Intelligent Service Robotics (2021) 14:47–60
https://doi.org/10.1007/s11370-020-00341-8

ORIGINAL RESEARCH PAPER

Fine semantic mapping based on dense segmentation network

Guoyu Zuo[1,2] · Tao Zheng[1,2] · Yuelei Liu[1,2] · Zichen Xu[1,2] · Daoxiong Gong[1,2] · Jianjun Yu[1,2]

Received: 5 December 2019 / Accepted: 19 October 2020 / Published online: 16 November 2020

Abstract
This paper proposes a fine semantic mapping method using dense segmentation network (DS-Net) to obtain good performance of semantic mapping fusion. First, the RGB image and the depth image are used to generate a dense indoor scene map via the state-of-the-art dense SLAM (ElasticFusion). Then, the DS-Net is constructed based on DenseNet's dense connection to perform precise semantic segmentation on the input RGB image. Finally, the long-term correspondence is established between the indoor scene map and the landmarks using continuous frames both in the visual odometer and in loop detection, and the final semantic map is obtained by fusing the indoor scene map with the semantic predictions of the RGB-D video frames of multiple angles. Experiments were performed on the NYUv2, PASCAL VOC 2012, CIFAR10 datasets and our laboratory environments. Results show that our method can reduce the error in dense map construction and obtain good semantic segmentation performance.

Keywords Semantic segmentation · RGB-D · DS-Net · DenseNet · ElasticFusion

1 Introduction

In the fields of robotics and computer vision, semantic map lays a foundation for realizing human–robot interaction and human–robot fusion, and it is widely used in robot navigation, robot manipulation and augmented reality. It is always an important research issue to construct an incremental and robust semantic map in real time. Owing to the rapid development of simultaneous localization and mapping (SLAM), the robot can use sparse or dense point clouds to map environments to the 2D or 3D grid maps [1]. Great achievements have been made in autonomous navigation and automatic obstacle avoidance by using geometric map and feature map. However, the robot agent cannot get more from these maps which only contain geometric and point cloud information. So, it is difficult and even impossible for the robot to understand complex environments and difficult to be competent for more complex tasks. In order to realize friendly understanding of complex environments, semantic map that integrates semantic and geometric information must be established to improve the ability of the robot in path planning and other more sophisticated tasks.

Currently, semantic mapping framework has two main parts: semantic segmentation performed by convolutional neural network (CNN) and map construction based on SLAM. Some CNN-based methods focus on improving the accuracy of semantic segmentation [2,3]. However, to maximally extract the information in the maps, we often deepen the network layers and build robotic systems with more rigorous computation, because large amounts of computing resources are needed to perform operations such as 3D reconstruction, camera pose estimation, and CNN-based semantics. To achieve the real-time performance, McCormac et al. [2] constructed a semantic 3D map by combining CNN and dense SLAM system, and Hermans et al. [4] proposed a 2D–3D label transformation based on Bayesian updates and dense pairwise 3D Conditional Random Fields. The above frame-skipping strategy can improve the run-time performance, but its application is limited. This is because it tends to bring inaccuracy in fast camera motions.

The SLAM systems generally establish the correspondence from 2D frames to globally consistent 3D maps. Compared with other mature SLAM systems such as RGB-D mapping [5], Kintinuous [6] and BundleFusion [7], ElasticFusion [8] is better to represent semantic information. ElasticFusion uses surface elements (surfels), which are

✉ Guoyu Zuo
zuoguoyu@bjut.edu.cn

1 Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

2 Beijing Key Laboratory of Computing Intelligence and Intelligent Systems, Beijing 100124, China

suitable for classifying point clouds and parsing semantic information, to generate and fuse point clouds, and deformation maps are used to ensure globally consistent mapping in the closed loop.

The latest developments in semantic mapping focus on two main aspects. One is to improve the accuracy of semantic segmentation. The other is to use the semantic information in the closed-loop detection module in SLAM to correct poses and obtain a more consistent map. Niko Sünderhauf at al. [9] proposed an object-oriented semantic mapping method and discussed its application in SLAM data association based on the established semantic map, but they only considered a few object classes in semantic segmentation. Bowman et al. [10] realized the integration of semantic information into SLAM. They formulated an optimization problem over sensor states and semantic landmarks, which integrate metric information, semantic information, and data associations. Then, they decomposed it into two interconnected problems: estimation for discrete data association and landmark class probabilities, and continuous optimization over metric states. McCormac et al. [2] implemented semantic mapping by performing probability fusion on semantic segmentation and dense 3D point clouds, in which probabilistic event model is adopted in semantic segmentation. However, the method fails to generate accurate semantic maps by probability multiplication and does not perform well for 3D semantic segmentation in complex environments.

To construct real-time and efficient semantic map, we propose a semantic segmentation network based on DenseNet [11] by combining ElasticFusion and the DS-Net segmentation network, in which 2D segmentation of input frames is performed by using the built DS-Net framework, and the mapping from 2D segmentation to 3D point clouds is realized by using the Bayesian framework.

This paper is organized as follows: Section 2 describes the semantic map-related works in recent years. Section 3 introduces our method. In Sect. 4, experiments on the CIFAR10, NYUv2 [12] and the real environments of our laboratory are performed to verify the effectiveness of our approach. This paper is concluded in the last section.

2 Related work

2.1 Semantic mapping

Semantic mapping is a challenging task in semantic SLAM, which is to construct semantically annotated dense 3D maps for indoor scenes. 3D semantic maps are mostly built according to the following three main stages [4,13] : (i) frame-wise segmentation for estimating the per-pixel class probabilities of input frames, (ii) 2D–3D label transfer from 2D semantic segmentation to 3D maps, (iii) 3D refinement to denoise the class probabilities of 3D maps. Hermans et al. [4] used Random Decision Forest (RDF), Bayesian framework and Conditional Random Field (CRF) to carry out the above three stages, respectively. CRF works on each element of the 3D map reconstructed via SLAM, and it helps to obtain high-precision semantic map but it has a heavy computation load. SemanticFusion [2] also uses CRF to optimize the 3D semantic map after Bayesian fusion but has no obvious performance improvement.

Some works only identified partial 3D maps without generating dense semantic 3D maps. Bowman et al. [10] improved the performance of RGB SLAM in camera pose and scale estimation by utilizing not only low-level geometric features such as points, lines and surfaces but also the detected target landmarks. Salas-Moreno et al. [14] mapped indoor scenes at the level of semantically defined objects, but this method is limited to mapping objects in pre-defined databases. It does not provide dense labeling for the entire scene that contains walls, floors, doors, and windows. Nakajima et al. [15] proposed a semantic mapping approach by assigning class probability to each region of the 3D map established by a SLAM framework with a ResNet-based structure and geometric-based segmentation.

2.2 2D semantic segmentation

CNN can greatly reduce the input resolution through successive pooling operations and it is well suitable for image classification task and semantic segmentation. The semantic segmentation structure is generally a convolutional neural network with deconvolution modules, in which CNN realizes feature extraction of input image and assigns an initial category label to each pixel, and the deconvolution module is to output a probability density map with the same resolution as the input image. The structure with deconvolution and convolution has been applied successfully in semantic segmentation [16–18]. In [19], the fully convolutional network (FCN) is proposed to extract features by using the Visual Geometry Group (VGG) model [20] and output the segmentation result. In FCN, the 1×1 convolutional layer is used to represent spatial information, the unpooling layers are used to preserve the original resolution of input image, and the skipping connections are used to improve the robustness of semantic segmentation.

