Spatial Information Guided Convolution for Real-Time RGBD Semantic Segmentation

Lin-Zhuo Chen, Zheng Lin, Ziqin Wang, Yong-Liang Yang, and Ming-Ming Cheng, *Senior Member, IEEE*

***Abstract*— 3D spatial information is known to be beneficial to the semantic segmentation task. Most existing methods take 3D spatial data as an additional input, leading to a two-stream segmentation network that processes RGB and 3D spatial information separately. This solution greatly increases the inference time and severely limits its scope for real-time applications. To solve this problem, we propose Spatial information guided Convolution (S-Conv), which allows efficient RGB feature and 3D spatial information integration. S-Conv is competent to infer the sampling offset of the convolution kernel guided by the 3D spatial information, helping the convolutional layer adjust the receptive field and adapt to geometric transformations. S-Conv also incorporates geometric information into the feature learning process by generating spatially adaptive convolutional weights. The capability of perceiving geometry is largely enhanced without much affecting the amount of parameters and computational cost. Based on S-Conv, we further design a semantic segmentation network, called Spatial information Guided convolutional Network (SGNet), resulting in real-time inference and state-of-the-art performance on NYUDv2 and SUNRGBD datasets.**

***Index Terms*— Spatial information, receptive field, RGBD semantic segmentation.**

I. Introduction

With the development of 3D sensing technologies, RGBD data with spatial information (depth, 3D coordinates) is easily accessible. As a result, RGBD semantic segmentation for high-level scene understanding becomes extremely important, benefiting a wide range of applications such as automatic driving [1], SLAM [2], and robotics. Due to the effectiveness of Convolutional Neural Network (CNN) and additional spatial information, recent advances demonstrate enhanced performance on indoor scene segmentation tasks [3]–[5]. Nevertheless, there remains a significant challenge caused by the complexity of the environment and the extra efforts for considering spatial data, especially for applications that require real-time inference.

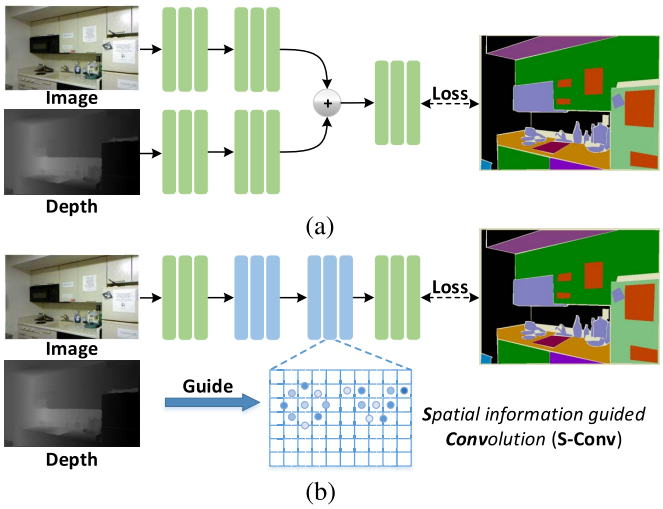

Fig. 1. The network architecture of different multi-modal fusion approaches. (a) The conventional two-stream structure [6]–[10]. (b) The proposed SGNet. It can be seen that the approach in (a) largely increases parameter number and inference time due to processing spatial information, thus less suitable for real-time applications. We replace the convolution with our S-Conv in (b) where the kernel distribution and weights of the convolution are adaptive to the spatial information. S-Conv greatly enhances the spatial awareness of the network with few additional parameters and computations, thus can efficiently utilize spatial information. Best viewed in color.

A common approach treats 3D spatial information as an additional input, followed by combining the features of RGB images to fuse multi-modal information [6]–[10] (see Fig. 1(a)). This approach achieves promising results at the cost of significantly increasing the parameter number and computational time, thus being unsuitable for real-time tasks. Meanwhile, several works [3], [6], [9], [11], [12] encode raw spatial information into three channels (HHA) composed of horizontal disparity, height above ground, and norm angle. However, the conversion from raw data to HHA is also time-consuming [9].

It is worth noting that indoor scenes have more complex spatial relations than outdoor scenes. This requires a stronger adaptive ability of the network to deal with geometric transformations. However, due to the fixed structure of the convolution kernel, the 2D convolution in the aforementioned methods cannot well adapt to spatial transformation and adjust the receptive field inherently, limiting the accuracy of semantic

Manuscript received April 16, 2020; revised October 15, 2020; accepted December 19, 2020. Date of publication January 22, 2021; date of current version January 28, 2021. This work was supported in part by the Major Project for New Generation of AI under Grant 2018AAA0100400, in part by the NSFC under Grant 61620106008, in part by the Tianjin Natural Science Foundation under Grant 17JCJQJC43700, and in part by the S&T Innovation Project from Chinese Ministry of Education. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Weiyao Lin. *(Corresponding author: Ming-Ming Cheng.)*

Lin-Zhuo Chen, Zheng Lin, and Ming-Ming Cheng are with the TKLNDST, College of Computer Science, Nankai University, Tianjin 300350, China (e-mail: linzhuochen@mail.nankai.edu.cn; cmm@nankai.edu.cn).

Ziqin Wang is with the Faculty of Medicine and Health, the University of Sydney, Sydney, NSW 2006, Australia.

Yong-Liang Yang is with the Department of Computer Science, University of Bath, Bath BA2 7AY, U.K. (e-mail: y.yang@cs.bath.ac.uk).

Digital Object Identifier 10.1109/TIP.2021.3049332

segmentation. Although alleviation can be made by revised pooling operation and prior data augmentation [13], [14], a better spatially adaptive sampling mechanism for conducting convolution is still desirable.

Moreover, the color and texture of objects in indoor scenes are not always representative [15]. Instead, the geometry structure often plays a vital role in semantic segmentation. For example, to recognize the fridge and wall, the geometric structure is the primary cue due to the similar texture. However, such spatial information is ignored by 2D convolution on RGB data. The depth-aware convolution [16] is proposed to address this problem. It forces pixels with similar depths as the center of the kernel to have higher weight than others. Nevertheless, this prior is handcrafted and may lead to sub-optimal results.

It can be seen that there is a contradiction between the fixed structure of 2D convolution and the varying spatial transformation, along with the efficiency bottleneck of separately processing RGB and spatial data. To overcome the limitations mentioned above, we propose a novel operation, called ***S**patial information guided **Conv**olution*(S-Conv), which adaptively changes according to the spatial information (see Fig. 1(b)). Specifically, this operation can generate convolution kernels with different sampling distributions adapting to spatial information, boosting the spatial adaptability and the receptive field regulation of the network. Furthermore, S-Conv establishes a link between the convolution weights and the underlying spatial relationship with their corresponding pixel, incorporating the geometric information into the convolution weights to better capture the spatial structure of the scene. Due to the input of spatial information in S-Conv, the scale and spatial transformation of objects can be directly analyzed to generate spatially adaptive offsets and weight.

The proposed S-Conv is light yet flexible and achieves significant performance improvements with only few additional parameters and computation costs, making it suitable for real-time applications. It can be seen as a novel and efficient method for multi-modal fusion task. Concretely, compared with other two-stream methods, we guide the convolution process by utilizing spatial information to achieve the purpose of multi-modal fusion. It performs better than other methods relying on two-stream network, and greatly reduces the amount of parameters and calculation compared with two-stream methods, enabling real-time application. We conduct extensive experiments to demonstrate the effectiveness and efficiency of S-Conv. We first design the ablation study and compare S-Conv with two-stream methods, deformable convolution [13], [14] and depth-aware convolution [16], exhibiting the advantages of S-Conv. We also verify the applicability of S-Conv to spatial transformations by testing its influence on different types of spatial data with depth, HHA and 3D coordinates. We demonstrate that spatial information is more suitable to generate offset than RGB feature which is used by deformable convolution [13], [14]. Finally, benefiting from the adaptability to spatial transformation and the effectiveness of perceiving spatial structure, our network equipped with S-Conv, named **S**patial information **G**uided convolutional **Net**work (SGNet), achieves high-quality results with real-time inference on NYUDv2 [17] and SUNRGBD [18], [19] datasets.

