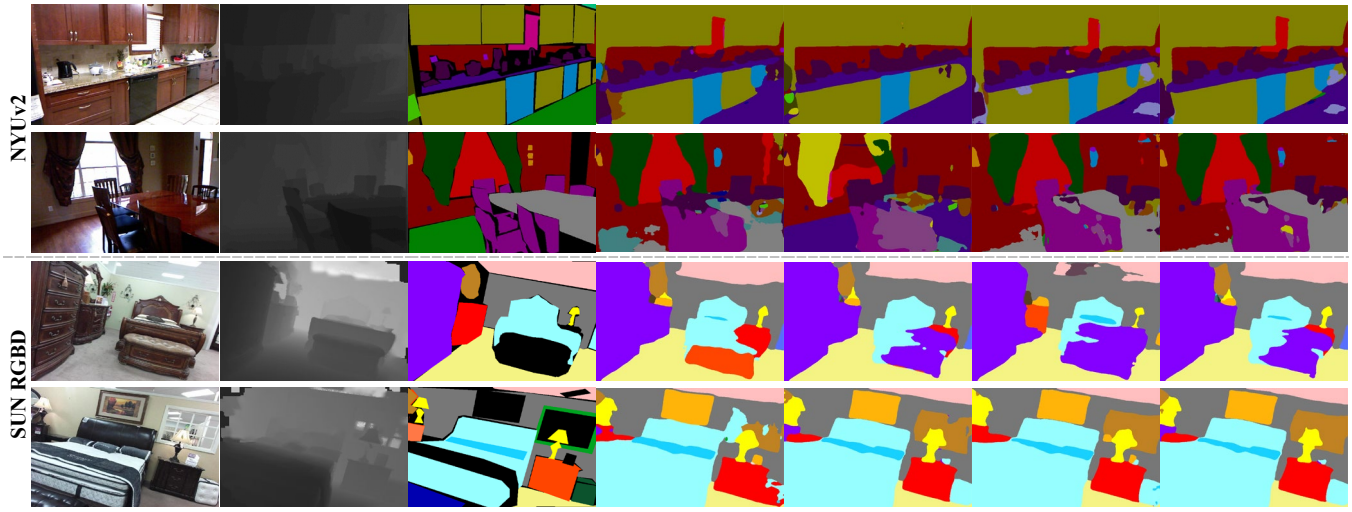
† Testing on our experiment.

TABLE II: FEAGNET COMPARED TO SOTA NETWORKS ON CITYSCAPES FOR BOTH COMMON RESOLUTIONS. NOTE: R DENOTES RESNET, △ DENOTES THE MODIFIED RESNET. THE BEST RESULT IN EACH METRIC IS IN BOLD.

	Method	1024×512 (val)		2048×1024 (val)		Params (M)
		MIoU	FPS	MIoU	FPS	
RGB	DABNet$_{19}$	-	104.2	69.1	27.7	**0.8**
	SwiftNet$_{19}$	70.2	**134.9**	75.4	39.9	11.8
	LDN$_{21}$	-	-	79.0	13.6	10.3
	DDRNet-S$_{22}$	-	108†	79.3	**60.6**†	5.7
	PIDNet-S$_{23}$	-	61.2†	79.9	57.5†	7.6
	FEAGNet-S	70.4	43.5	76.4	36.7	2.8
	FEAGNet-S*	70.4	56.5	76.4	47.7	2.8
	FEAGNet-L	71.3	39.6	77.3	22.0	2.8
	FEAGNet-L*	71.3	50.6	77.3	31.3	2.8
RGB-D	SSMA$_{19}$	-	-	**82.1**	-	56.4
	LDFNet$_{19}$	68.5	-	-	-	2.3
	ESANet-R34△$_{21}$	**75.2**	17.6†	80.1	7.4†	46.4
	SGACNet-R34△$_{23}$	74.1	13.7†	79.7	5.8†	35.7
	FEAGNet-V1	72.9	36.9	78.2	21.2	4.5
	FEAGNet-V1*	72.9	46.4	78.2	25.4	4.5
	FEAGNet-V2	73.6	29.7	79.1	16.4	4.5
	FEAGNet-V2*	73.6	36.4	79.1	18.1	4.5

* Speed measured on Jetson Agx Xavier with TensorRT acceleration.

† Testing on our experiment.



(a) RGB (b) Depth (c) GT (d) ACNet (e) ESANet-R34△ (f) FEAGNet-V1 (e) FEAGNet-V2

Fig. 6: Qualitative comparison results of semantic segmentation on the NYUv2 and SUN RGBD dataset.