Some robust 2D semantic segmentation networks, such as Deeplab [21], ICNet [22] and SegNet [23], are implemented by either adding more layers or fine-tuning the network structure. These methods improve the convolution layers by using global and context information to extract more sufficient features. In these sparsely connected network structures, since a large number of neurons and the deeper network structure are introduced, it is difficult to train the network and achieve the expected goal in real-time applications. For example, Ivan

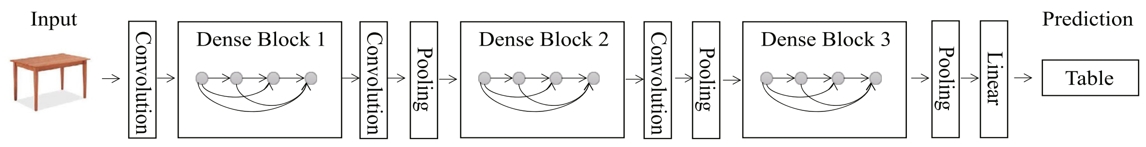

Fig. 1 A DenseNet with three dense blocks

Krešo et al. [24] put forward a ladder-style segmentation network and improved the semantic segmentation accuracy to a certain extent, but the network structure of jump connection also greatly increases the number of layers and parameters of the network.

2.3 DenseNet

The traditional object classification methods generally extract the same features for the same type of objects and output them as the recognition probabilities or positions of the objects in the image. Since sparse connections in traditional neural networks such as FractalNets [25], Highway network [26] and ResNets [27] cannot fully extract features, Huang G et al. proposed a new convolution network architecture, Dense Convolutional Network (DenseNet), as shown in Fig. 1.

To facilitate downsampling, the network is divided into multiple densely connected dense blocks. In each dense block, the direct connections between any two layers with the same feature map size can greatly reduce the number of parameters and improve the utilization of parameters. Such dense connections also effectively reduce the loss of some information about the image during conversion due to the pooling operation. Different from traditional CNN schemes, DenseNet can achieve good performance without increasing the depth of the network and the number of neurons, so it can alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters. The convolution and pooling used in DenseNet consist of a batch normalization layer and an 1×1 convolutional layer followed by a 2×2 average pooling layer. Unlike deeply supervised nets (DSN) [28], DenseNet has a single loss function, making model construction and gradient calculation easier.

Since the feature maps of different layers are connected in DenseNet, they are required to maintain the same feature size, which in turn limits the implementation of downsampling in the network. So, the transition layers are set between different dense blocks to fully extract the features of all previous layers and reduce the number of redundant features. For example, Zhang at al [29] reconstructed CT images from sparsely sampled sinusoids by using filter back projection (FBP) and then fed the FBP results to the DenseNet-based deep neural network. To reduce the compression artifacts of high efficiency video coding, Li at al. [30] proposed a DenseNet-based approach as the in-loop filter of efficient video coding.

In traditional semantic mapping methods, the semantic segmentation network based on sparse convolution cannot extract sufficient information and the accuracy of semantic map is not too high. So, the deeper network structure is used to fully extract image features, but it is difficult to guarantee the real-time performance of the system. Therefore, we propose a semantic segmentation network based on DenseNet to obtain fine semantic segmentation effect.

3 Method

As shown in Fig. 2, our method consists of three main parts: a SLAM framework for real-time reconstruction of indoor scenes, a specially designed 2D semantic segmentation network DS-Net, and a Bayesian update scheme. First, a geometric edge map is generated from the current depth frame, and the RGB image and the depth image obtained from the input RGB-D image are used to generate a dense indoor scene map via the ElasticFusion system. Second, precise semantic segmentation is performed on the input RGB image via DS-Net and returns the class probability of each set of pixels. Finally, the update class probability is assigned to each surfel in the 3D map by the Bayesian update scheme, and the final semantic map is generated by updating these probabilities using the correspondence between the frames provided by SLAM. In this method, camera pose based on SLAM is used to establish the relationship between the pixels of each keyframe and realize incremental fusion by stacking the semantic label information of keyframes with the spatial coordinates of the SLAM landmarks. By Bayesian update, the long-term correspondence between the indoor scene map and the landmarks is established using continuous frames both in the visual odometer and loop detection, and the final fused semantic map is obtained by integrating the indoor scene map with the semantic predictions of the RGB-D video frames of multiple angles.

3.1 Scene map construction

The dense three-dimensional reconstruction based on RGB-D images generally uses the grid model to fuse point clouds, and directly realizes the reconstruction in the spatial grid based on the extracted dense feature points. However, since

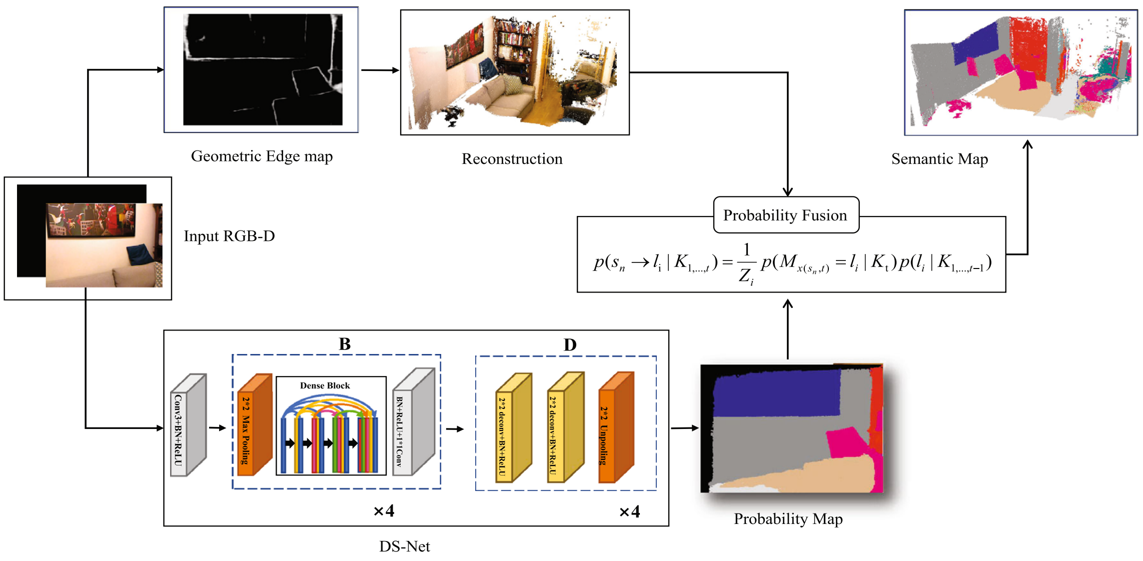

Fig. 2 The framework of our method

there is the lack of management of point cloud information, it is very difficult to make subsequent semantic annotation. Traditional SLAM algorithms generally improve the positioning accuracy of the robot by continuously optimizing the camera trajectory or feature points. When calculating camera pose, the traditional methods usually use 3D feature matching to perform rough pose calculation and then use the Iterative Closest Point (ICP) [31] algorithm to fine-tune the pose. The ICP algorithm calculates the pose by minimizing the distance from the point to the plane. In ElasticFusion, however, it not only calculates the pose of the RGB image through color consistency constraints, but also calculates the pose of the point cloud through the ICP algorithm.

In the geometric pose calculation, we use the depth image to estimate the camera pose transformation. The motion parameter ξ can be obtained through the ICP algorithm, which is to minimize the point-to-face error between the 3D back projection vertices:

$$E_{icp} = \sum_k \left(\left(v^k - \exp(\hat{\xi}) T v_t^k \right) \cdot n^k \right)^2 \tag{1}$$

where v^k and n^k are the corresponding vertex and normal in the previous camera coordinate frame. T is the current estimate of the transformation from the previous camera pose to the current one, and $\exp(\hat{\xi})$ is the matrix exponential that maps the Lie algebra to the corresponding Lie group.