We highlight our contributions as follows:

- We propose a novel S-Conv operator that can adaptively adjust receptive field while effectively adapting to spatial transformation, and can perceive intricate geometric patterns with low cost.
- Based on S-Conv, we propose a new SGNet that achieves competitive RGBD segmentation performance in real-time on NYUDv2 [17] and SUNRGBD [18], [19] datasets.

II. Related Work

A. Semantic Segmentation

The recent advances of semantic segmentation benefit a lot from the development of convolutional neural network (CNN) [20], [21]. FCN [3] is the pioneer of leveraging CNN for semantic segmentation. It leads to convincing results and serves as the basic framework for many tasks. With the research efforts in the field, the recent methods can be classified into two categories according to the network architecture, including atrous convolution based methods [4], [22]–[24], and encoder-decoder based methods [25]–[30].

1) Atrous Convolution: The standard approach relies on stride convolutions or poolings to reduce the output stride of the CNN backbone and enables a large receptive field. However, the resolution of the resulting feature map is reduced [4], and many details are lost. One approach exploits atrous convolution to alleviate the conflict by enhancing the receptive field while keeping the resolution of the feature map [4], [22], [26], [31]. We use atrous convolution based backbone in the proposed SGNet.

2) Encoder-Decoder Architecture: The other approach utilizes the encoder-decoder structure [25]–[30], [32], which learns a decoder to recover the prediction details gradually. DeconvNet [28] employs a series of deconvolutional layers to produce a high-resolution prediction. SegNet [27] achieves better results by using pooling indices in the encoder to guide the recovery process in the decoder. RefineNet [25] fuses low-level features in the encoder with the decoder to refine the prediction. [29], [30] propose a scheme of gated sum, which can control the information flow of different scale in the encoder-decoder architecture. While this method can achieve more precise results, it requires longer inference time.

B. RGBD Semantic Segmentation

How to effectively use the extra geometry information (depth, 3D coordinates) is the key of RGBD semantic segmentation. A number of works focus on how to extract more information from geometry, which is treated as additional input in [7]–[10], [33]. Two-stream network is used in [6], [8]–[10], [12] to process RGB image and geometry information separately, and combines the two results in the last layer. These methods achieve promising results at the expense of doubling the parameters and computational cost. 3D CNNs or 3D KNN graph networks are also used to take

geometry information into account [34]–[36]. Besides, various deep learning methods on 3D point cloud [37]–[42] are also explored. However, these methods cost a lot of memory and are computationally expensive. Another stream incorporates geometric information into explicit operations. [43] proposes to perform 3D object detection based on depth-guided convolution, whose weights are location-variant and depth-adaptive. Cheng *et al.* [44] use geometry information to build a feature affinity matrix acting in average pooling and up-pooling. Lin *et al.* [45] splits the image into different branches based on geometry information. Wang and Neumann [16] propose Depth-aware CNN, which adds depth prior to the convolutional weights. Although it improves feature extraction by convolution, the prior is handcrafted but not learned from data. Other approaches, such as multi-task learning [7], [46]–[50] or spatial-temporal analysis [51], are further used to improve segmentation accuracy. The proposed S-Conv aims to efficiently utilize spatial information to improve the feature extraction ability. It can significantly enhance the performance with high efficiency due to using only a small amount of parameters.

C. Dynamic Structure in CNN

Using dynamic structure to deal with varying input of CNN has also been explored. Dilation Convolution is used in [4], [22] to increase the receptive field size without reducing feature map resolution. Spatial transformer network [52] adapts spatial transformation by warping feature map. Dynamic filter [53] adaptively changes its weights according to the input. Besides, self-attention based methods [54]–[57] generate attention maps from the intermediate feature map to adjust response at each location or capture long-range contextual information adaptively. Focusing on the understanding of contextual semantics, shape-variant convolution [57] confines its contextual region by location-variant convolution based on semantic-correlated region. Some generalizations of convolution from 2D image to 3D point cloud are also presented. PointCNN [42] is a seminal work that enables CNN on a set of unordered 3D points. There are other improvements [39]–[41] on utilizing neural networks to effectively extract deep features from 3D point sets. Deformable convolution [13], [14] can generate different distribution with adaptive weights. Nevertheless, their input is an intermediate feature map rather than spatial information. Our work experimentally verifies that better results can be obtained based on spatial information in Sec. IV.

III. S-Conv and SGNet

In this section, we first elaborate on the details of ***S**patial information guided **Conv**olution (S-Conv)*, which is a generalization of conventional RGB-based convolution by involving spatial information in the RGBD scenario. Then, we discuss the relation between our S-Conv and other approaches. Finally, we describe the network architecture of **S**patial information **G**uided convolutional **Net**work (SGNet), which is equipped with S-Conv for RGBD semantic segmentation.

A. Spatial Information Guided Convolution

For completeness, we first review the conventional convolution operation. We use $\mathbf{A}_i(\mathbf{j}), \mathbf{A} \in \mathbb{R}^{c\times h\times w}$ to denote a tensor, where i is the index corresponding to the first dimension, and $\mathbf{j} \in \mathbb{R}^2$ indicates the two indices for the second and third dimensions. Non-scalar values are highlighted in bold for convenience.

For an input feature map $\mathbf{F} \in \mathbb{R}^{c\times h\times w}$. We describe it in 2D for simplicity, thus we note $\mathbf{X}$ as input feature map. $\mathbf{X} \in \mathbb{R}^{1\times h\times w}$. Note that it is straightforward to extend to the 3D case. The conventional convolution applied on $\mathbf{X}$ to get $\mathbf{Y}$ can be formulated as the following:

$$\mathbf{Y}(\mathbf{p}) = \sum_{i=1}^{K} \mathbf{W}_i \cdot \mathbf{X}(\mathbf{p} + \mathbf{d}_i), \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^K$ represents the weight of convolution kernel with kernel size $k_h \times k_w$, and $K = k_h \times k_w$. $\mathbf{p} \in \mathbb{R}^2$ is the 2D convolution center, $\mathbf{d} \in \mathbb{R}^{K\times 2}$ denotes the kernel distribution around $\mathbf{p}$. For 3×3 convolution, d is defined as Equ. (2):

$$\mathbf{d} = \{[-1,-1],[-1,0],\ldots,[0,1],[1,1]\}. \tag{2}$$

From the above equation, we can see that the convolution kernel is constant over $\mathbf{X}$. In other words, $\mathbf{W}$ and $\mathbf{d}$ are fixed, meaning the convolution is location-invariant and spatially-agnostic.

In the RGBD context, we want to involve 3D spatial information efficiently by using adaptive convolution kernels. We first generate the offset according to the spatial information, then use the spatial information corresponding to the given offset to generate new spatially adaptive weights. Our S-Conv requires two inputs. One is the feature map $\mathbf{X}$ which is the same as conventional convolution. The other is the spatial information $\mathbf{S} \in \mathbb{R}^{c'\times h\times w}$. In practice, $\mathbf{S}$ can be HHA ($c' = 3$), 3D coordinates ($c' = 3$), or depth ($c' = 1$). The method of encoding depth into 3D coordinates and HHA is the same as [36]. Note that the input spatial information is not included in the feature map.

As the first step of S-Conv, we project the input spatial information into a high-dimensional feature space, which can be expressed as:

$$\mathbf{S}' = \phi(\mathbf{S}), \tag{3}$$

where ϕ is a spatial transformation function, and $\mathbf{S}' \in \mathbb{R}^{64\times h\times w}$, which has a higher dimension than $\mathbf{S}$.