TABLE III: ABLATION STUDY OF OUR PROPOSED MODULES ON NYUV2 DATASET. 'BASEL.' DENOTES THE BASELINE. 'WA&HA' DENOTES THE USE OF BOTH WA AND HA IN THE DECODER. 'ML' DENOTES MULTI-SCALE SUPERVISION.

Basel.	CFIM	DAPPF	WA&HA	ML	MIoU	Params (M)	FLOPs (G)
√					44.35	3.44	7.11
√	√				45.78	3.91	7.11
√		√			45.82	3.70	7.15
√	√	√			46.84	4.17	7.15
√	√	√	√		47.12	4.45	7.85
√	√	√	√	√	47.64	4.45	7.85

Here, we set θ to 0.6, and Alg 1 provides the pseudocode for this process. Additionally, a node will be pruned when all eight of its sub-nodes have identical confidence levels. With this semantic fusion strategy, EVSMap not only efficiently completes mapping but also closely aligns with the real-world semantic information.

IV. Experiments

A. Datasets and Metrics

To evaluate the performance of our proposed FEAGNet, we conducted experiments on three different datasets, namely, the indoor datasets NYUv2 and SUN RGBD, and the outdoor dataset Cityscapes.

1) Indoor datasets: The NYUv2 dataset consists of 1,449 RGB-D images, for which we adopted the standard train/test split, with 795 images for training and 654 for testing, and we utilized the common set of 40 class labels. The SUN RGBD dataset includes 37 class labels and 10,335 RGB-D images, with a standard train/test split of 5,285 images for training and 5,050 for testing.

2) Outdoor datasets: The Cityscapes dataset comprises 5,000 high-resolution images of 1024 × 2048 pixels, with 2,975 images for training, 500 for validation, and 1,525 for testing. The dataset includes 19 categories.

3) Evaluation Metrics: In our experiments, we used mean intersection over union (MIoU), frames per second (FPS), the parameters of the network (Params), and computational complexity (FLOPs) as our evaluation metrics.

