A NEW REAL-TIME 3D DENSE SEMANTIC MAPPING SYSTEM FOR LARGE-SCALE ENVIRONMENTS

Zhiwei Xing,* Xiaorui Zhu,* and Yudong Wu*

Abstract

In this paper, a novel real-time 3D dense semantic mapping system, SemanticSurfel, is proposed to integrate semantic segmentation results, poses, depth graphs into the constructed map, and to scale well in large-scale environments. First, a lightweight semantic segmentation network, HybridNet, is designed with efficient Hybrid Basic Blocks and Hybrid Dilated Blocks in the encoder and Attention Pyramid Module in the decoder to accurately and efficiently segment the input image at pixel level. Then, super-pixels extracted from semantic, depth, and intensity graphs are used to construct surfels to build the 3D dense semantic map according to the pose graph of a sparse SLAM system. Extensive experiments were carried out to evaluate the performance of HybridNet and SemanticSurfel. Experimental results demonstrate that HybridNet achieves a good balance between accuracy and hybrid efficiency, and the SemanticSurfel system achieves great accuracy and scales well in large-scale environments.

Key Words

Semantic segmentation, lightweight network, super-pixel extraction, semantic mapping

1. Introduction

The inclusion of semantic information within a dense map is essential for mobile robots to work autonomously in the surrounding world. Simultaneous localisation and mapping (SLAM) algorithm is an important technology to estimate the robot poses and map representation of the environment. However, the maps obtained by most SLAM systems contain only the geometric information, which is susceptible to changes of illumination and scene appearance of the environment. Besides, these maps are not suitable for autonomous tasks other than localisation, such as autonomous exploration in an unknown environment, human activity recognition [1], or scene understanding [2]. To address these issues, robots need to be able to understand the environment more intelligently, *i.e.*, at the semantic level. This can be achieved by constructing a 3D dense semantic map of the environment with sufficient semantic and geometric information. In order to be deployed on mobile robots with limited computing and memory resources, especially in large-scale environments, the mapping system needs to be highly efficient in computing time and memory footprint.

The first step of the semantic mapping system is to extract the semantic information from the environment. Traditional image segmentation methods, such as Texton Forest [3] and Random Forest [4], relied on the brightness, colour, and texture of the image, and they achieved poor performance on complex images and could not get pixelwise segmentation. More recently, deep learning-based methods have made significant progress. He *et al.* [5] proposed Mask-RCNN to get bounding box of objects in the scene and then perform pixel-level semantic labelling. It achieved good segmentation accuracy but required very high computing resources. Aimed at obtaining the best accuracy under a limited computational budget, many works have been dedicated to design lightweight networks. Badrinarayanan *et al.* [6] proposed to use an encoder–decoder network architecture in SegNet, and Paszke *et al.* [7] proposed ENet which uses dilated convolutions as the main convolution unit to further improve computational efficiency. Zhao *et al.* [8] incorporated multi-resolution branches under proper label guidance in ICNet to reduce the amount of computation. Romera *et al.* [9] proposed to use residual connections and factorised convolutions in ERFNet to improve network accuracy and efficiency. In the same year, Zhang *et al.* [10] proposed ShuffleNet, which improved the accuracy and efficiency of the network using channel shuffle and split. Besides computational efficiency, many researches focused on reducing network parameter size to save memory footprint. Wang *et al.* [11] proposed an attention pyramid network in LEDNet which had a lighter network structure and smaller network parameter size. Wu *et al.* [12] developed CGNet with several context-guided blocks that could learn joint features from local features and surrounding contexts and then improved the joint feature with global context, thus reduced network

* Harbin Institute of Technology, Shenzhen, Guangdong, China; e-mail: williamxing@foxmail.com; zhuxiaorui@hotmail.com; don.wu@dadaoii.com
Corresponding author: Xiaorui Zhu

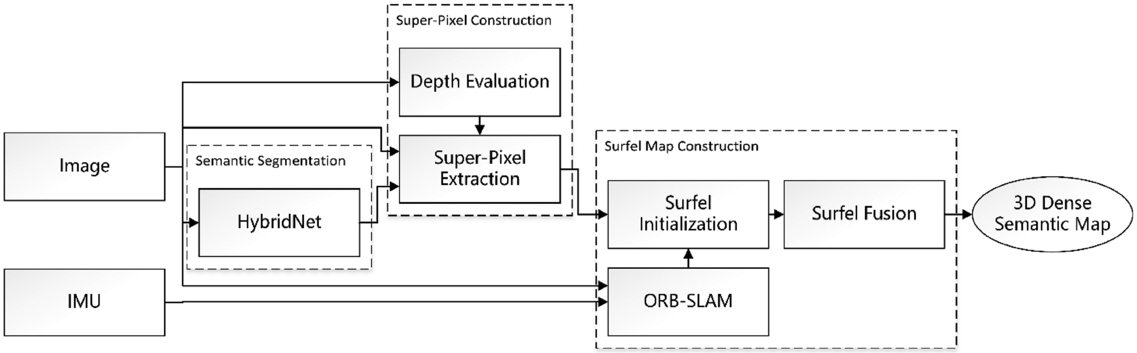

Figure 1. SemanticSurfel system framework.

parameter size. However, these above networks cannot achieve good results in improving segmentation accuracy, improving computational efficiency, or saving memory footprint in one network.

The second step of the semantic mapping system is to integrate the 2D semantic segmentation result into the SLAM system to get 3D dense semantic map of the environment. Salas-Moreno *et al.* [13] proposed SLAM++, which mapped indoor scenes at the level of semantically defined objects. However, this method was limited to mapping objects in a pre-defined database. Kundu *et al.* [14] proposed a conditional random field (CRF) model that jointly inferred the semantic category and occupancy for each voxel, and produced a 3D volumetric semantic map. Grinvald *et al.* [15] proposed an approach to incrementally build volumetric object-aware maps. It segmented the input frame with an unsupervised geometric approach in combination with an instance-aware semantic prediction to detect objects and fused information about their 3D shape, location, and semantic class into a global volume. However, these volumetric maps required powerful computing resources. Surfel-based or mesh-based methods were more efficient as these methods did not store all the voxels of objects in the environment but only the surfaces. Stückler and Behnke [16] proposed random decision forests to obtain pixelwise semantic predictions of each incoming frame and then fuse the segmentation results into a surfel map by Bayesian estimation. In order to further improve the computational efficiency, McCormac *et al.* [17] proposed SemanticFusion, which used a convolutional neural network (CNN) to obtain pixelwise semantic segmentation result and integrated them into the surfel map constructed by ElasticFusion through Bayesian estimation. But the network parameter size was too large. And Dung and Capi [18] proposed to use depth image camera to do semantic mapping, however, depth camera has a small range and is only suitable for small-scale scenes. Hence these methods were not suitable for mobile robots in large-scale environments.

Therefore, the main contributions of this paper are:

- A lightweight semantic segmentation network, HybridNet, is proposed for segmenting the scene at pixel level. In HybridNet, an asymmetric encoder–decoder network structure is designed with efficient Hybrid Basic Blocks and Hybrid Dilated Blocks in the encoder and Attention Pyramid Module in the decoder to achieve a good balance in the segmentation accuracy, computational efficiency, and memory footprint.
- A novel SemanticSurfel system integrating HybridNet and super-pixel-based surfel map construction is proposed to construct 3D dense semantic map of large-scale environments in real time. In the SemanticSurfel system, super-pixels are extracted from semantic, depth, and intensity graphs to model surfels. Then these surfels are constructed into the map according to the pose graph of a sparse SLAM system.