Then, we take the transformed spatial information $\mathbf{S}'$ into consideration, perceive its geometric structure, and generate the distribution (offset of pixel coordinate in $x-$ and $y-$axis) of convolution kernels at different $\mathbf{p}$. This processes can be expressed as:

$$\Delta\mathbf{d} = \eta(\mathbf{S}'), \tag{4}$$

where $\Delta\mathbf{d} \in \mathbb{R}^{K\times h'\times w'\times 2}$, For the sake of simplicity, we do not show the reshaping process of $\Delta\mathbf{d}$ in Equ. (4). $\Delta\mathbf{d} \in \mathbb{R}^{2K\times h'\times w'}$ before reshaping. h', w' represent the feature map size after convolution. $K = k_h \times k_w$, which k_h and k_h are the kernel size. For 3×3 convolution, $\Delta\mathbf{d} \in \mathbb{R}^{9\times h'\times w'\times 2}$. η is a

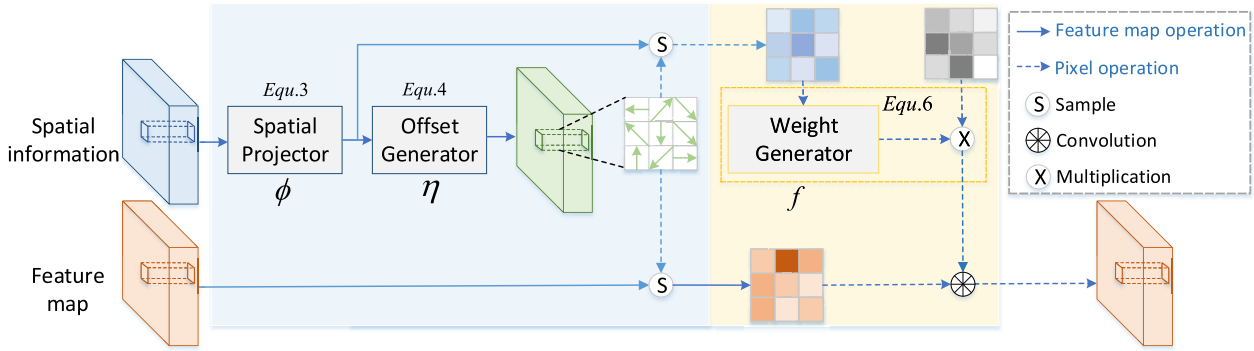

Fig. 2. The illustration of the Spatial information guided Convolution (S-Conv). Firstly, the input 3D spatial information is projected by the **spatial projector** to match the input feature map. Secondly, the adaptive convolution kernel distribution is generated by the **offset generator**. Finally, the projected spatial information is sampled according to the kernel distribution and fed into the **weight generator** to generate adaptive convolution weights.

non-linear function which can be implemented by a series of convolutions.

After generating the distribution of kernel for each possible $\mathbf{p}$ using $\Delta\mathbf{d}(\mathbf{p})$, we boost its feature extraction ability by establishing the link between the geometric structure and the convolution weight. Due to the shifting of convolution kernel in Equ. (4), the corresponding depth information of the convolution kernel has also changed. We want to collect the depth information corresponding to the convolution kernel after shifting for generating spatially adaptive weight. More specifically, we sample the geometric information of the pixels corresponding to the convolution kernel after shifting:

$$\mathbf{S}^*(\mathbf{p}) = \{\mathbf{S}'(\mathbf{p} + \mathbf{d}_i + \Delta\mathbf{d}_i(\mathbf{p}))|_{i=1,2,\ldots,K}\}, \tag{5}$$

where $\Delta\mathbf{d}(\mathbf{p})$ is the spatial distribution of convolution kernels at $\mathbf{p}$. $\mathbf{S}^*(\mathbf{p}) \in \mathbb{R}^{64K}$ is the spatial information corresponding to the feature map of the convolution kernel centered on $\mathbf{p}$ after transformation.

Finally, we generate convolution weights according to the final spatial information as the following:

$$\mathbf{W}^*(\mathbf{p}) = \sigma(f(\mathbf{S}^*(\mathbf{p}))) \cdot \mathbf{W}, \tag{6}$$

where f is a non-linear function that can be implemented as a series of fully connected layers with non-linear activation function, σ is sigmoid function, $\cdot$ is element-wise product, $\mathbf{W} \in \mathbb{R}^K$ indicates the convolution weights, which can be updated by the gradient descent algorithm. $\mathbf{W}^*(\mathbf{p}) \in \mathbb{R}^K$ denotes the spatially adaptive weights for convolution after shifting centered at $\mathbf{p}$.

Overall, our generalized S-Conv is formulated as:

$$\mathbf{Y}(\mathbf{p}) = \sum_{i=1}^{K} \mathbf{W}_i^*(\mathbf{p}) \cdot \mathbf{X}(\mathbf{p} + \mathbf{d}_i + \Delta\mathbf{d}_i(\mathbf{p})). \tag{7}$$

We can see that $\mathbf{W}_i^*(\mathbf{p})$ establishes the correlation between spatial information and convolution weights. Moreover, convolution kernel distribution is also relevant to the spatial information through $\Delta\mathbf{d}$. Note that $\mathbf{W}_i^*(\mathbf{p})$ and $\Delta\mathbf{d}_i(\mathbf{p})$ are not constant, meaning the generalized convolution is adaptive to different $\mathbf{p}$. Also, as $\Delta\mathbf{d}$ is typically fractional, we use bilinear interpolation to compute $\mathbf{X}(\mathbf{p} + \mathbf{d}_i + \Delta\mathbf{d}_i(\mathbf{p}))$ as in [13], [52]. The main formulae discussed above are labeled in Fig. 2.

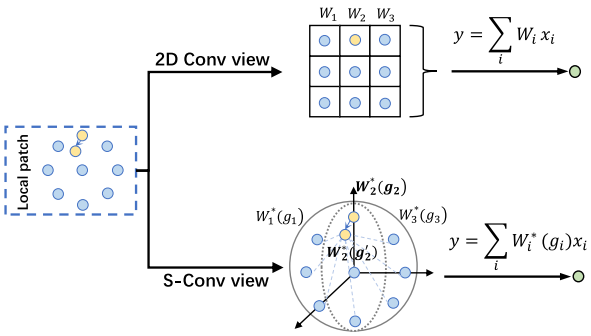

Fig. 3. The illustration of weights W in 2D convolution and W^* in S-Conv. The yellow dot indicates the point whose spatial position changes along the arrow. Illustration of 2D convolution is on the top, and S-Conv is on the bottom. The conventional 2D convolution operation orderly places local points in a regular grid with fixed weights, while ignoring the spatial information. We can see that the spatial position variation of the yellow point can not be reflected in the weight. Our S-Conv can be regarded as placing a local patch into a weight space, which is generated by the spatial guidance of that patch. Hence the weight of each point establishes a link with its spatial location, effectively capturing the spatial variation of the local patch. The spatial relationship between the yellow point and other points can be reflected in the adaptive weights.

B. Relation to Other Approaches

2D convolution is the special case of the proposed S-Conv without geometry information. Specifically, without geometry information, if we remove the $\mathbf{W}_i^*(\mathbf{p})$ and $\Delta\mathbf{d}_i(\mathbf{p})$ which are generated by geometry information in Equ. (7), this process degenerates to 2D convolution. While for the RGBD case, our S-Conv can extract feature at the point level and is not limited to the discrete grid by introducing spatially adaptive weights as shown in Fig. 3. Deformable convolution [13], [14] also alleviates this problem by generating different distribution weights. Nevertheless, their distributions are inferred from 2D feature maps instead of 3D spatial information as in our case. We will verify through experiments that our method achieves better results than deformable convolution [13], [14]. Compared with shape-variant (SV) convolution [57], SV convolution confines