Photometric pose estimation is realized in our method, in which the scene representation is an unordered list of surfels $\mathcal{M}$ (similar to the representation in Keller et al. [32]), where each surfel $\mathcal{M}^s$ has the following attributes: position $\mathbf{p} \in \mathbf{R}^3$, normal $\mathbf{n} \in \mathbf{R}^3$, color c $\in \mathbf{N}^3$, weight $w \in \mathbf{R}$, radius $w \in \mathbf{R}$, initial time stamp t_0 and last updated time stamp t. We try to find the motion parameters ξ to minimize the cost over the photometric error (intensity difference) between pixels:

$$E_{\text{rgb}} = \sum_{u \in \Omega} \left(I\left(u, c_t^l\right) - I\left(\pi\left(K \exp(\hat{\xi}) T p(u, D_t^l)\right), \hat{c}_{t-1}^a\right)\right)^2 \tag{2}$$

where the value of $\hat{c}_{t-1}^a$ is from the estimated active model part, not just from the previous frame. c_t^l is a new frame of color image. By mapping the Lie algebra to the corresponding Lie group, the spatial points in the coordinate system of depth map are converted into the world coordinate system.

When completing geometric pose calculation and photometric pose estimation, we minimize the joint cost function:

$$E_{\text{track}} = E_{\text{icp}} + w_{rgb} E_{rgb} \tag{3}$$

with the same $\boldsymbol{w}_{rgb}$=0.1 as in [33]. The Gauss–Newton nonlinear least-squares method with a three-level coarse-to-fine pyramid scheme is used to obtain the minimum joint cost function.

In our method, ElasticFusion uses the surfels model to integrate the point clouds, which contain position information, normal vectors, and color information, and then updates, merges, displays, and projects point clouds based on OpenGL. On this basis, continuous optimization of reconstruction map is performed to improve the reconstruction accuracy. Therefore, it is suitable to use the surfels model to represent the observed scene for semantic annotation. Figure 3 shows the surfels model.

The ElasticFusion algorithm consists of four steps: (1) converting the RGB image and depth image into point clouds, (2) acquiring coordinates and normal vectors of point clouds,

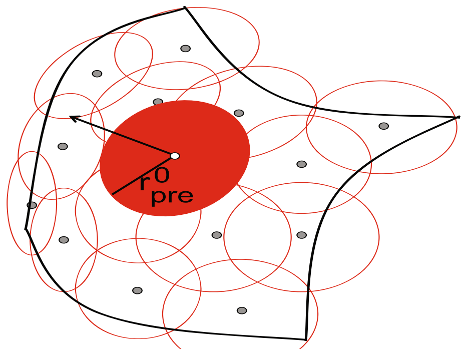

Fig. 3 The surface elements (surfels) model

(3) estimating the camera pose parameters by the ICP algorithm and the photometric method [34] for point clouds registration, and (4) using the random ferns algorithm [35] to achieve loop detection, integration and point cloud updates.

Local loop detection is the basis for reconstruction and segmentation. If the loop has not been detected when the camera moves, the reconstructed map will become a ghost image. To ensure the local surface consistency of the whole map, when the loops are revisited, the visual odometer and loop detection in SLAM establish the long-term correspondence between the landmark objects and the map, therefore reducing the error of the system in constructing dense map. The semantic predictions from multiple angles obtained from the RGB-D video frames are integrated into a map to improve the effect of semantic segmentation. In ElasticFusion, the reconstructed surfels are divided into ACTIVE and INACTIVE by time nodes. ACTIVE is the reconstructed surfels of the current frame, and INACTIVE is the reconstructed surfels of the previous frame. With the movement of the camera, the current frame pose T_{cur} of the current moment t_{cur} is calculated. At the position T_{inc} of the previous frame of time t_{ina}, the surfels are projected onto the plane, and the coordinates of the previous frame and the surfels of the current frame in the world coordinate system are calculated as $T_{cur}P(u, D^a_t)$ and $T_{ina}P(u, D^a_t)$, respectively, and the two frames are registered to establish the following constraint:

$$\begin{aligned} Q_p &= (T_{cur}P(u, D^a_t); T_{ina}P(u, D^a_t); t_{cur}; t_{ina}) \\ &= (Q^p_s; Q^p_d; t_s; t_d) \end{aligned} \tag{4}$$

where T_{inc} is the camera pose obtained by aligning the projected surfels of the ACTIVE frame of the camera system to the INACTIVE frame of the world system. The camera pose is obtained by setting the surfels point of the INACTIVE frame in the world system as the registration model, and setting the surfels point of the ACTIVE frame in the camera system as the frame to be registered. If the two point clouds overlap and there is a loop, the projected two point clouds with the overlapping points can be accurately registered and aligned, thereby providing an accurate dense map for the system.

3.2 The DS-Net architecture

Figure 4 shows the DS-Net network structure designed on the basis of DenseNet. In DS-Net, there are four dense blocks, eight deconvolutions, four unpoolings, and other operations, such as batch normalization (BN) [36] and rectified linear units (ReLU) [37]. Four B modules (B1, B2, B3, B4) and four D modules (D1, D2, D3, D4) transmit features in a cascading way. Each dashed box B consists of a pooling operator, a dense block, and a convolution layer; Each dashed box D includes an unpooling operator and two deconvolution layers. The pooling operators, the dense blocks, and the convolution layers in the B module are used to fully extracts both high-level and low-level features of the input image by the dense connections between dense blocks. The deconvo-

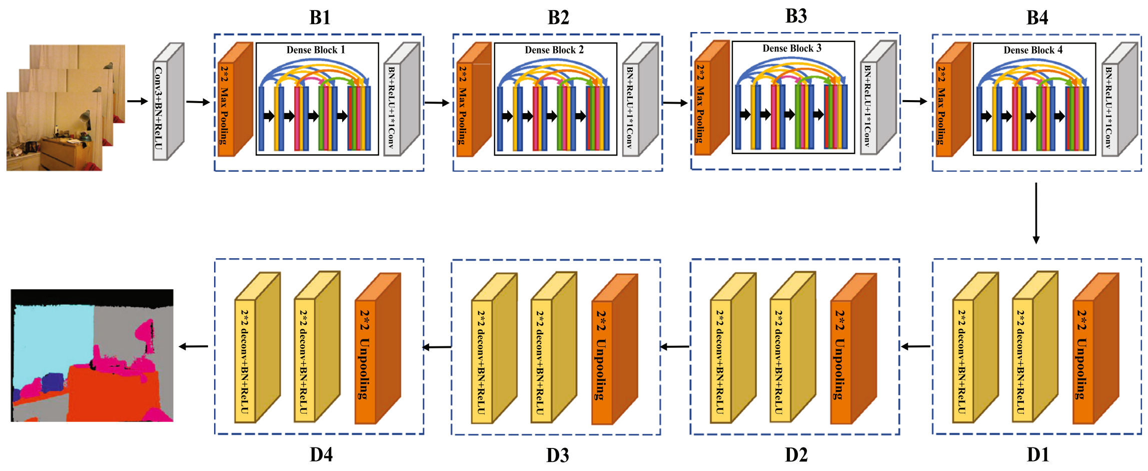

Fig. 4 The network architecture of DS-Net

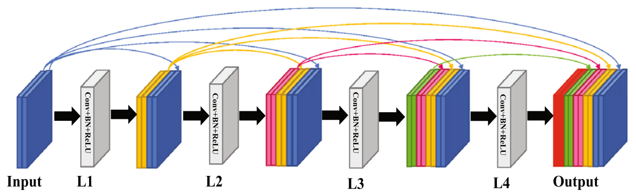

Fig. 5 A dense block with four convolution layers

lution and unpooling operations in the D module are used to recover high-quality images from previously extracted feature maps. Our method generates object segmentation masks using the deconvolution network, in which a dense pixel-wise class probability map is obtained by successive operations of unpooling, deconvolution, BN and ReLU.

3.2.1 Dense block

Figure 5 shows the network structure of a dense block we defined in DS-Net. Here, BN normalizes the corresponding feature map x to a specific distribution of y by using the formula $y=\gamma(x\text{-}\mu)/\delta+\beta$, where μ is the average of x, δ is the variance, γ is the scale factor, and β is the offset value.