This paper is organised as follows. Section 1 introduces the background and related works. Section 2 provides an overview of the SemanticSurfel system framework. Section 3 introduces the lightweight semantic segmentation network HybridNet. Super-pixel construction and surfel map construction are described in detail in Sections 4 and 5, respectively. Experiments and discussion are presented in Section 6. Finally, a conclusion is drawn.

2. System Framework

The framework of the proposed SemanticSurfel system is shown in Fig. 1 with three stages, such as semantic segmentation, super-pixel construction, and surfel map construction.

In this framework, the input is each image and IMU frame at the current time period. During semantic segmentation stage, the proposed HybridNet assigns semantic label to each pixel of the input image frame. In the super-pixel construction stage, the depth graph is evaluated by the depth evaluation network, and the super-pixels are extracted by an augmented SLIC algorithm using semantic, depth, and intensity information comprehensively. Finally, in the surfel map construction stage, ORB-SLAM [19] is used as the localisation system to provide pose graph of the camera. With the poses of the camera, the 2D super-pixels are constructed into the surfels in 3D space. And as the camera moves, all the surfels will be fused together to obtain the 3D dense semantic map of the environment.

3. Semantic Segmentation: HybridNet

In this section, a lightweight semantic segmentation network, HybridNet, is proposed with an asymmetric

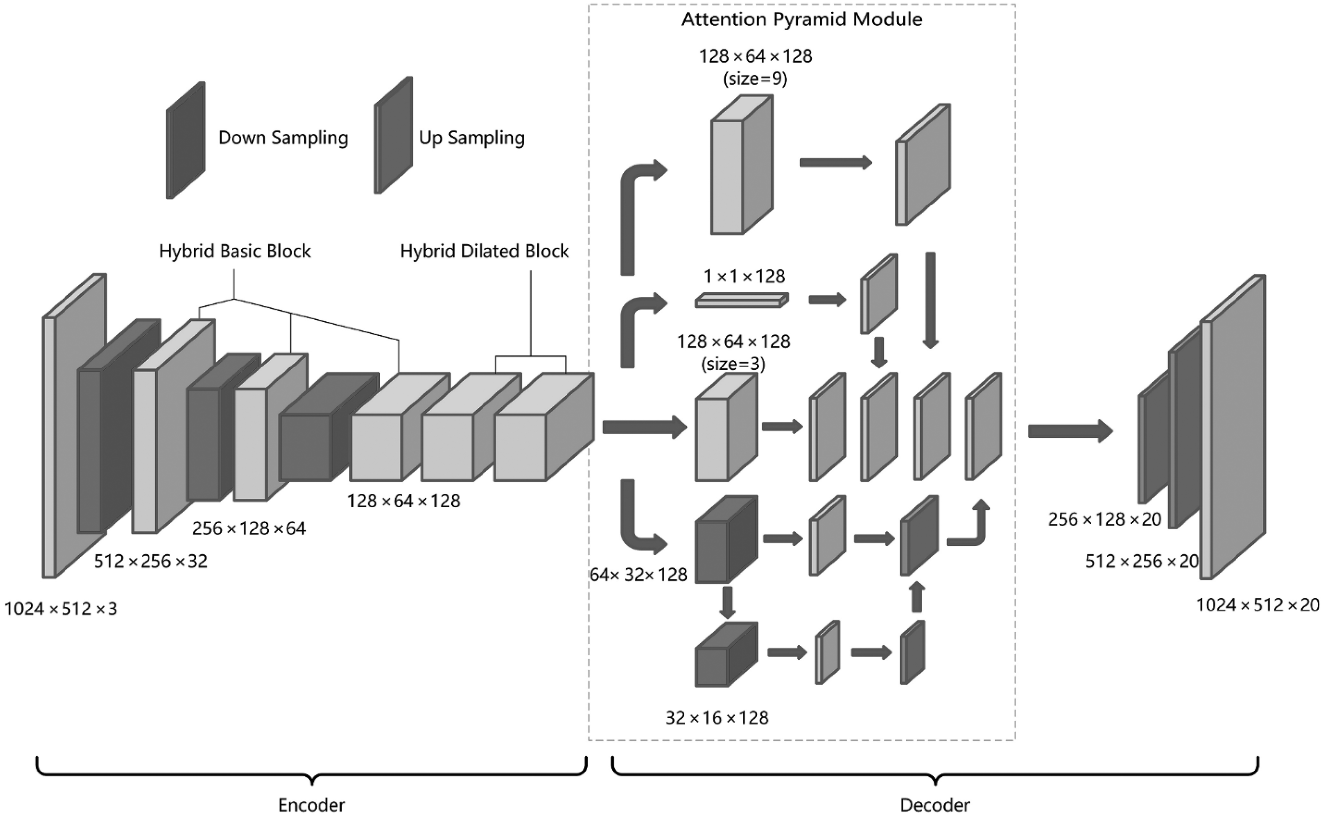

Figure 2. HybridNet network structure.

encoder–decoder network structure in Fig. 2. The encoder extracts feature maps of different levels from the input image. Two complementary convolution blocks, Hybrid Basic Block and Hybrid Dilated Block, are designed in the encoder to guarantee the continuity of the extracted features and large receptive field of the kernel. Hybrid Basic Block can extract features continuously from the image, but its receptive field is limited in size. On the contrary, Hybrid Dilated Block has a larger receptive field but the kernel is discontinuous. In the subsequent decoder, an Attention Pyramid Module is designed to capture the context information at different scales without introducing additional computational requirements. Then the extracted feature maps are upsampled to ensure the input and output have the same resolution. Table 1 is a detailed description of the network structure and parameters of each layer.

In the encoder, Downsampling, Hybrid Basic Blocks, and Hybrid Dilated Blocks are used in combination to extract features from the input image. From layer 1 to layer 8 of the network, the resolution of images at each layer is relatively large. Therefore, Hybrid Basic Block with a small kernel is adopted to perform convolution operation, which reduces the requirements of computing resources and is conducive to extracting the basic features of images.

Hybrid Basic Block: As shown in Fig. 3, three techniques, such as channel split and shuffle, spatial separable convolution, and residual connection, are adopted in Hybrid Basic Block. The input feature channels are splitted into two equally sized groups at the beginning of each block. By this split operation, each group has half number of channels, hence the computational cost of subsequent convolution operation is affordable under a limited computation budget. At each group, spatial separable convolution is used to extract features.