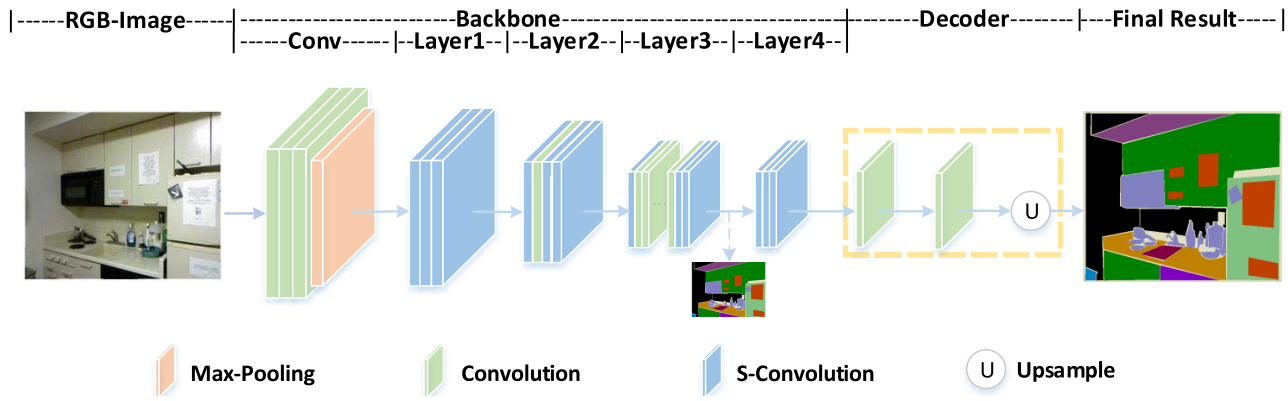

Fig. 4. The network architecture of SGNet equipped with S-Conv for RGBD semantic segmentation. The SGNet consists of a backbone network and a decoder. The deep supervision is added between layer 3 and layer 4 to improve network optimization.

TABLE I

THE RESULTS OF REPLACING CONVOLUTION (OF 3 × 3 FILTER) OF DIFFERENT LAYERS WITH S-CONV ON NYUDV2 DATASET. "LAYERX_Y" MEANS THE 3 × 3 CONVOLUTION OF Y-TH RESIDUAL BLOCK IN X-TH LAYER

S-Conv	layer3_0	layer3_1	layer3_2	layer3_20	layer3_21	layer3_22	other layers	mIoU(%)	param(M)	FPS
Baseline (ResNet101)								43.0	56.8	37
	✓							47.0	56.9	37
	✓	✓	✓					46.6	57.2	36
				✓	✓	✓		46.5	57.2	36
	✓				✓	✓		47.8	57.2	36
	✓				✓	✓	✓	**49.0**	58.3	28

its contextual region by location-variant convolution based on semantic-correlated region. It implements a location-variant convolution operator whose weights are location-variant and generated by feature map, focusing on the understanding of contextual semantics. Our S-Conv utilizes depth map rather than feature map to generate spatially adaptive offsets and weights. And the weights and offset of S-Conv are defined by spatial information (depth map). This helps the convolutional layer to adjust the receptive field and adapt to geometric transformation according to the spatial information. Compared with the 3D KNN graph-based method, our S-Conv selects neighboring pixels adaptively instead of using the KNN graph, which is not flexible and computationally expensive.

C. SGNet Architecture

Our semantic segmentation network, called SGNet, is equipped with S-Conv and consists of a backbone and decoder. The structure of SGNet is illustrated in Fig. 4. We use ResNet101 [58] as our backbone, and replace the first and the last two conventional convolutions (3 × 3 filter) of each layer with our S-Conv. We add a series of convolutions to extract the feature further and then use bilinear up-sampling to produce the final segmentation probability map, which corresponds to the decoder part of the SGNet. The ϕ in Equ. (3) is implemented as three 3 × 3 convolution layers, *i.e.* Conv(3, 64) - Conv(64, 64) - Conv(64, 64) with non-linear activation function. The η in Equ. (4) and the f in Equ. (6) are implemented as single convolution layer and two fully connected layers separately. The S-Conv implementation is modified from deformable convolution [13], [14]. We add deep supervision between layer 3 and layer 4 to improve the network optimization capability, which is the same as PSPNet [59].

IV. EXPERIMENTS

In this section, we first validate the performance of S-Conv by analyzing its usage in different layers; conducting ablation study/comparison with its alternatives; evaluating results of using different input information to generate offset; and testing inference speed. Then we compare our SGNet equipped with S-Conv with other state-of-the-art semantic segmentation methods on NYUDV2 and SUNRGBD datasets. Finally, we visualize the depth adaptive receptive field in each layer and the segmentation results, demonstrating that the proposed S-Conv can well exploit spatial information.

Datasets and Metrics: We evaluate the performance of S-Conv operator and SGNet segmentation method on public datasets:

- NYUDv2 [17]: This dataset has 1449 RGB images with corresponding depth maps and pixel-wise labels. 795 images are used for training, while 654 images are used for testing as in [17]. The 40-class settings are used for experiments.
- SUN-RGBD [18], [19]: This dataset contains 10335 RGBD images with semantic labels organized

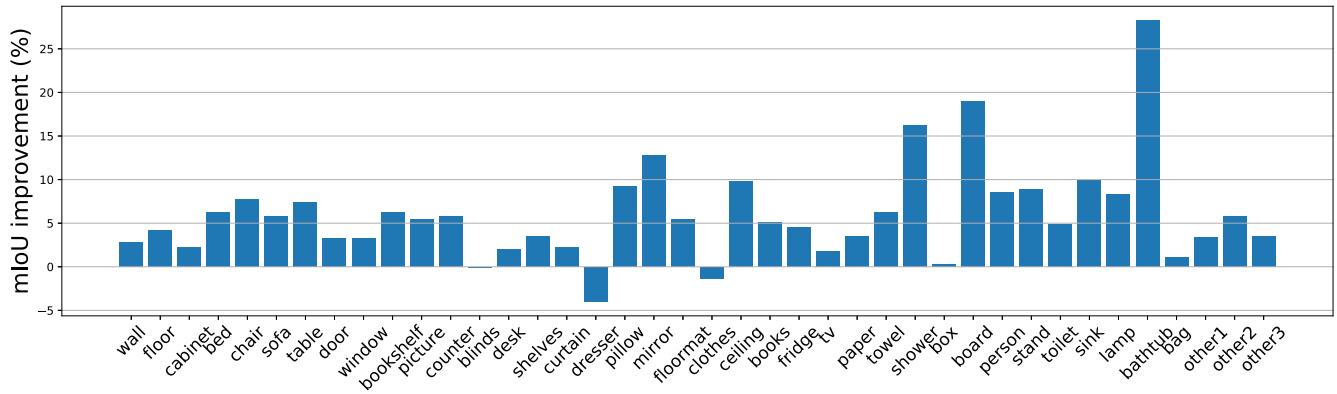

Fig. 5. Per-category IoU improvement of S-Conv on NYUDv2 dataset.

TABLE II

THE RESULTS OF REPLACING CONVOLUTION (OF 3 × 3 FILTER) OF DIFFERENT LAYERS WITH S-CONV ON NYUDV2 DATASET

layer1	layer2	layer3	layer4	mIoU(%)	param(M)
				43.0	56.8
✓				46.5	57.2
✓	✓			47.0	57.5
✓	✓	✓		48.8	57.9
✓	✓	✓	✓	**49.0**	58.3

TABLE III

ABLATION STUDY OF SGNET ON NYUDV2 [17] DATASET. OG: OFFSET GENERATOR OF S-CONV, WG: WEIGHT GENERATOR OF S-CONV, SP: SPATIAL PROJECTION OF S-CONV

Model	Acc	mAcc	mIoU
Baseline	72.1	54.6	43.0
Baseline+OG	73.9	58.2	46.3
Baseline+SP+OG	75.2	60.0	48.4
Baseline+SP+WG	74.5	58.4	46.8
Baseline+SP+OG+WG	**75.5**	**60.9**	**49.0**

in 37 categories. 5285 images are used for training, and 5050 images are used for testing.

- Cityscapes [60]: We split the dataset into training, validation and test. The training set, validation set, and test set contain 2975, 500, and 1525 images respectively.

We use three common metrics for evaluation, including pixel accuracy (Acc), mean accuracy (mAcc), and mean intersection over union (mIoU). The three metrics are defined as the following:

$$\begin{aligned} Acc &= \sum_i \frac{p_{ii}}{g}, \\ mAcc &= \frac{1}{p_c} \sum_i \frac{p_{ii}}{g_i}, \\ mIoU &= \frac{1}{p_c} \sum_i \frac{p_{ii}}{g_i + \sum_j p_{ji} - p_{ii}}, \end{aligned} \tag{8}$$

where p_{ij} is the amount of pixels which are predicted as class j with ground truth i, p_c is the number of classes, and g_i is the number of pixels whose ground truth class is i. $g = \sum_i g_i$ is the number of pixels. The depth map is used as the default format of spatial information unless specified otherwise.