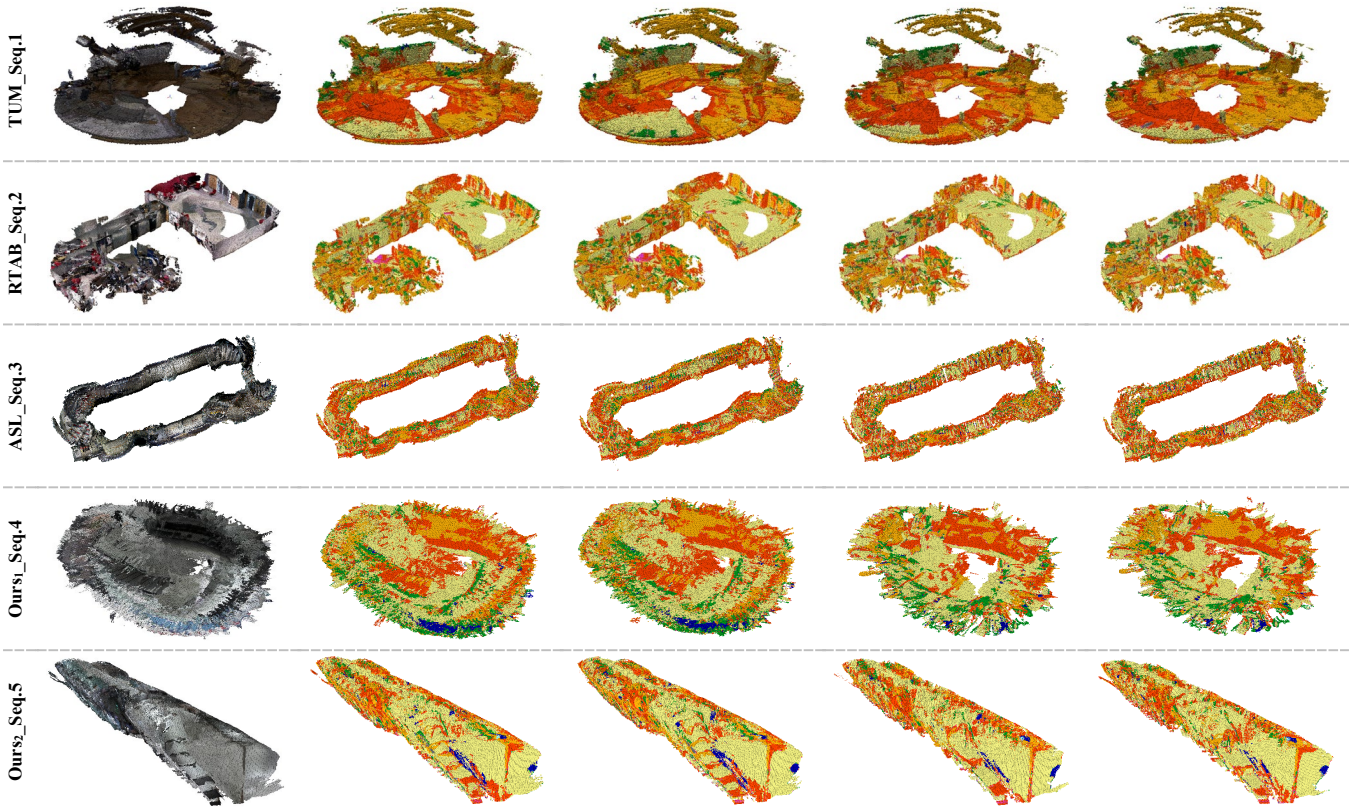
B. Implementation Details

Our experiments were conducted using PyTorch 1.2, with all models subjected to identical experimental configurations. Training was carried out over 500 epochs with a batch size of 8. Stochastic Gradient Descent (SGD) with a momentum value of 0.9 was employed as our optimizer. Weight decay and the initial learning rate were set to 0.0005 and 0.005, respectively. Upon reaching 150 epochs, we employ a polynomial learning rate strategy $\mathrm{lr} = \mathrm{initial\_lr} \times \left(1 - \frac{\mathrm{ep}}{\mathrm{max\_ep}}\right)^{0.9}$, to adjust the learning rate. For data augmentation, methods such as random scaling, cropping, and flipping were utilized. Furthermore, unless specifically stated otherwise, all encoder blocks in the experiments were pre-trained on the ImageNet dataset [53].

C. Results on 2D Semantic Segmentation

1) Comparison Results on NYUv2 &SUN RGBD: Table I lists our experimental results on two indoor datasets. Compared with SOTA models, although our MIoU did not achieve the best results, in terms of the number of parameters, both versions of our FEAGNet reached 4.5M, with computational costs of 1.9 GFLOPs and 7.8 GFLOPs, respectively, achieving a significant reduction compared to other networks, while also meeting the requirements for real-time performance. As shown in Fig. 6, the segmentation results of our network were on par with those of ACNet and ESANet, which have higher MIoU than ours. These minor differences are acceptable considering the substantial improvement in network processing speed for complex mapping tasks.

2) Comparison Results on Cityscapes: To further test the performance of FEAGNet, we also achieved commendable results on the outdoor dataset Cityscapes, the results of the experiment are shown in Table II. With only RGB input, our



(a) ORGBMap (b) TESAMap (c) TFEAGMap (d) MESAMap (e) EVSMap

Fig. 7: Qualitative comparison results of 3D semantic mapping on the *TUM*, *RTABMap*, *ASL* and *our* sequences. *ORGBMap* refers to mapping using the original OctoMap approach, with voxel grids being assigned RGB information. *TE*SAMap and *MESAMap* utilize ESANet-R34$^{\triangle}$ as the semantic segmentation network, employing the traditional voxel node update strategy from [14] and our update strategy for mapping, respectively. *TFEAGMap* and *EVSMap* build upon the foundation of *TESAMap* and *MESAMap* by replacing the semantic segmentation network with FEAGNet, while all other aspects remain consistent.

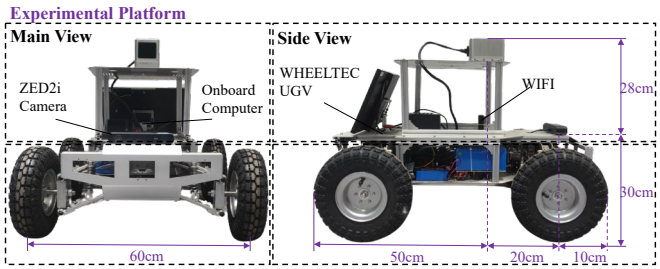


Fig. 8: Our experimental platform, equipped with a ZED2i Camera.

TABLE IV: ANALYSIS OF THE MAPPING EFFICIENCY OF EVSMAP COMPARED TO OTHER METHODS ON DIFFERENT SEQUENCES.