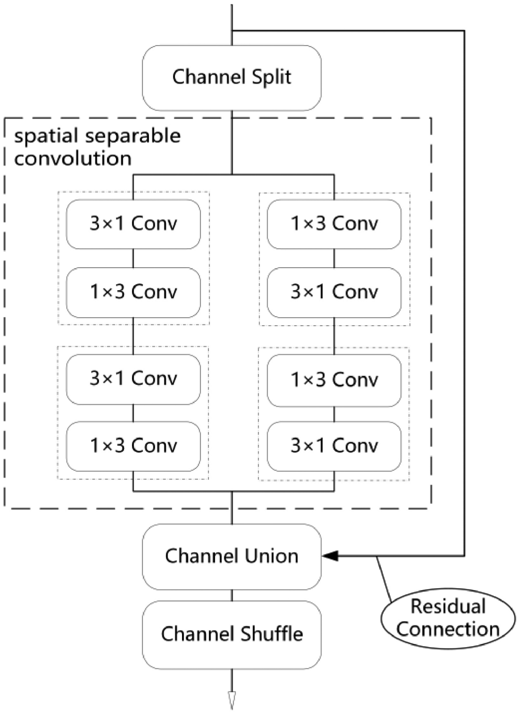

Figure 3. Hybrid Basic Block.

Table 1
Hybrid Network Structure Description

	Layer	Module	Channel	Output Size
Encoder	1	Down sampling	32	512 × 256
	2–3	Hybrid basic block	64	256 × 128
	4	Down sampling	64	256 × 128
	5–6	Hybrid basic block	128	128 × 64
	7	Down sampling	128	128 × 64
	8	Hybrid basic block	128	128 × 64
	9	Hybrid dilated block (dilation = 2)	128	128 × 64
	10	Hybrid dilated block (dilation = 5)	128	128 × 64
	11	Hybrid dilated block (dilation = 7)	128	128 × 64
	12	Hybrid dilated block (dilation = 9)	128	128 × 64
	13	Hybrid dilated block (dilation = 11)	128	128 × 64
Decoder	14	Attention pyramid module	20	256 × 128
	15	Upsampling	20	512 × 256
	16	Upsampling	20	1024 × 512

The spatial separable convolution replaces the 3 × 3 convolution kernel with the joint convolution of 3 × 1 and 1 × 3, which reduces the number of parameters by 33%. After spatial separable convolution, two groups are combined to ensure the input and output have the same number of channels. And to avoid the vanishing gradient problem during the training phase, residual connection is adopted to superimpose the input feature channels and the feature channels after spatial separable convolution. Finally, channels are reordered by channel shuffle to enable information communication.

At layers 9–13 of the network, the image resolution becomes smaller after downsampling, and the computation requirements are greatly reduced. Hence Hybrid Dilated Blocks with large dilation rate are adopted for convolution operation, so as to obtain a large receptive field and improve the distinction between small objects and background as well as the segmentation accuracy.

Hybrid Dilated Block: For the sake of efficiency, small convolution kernels are adopted in Hybrid Basic Block. The drawback is that the receptive field is limited and is not conducive to the communication of context information. To address this issue, dilated convolution is applied in Hybrid Dilated Block, as shown in Fig. 4. Dilated convolution improves the receptive field of the convolution operation without introducing additional parameters, which is beneficial to improve accuracy while maintaining efficiency. Compared with larger convolution kernels, the receptive field of dilate convolution increases exponentially as the dilation rate rises, but it does not bring additional requirements for computation and memory resources, as the number of parameters remains unchanged. The potential problem of dilated convolution is that the kernel is discontinuous, which means not all pixels are involved in the convolution operation, so there will be information loss. To ensure the richness of feature information, dilated convolution is used only for the latter two units in spatial separable convolution, and the dilation rates of the Hybrid Dilated Block used in layers 9–13 are pairwise co-prime.

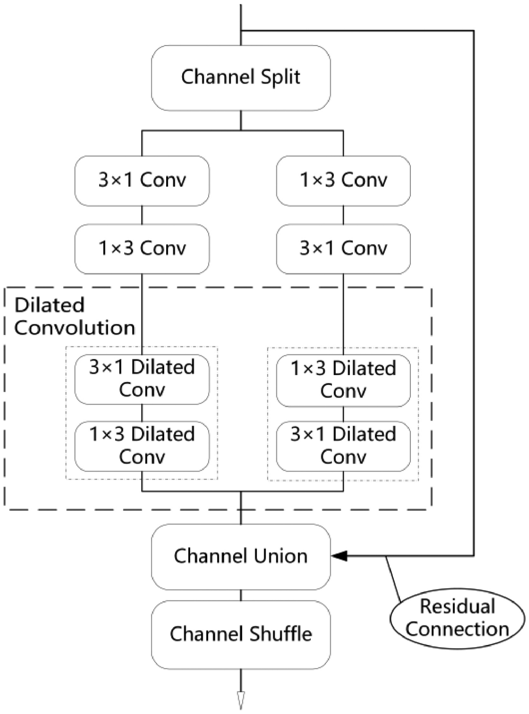

Figure 4. Hybrid Dilated Block.

Attention Pyramid Module: Attention Pyramid Module is designed in the decoder (layer 14 of the network), so as to capture multi-scale context information without affecting the efficiency. The Attention Pyramid Module has a multi-branch structure, which transfers the feature maps obtained by the encoder to four branches, such as a 9 × 9 convolution branch, a 1 × 1 convolution branch, a maximum pooling branch, and a downsampling branch, as shown in the middle of Fig. 2. The 9 × 9 convolution kernel has a larger receptive field than the convolution kernel in the encoder, hence has a stronger ability to obtain features. Moreover, after downsampling, the input feature maps of this layer have small resolution, so the 9 × 9 convolution will not bring too much computation burden. The role of the 1 × 1 convolution branch is to enable information communication between different feature channels, hence improving the richness of the features. The maximum pooling branch filters out small values to obtain the information of high global priority. The downsampling branch reduces the image resolution twice and obtains information at different scales, then the feature information are cascaded through pyramid structure.

At the end of the Attention Pyramid Module, the features outputted by each branch are cascaded together and then upsampled at layers 15 and 16 to restore the image resolution and obtain the final semantic segmentation result. Bilinear interpolation is adopted for upsampling, which does not require convolution operation. Compared with the commonly used transpose convolution method, bilinear interpolation requires fewer parameters and has high computational efficiency.

4. Super-pixel Construction

In the SemanticSurfel system, super-pixel-based surfels are used to construct 3D dense semantic map of the environment. Using super-pixels instead of pixels as the basic unit can effectively reduce the amount of data to be processed and reduce the memory footprint, which is more suitable for mobile robots working in large-scale environments. In addition, in the process of super-pixel extraction, semantic, depth, and intensity information of pixels are comprehensively utilised, and only the dominant information in the super-pixel region is considered. Therefore, the noise and image blur caused by the moving camera will be reduced and the map construction accuracy will be improved.

4.1 Depth Evaluation

Depth information can be obtained directly by RGB-D camera or calculated by the binocular matching algorithm. However, RGB-D camera has limited range and is easily disturbed by sunlight, so it is not suitable for large-scale outdoor environment. And binocular matching algorithm is usually difficult to achieve good balance between speed and accuracy. In the SemanticSurfel system, StereoDNN [20] network is adopted to estimate the dense depth graph because of its high accuracy and efficiency.

4.2 Super-pixel Extraction

SLIC is a traditional k-means algorithm for super-pixel extraction [21]. In this subsection, an augmented SLIC algorithm that fuses semantic, depth, and intensity graphs together is proposed.