In Fig. 5, the direct connection of different convolution layers in dense block is used to fully extract both the advanced and low-level features from the input image. In this process, the resolution of the image does not change. Any two adjacent layers in dense block are directly connected by multiple operations such as BN, ReLU and convolution layers. The 1×1 convolution layer in the B module is to decrease the large number of input feature maps caused by too many input features after concatenations, and improve the computational efficiency. In the B module, the linear growth in the number of features is compensated by the reduction in the spatial resolution of each feature map after the pooling operation. Therefore, each layer in dense block can not only benefit from both low-level and advanced functions in feedforward setting, but also reduce the risk of gradient explosion or disappearance.

In dense block, the feature maps obtained from all its previous layers and the map learned from the current layer are both input into the subsequent layer. This structure allows the gradients to be sent back to their respective places in the network more quickly. The number of network layers in dense block also has an important impact on the segmentation effect and the real-time performance of the network.

3.2.2 Deconvolution network

In the deconvolution network, images are reconstructed from the extracted features obtained from dense blocks. In Fig. 4, each dashed box D is a deconvolution network containing the deconvolution and unpooling modules. The deconvolution network upsamples the previous feature map, expands the image pixels, performs deconvolution, and obtains the weights. As the upsampling path increases the spatial resolution of feature map, the linear increase in the number of features brings a high memory requirement, especially for full resolution features in the pre-softmax layer. To reduce this limitation, the crossover structure with unpooling and deconvolution is used in the deconvolution network. We do not use the skip structure here because its performance gain is not too significant; instead, it will bring a lot of parameters. Therefore, in our network structure, the deconvolution is only connected to the last dense block, not to all previous dense blocks. The feature map that the last dense block outputs already contains the information from all previous dense blocks with different resolutions.

The pooling operation in convolution network is to filter noisy activations in the lower layers by abstracting activations in a receptive field with a single representative value. Although it contributes to classification by retaining only robust activations in the upper layers, spatial information within a receptive field will be lost during pooling, which will have a great influence on precise localization required for semantic segmentation. In our network, the pooling corresponds to the unpooling, and the network remembers the output positions of the pooling and determines the output unpooling locations. The unpooling layer outputs an enlarged but sparse activation map. The deconvolution layers densify the sparse activations in unpooling through the convolution-like operations with multiple learned filters and output an enlarged and dense activation map. Therefore, all available feature maps with given resolutions can be used to recover the final semantic segmentation map of the input image.

3.2.3 Network parameter selection

The parameter setting has an important influence on the performance of the network. A major difference between DenseNet and other CNNs is that the output of the network layer of DenseNet has a small number of feature maps, but CNNs has hundreds or even thousands of feature maps, including a large number of redundant feature maps. In order to obtain the number of feature maps suitable for our network, we conducted a comparative experiment on the networks with different feature map numbers in Sect. 4.1, and obtained the best number of feature maps K as shown in Table 1.

In Fig. 4, the B module is composed of one 2×2 MaxPooling, one dense block, followed by one BN, one ReLU, one 1×1 convolution, and one dropout with $p=0.2$. The D module consists of one 2×2 Unpooling, two 2×2 deconvolutions, two BNs, and two ReLUs. Table 1 shows the parameters of DS-Net, in which the convolution module is a 3×3 convolution layer with a step size of 2.

Table 1 Parameters of DS-Net

Layers	Parameters	Output size
Convolution	3×3 conv	224×224×16
B1	2×2 MaxPooling	112×112×16
	Dense Block1	112×112×80
B2	BN+ReLU+1×1 conv	112×112×16
	2×2 MaxPooling	56×56×16
	Dense Block2	56×56×80
B3	BN+ReLU+1×1 conv	56×56×16
	2×2 MaxPooling	28×28×16
	Dense Block3	28×28×80
B4	BN+ReLU+1×1 conv	28×28×16
	2×2 MaxPooling	14×14×16
	Dense Block4	14×14×80
D1	BN+ReLU+1×1 conv	14×14×16
	2×2 Unpooling	28×28×16
	2×2 deconv+BN+ReLU	28×28×16
D2	2×2 deconv+BN+ReLU	28×28×16
	2×2 Unpooling	56×56×16
	2×2 deconv+BN+ReLU	56×56×16
D3	2×2 deconv+BN+ReLU	56×56×16
	2×2 Unpooling	112×112×16
	2×2 deconv+BN+ReLU	112×112×16
D4	2×2 deconv+BN+ReLU	112×112×16
	2×2 Unpooling	224×224×16
	2×2 deconv+BN+ReLU	224×224×16
	2×2 deconv+BN+ReLU	224×224×14

In dense block, the concept of receptive fields, i.e., filter, is introduced. A large-sized filter will help the network extract a more abstract feature map, but it will cause the loss of information and the increase in the number of parameters. Generally, multiple small-sized filters have fewer parameters than a large-sized filter. For example, assuming that the input and output feature maps of the convolutional layer have the same size of C, if we use three 3×3 filters, there are a total of $3\times(3\times3\times C\times C) = 27CC$ parameters, while the number of parameters of a 7×7 filter is $49CC$. So, we use three 3×3 convolution layers in DS-Net, rather than a large-sized filter of 7×7 convolution layer.

3.3 Probability fusion

Due to the uncertainty of the environment and the sensor, the keyframe correspondence given by the SLAM system can use recursive Bayesian updates to update the labels of multiple keyframes. Camera pose can be used to establish the relationship between keyframes, realizing incremental fusion by stacking the semantic label information of keyframes with the spatial coordinates of the SLAM landmarks. However, a single 2D semantic segmentation will generate inconsistent labels between successive frames during camera movement. For the keyframe K_t of the camera video at time t, we define the category label distributed on the surfels S_n in the 3D map as l_i. If the spatial coordinates of S_n are not fused in the keyframe K_t, we use recursive Bayesian to update the probability of the corresponding category label as follows:

$$p(l_i|K_{1,\ldots,t}) = \frac{1}{Z_i} p(l_i|K_t) p(l_i|K_{1,\ldots,t-1}) \tag{5}$$

where Z is normalized constant. In order to obtain the probability of the corresponding category label on S_n given the corresponding keyframe, we define the pixel coordinates of S_n in the keyframe K_t as $x(S_n, t)$ and the coordinate matrix of the corresponding category label as $M_{x(S_n,t)}$, and after recursive Bayesian update we get the probability:

$$p(S_n \to l_i|K_{1,\ldots,t}) = \frac{1}{Z_i} p(M_{x(S_n,t)} = l_i|K_t) p(l_i|K_{1,\ldots,t-1}) \tag{6}$$

4 Experiments

In this section, three main experiments are conducted. First, as the encoder of the semantic segmentation system, our classification network is built based on DenseNet and tested on the CIFAR10 dataset. Second, because the amount of data with category labels in NYUv2 cannot meet the requirement to train our segmentation network, it is trained on the VOC2012 split dataset [38] and fine-tuned on the NYUv2 training set in the Caffe framework with 13 semantic classes defined by Couprie et al. [39]. Finally, our semantic segmentation system and SemanticFusion [2] are compared on NYUv2 and our laboratory environment to show the performance of fine-grained segmentation. The test sequences we selected include the bathroom_0003 scene, bedroom_0014 scene, bedroom_0003 scene, livingroom_0005 scene, and livingroom_0013 scene in NYUv2. In addition, numerical results for several segmentation networks are described to further prove the performance of our semantic system. All the experiments are performed on an Intel Core i7 3.3GHz CPU and Nvidia GTX 1060 GPU.