Method	Runtime (Hz)	Map Size (MB)				
		Seq.1	Seq.2	Seq.3	Seq.4	Seq.5
ORGBMap	1.5	3.54	24.96	10.37	16.12	40.32
TESAMap	2.1	5.43	24.62	13.82	16.75	38.23
TFEAGMap	5.4	5.71	24.67	13.74	16.73	38.02
MESAMap	10.9	5.89	15.39	10.11	5.99	14.85
MVSMap	14.4	5.30	15.34	9.99	5.95	15.04

Model size is 2.8M, and at an input resolution of 2048×1024 , our MIoU can reach 77.3. When integrating depth data, our network can achieve results comparable to the currently best-performing PIDNet, with our model size being only 4.5M. This also demonstrates that incorporating depth data can enhance the network's segmentation performance.

D. Ablation Study

To demonstrate the effectiveness of the modules we proposed, we conducted ablation experiments on the NYUv2 dataset using FEAGNet, which only consists of an Encoder block and a Decoder block (without HA and WA), as our baseline. This was to investigate the effectiveness of each module and the multi-scale supervision strategy within a fixed structure. Our experimental data, as shown in Table III, indicate that, as expected, the inclusion of our proposed CFIM, DAPPF, and WA&HA modules led to improvements in MIoU. Furthermore, when the network was trained using the multi-scale supervision strategy, the network's Miou experienced further enhancement. Therefore, it can be concluded that our approach is effective.

E. Results on 3D Semantic Mapping

As shown in Fig. 7, we utilize the proposed EVSMap and compare it with several other systems for semantic mapping in indoor scenes on TUM, RTABMap, ASL and our own sequences. Compared to other systems, our system not only maintains equivalent capabilities in semantic segmentation and information extraction but also compresses and prunes the results of semantic mapping to reduce map storage consu-

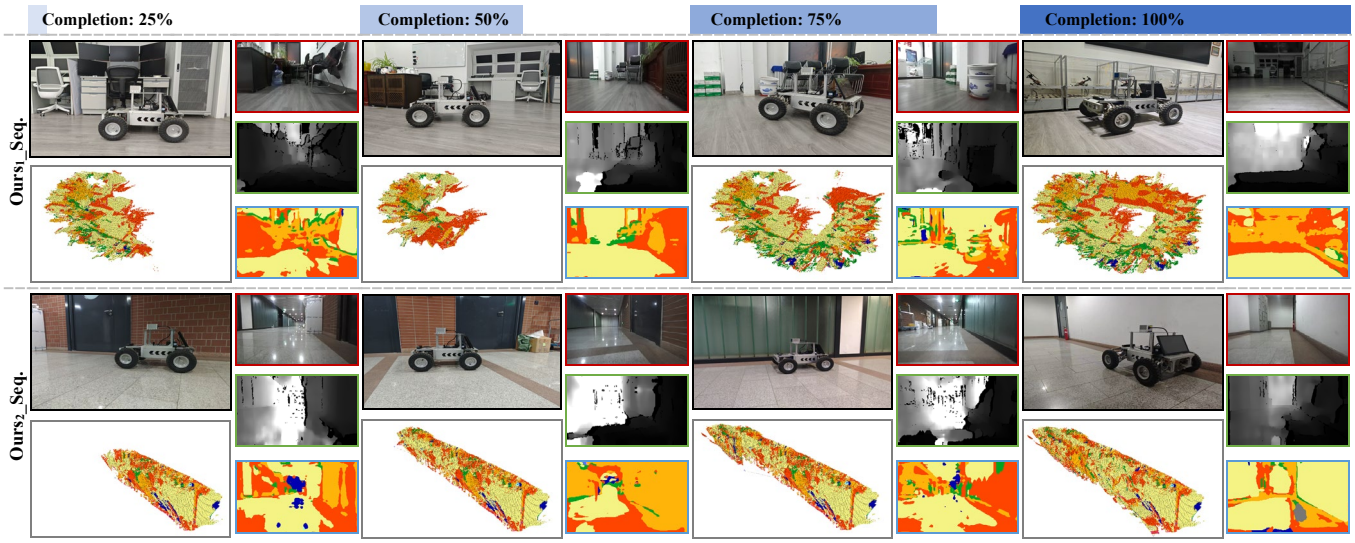


Fig. 9: EVSMap in real scenarios. we tasked a mobile robot with exploring its surroundings and performing semantic mapping in both small-scale bright scenes *Ours1_Seq.* and large-scale dim scenes *Ours2_Seq.*, respectively, to test the robustness of our system.

mption. As seen in Table IV, our EVSMap achieves the fastest runtime. Furthermore, in terms of map storage size for sequence mapping, our EVSMap stores maps in the smallest memory across five sequences. Due to the cross-utilization of FEAGNet and multi-class bayesian fusion strategy in several other approaches, it is observable that they can optimize the mapping results of other systems, which also substantiates the efficiency of the mapping method we proposed.