During initialisation, the input image is divided into K evenly distributed grids, and each grid is initialised as a cluster $C_i = [x_i, y_i, d_i, c_i, s_i, n_i, r_i]^T$, where $\{x_i, y_i\}$ is the centre coordinate, d_i is the average depth, c_i is the average intensity, s_i is the number of the pixels in this cluster with most commonly semantic label, n_i is the number of all the pixels in the cluster, and r_i is the cluster radius, which is the largest distance between the pixels in the cluster to the centre.

After initialisation, iterations are carried out on all the pixels to add them to their nearest cluster. In the iteration process, (1) and (2) are used to judge the distance of one pixel from the cluster centre, including the location distance, intensity distance, depth distance, semantic information, and other factors.

$$\begin{aligned} D &= \frac{(x_i - x_u)^2 + (y_i - y_u)^2}{N_l^2} \\ &\quad + \frac{(c_i - c_u)^2}{N_c^2} + \frac{n_i}{s_u N_s^2} \qquad (1) \\ D_d &= D + \frac{(1/d_i - 1/d_u)^2}{N_d^2} \qquad (2) \end{aligned}$$

where $\{x_u, y_u\}$, c_u, d_u is the coordinate, intensity, and depth of a pixel u, s_u is the number of pixels with same semantic label as pixel u in this cluster, and D_d is the distance between the pixel and the cluster centre. N_l^2, N_c^2, N_s^2, and N_d^2 are used to normalise location, intensity, semantic information, and depth, respectively.

When the iteration converges, all pixels are finally divided, and the final cluster centre $\{x_i, y_i\}$ and c_i is update to the mean value of the cluster, depth d_i is optimised by Gauss–Newton iterative, with optimisation goal $E_d = \sum_u L_\delta(d_u - d_i)$, where L_δ is the Huber kernel function. The resulting clusters constitute the super-pixels.

5. Surfel Map Construction

In the SemanticSurfel system, the super-pixels extracted from the current image are used together with the current pose of the camera to construct surfels in the 3D global coordinate frame. And as the camera moves, the newly created surfels are merged with the existing surfels in the map to form a consistent 3D dense semantic map of the environment.

5.1 Surfel Initialisation

The definition of surfel in SemanticSurfel system is $C_i = [S_p, S_n, S_c, S_s, S_r, S_w, S_t, S_i]^T$, which represents the 3D coordinate, normal vector, intensity, semantic information, radius, weight, fusion counter, and associated keyframe id of the surfel, respectively.

For each super-pixel, a surfel is initialised with it and the pose of the associate keyframe. Intensity S_c and semantic label S_s are initialised to c_i and s_i of the super-pixel. S_i is the keyframe ID given by SLAM system, and S_t is initialised to 0, indicating that the surfel has not been fused. The weight S_w is initialised to $\frac{s_i}{n_i}$, which is the ratio between the number of the pixels with most commonly semantic label to the total number of pixels in this super-pixel.

The normal vector S_n, position S_p, and radius S_r are initialised using the methods in [22]. S_n is initialised as the average normal vector of all pixels firstly, and then more accurate result is obtained by means of optimisation, as shown in (3):

$$E_s = \sum_u L_\delta \left(S_n \cdot (p_u - \bar{p}) + b\right) \tag{3}$$

where $p_u = R \cdot p_{2d} + t$ is the 3D coordinates of pixel point, p_{2d} is the 2D coordinates of the pixel in the current image, R and t are the rotation matrix and translation vector of the current camera pose estimated by ORB-SLAM, $\bar{p}$ is the average coordinate of all 3D points, and b is used to estimate bias.

S_p and S_r are initialised using (4) and (5):

$$S_p = \frac{S_n \cdot \bar{p} - b}{S_n \cdot (K^{-1}[x_i, y_i, 1]^T)} K^{-1}[x_i, y_i, 1]^T \tag{4}$$

$$S_r = \frac{S_p\,(z) \cdot r_i \cdot \parallel K^{-1}[x_i, y_i, 1]^T \parallel}{f \cdot S_n \cdot (K^{-1}[x_i, y_i, 1]^T)} \tag{5}$$

where K is the camera intrinsic parameter matrix, $S_p(z)$ is the depth of the surfel, r_i is the radius of the super-pixel, and f is the focal length of the camera.

5.2 Surfel Fusion

During the movement of the camera, keyframes are updated and new surfels are extracted gradually. The same surfel observed in different keyframes is fused in this subsection.

When a new surfel S^n is initialised, if a surfel S^l in the map has the same semantic label with S^n, and the distance between them and the angle difference between their normal vectors is less than a threshold, they are considered to be the same surfel. The judgement criterion is shown in (6):

$$\begin{cases} S_s^n = S_s^l \\ \left|S_p^n - S_p^l\right| < \sigma_p \\ S_n^n \cdot S_n^l > \sigma_n \end{cases} \tag{6}$$

If (6) is satisfied, S^n and S^l are fused into one surfel S^f by (7), where the coordinate, normal vector, intensity, and radius of S^f are the weighted sum of the corresponding parameters of S^n and S^l, S_i^f is set to the associate keyframe id of S^n, S_t^f is set to the fusion counter of S^l plus 1, and S_w^f is the sum of the weight of S^n and S^l.

$$\begin{cases} S_k^f = \frac{S_k^l S_w^l + S_k^n S_w^n}{S_w^l + S_w^n}, k \in \{p, n, c, r\} \\ S_s^f = S_s^n \\ S_i^f = S_i^n \\ S_t^f = S_t^l + 1 \\ S_w^f = S_w^l + S_w^n \end{cases} \tag{7}$$

Through the surfel fusion process, all surfels are added into the global semantic map. In order to reduce the size of the map and deal with outliers, surfels that are more than 10 keyframes away from the current keyframe but updated less than 5 times will be removed from the map.

6. Experiments and Discussion

6.1 Experiment 1: Semantic Segmentation Accuracy and Efficiency

6.1.1 Methods and Procedures

Experiment 1 was designed to evaluate the accuracy and efficiency of the proposed semantic segmentation network, HybridNet using the Cityscape dataset.

Cityscape Dataset: The Cityscape dataset contains 5,000 images collected in street scenes from 50 different cities. 2,975, 500, and 1,525 images in Cityscape are used as the training set, verification set, and test set, respectively. High-quality pixel-level annotations of 19 semantic classes are provided in this dataset.

Methods for Comparison: The segmentation result of HybridNet was compared with several other state-of-the-art lightweight semantic segmentation networks, such as ENet, ICNet, ERFNet, and CGNet. ENet and CGNet have the smallest network parameter size while ICNet and ERFNet have the highest accuracy.

Evaluation Index: Segmentation accuracy is evaluated using the common mean-Intersection-over-Union (mIoU). In this paper, the hybrid efficiency is defined as the index weighting the computing time and network parameter size to comprehensively evaluate the computational efficiency and memory footprint of these networks:

$$\begin{cases} \text{in}v_{\text{time}} = \frac{1}{\text{time}} \\ \text{in}v_{\text{size}} = \frac{1}{\text{paramete}r_{\text{size}}} \\ \text{efficiency} = \alpha \cdot \text{normalize}\left(\text{in}v_{\text{time}}\right) \\ \qquad + (1-\alpha) \cdot \text{normalize}\left(\text{in}v_{\text{size}}\right) \end{cases} \tag{8}$$

In practise, time efficiency is more important than memory footprint, and the speed of CPU and GPU is evolving slower than the size of memory. Therefore, the coefficient α used in this experiment was 0.75, which means we valued more on computational efficiency.