Implementation Details: We use dilated ResNet101 [58] pretrained on ImageNet [61] as the backbone network for feature extraction following [4], and the output stride is 16 by default. The whole system is implemented based on PyTorch. The SGD optimizer is adopted for training with the same learning rate schedule ("poly" policy) as [4], [26], where the initial learning rate is 5e-3 for ablation study, 8e-3 for NYUDv2 and 1e-3 for SUNRGBD, and the weight decay is 5e-4. This learning policy updates the learning rate for every 40 epochs for NYUDv2 and ablation study and 10 epochs for SUNRGBD. We use ReLU activation function, and the batch size is 8. Following [6], we employ general data augmentation strategies, including random scaling, random cropping, and random flipping. The crop size is 480 × 640. During testing, we down-sample the image to the training crop size (480 × 640), and its prediction map is upsampled to the original size. We use cross-entropy loss in both datasets, and reweight [62] training loss of each class in SUNRGBD due to its extremely unbalanced label distribution. We train the network by 480 epochs for the NYUDv2 dataset and 200 epochs for the SUNRGBD dataset on two NVIDIA 1080Ti GPUs.

A. Analysis of S-Conv

We design ablation studies on NYUDv2 [17] dataset. The ResNet101 with a simple decoder and deep supervision is used as the baseline.

1) Replace Convolution With S-Conv: We evaluate the effectiveness of S-Conv by replacing the conventional convolution (of 3 × 3 filter) in different layers. We first replace convolution in layer 3, then extend the explored rules to other layers. The FPS (Frames per Second) is tested on NVIDIA 1080Ti with input image size 425 × 560 following [16]. The results are shown in Tab. I.

We can draw the following two conclusions from the results in the Tab. I. 1) The inference speed of the baseline network is fast, but its performance is poor. Replacing convolution

TABLE IV

THE COMPARISON RESULTS ON NYUDV2 TEST DATASET. DCV2: DEFORMABLE CONVOLUTION V2 [14], DAC: DEPTH-AWARE CONVOLUTION [16], SP: SPATIAL PROJECTOR IN S-CONV, WG: WEIGHT GENERATOR IN S-CONV

Model	Acc	mAcc	mIoU
Baseline	72.1	54.6	43.0
Baseline+DCV2	73.0	56.1	44.5
Baseline+HHANet	73.5	56.8	45.4
Baseline+DAC	73.8	57.1	45.4
Baseline+HHANet+DCV2	74.3	58.4	47.0
Baseline+DAC+DCV2	74.5	58.3	46.5
Baseline+SP+WG	74.5	58.4	46.8
Baseline+S-Conv(SGNet)	**75.5**	**60.9**	**49.0**

TABLE V

COMPARISON OF USING DIFFERENT TYPES OF SPATIAL INFORMATION ON NYUDV2 DATASET

Information	Acc	mAcc	mIoU
Depth	75.5	60.9	**49.0**
RGB Feature	73.9	58.5	46.4
HHA	**75.7**	60.8	48.9
Coordinates	75.3	**61.2**	48.5

TABLE VI

INFERENCE SPEED TEST OF SGNET ON NYUDV2 DATASET WITH INPUT IMAGE SIZE 480×640. OG: OFFSET GENERATOR OF S-CONV, †: WITHOUT APPLYING GENERATED LOCATION-VARIANT WEIGHT AND OFFSET IN SGNET, HHANET: ADDITIONAL STREAM BACKBONE (RESNET101) TO UTILIZE SPATIAL INFORMATION

Model	time(s)	FPS	param(M)
Baseline	0.029	34	56.8
Baseline+OG	0.030	33	57.7
Baseline+HHANet	0.053	18	99.4
SGNet(ResNet50)	0.028	36	39.3
SGNet†	0.032	31	58.3
SGNet	0.037	26	58.3

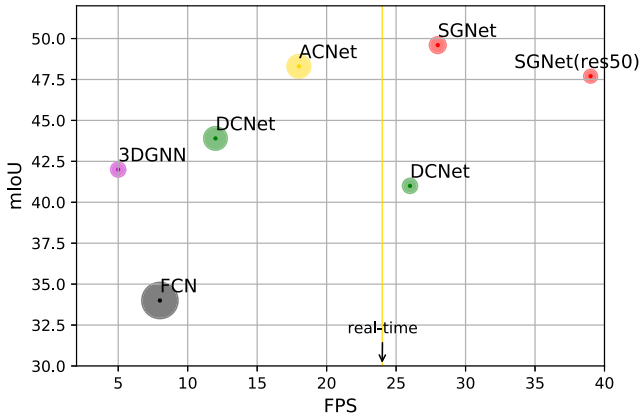

Fig. 6. FPS, mIoU, and the number of parameters of different methods on NYUDv2. The input image size for all single-scale speed comparisons is 425×560 following [16]. The radius of the circle corresponds to the number of parameters of the model. The results of DCNet [16] and 3DGNN [36] are from [16]. Our SGNet can achieve fastest inference time and state-of-the-art performance.

with S-Conv can improve the results of the baseline network with a little bit more parameters and computational time. 2) In addition to the first convolution in layer 3 whose stride is 2, the effect of replacing the later convolution is better. The main reason would be that spatial information can better guide down-sampling operation in the first convolution. Thus we choose to replace the first convolution and the last two convolutions of each layer with S-Conv. We generalize the rules found in layer 3 to other layers and achieve better results. The above experiments show that our S-Conv can significantly improve network performance with only a few parameters. It is worth noting that our network has no spatial information stream. The spatial information only affects the distribution and weight of convolution kernel. We also explore the performance of S-Conv embedded into different layers. The results are shown in Tab. II. We can observe that the performance enhances with the number of layers equipped with S-Conv.

We also show the IoU improvement of S-Conv on most categories in Fig. 5. It's obvious that our S-Conv improves IoU in most categories, especially for objects lacking representative texture information such as mirror, board and bathtub. There are also clear improvements for objects with rich spatial transformation, such as chairs and tables. This shows that our S-Conv can make good use of spatial information during the inference process.

2) Architecture Ablation: To evaluate the effectiveness of each component in our proposed S-Conv, we design ablation studies. The results are shown in Tab. III. By default, we replace the first convolution and the last two convolutions of each layer according to Tab. I. We can see that the offset generator, spatial projection module, and weight generator of S-Conv all contribute to the improvement of the results.

3) Comparison With Alternatives: Most methods [6], [9], [33], [62] use a two-stream network to extract features from two different modalities and then combine them. Our S-Conv focuses on advancing the feature extraction process of the network by utilizing spatial information. Here we compare our S-Conv with two-stream network, deformable convolution [13], [14], and depth-aware convolution [16]. We use a simple baseline which consists of a ResNet101 network with deep supervision and a simple decoder. We add an additional ResNet101 network, called HHANet, to extract HHA features and fuse it with our baseline features at the final layer of a two-stream network. To compare with depth-aware convolution and deformable convolution, similar to SGNet, we replace the first convolution and the last two convolutions of each layer. For "Baseline + DAC + DCV2", we replace convolution with depth-aware convolution [16] (DAC) in first two layers and replace convolution with deformable convolution [13] (DCV2) in last two layers, because DCV2 does not work for the lower layers [13]. The results are shown in Tab. IV. We find that our S-Conv achieves better results than two-stream networks, deformable convolution [13], depth-aware convo-

TABLE VII

COMPARISON RESULTS ON NYUDV2 TEST DATASET. MS: MULTI-SCALE TEST; SI: SPATIAL INFORMATION. THE INPUT IMAGE SIZE FOR FORWARD SPEED COMPARISON IS 425 × 560 USING NVIDIA 1080TI FOLLOWING [16]. WE ADD ASPP MODULE [4] AFTER THE FINAL LAYER OF SGNET, NOTED AS"SGNET*"