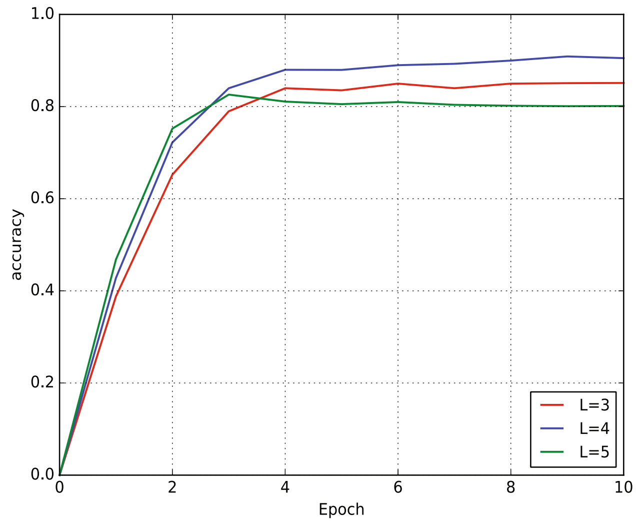

Fig. 6 Accuracy for different numbers of layers in dense block

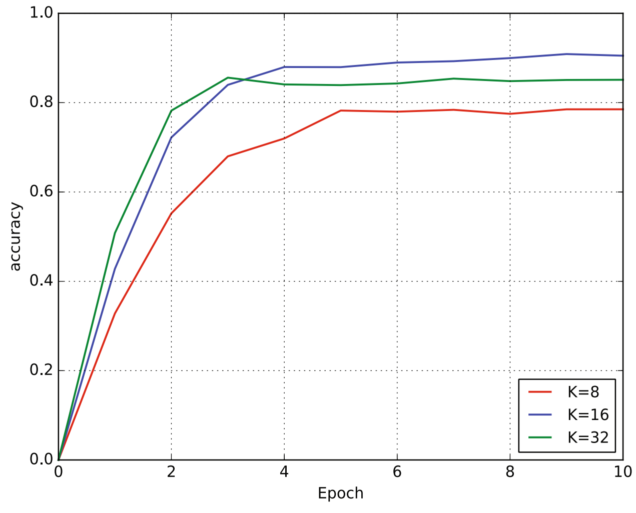

Fig. 7 Accuracy for different numbers of feature maps in dense block

4.1 Network parameter tuning

The parameters to be optimized in DS-Net include the number of dense blocks, the number of convolution layers in each dense block, and the number of feature maps in each dense block. Same as in DenseNet, the number of dense blocks is set to 4. So, the experiment is conducted to choose the number of layers in dense block and the number of feature maps in the convolution layer. The optimal network parameters are selected according to the experimental effects by trying different numbers. In experiment, the number of network layers L in dense block is set to 3, 4 and 5, respectively, and the number of feature maps K in dense block is set to 8, 16 and 32, respectively. The CIFAR10 dataset has a total of 60,000 color images of ten categories, with 6000 images each category. When building our dataset, we divided these 60,000 color images into 10 epochs by using k-fold cross validation. The numbers of different types of images in each epoch need not to be the same. We randomly disrupted the order of the training samples in order to prevent the occurrence of excessively regular data, for it may lead to overfitting or non-convergence and ultimately affect the accuracy of the network. Figure 6 shows the accuracy for different numbers of layers in dense block. Figure 7 shows the accuracy for different numbers of feature maps in dense block.

From Fig. 6, we can see that when L changes from 3 to 4, the prediction accuracy of the network is improved significantly with the increase in convergence speed. When L changes from 4 to 5, the network prediction accuracy of $L = 5$ increases in the early epochs. However, as the time increases, the prediction accuracy has no more improvement, even lower than the accuracy of $L = 3$. When $L = 5$, the convergence speed of the network becomes slower and more unstable than the other two curves. This is because more training samples are needed to avoid overfitting as the depth of the network increases. Moreover, the $L = 5$ network needs longer time to train more parameters, and so influences real-time calculation. Therefore, we choose $L = 4$ as the number of layers in dense block.

From Fig. 7, we find that the network can achieve the best results when $K = 16$. In dense block, the output of each nonlinear transform has K feature maps, and the input of the l^{th} layer is $K_0 + (l - 1) * K$. If we take the feature map as the global state of dense block, the training goal of each layer is to determine the updated value to be added to the global state. Thus, the number K of feature maps generated by each layer determines how much information each layer needs to update the global state. In the later stage of network training, there is very slightly decrease in accuracy. The reason is that the number of feature maps is not enough to achieve full extraction of image features. For $K = 32$, as the network width increases too much, the number of network parameters increases significantly, and the convergence becomes more difficult. Therefore, considering both accuracy and training time, we choose $K = 16$ as the optimal number of feature maps.

We compared the classification network with the best L and K obtained as above with the VGG-16-based network constructed by Noh et al. [16] on the CIFAR10 dataset. Figure 8 shows the accuracy of ours and Noh et al. It can be seen that our network can get higher accuracy than Noh et al. In the early four epochs, the convergence is very fast. After then, the network reaches a high and stable accuracy value. In general, real-time segmentation systems should have fast and accurate object detection ability.

Table 2 Performance on PASCAL VOC 2012 test set

Method	$L=3$	$L=4$	$L=5$	$K=8$	$K=16$	$K=32$	mIoU
$L=3, K=8$	√			√			78.8
$L=3, K=16$	√				√		79.5
$L=3, K=32$	√					√	78.3
$L=4, K=8$		√		√			82.1
$L=4, K=16$		√			√		83.2
$L=4, K=32$		√				√	81.6
$L=5, K=8$			√	√			80.7
$L=5, K=16$			√		√		81.4
$L=5, K=32$			√			√	80.9
PSPNet [42]	n/a						82.6
DeepLabv3 [43]	n/a						81.6
EncNet[44]	n/a						82.9
DFN[45]	n/a						82.7

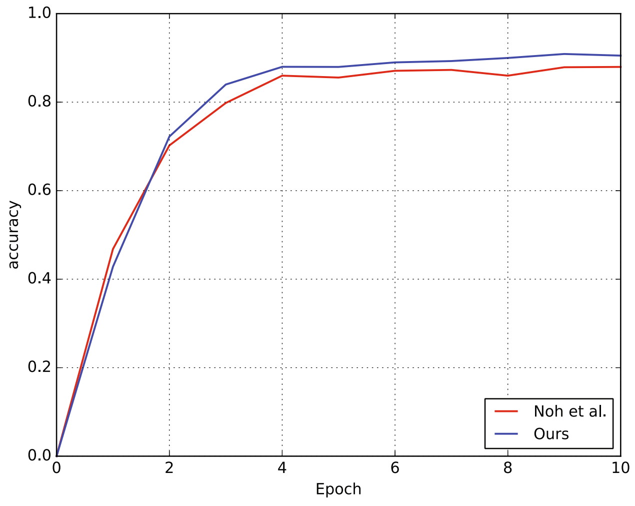

Fig. 8 Accuracy for ours and Noh et al. on the validation dataset

4.2 Semantic segmentation network

After establishing the classification network, the semantic segmentation network can be constructed. We trained it on the PASCAL VOC 2012 dataset and compared it with most current advanced segmentation networks. A total of 6929 objects in 1464 training images and 1449 verification images in PASCAL VOC 2012 can be used for segmentation tasks. And there are 1456 test images of variable sizes in the dataset. Since there are a large number of images without split labels in the PASCAL VOC 2012 dataset, we use the annotations in Hariharan et al. [40] to obtain 10,582 enhancement images for training. In the experiment, the 512×512 crop is used, and the learning rate of pretrained weights is divided by 8. The results are measured with the Intersection over Union (IoU) metric. For a given class c, predictions (o_i), and targets (y_i), the IoU is defined as

$$\mathrm{IoU}(c) = \frac{\sum_i (o_i == c \wedge y_i == c)}{\sum_i (o_i == c \vee y_i == c)} \tag{7}$$

where $\wedge$ is the logical AND operation, while $\vee$ is the logical OR operation. IoU is computed over all the pixels i of the dataset.