Implementation Protocol: During the training phase, HybridNet was trained 300 iterations on the Cityscape dataset. All parameters of the convolution kernels were initialised by normal distribution with the mean value be

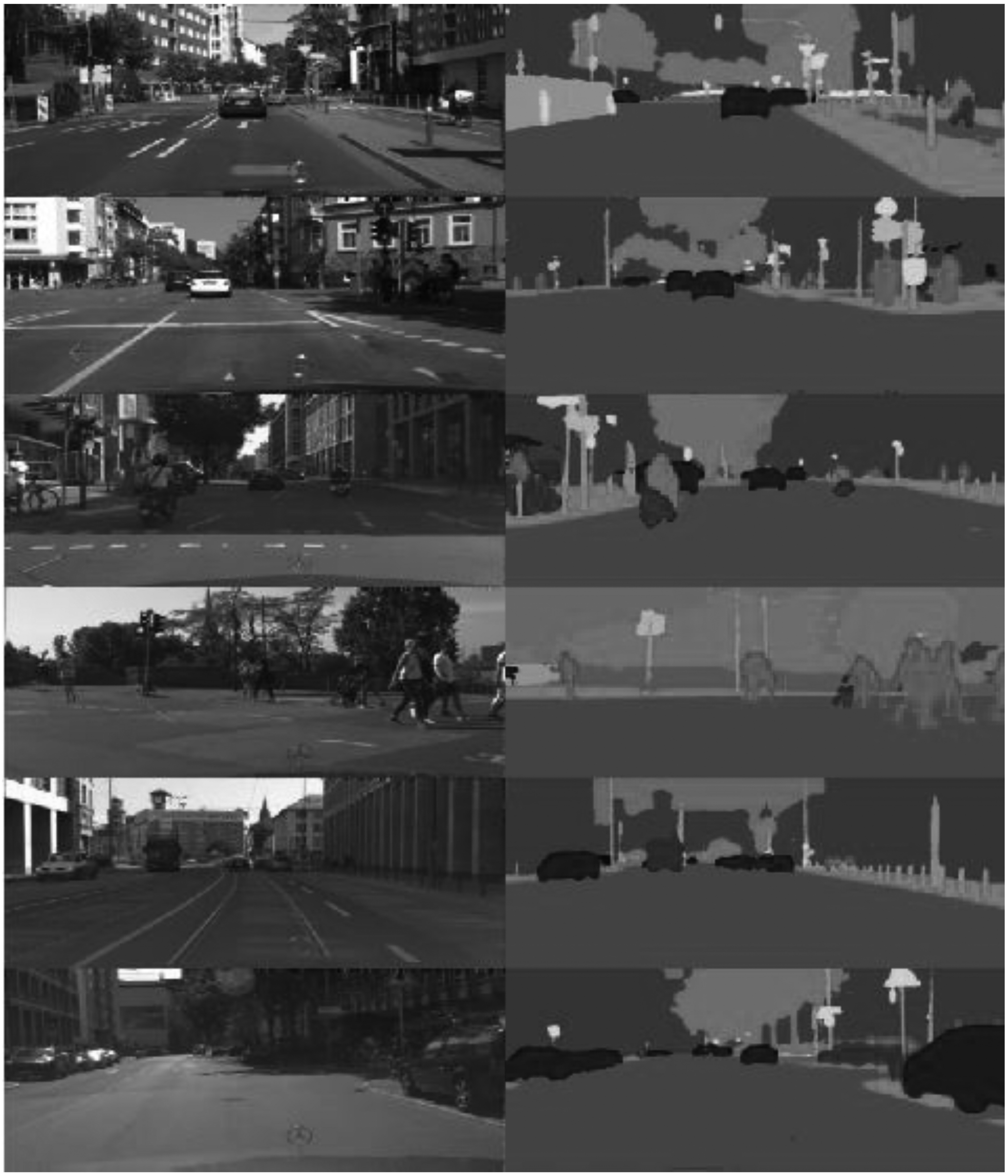

Figure 5. Semantic segmentation result on the Cityscape dataset.

0 and the standard deviation be 1.0, and the batch size of each iteration was set to 5. Adam optimizer was used for training because it has no stationary requirement for the cost function and handles the sparse gradient well. The initial value of the learning rate was set to 0.0005, and adaptive attenuation was used to ensure the early acceleration and the late convergence. In order to improve the generalisability of HybridNet, the training set was augmented by randomly dividing the training set into two groups, and then move the image 0, 1, or 2 pixels towards a random direction. All of these networks were deployed on the same 1080Ti platform.

6.1.2 Experimental Results and Discussion

The semantic segmentation result of HybridNet on the Cityscape dataset is shown in Fig. 5, where the left column is the original images, and the right column is the segmentation results. In the results, different colors indicate different objects. As Fig. 5 shows, the road, the pavement, motors, pedestrians, street lamps, road signs, trees, walls, and the sky are segmented correctly.

Table 2 lists the comparison of the segmentation results of HybridNet and other networks in terms of the mIoU, the mean computing time, the network parameter size, and the hybrid efficiency. According to Table 2, the accuracy of HybridNet is 68.3%, and the hybrid efficiency is 0.80. Compared with ICNet (the best accuracy), the accuracy of ERFNet, HybridNet, CGNet, and ENet is 1.01%, 1.73%, 6.76%, and 16.12% lower, respectively. So the accuracy of the proposed HybridNet and ERFNet are very close to ICNet. And HybridNet achieves the highest hybrid efficiency, which is much higher than other networks. For instance, CGNet (second rank) is 25.9% lower than HybridNet.

Table 2
Semantic Segmentation Comparison on the Cityscape Dataset

Network	mIoU (%)	Time (ms)	Parameter size (Mb)	Hybrid Efficiency
Enet	58.3	34	0.36	0.493
ICNet	**69.5**	33	7.8	0.262
ERFNet	68.8	27	3.0	0.336
CGNet	64.8	20	**0.5**	0.593
HybridNet	68.3	**11**	1.8	**0.80**

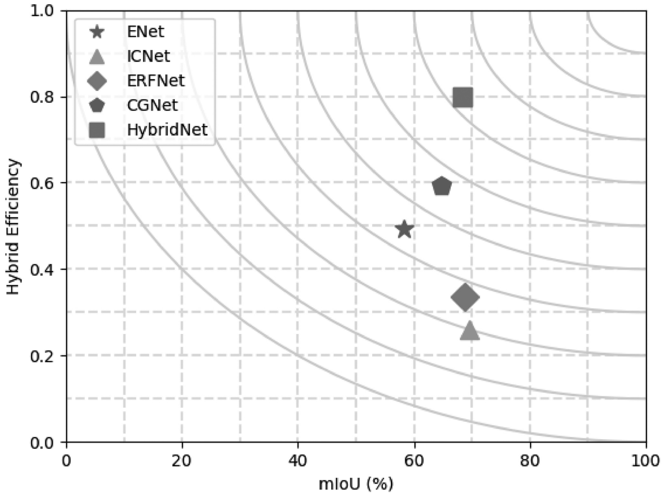

Figure 6. Semantic segmentation comparison in terms of mIoU and hybrid efficiency. Segmentation result on the Cityscape dataset.