Network	Backbone	MS	SI	Acc	mAcc	mIoU	FPS	param (M)
FCN [3]	2×VGG16		HHA	65.4	46.1	34.0	8	272.2
LSD-GF [44]	2×VGG16		HHA	71.9	60.7	45.9	-	-
3DGNN [36]	VGG16		HHA	-	55.2	42.0	5	47.2
D-CNN [16]	VGG16		Depth	-	53.6	41.0	26	47.0
D-CNN [16]	2×ResNet152		Depth	-	61.1	48.4	-	-
ACNet [33]	2×ResNet50		Depth	-	-	48.3	18	116.6
RefineNet [25]	ResNet152	✓	-	73.6	58.9	46.5	16	129.5
RDFNet [6]	2×ResNet152	✓	HHA	76.0	62.8	50.1	9	200.1
RDFNet [6]	2×ResNet101	✓	HHA	75.6	62.2	49.1	11	169.1
CFNet [45]	2×ResNet152	✓	HHA	-	-	47.7	-	-
SGNet	ResNet50		Depth	75.0	60.9	47.7	**39**	**39.3**
SGNet	ResNet101		Depth	75.6	61.9	49.6	28	58.3
SGNet*	ResNet101		Depth	76.1	62.7	50.2	26	64.7
SGNet*	ResNet101	✓	Depth	**76.8**	**63.3**	**51.1**	26	64.7

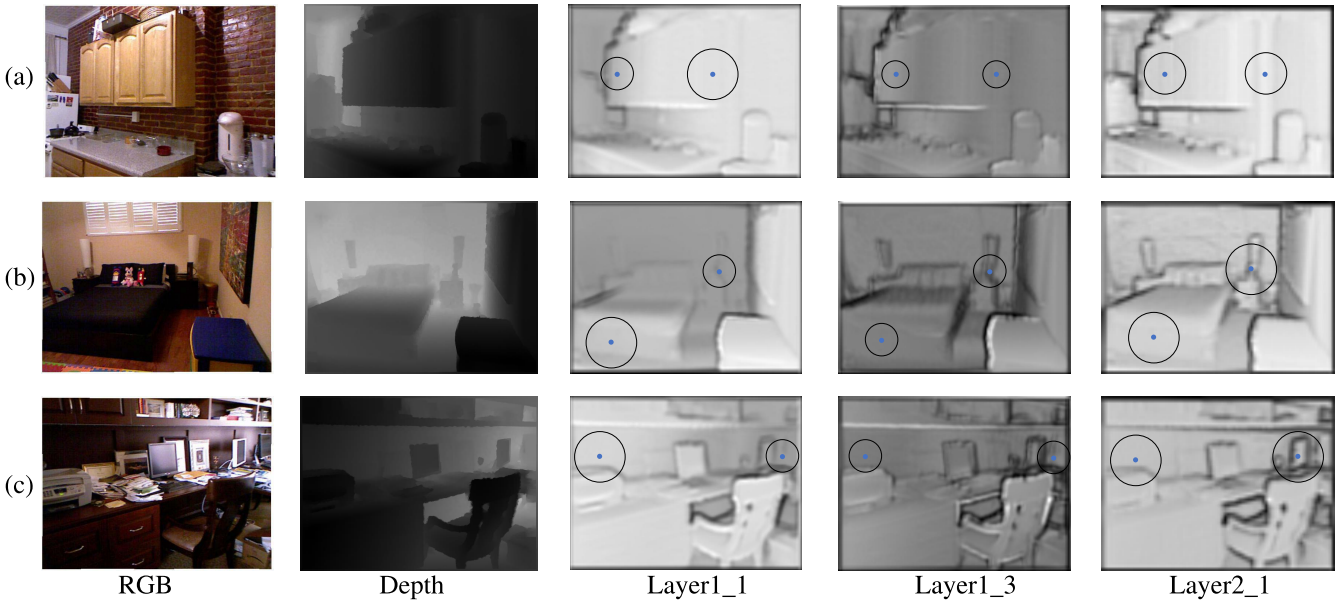

Fig. 7. The visualization of relative receptive field in S-Conv.

lution [16], and their combination. This demonstrates that our S-Conv can effectively utilizes spatial information. The baseline equipped with weight generator can also achieve better results than depth-aware convolution, indicating that learning weights from spatial information is necessary.

4) Spatial Information Comparison: We also evaluate the impact of different formats of spatial information on S-Conv. The results are shown in Tab. V. We can see that depth information leads to comparable results with HHA and 3D coordinates, and better results than intermediate RGB features which are used by deformable convolution [13], [14]. This shows the advantage of using spatial information for offset and weight generation over RGB features. However, converting depth to HHA is time-consuming [9]. Hence 3D coordinates and depth map are more suitable for real-time segmentation using SGNet. It can be seen that even without spatial information input (with RGB features), our S-Conv has more than 3.4% improvement than the baseline.

5) Inference Speed Test: To demonstrate the light weight of S-Conv, we test the inference speed of SGNet in this part. We also compare our S-Conv with two-stream methods. The input size of image is 480×640. Results are shown in Tab. VI. We can observe that S-Conv only requires a small amount of additional computation compared with two-stream methods. Our SGNet can also achieve real-time inference speed using ResNet101 and ResNet50 [58] backbone.

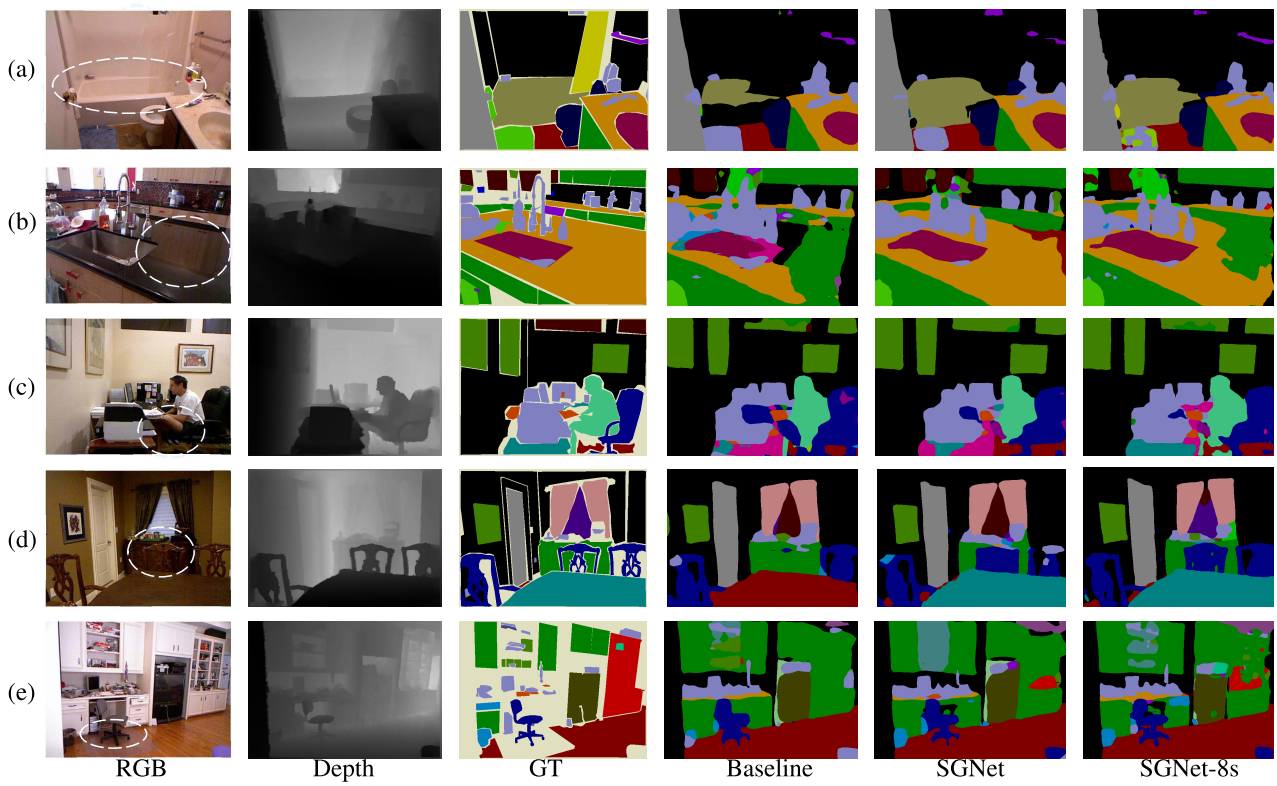

Fig. 8. The qualitative semantic segmentation comparison results on NYUDv2 test dataset. SGNet-8s: output stride is 8.