To validate the feasibility of EVSMap in real-world scenarios, we chose the mobile robot shown in Fig. 8 as our experimental platform, equipped with a Jetson Agx Xavier (Jetpack 4.5, TensorRT 7.1) embedded platform and a ZED 2i RGBD camera. Semantic mapping was performed in both a small-scale bright scene and a large-scale dim scene, to test the robustness of our system. The mapping process, as depicted in Fig. 9, shows that EVSMap can efficiently map the surrounding environment in both scenarios, guiding the work of the mobile robot.

V. Conclusion

In this paper, we propose a novel end-to-end architecture for efficient and real-time volumetric-semantic mapping, named EVSMap. It primarily utilizes our proposed lightweight and robust RGB-D segmentation network FEAGNet to extract semantic probability information from voxel space nodes. The output results are continuously updated within the voxel space using an improved multi-category Bayesian fusion strategy, which not only enhances the speed of mapping but also reduces the map's storage size, thereby achieving efficient mapping. Experiments conducted on a mobile robot platform have demonstrated the feasibility of our EVSMap.

References

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. on Comput. Vis. Patte rn Recognit. (CVPR)*, 2015, pp. 3431-3440.

[2] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[3] L.-C. Chen, et al., "Rethinking Atrous Convolution for Semantic Image Segmentation," *arXiv preprint arXiv:1706.05587, 2017*.

[4] C. Hazirbas, L. N. Ma, C. Domokos, and D. Cremers, "FuseNet: incorporating depth into semantic segmentation via fusion-based CNN architecture," in *Proc. of the 13th Asian Conf. on Comput. Vis.*, Taipei, China: Springer, 2017, pp. 213-228.

[5] S. W. Hung, S. Y. Lo, and H. M. Hang, "Incorporating luminance, depth, and color information by a fusion-based network for semantic segmentation," in *Proc. of 2019 IEEE Int. Conf. on Image Process*, Taipei, China: IEEE, 2019, pp. 2374-2378.

[6] J. Jiang, L. Zheng, F. Luo, and Z. Zhang, "RedNet: Residual encoder-decoder network for indoor RGB-D semantic segmentation," *arXiv preprint arXiv:1806.01054, 2018*.

[7] D. Seichter et al., "Efficient rgb-d semantic segmentation for indoor scene analysis," in *2021 IEEE Int. Conf. on Rob. and Autom (ICRA)*, IEEE, 2021.

[8] Y. Zhang et al., "Spatial-information Guided Adaptive Context-aware Network for Efficient RGB-D Semantic Segmentation," *IEEE Sens. J.*, 2023.

[9] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, "PanopticFusion: Online volumetric semantic mapping at the level of stuff and things," *arXiv preprint arXiv:1903.01177, 2019.*

[10] J. Huang, S. Yang, T.-J. Mu, and S.-M. Hu, "ClusterVO: Clustering moving instances and estimating visual odometry for self and surroundings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2165–2174.

[11] P. Chao, C.-Y. Kao, Y. Ruan, C.-H. Huang, and Y.-L. Lin, "HarDNet: A low memory traffic network," in *Proceedings of the IEEE/CVF Int. Conf. on Comput. Vis. (ICCV)*, Oct. 2019, pp. 3552-3561.

[12] Y. Wang et al., "Spatial-Assistant Encoder-Decoder Network for Real Time Semantic Segmentation," *arXiv preprint arXiv:2309.10519, 2013*.

[13] A. Asgharivaskasi and N. Atanasov, "Semantic OcTree Mapping and Shannon Mutual Information Computation for Robot Exploration," in *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1910-1928, June 2023.

[14] T. Ran, L. Yuan, J. Zhang, D. Tang, and L. He, "RS-SLAM: A robust semantic SLAM in dynamic environments based on RGB-D sensor," *IEEE Sens. J.*, vol. 21, no. 18, pp. 20657-20664, 2021.