Visualised comparison between these networks in terms of mIoU and hybrid efficiency is plotted in Fig. 6. For each network, the closer it is to the top right corner, the better its balance between accuracy and hybrid efficiency. As Fig. 6 shows, HybridNet achieves the best balance, which is suitable for the real-time operation of mobile robots.

6.2 Experiment 2: Mapping Accuracy and Efficiency

6.2.1 Methods and Procedures

Experiment 2 was designed to evaluate the performance of the proposed SemanticSurfel system using the ICL-NIUM dataset and the KITTI dataset.

ICL-NIUM and KITTI Datasets: ICL-NIUM is a dataset derived from a rendered room model to evaluate the accuracy of SemanticSurfel because it includes the ground truth of the model. The mapping accuracy is evaluated using the mean difference between the constructed map and the ground truth. The KITTI dataset contains sequences from different scenes, such as villages, urban areas, and highways. The 00 sequence of the KITTI dataset was used in this experiment to show the real-time application of SemanticSurfel in large-scale environments.

Methods for Comparison: The mapping accuracy of SemanticSurfel on the ICL-NIUM dataset was compared with state-of-the-art methods BundleFusion [23], ElasticFusion, and InfiniTAM [24].

Implementation Protocol: During this experiment, the SemanticSurfel system was deployed on an Intel i5-7500 CPU platform with an additional embedded Jetson TX2 GPU.

6.2.2 Experimental Results and Discussion

Table 3 shows the performance of SemanticSurfel compared with other methods on the ICL-NIUM dataset, including mapping accuracy and required computing resources. According to Table 3, in terms of mapping accuracy, BundleFusion has the best average while SemanticSurfel has the smallest variance. So BundleFusion and SemanticSurfel are comparable in accuracy. In terms of computing resources, BundleFusion, ElasticFusion, and InfiniTAM need two or one desktop GPUs, which is not applicable for real-time implementations in large-scale environments. Therefore, the proposed SemanticSurfel can achieve comparable accuracy and also can be applied to real-time scenarios.

The close shot and overview of the mapping results of the SemanticSurfel system on KITTI-00 sequence are shown in Figs. 7 and 8, where red, blue, green, and light grey area represent the road, buildings, trees, and other kinds of objects in the environment, respectively. Although the density of the constructed map is satisfactory for most applications, more detailed map can be constructed by introducing high-resolution Lidar as complementary information in super-pixel construction process in the near future.

Figure 9 shows the time-consuming of SemanticSurfel on KITTI-00 sequence. Among all the modules, the SLAM system, semantic segmentation, and depth estimation process each frame at about 50 ms, 20 ms, and 11 ms, respectively. The average processing time of SemanticSurfel per frame is 88.8 ms, and then the frequency is about 11.3 fps, which satisfies the real-time requirement. And the processing time remains relatively stable as the number of frames accumulates. Therefore, it is concluded that SemanticSurfel system can scale well in large-scale environments.

Table 3
Mapping Accuracy and Computing Resource Comparison on the ICL-NIUM Dataset (cm)

Method	ICL-NIUM Sequence						Computing Resource	Real-time Implementation
	kt0	kt1	kt2	kt3	Average	Variance		
BundleFusion	0.5	0.6	0.7	0.8	**0.65**	0.0167	Two desktop GPU	Not applicable
ElasticFusion	0.7	0.7	0.8	2.8	1.25	1.07	One desktop GPU	Not applicable
InfiniTAM	1.3	1.1	0.1	2.8	1.325	1.2425	One desktop GPU	Not applicable
SemanticSurfel	0.6	0.7	0.8	0.8	0.725	**0.0092**	**Embedded GPU**	**Applicable**

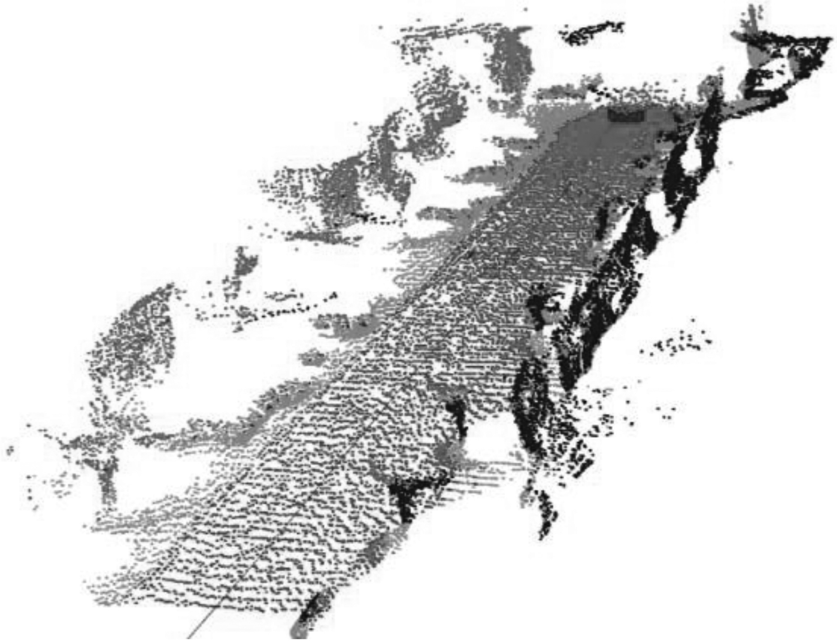

Figure 7. KITTI-00 mapping result in close shot.

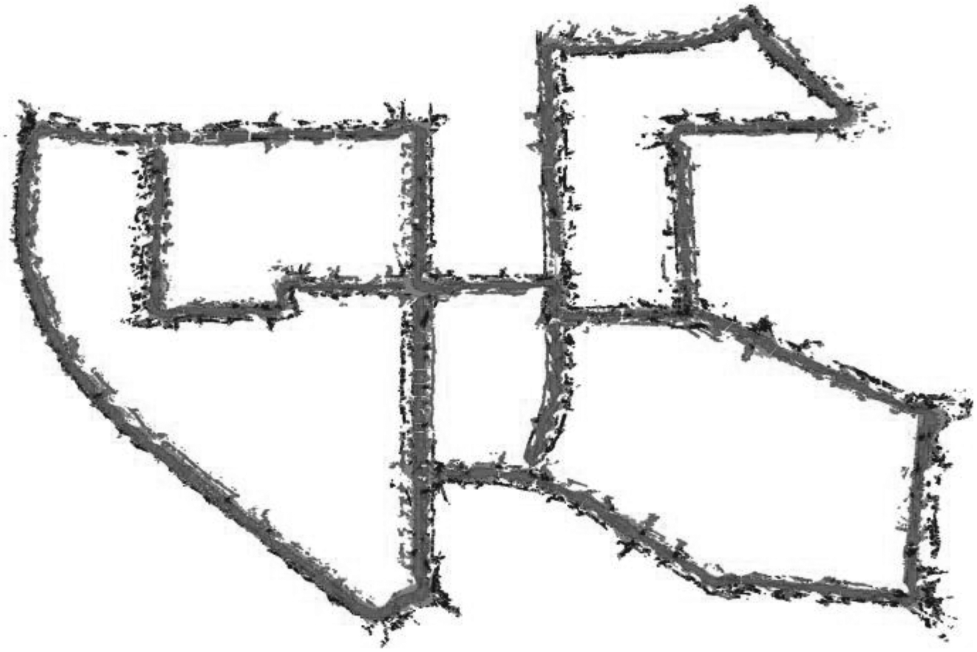

Figure 8. KITTI-00 mapping result in overview.