TABLE VIII
COMPARISON RESULTS ON SUNRGBD TEST DATASET. MS: MULTI-SCALE TEST, SI: SPATIAL INFORMATION. WE ADD ASPP MODULE [4] AFTER THE FINAL LAYER OF SGNET, NOTED AS "SGNET*"

Network	Backbone	MS	SI	Acc	mAcc	mIoU	param (M)
LSD-GF [44]	2×VGG16		HHA	-	58.0	-	-
RefineNet [25]	ResNet152	✓	-	80.6	58.5	45.9	129.5
CGBNet [30]	ResNet101		-	82.3	61.3	48.2	-
3DGNN [36]	VGG16	✓	HHA	-	57.0	45.9	47.2
D-CNN [16]	2×VGG16		HHA	-	53.5	42.0	92.0
ACNet [33]	2×ResNet50		HHA	-	-	48.1	272.2
RDFNet [6]	2×ResNet152	✓	HHA	81.5	60.1	47.7	200.1
CFNet [45]	2×ResNet152	✓	HHA	-	-	48.1	-
SGNet	ResNet101		Depth	81.0	59.6	47.1	58.3
SGNet*	ResNet101		Depth	81.0	59.8	47.5	64.7
SGNet*	ResNet101	✓	Depth	82.0	60.7	**48.6**	64.7

B. Comparison With state-of-the-art

We compare our SGNet with other state-of-the-art methods on NYUDv2 [17] and SUNRGBD [18], [19] datasets. The architecture of SGNet is shown in Fig. 4.

1) NYUDv2 Dataset: The comparison results can be found in Tab. VII and Fig. 6. We change the learning rate from 5e-3 to 8e-3. We down-sample the input image to 480 × 640 and upsample its predict map to get final results during test. To compare inference speed with other methods, the input image size for all single-scale speed comparisons in Tab. VII is 425 × 560 following [16]. The inference speed results of DCNet [16] and 3DGNN [36] are from [16]. We tested the single-scale speed of other methods under the same conditions using NVIDIA 1080Ti. Furthermore, inference speed test of SGNet with input size 480×640 is shown in Tab. VI. Note that some methods in Tab. VII do not report parameter quantities or open source. So we just listed the mIoU of these methods. We can draw the following conclusions from Tab. VI and Tab. VII. Instead of using additional networks to extract spatial features, our SGNet (ResNet50) can achieve competitive performance and fastest inference with minimum number of parameters. Our SGNet (ResNet101) can achieve more competitive performance and real-time inference. This benefits from S-Conv which can make use of spatial information

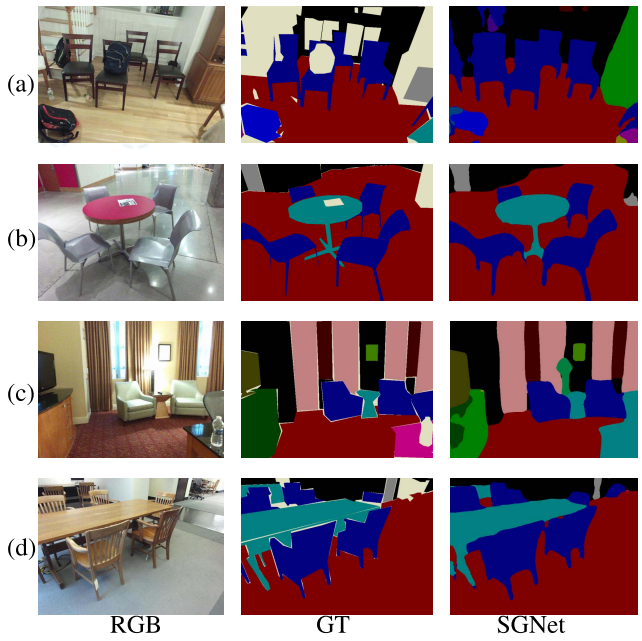

Fig. 9. The qualitative semantic segmentation comparison results on SUNRGBD test dataset.

efficiently with only a small amount of extra parameters and computation cost. Moreover, our S-Conv can achieve good results without using HHA information, making it suitable for real-time tasks. This verifies the efficiency of our S-Conv in utilizing spatial information. At the expense of a little bit more reasoning time by adding ASPP module [4] after SGNet noted as SGNet*, the proposed SGNet can achieve better results than other methods and RDFNet which uses multi-scale test, HHA information and two ResNet152 backbones. After using multi-scale test which is used by other methods, SGNet's performance can be further improved.

2) SUNRGBD Dataset: The comparison results on the SUNRGBD dataset are shown in Tab. VIII. It is worth noting that some methods in Tab. VII did not report results on the SUNRGBD dataset. The inference time and parameter number of models in Tab. VIII are the same as those in Tab. VII. Our SGNet can achieve state-of-the-art results in real-time compared with models that do not have real-time performance.

3) Cityscapes Dataset: We add ASPP [4] module after SGNet and set $output\ stride = 8$. We training with 2975 images on training set for validation. We also provide our test result on Cityscapes server. The comparison results on the Cityscapes dataset are shown in Tab. IX. It is worth noting that due to the serious noise of depth map in Cityscapes, most of previous RGB-D based methods perform worse than RGB based methods. We can observe that our network benefiting from S-Conv can achieve better results than baseline and achieve competitive results on Cityscapes.

C. Qualitative Performance

1) Visualization of Receptive Field in S-Conv: Appropriate receptive field is very important for scene recognition. We visualize the input adaptive receptive field of SGNet in different layers generated by S-Conv. Specifically, we get the receptive field of each pixel by summing up the norm of their offsets during the S-Conv operation, then we normalize each value to [0, 255] and visualize the result using a gray-scale image. The results are shown in Fig. 7. The brighter the pixel, the larger the relative receptive field. We also use the radius of circle to represent the size of the relative receptive field. We observe that the receptive fields of different convolutions vary adaptively with the depth of the input image. For example, in layer1_1, the receptive field is inversely proportional to the depth. The combination of the adaptive receptive field learned at each layer can help the network better resolve indoor scenes with complex spatial relations.

TABLE IX
COMPARISON RESULTS ON CITYSCAPES VALIDATION DATASET. ‡: RESULTS ON TEST DATASET

Network	Backbone	iterations	MS	mIoU
Baseline	ResNet101	40k		78.2
SGNet	ResNet101	40k		79.2
SGNet	ResNet101	65k	✓	80.6
SGNet	ResNet101	65k	✓	81.2‡

2) Qualitative Comparison Results: We show qualitative comparison results on NYUDv2 test dataset in Fig. 8. For the visual results in Fig. 8(a), the bathtub and the wall have insufficient texture, which cannot be easily distinguished by the baseline method. Some objects may have reflections such as the table in Fig. 8(b), which is also challenging for the baseline. SGNet, however, can recognize it well by incorporating spatial information with the help of S-Conv. The chairs in Fig. 8(c, d) are hard to be recognized by RGB data due to the low contrast and confused texture, while they can be easily recovered by SGNet benefiting from the equipped S-Conv. In the meantime, SGNet can recover the object's geometric shape nicely, as demonstrated by the chairs of Fig. 8(e). We also show qualitative results on SUNRGBD test dataset in Fig. 9. It can be seen that our SGNet can also achieve precise segmentation on SUNRGBD.