We compared our segmentation network DS-Net on VOC 2012 with the new state-of-the-art methods such as PSPNet [42], DeepLabV3 [43], EncNet [44] and DFN [45], which have not been pretrained on COCO [41]. Table 2 shows the comparative results, as well as those with regard to the number of layers L and the number of feature maps K in dense block. We can find that when L and K are 4 and 16, respectively, our segmentation network can get better segmentation accuracy, which is in accord with the classification results as shown in Figs. 6 and 7. The mean IoU (mIoU) values in regard to the number of layers L and the number of feature maps K illustrate the role of the classification module on the segmentation network. Under the same experimental conditions, due to the dense convolution in our classification network, the mIoU of our DS-Net segmentation network is already slightly higher than other networks.

4.3 Semantic mapping

In our semantic system, ElasticFusion is used to generate dense indoor scene maps by using RGB images and depth images. ElasticFusion calculates the pose through color consistency constraints on RGB image and the ICP algorithm on point clouds and then continuously optimizes and reconstructs 3D maps of environment through the surfels model. Inspired by Amiri's work [46], we save time stamp, position (x, y, z), and pose quaternion (qx, qy, qz, qw) of each frame

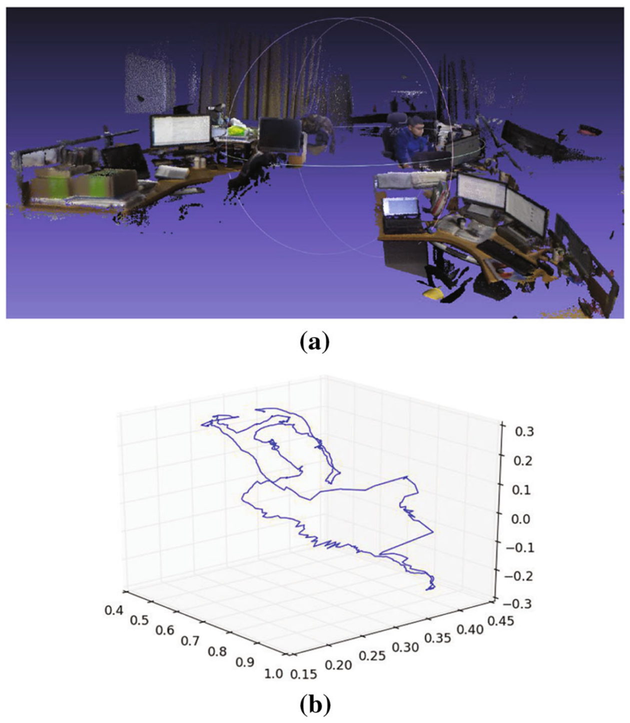

Fig. 9 **a** The dense map of our laboratory and **b** its corresponding camera trajectory estimation (m)

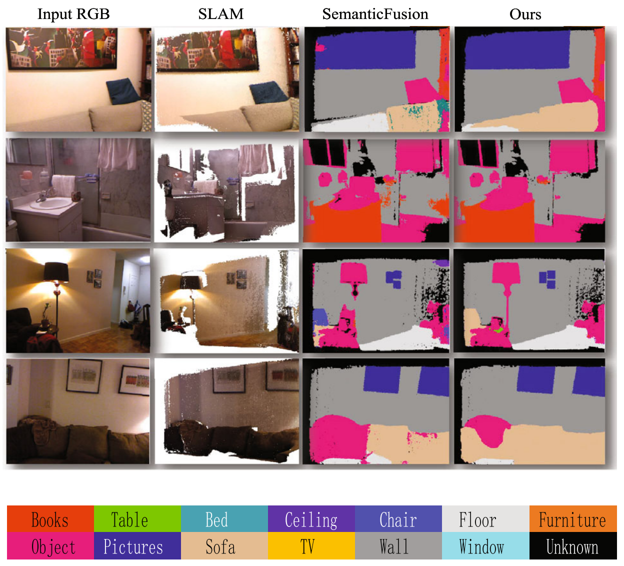

Fig. 10 The semantic segmentation results of our method with SemanticFusion on the NYUv2 dataset

while completing map construction of real scenes and plot the camera trajectory.

The experimental results are shown in Fig. 9. The upper is the real-time dense scene map about 20 square meters in our laboratory by using ElasticFusion and RGB-D camera, and the bottom is the camera trajectory plotted with the saved camera pose of each frame. The calculation of camera pose and the composition method based on the surfels model in ElasticFusion contribute to this accurate dense map in Fig. 9. For either large-sized items such as people and curtains or relatively small items such as display screens, boxes, and keyboards, they are all reconstructed well in the 3D dense map. It is evident that accurate trajectory estimation for camera output accounts well for accurate 3D dense maps. The above results provide a solidation for constructing good semantic map for our semantic fusion system.

After completing the construction of 3D dense maps, we can perform precise semantic segmentation in Caffe and compare our semantic fusion system with SemanticFusion on NYUv2. Figure 10 shows the results of single video frames. In the first column are the original RGB images, the second are dense maps of indoor environments constructed by ElasticFusion, the third are semantic segmentation maps obtained by SemanticFusion, and the last are semantic segmentation maps obtained by our method.

For the objects not appearing completely in video frames, for example, the rightmost pillow in the first scene, ElasticFusion can realize its reconstruction, but SemanticFusion has an erroneous probability prediction and generates the semantic map with segmentation error. Since the densely connected network in DS-Net fully extracts image edge features of objects, our system can generate more accurate probability maps and then get better results. For the more complex bathroom environment of the second scene, our system also has better performance in boundary segmentation of different objects and obtains a more satisfactory semantic map. For the third scene with strong light, which has a great influence on object detection, our system also has a good segmentation result. Strong light makes it difficult to detect the object from a certain angle of view. But we can get more probability maps for the same object from multiple views where it appears in the video frame sequence, and get more decisions for the final Bayesian update. For the dimly lit scene shown as the fourth scene, our system shows a better segmentation effect for similar objects than SemanticFusion. This is because our method can realize the reuse of high- and low-level features of different objects through dense connections. In contrast, for most sparse CNNs, some features may lose during their transmission process. We can also see that both our system and SemanticFusion failed to recognize the objects like glass. This shows that the ElasticFusion algorithm has some difficulty in achieving better reconstruction effect in this condition.

To fully verify the segmentation performance of our system, several more complex scenes in NYUv2 are chosen for experiment. Due to the depth loss and frame rate degradation in video sequences of the NYUv2 test set, we selected the video sequences with the camera sampling rates from 20 to 30 FPS for tracking and reconstruction. The depth map on the nearest time stamp is projected onto the RGB coordinate