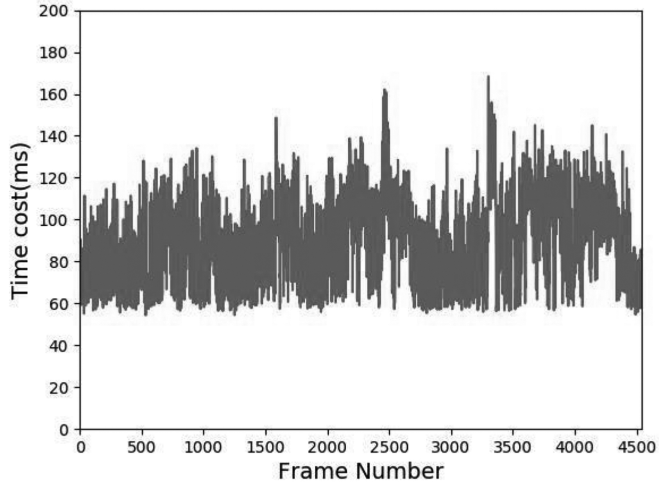

Figure 9. Time -consuming on KITTI-00 sequence.

6.3 Experiment 3: Mapping in Real Campus Environment

6.3.1 Methods and Procedures

In order to verify the effectiveness of SemanticSurfel in real environment, this experiment was carried out in the campus of Harbin Institute of Technology (Shenzhen) where the mapping area was about 7000 m^2. The hardware platform used in this experiment was a mobile robot with a ZED stereo camera and an Intel i5-7500 CPU platform with an additional embedded Jetson TX2 GPU. During the experiment, 1,831 pairs of stereo images were collected at the resolution 1280 × 720.

6.3.2 Experimental Results and Discussion

The mapping results in the real campus are shown in Figs. 10 and 11, where purple, pink, blue, dark grey, and green areas represent the road, pavement, sky, buildings, and trees, respectively. In the close shot of the mapping result in Fig. 10, the top row is the original image, the middle row is the corresponding semantic segmentation result, and the bottom row is the corresponding map. The map overview is shown in Fig. 10. Figure 11 shows the time-consuming of SemanticSurfel in this experiment. As shown, the average processing time per frame is 123.4 ms, and then the frequency is about 8.1 fps. Moreover, the processing time remains relatively stable as the number of frames accumulates. Therefore, the experimental results demonstrate that SemanticSurfel system can effectively build the 3D dense semantic map of real large-scale environment on a mobile robot.

7. Conclusion

This paper proposes HybridNet and SemanticSurfel, which can semantically segment the input image at pixel-level and construct the 3D dense semantic map of the large-scale environment in real time. In the future, we will focus

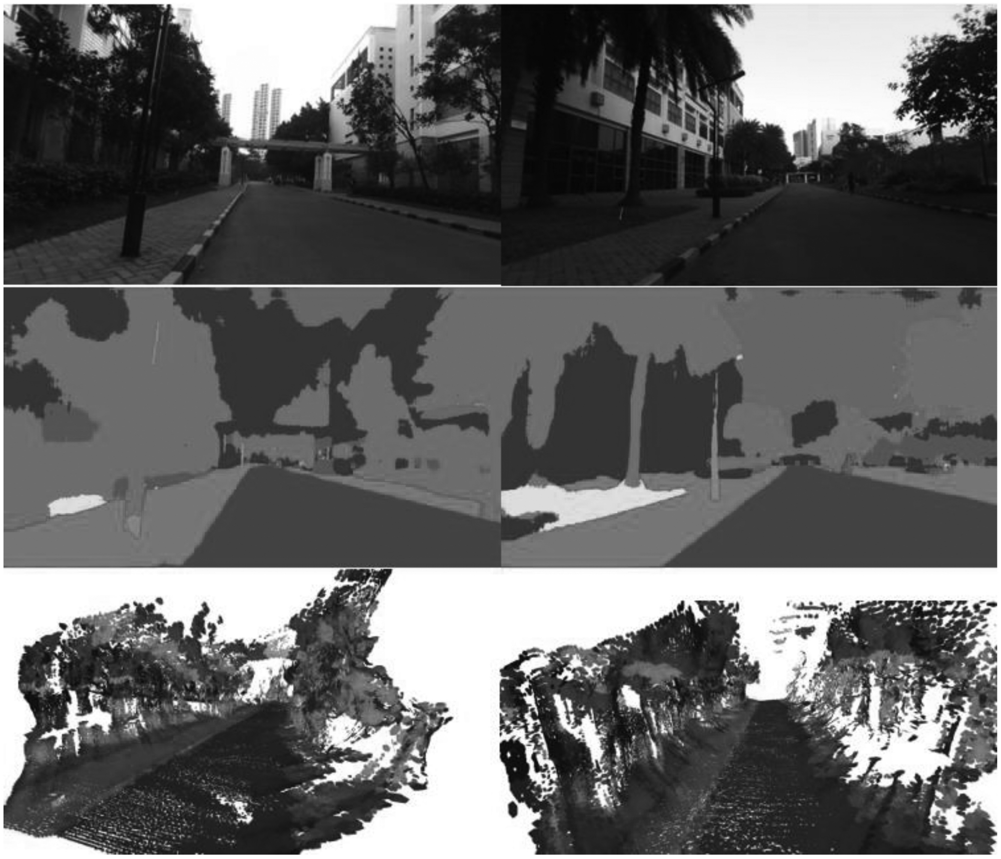

Figure 10. Campus mapping result in close shot.

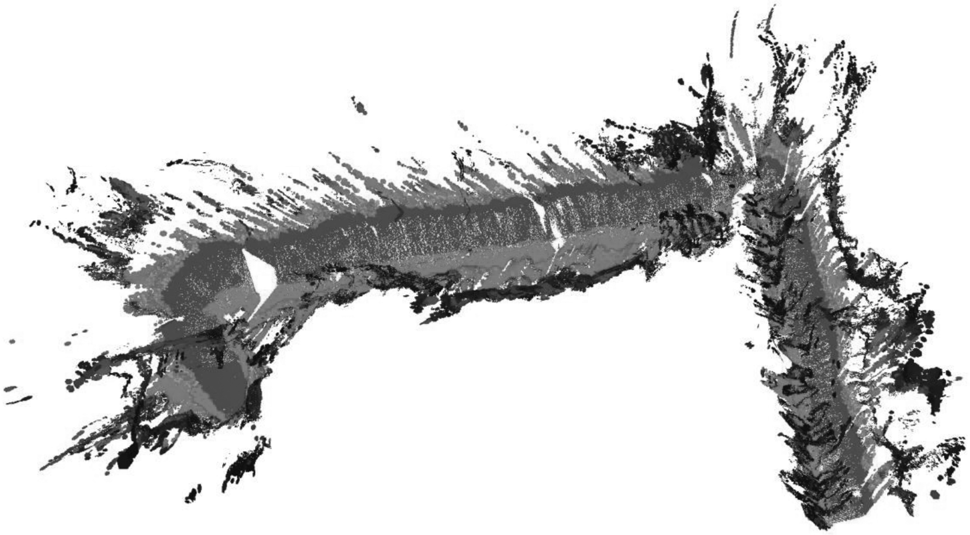

Figure 10. Campus mapping result in overview.