V. CONCLUSION

In this paper, we propose a novel ***S**patial information guided **Conv**olution (S-Conv)* operator. Compared with conventional 2D convolution, it can adaptively adjust the convolution weights and distributions according to the input spatial information, resulting in better awareness of the geometric structure with only a few additional parameters and computation cost. We also propose **S**patial information **G**uided convolutional **Net**work (SGNet) equipped with S-Conv that yields real-time inference speed and achieves competitive results on NYUDv2 and SUNRGBD datasets for RGBD semantic segmentation. We also compare the performance of using different inputs to generate offset, demonstrating the advantage of using spatial information over RGB feature. Furthermore, we visualize the depth-adaptive receptive field in each layer to show effectiveness. In the future, we will investigate the fusion

of different modal information and the adaptive change of S-Conv structure simultaneously, making these two approaches benefit each other. We will also explore the application of S-Conv in different fields, such as pose estimation and 3D object detection.

REFERENCES

[1] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 405–420.

[2] B. Bescos, J. M. Facil, J. Civera, and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[5] Q. Hou, L. Han, and M.-M. Cheng, "Autonomous learning of semantic segmentation from Internet images," (in Chinese), *Sci Sinica Inf.*, 2021.

[6] S. Lee, S.-J. Park, and K.-S. Hong, "RDFNet: RGB-D multi-level residual feature fusion for indoor semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4980–4989.

[7] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2650–2658.

[8] L. Ma, J. Stuckler, C. Kerl, and D. Cremers, "Multi-view deep learning for consistent semantic mapping with RGB-D cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 598–605.

[9] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "FuseNet: Incorporating depth into semantic segmentation via fusion-based CNN architecture," in *Proc. Asian Conf. Comput. Vis.*, 2016, pp. 213–228.

[10] J. Wang, Z. Wang, D. Tao, S. See, and G. Wang, "Learning common and specific features for RGB-D semantic segmentation with deconvolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 664–679.

[11] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 345–360.

[12] Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, and L. Lin, "LSTM-CF: Unifying context modeling and fusion with LSTMs for RGB-D scene labeling," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 541–557.

[13] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[14] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9308–9316.

[15] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li, "Salient object detection: A survey," *Comput. Vis. Media*, vol. 5, no. 2, pp. 117–150, Jun. 2019.

[16] W. Wang and U. Neumann, "Depth-aware CNN for RGB-D segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 135–150.

[17] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 746–760.

[18] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 567–576.

[19] A. Janoch *et al.*, "A category-level 3D object dataset: Putting the Kinect to work," in *Consumer Depth Cameras for Computer Vision*. London, U.K.: Springer, 2013, pp. 141–165.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inform. Process. Syst.*, 2012, pp. 1097–1105.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[22] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–13.

[23] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-aware feature propagation for scene segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6819–6829.

[24] B. Shuai, H. Ding, T. Liu, G. Wang, and X. Jiang, "Toward achieving robust low-level and high-level scene parsing," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1378–1390, Mar. 2019.

[25] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.

[26] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 801–818.

[27] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[28] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.

[29] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang, "Context contrasted feature and gated multi-scale aggregation for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2393–2402.

[30] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang, "Semantic segmentation with context encoding and multi-path decoding," *IEEE Trans. Image Process.*, vol. 29, pp. 3520–3533, 2020.

[31] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.

[32] R. Fan, M.-M. Cheng, Q. Hou, T.-J. Mu, J. Wang, and S.-M. Hu, "S4Net: Single stage salient-instance segmentation," *Comput. Vis. Media*, vol. 6, no. 2, pp. 191–204, Jun. 2020.

[33] X. Hu, K. Yang, L. Fei, and K. Wang, "ACNET: Attention based network to exploit complementary features for RGBD semantic segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 1440–1444.

[34] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1746–1754.

[35] S. Song and J. Xiao, "Deep sliding shapes for amodal 3D object detection in RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 808–816.

[36] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3D graph neural networks for RGBD semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5199–5208.

[37] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[38] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inform. Process. Syst.*, 2017, pp. 5099–5108.

[39] L.-Z. Chen, X.-Y. Li, D.-P. Fan, K. Wang, S.-P. Lu, and M.-M. Cheng, "LSANet: Feature learning on point sets by local spatial aware layer," 2019, *arXiv:1905.05442*. [Online]. Available: http://arxiv.org/abs/1905.05442

[40] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 87–102.

[41] C. Wang, B. Samari, and K. Siddiqi, "Local spectral graph convolution for point set feature learning," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 52–66.

[42] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on x-transformed points," in *Proc. Adv. Neural Inform. Process. Syst.*, 2018, pp. 828–838.

[43] M. Ng *et al.*, "Learning depth-guided convolutions for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1000–1001.

[44] Y. Cheng, R. Cai, Z. Li, X. Zhao, and K. Huang, "Locality-sensitive deconvolution networks with gated fusion for RGB-D indoor semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3029–3037.

[45] D. Lin, G. Chen, D. Cohen-Or, P.-A. Heng, and H. Huang, "Cascaded feature network for semantic segmentation of RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1311–1319.

[46] J. Jiao, Y. Wei, Z. Jie, H. Shi, R. Lau, and T. S. Huang, "Geometry-aware distillation for indoor semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2869–2878.
[47] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille, "Towards unified depth and semantic prediction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2800–2809.
[48] J. Hoffman, S. Gupta, and T. Darrell, "Learning with side information through modality hallucination," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 826–834.
[49] I. Kokkinos, "UberNet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6129–6138.
[50] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang, "Pattern-affinitive propagation across depth, surface normal and semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4106–4115.
[51] Y. He, W.-C. Chiu, M. Keuper, and M. Fritz, "STD2P: RGBD semantic segmentation using spatio-temporal data-driven pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4837–4846.
[52] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Proc. Adv. Neural Inform. Process. Syst.*, 2015, pp. 2017–2025.
[53] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Proc. Adv. Neural Inform. Process. Syst.*, 2016, pp. 667–675.
[54] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.
[55] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.
[56] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[57] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang, "Semantic correlation promoted shape-variant context for segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8885–8894.
[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[59] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.
[60] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.
[61] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[62] J. Jiang, L. Zheng, F. Luo, and Z. Zhang, "RedNet: Residual encoder-decoder network for indoor RGB-D semantic segmentation," 2018, *arXiv:1806.01054*. [Online]. Available: http://arxiv.org/abs/1806.01054

Lin-Zhuo Chen is currently pursuing the master's degree with the College of Computer Science, Nankai University. His research interests include computer vision and deep learning.

Zheng Lin is currently pursuing the Ph.D. degree with the College of Computer Science, Nankai University, under the supervision of Prof. M.-M. Cheng. His research interests include deep learning, computer graphics, and computer vision.

Ziqin Wang received the bachelor's degree in information engineering and the master's degree in control engineering form Xian Jiaotong University, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree with The University of Sydney. His research interests include computer vision, deep learning, image segmentation, and pattern recognition.

Yong-Liang Yang is currently a Senior Lecturer with the Department of Computer Science, University of Bath. His research interests include broadly in visual computing and interactive techniques. His work has led to more than 40 publications in major journals and conferences, including SIGGRAPH, SIGGRAPH Asia, CHI, UIST, NeurIPS, and ICCV. He has served on program committees of multiple major conferences, including Symposium on Geometry Processing, Pacific Graphics, and Solid Physical Modeling. He was a recipient of the Computer Aided Geometric Design Most Cited Paper Award in 2011 and 2012. His work has also been selected as research highlights by the Communications of the ACM.

Ming-Ming Cheng (Senior Member, IEEE) received the Ph.D. degree from Tsinghua University, in 2012. Then, he did two years research fellow, with Prof. P. Torr in Oxford. He is currently a Professor with Nankai University, leading the Media Computing Laboratory. His research interests include computer graphics, computer vision, and image processing. He has published more than 60 refereed research articles, with more than 15 000 Google Scholar citations. He is on the Editor Board of IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP). He received research awards, including the ACM China Rising Star Award and the IBM Global SUR Award.