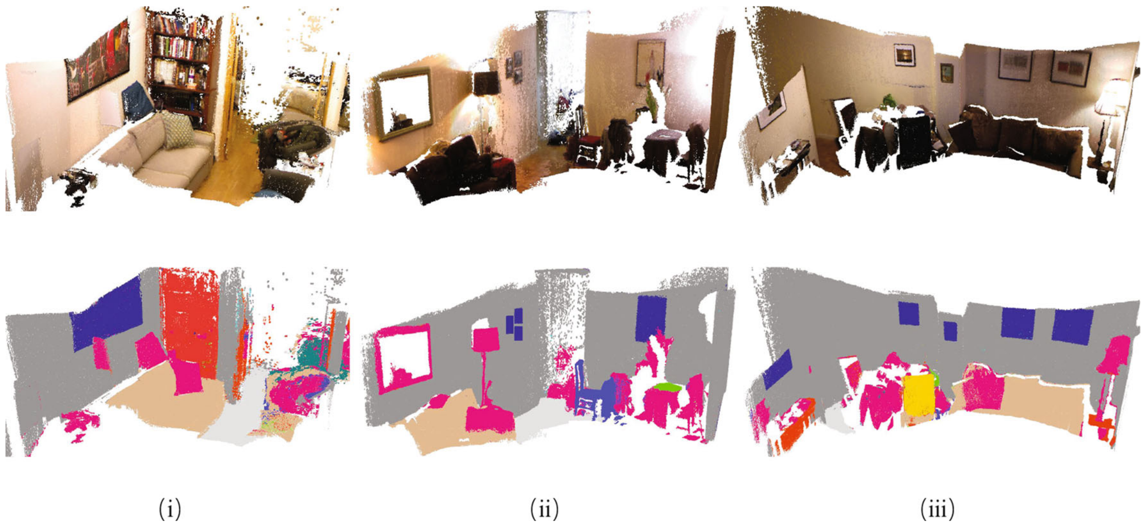

Fig. 11 The semantic maps by our system on the NYUv2 dataset

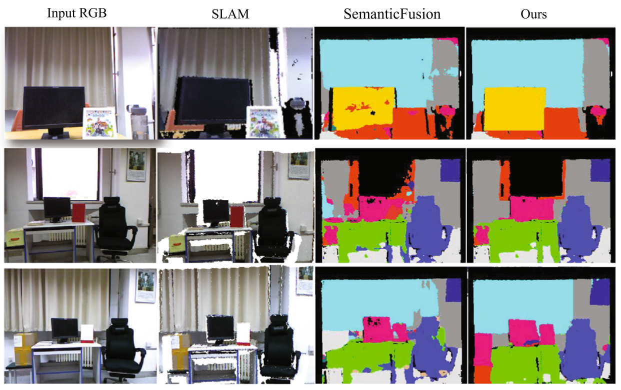

Fig. 12 The semantic maps of our laboratory environments

space to achieve image alignment and generate the video sequence of the scene.

Figure 11 shows dense maps (upper) and semantic maps (bottom) by our semantic system, respectively, of (i) the bedroom_0003 sequence, (ii) livingroom_0005 sequence and (iii) livingroom_0013 sequence in NYUv2. Since ElasticFusion can provide precise camera trajectory and real-time reconstruction, continuous frames in the visual odometry and loop detection can be used to establish the long-term correspondence between the landmarks and the dense map during the movement of the RGB-D camera. We can see that our fusion algorithm can build precise semantic segmentation maps for the scenes reconstructed by ElasticFusion. Large objects in the scenes, such as sofas, walls, and floors, have not been miss-segmented by our system, while small objects that continuously appear in the scenes are reconstructed more completely in our semantic maps. Moreover, our system realizes the real-time construction of indoor large-scale semantic maps on the public dataset using the laptop, validating its better online performance.

We evaluated the effectiveness of our method with the real scenes about 6 square meters in our laboratory with different distances from the RGB-D camera. Figure 12 shows the results of our fusion system and SemanticFusion. In the first column are the original RGB images, the second are dense SLAM maps constructed by ElasticFusion, the third are semantic segmentation maps by SemanticFusion, and the last are semantic segmentation maps obtained by our system.

We can see that different objects can be represented clearly in the semantic maps. In the first scene, our method shows a better segmentation effect for the intersection part of the curtain and the wall, as well as that of the display and the photo frame. In the second scene, our system also shows superior results for the table and chair with complex shapes. In the

Table 3 Information about test data

Test data	Frames	Surfels	Graph nodes	Local nodes
bedroom_0003	2265	1.2×10^6	98	8
livingroom_0005	1139	5.7×10^5	49	6
livingroom_0013	1195	6.1×10^5	52	5
lab	54	4.3×10^4	4	2
lab_map	605	4.8×10^5	39	3

third scene, although the measured objects are far from the camera, we can see that our system not only has a better effect on large objects such as window, table, chair, but also on small objects such as the photo frame on the table and the picture hanging on the wall. Owing to the sufficient extraction and feature reuse of object information in our network, continuous frames in visual odometry and loop detection can be used to establish the long-term correspondence between the landmarks and the dense map. So, our system has better robustness and better performance in feature extraction and edge detection for complex objects. However, we can also see in Fig. 12 that there are unrecognized parts in the semantic maps representing by black areas, showing that ElasticFusion used in our method cannot realize the real-time reconstruction for complex scenes.

In order to prove the generality of our semantic SLAM system, we list in Table 3 our information as detailed in [47] about our laboratory scenes and the NYUv2 data we used. Quantitative evaluations are performed on several systems including ours. In Table 3, bedroom_0003, livingroom_0005 and livingroom_0013 are the 3D space map information about NYUv2, and lab and lab_map as shown, respectively, in Figs. 9 and 12, are the information about the real environments of our laboratory.

All the information in Table 3 contains loop detection, including the laboratory semantic map of lab with only 54 frames. On the one hand, loop detection can eliminate accumulated construction errors, and continuous frames in loop detection can establish the long-term correspondence between the landmarks and the map. So, semantic predictions from multiple angles of RGB-D video frames can be integrated into a map to reduce the errors of dense maps and optimize the effect of semantic segmentation. When we construct a three-dimensional space map or a semantic map for a real environment, more surfels models are needed and more computing resources are required. This is why we can complete a large-scale semantic map construction for the public dataset, but only a part of real laboratory scene semantic map can be completed in real time.

In order to give a quantitative evaluation, we conducted a comparative experiment with different methods. From the NYUv2 dataset, we selected a video sequence with the frame rate higher than 2Hz as the test sequence. Table 4 shows the experimental results of different systems. RGBD-SF-CRF and RGBD-SF are the segmentation accuracy of each type of objects, respectively, with and without using CRF in SemanticFusion system. Yoshikatsu-Geometric-Only and Yoshikatsu are the accuracy of each type of objects, respectively, by geometric segmentation only and by both geometric and semantic segmentation.

Compared with the Hermans method, our method achieves 11.3% higher accuracy on average. Although our method is nearly the same as SemanticFusion in average accuracy, it

Table 4 Quantitative results on the NYUv2 dataset. The same strategy as in SemanticFusion is used in evaluation on the images with the same 320×240 resolution

Method	**Books**	**Table**	**Bed**	**Ceiling**	**Chair**	**Floor**	**Furniture**	**Object**	**Pictures**	**Sofa**	**TV**	**Wall**	**Window**	**Class avg.**
Hermans [4]	45.4	27.7	68.4	83.4	41.9	91.5	37.1	8.6	35.8	28.5	38.4	71.8	46.1	48.0
RGBD-SF[2]	58.8	34.0	61.7	43.4	58.4	92.6	63.7	59.1	66.4	47.3	33.9	86.0	60.5	58.9
RGBD-SF-CRF [2]	58.4	34.3	62.0	43.3	59.5	92.7	64.4	58.3	65.8	48.7	34.3	86.3	62.3	59.2
Yoshikatsu-Geometric-Only[15]	6.4	56.6	83.7	32.0	52.8	83.1	73.5	40.0	4.3	75.3	53.1	75.0	50.2	52.8
Yoshikatsu[15]	15.6	53.0	83.7	24.4	56.7	83.3	76.1	52.5	40.8	77.7	57.3	75.3	64.4	58.5
Ours	37.8	45.2	64.8	38.5	64.2	91.8	74.3	65.5	68.2	50.8	39.1	85.4	45.8	59.3

has better segmentation for some categories. Particularly, it has the best segmentation accuracy for the table class with an increase of 10.9%. For furnitures, chairs, and other objects, the segmentation accuracy increases by 9.9%, 7.2% and 4.7%, respectively. Compared with the Yoshikatsu method, our method achieves only 0.8% higher accuracy on average. For some shape-consistent objects, it has relatively poor segmentation accuracy. This is because the Yoshikatsu method relies more on the geometric information of objects, such as bed, furniture, sofa, and television. Their border information is closely related to their category information. However, for some objects with irregular shapes such as books, the Yoshikatsu method performs poorly, while our method achieves relatively better results on these categories. One important reason is that our network can obtain more details about small objects and complex objects. So, it can recover each type of objects by transferring these features during the upsampling phase.