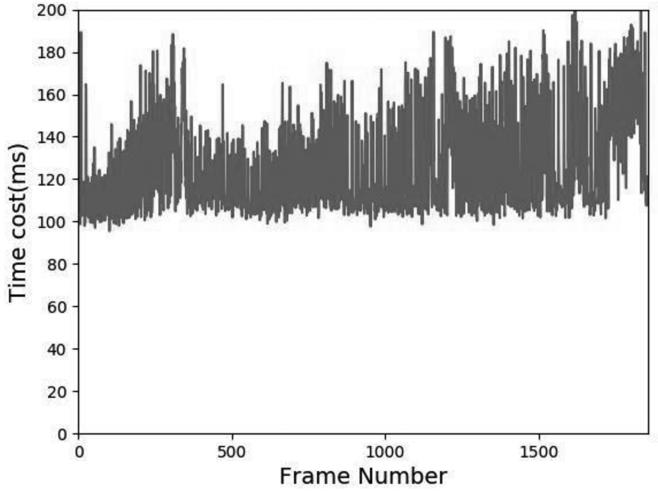

Figure 11. Time-consuming in real campus environment.

on introducing high-resolution Lidar as complementary information in super-pixel construction process to obtain more detailed map, and utilise the constructed semantic map in mobile robot auto-exploration tasks.

References

[1] F. Niu and M. Abdel-Mottaleb, View-invariant human activity recognition based on shape and motion features, *Proc. IEEE 6th International Symp. Multimedia Software Engineering*, Miami, FL, 2004, 546–556.

[2] H. Zheng, C. Sun, and H. Yin, A novel deep model with structure optimization for scene understanding, *International Journal of Robotics and Automation*, *36*(6), 2021, 392–401.

[3] J. Shotton, M. Johnson, and R. Cipolla, Semantic texton forests for image categorization and segmentation, *Proc. IEEE Computer Vision and Pattern Recognition*, *5*, Anchorage, AK, 2008, 1–8.

[4] C. Lindner, S. Thiagarajah, J. Wilkinson, G. Wallis, and T. Cootes, Fully automatic segmentation of the proximal femur using random forest regression voting, *IEEE Transactions on Medical Imaging*, *32*(8), 2013, 1462–1472.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, Mask R-CNN, *Proc. IEEE Computer Vision and Pattern Recognition*, Honolulu, HI, 2017, 1–12.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, SegNet: A deep convolutional encoder-decoder architecture for image segmentation, *CoRR*, *abs/1511.00561*, 2015. [Online]. Available: http://arxiv.org/abs/1511.00561

[7] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, ENet: A deep neural network architecture for real-time semantic segmentation, *CoRR*, *abs/1606.02147*, 2016. [Online]. Available: http://arxiv.org/abs/1606.02147

[8] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, ICNet for real-time semantic segmentation on high-resolution images, *CoRR*, *abs/1704.08545*, 2017. [Online]. Available: http://arxiv.org/abs/1704.08545

[9] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation, *IEEE Transactions on Intelligent Transportation Systems*, *19*(1), 2018, 263–272.

[10] X. Zhang, X. Zhou, M. Lin, and J. Sun, ShuffleNet: An extremely efficient convolutional neural network for mobile devices, *CoRR*, *abs/1707.01083*, 2017. [Online]. Available: http://arxiv.org/abs/1707.01083

[11] Y. Wang, Q. Zhou, J. Liu, J. Xiong, and L. J. Latecki, LedNet: A lightweight encoder-decoder network for real-time semantic segmentation, *Proc. IEEE International Conf. on Image Processing (ICIP)*, Taipei, 2019, 1860–1864.

[12] T. Wu, S. Tang, R. Zhang, J. Cao, and Y. Zhang, CGNet: A light-weight context guided network for semantic segmentation, *IEEE Transactions on Image Processing*, *30*(1), 2021, 1169–1179.

[13] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, SLAM++: Simultaneous localisation and mapping at the level of objects, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Portland, OR, 2013, 1352–1359.

[14] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg, Joint semantic segmentation and 3d reconstruction from monocular video, in D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars (eds.), *Computer Vision – ECCV*. (Cham: Springer International Publishing, 2014), 703–718.

[15] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto, Volumetric instance-aware semantic mapping and 3D object discovery, *IEEE Robotics and Automation Letters*, *4*(3), 2019, 3037–3044.

[16] J. Stückler and S. Behnke, Multi-resolution surfel maps for efficient dense 3D modeling and tracking, *Journal of Visual Communication and Image Representation, 25*(1), 2014, 137–147.

[17] J. McCormac, A. Handa, A. J. Davison, and S. Leutenegger, SemanticFusion: Dense 3D semantic mapping with convolutional neural networks, *CoRR*, *abs/1609.05130*, 2016. [Online]. Available: http://arxiv.org/abs/1609.05130

[18] T. D. Dung and G. Capi, Application of neural networks for robot 3D mapping and annotation using depth image camera, *International Journal of Robotics and Automation*, *37*(6), 2022, 529–536.

[19] R. Mur-Artal and J. Tardós, ORB-SLAM2: An open-source slam system for monocular, stereo and RGB-D cameras, *IEEE Transactions on Robotics*, *33*(15), 2017, 1255–1262.

[20] N. Smolyanskiy, A. Kamenev, and S. Birchfield, On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach, *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Salt Lake City, UT, 2018, 1120–1128.

[21] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, SLIC superpixels compared to state-of-the-art superpixel methods, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(11), 2012, 2274–2282.

[22] K. Wang, F. Gao, and S. Shen, Real-time scalable dense surfel mapping, *Proc. International Conf. on Robotics and Automation (ICRA)*, Montreal, QC, 2019, 6919–6925.

[23] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, BundleFusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration, *ACM Transactions on Graphics*, *36*(7), 2017, 1–18.

[24] O. Kähler, V. A. Prisacariu, and D. W. Murray, Real-time large-scale dense 3D reconstruction with loop closure, *Proc. European Conf. on Computer Vision*, Amsterdam, 2016, 500–516.

Biographies

Zhiwei Xing is currently pursuing the Ph.D. degree. The main research direction is the autonomous positioning and navigation of mobile robots in outdoor large-scale environments.

Xiaorui Zhu received the doctoral degree. She is a Professor. The main research direction is mobile robot system and technology. In 2012, she won the second prize of the National Science and Technology Progress Award.

Yudong Wu is currently pursuing the master's degree. The main research direction is lightweight neural network and semantic mapping of autonomous mobile robots.