5 Conclusions

In this paper, we propose an effective semantic segmentation method, in which the SLAM map and probabilistic map are fused to construct semantic map. We built the classification network based on DenseNet, the DS-Net segmentation network, and our final semantic mapping system. Experiments are conducted to evaluate the effectiveness of our method. For either single-frame RGB image or continuous video frames, our method shows its higher segmentation accuracy and effectiveness in semantic map construction. However, it also has some limitations. For example, some special objects like glass cannot be recognized, so the constructed semantic map cannot be fully labeled. It can also be seen that some reconstructed scenes are not too perfect, for these maps contain the areas that cannot be reconstructed in real time. Therefore, the next step we will take is to further improve its real-time performance and accuracy.

Acknowledgements This document is the results of the research projects funded by the National Natural Science Foundation of China (61873008) and Beijing Natural Science Foundation (4182008, 4192010).

References

1. Cadena C, Carlone L, Carrillo H et al (2016) Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. IEEE Trans Rob 32(6):1309–1332
2. McCormac J, Handa A, Davison A et al (2017) Semanticfusion: dense 3d semantic mapping with convolutional neural networks. In: IEEE international conference on robotics & automation (ICRA), pp 4628–4635
3. Yang S, Huang Y, Scherer S (2017) Semantic 3d occupancy mapping through efficient high order crfs. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 590–597
4. Hermans A, Floros G, Leibe B (2014) Dense 3d semantic mapping of indoor scenes from RGB-D images. In: IEEE international conference on robotics & automation (ICRA), pp 2631–2638
5. Henry P, Krainin M, Herbst E et al (2014) RGB-D mapping: Using depth cameras for dense 3d modeling of indoor environments. Int J Robot Res 31(5):647–663
6. Whelan T, Johannsson H, Kaess M et al (2013) Robust real-time visual odometry for dense RGB-D mapping. In: IEEE international conference on robotics & automation (ICRA), pp 5724–5731
7. Dai A, NieSSner M, Zollhöfer M et al (2017) Bundlefusion: real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. ACM Trans Gr 36(3):24
8. Whelan T, Salas-Moreno RF, Glocker B et al (2016) Elasticfusion: real-time dense slam and light source estimation. Int J Robot Res 35(14):1697–1716
9. Sunderhauf N, Pham TT, Latif Y et al (2017) Meaningful maps with object-oriented semantic mapping. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 5079–5085
10. Bowman SL, Atanasov N, Daniilidis K et al (2017) Probabilistic data association for semantic slam. In: IEEE international conference on robotics & automation (ICRA), pp 1722–1729
11. Huang G, Liu Z, Laurens VDM et al (2017) Densely connected convolutional networks. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 2261–2269
12. Silberman N, Hoiem D, Kohli P et al (2012) Indoor segmentation and support inference from RGBD images. In: European conference on computer vision (ECCV), pp 746–760
13. Vineet V, Miksik O, Lidegaard M et al (2015) Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In: IEEE international conference on robotics & automation, pp 75–82
14. Salas-Moreno RF, Newcombe RA, Strasdat H et al (2013) Slam++: simultaneous localisation and mapping at the level of objects. In: Computer vision pattern recognition (CVPR), pp 1352–1359
15. Nakajima Y, Tateno K, Tombari F et al (2018) Fast and accurate semantic mapping through geometric-based incremental segmentation. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 385–392
16. Noh H, Hong S, Han B (2015) Learning deconvolution network for semantic segmentation. In: IEEE international conference on computer vision (ICCV), pp 1520–1528
17. Hong S, Noh H, Han B (2015) Decoupled deep neural network for semi-supervised semantic segmentation. In: Advances in neural information processing systems, pp 1495–1503
18. Schuler CJ, Hirsch M, Harmeling S et al (2016) Learning to deblur. IEEE Trans Pattern Anal Mach Intell 38(7):1439–1451
19. Shelhamer E, Long J,Darrell T (2017) Fully convolutional networks for semantic segmentation. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 3431–3440
20. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations, pp 472–483
21. Chen LC, Papandreou G, Kokkinos I et al (2018) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
22. Zhao H, Qi X, Shen X et al (2018) Icnet for real-time semantic segmentation on high-resolution images. In: Proceedings of the European conference on computer vision (ECCV), pp 405–420
23. Badrinarayanan V, Kendall A, Segnet Cipolla R (2017) A deep convolutional encoder-decoder architecture for scene segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495

24. Kreo I, Krapac J, šegvić S (2019) Efficient ladder-style densenets for semantic segmentation of large images. arXiv preprint arXiv:1905.05661
25. Larsson G, Maire M, Shakhnarovich G. Fractalnet (2016) Ultra-deep neural networks without residuals. In: International conference on learning representations, pp 485–495
26. Srivastava RK, Greff K, Schmidhuber J (2015) Training very deep networks. In: Neural information processing systems (NIPS), pp 2377–2385
27. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
28. Lee CY, Xie S (2015) Gallagher P et al. Deeply-supervised nets. In: Artificial intelligence and statistics, pp 562–570
29. Zhang Z, Liang X, Dong X et al (2018) A sparse-view CT reconstruction method based on combination of densenet and deconvolution. IEEE Trans Med Imaging 37(6):1407–1417
30. Li T, Xu M, Yang R et al (2019) A densenet based approach for multi-frame in-loop filter in HEVC. In: Data compression conference, pp 270–279
31. Joaquim S, Matabosch C et al (2007) A review of recent range image registration methods with accuracy evaluation. Image Vis Comput 25(5):578–596
32. Keller M, Lefloch D, Lambers M, Izadi S, Weyrich T. Kolb A (2013) Real-time 3D Reconstruction in Dynamic Scenes using Point-based Fusion. In: Proceedings of joint 3DIM/3DPVT conference (3DV), pp 1–8
33. Whelan T, Kaess M, Johannsson H, Fallon MF, Leonard JJ, McDonald JB (2015) Real-time large scale dense RGB-D SLAM with volumetric fusion. IJRR 34(4–5):598–626
34. Woodham RJ et al (1992) Photometric method for determining surface orientation from multiple images. Opt Eng 19(1):139–144
35. Glocker J, Criminisi Shotton A, Izadi S (2015) Real-Time RGB-D camera relocalization via randomized ferns for Keyframe encoding. IEEE Trans Visual Comput Graphics 21(5):571–583
36. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on international conference on machine learning, pp 448–456
37. Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: International conference on international conference on machine learning, pp 807–814
38. Everingham M, Winn J (2006) The Pascal visual object classes challenge 2007 (voc2007) development kit. Int J Comput Vis 111(1):98–136
39. Couprie C, Farabet C, Najman L et al (2013) Indoor semantic segmentation using depth information. arXiv preprint arXiv:1301.3572
40. Hariharan, B et al (2011) Semantic contours from inverse detectors. In: IEEE international conference on computer vision, ICCV, Barcelona, Spain, November 6–13, pp 991–998
41. Lin T-Y, Maire M, Belongie S, et al (2014) Microsoft COCO: common objects in context. In: ECCV, pp 740–755
42. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: CVPR, pp 6230–6239
43. Chen L, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV, pp 833–851
44. Zhang H, Dana KJ, Shi J, Zhang Z, Wang X, Tyagi A, Agrawal A (2018) Context encoding for semantic segmentation. In: CVPR, pp 7151–7160
45. Yu C, Wang J, Peng C, Gao C, Yu G, Sang N (2018) Learning a discriminative feature network for semantic segmentation. In CVPR, pp 1857–1866
46. Amiri Atashgah MA, Malaek SMB et al (2012) Prediction of aerial-image motion blurs due to the flying vehicle dynamics and camera characteristics in a virtual environment. J Aerosp Eng 227(7):1055–1067
47. Amiri Atashgah MA, Gholampour P, Malaek SMB (2013) Integration of image de-blurring in an aerial Mono-SLAM. J Aerosp Eng 228(8):1348–1462