24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Semantic Point Completion Network for 3D Semantic Scene Completion

Min Zhong and **Gang Zeng** [1]

Abstract. Semantic scene completion (SSC) is composed of scene completion (SC) and semantic segmentation. Most of the existing methods carry out SSC in a regular 3D grid space, where 3D CNNs cause unnecessary computational cost on empty voxels. In this work, a Semantic Point Completion Network (SPCNet) is proposed to address SSC in the point cloud space. Specifically, SPCNet is an Encoder-decoder architecture, in which an Observed Point Encoder is applied to extract the features of observed points, and an Observed to Occluded Point Decoder is responsible for mapping the features to the occluded points. Based on the SPCNet, we further introduce an Image-point Fused Semantic Point Completion Network (IPF-SPCNet), which aims to boost the performance of SSC by combining the texture with geometry information. Evaluations are conducted on two public datasets. Experimental results show that our method can address the SC problem in the point cloud space. Compared to state-of-the-art approaches, our method can achieve satisfying results on the SSC task.

1 INTRODUCTION

Everything in the real-world occupies part of the 3D space. Humans are capable of understanding the observed 3D object and inferring the object behind the occlusion. So it is also a basic and important capability for agents to explore and interact with the circumstance. In order to meet this demand, Semantic scene completion (SSC) [20] is put forward, which predict the occupancy and semantic labels of a volumetric 3D scene from a single depth image.

In previous works, scene completion [2, 13] and scene understanding [6, 17] are considered separately. Recently, [20] points out that these two individual tasks are strongly coupled. Jointly learning semantic and geometric knowledge in the network training process can improve the performance of both tasks simultaneously. Motivated by this key idea, quite a lot of excellent works [3, 5, 8, 11, 24, 25] have been proposed in the following years and remarkable gains have been achieved.

However, there are several problems faced SSC. the first problem is brought by the 3D data representation. Conventional methods typically carry out SSC in a regular 3D grids space, where the input depth map is encoded as a 3D voxel and TSDF (Truncated Signed Distance Function). Most of the voxels are empty especially in the room scene, which renders data unnecessarily voluminous (as shown in Figure 1). Thus 3D CNN methods are not efficient in extracting the redundant 3D voxel representation, which limited the SSC performance.

The other problem is that most of the existing SSC methods only use depth image as input, where only the geometry information is considered. However, the depth image lacks of many object details, which makes it difficult to recognize some geometrical similar objects, such as wall and window. The color image carries more texture information, which can be served as assistant information to distinguish geometrical similar objects.

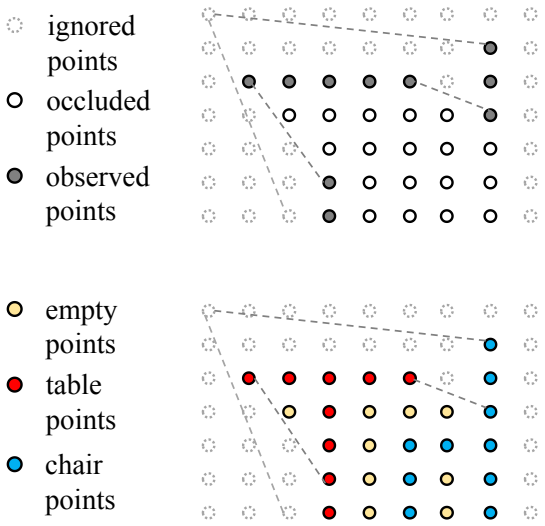

Figure 1: Intrinsic sparsity in 3D voxels. The top figure is the input for SSC and the bottom is the output. Most of the existing methods carry out SSC in the whole 3D grid space which wastes a lot of computations on the ignored and occluded points. Our method only consumes the observed points and predict a label for each observed and occluded point, which is memory and computation saving.

To overcome the first problem, we design an efficient network called SPCNet which consumes the observed points and simultaneously generates shapes and their categories for both observed and occluded points. The advantages of encoding the input depth images into point cloud representation are obvious: it can avoid unnecessary computations on empty voxels. To overcome the second problem, we propose an IPF-SPCNet based on the SPCNet, which aims to utilize texture information carried by color images.

However, to design a point consuming network for SSC task, we are faced with the issue that must be addressed: how can we take the observed points as inputs and generate new semantic points in the occluded regions. To address the issue, we design a SPCNet which contains an Observed Point Encoder (OP-Encoder) and an Observed to Occluded Point Decoder (O2OP-Decoder). The OP-Encoder extract point features from the observed points. O2OP-Decoder mapped

[1] Key Laboratory on Machine Perception, Peking University, email: minzhong, zeng@pku.edu.cn

these features to the occluded point and integrates them to infer the semantics and shape of the point cloud.

To sum up, our main contributions are as follows:

- We introduce a point cloud based network for SSC, which contains a novel Observed Point Encoder and Observed to Occluded Point Decoder.
- Based on the SPCNet, a practical IPF-SPCNet is introduced to boost the semantic scene completion performance by combining the texture with geometry information.

2 RELATED WORK

2.1 3D Deep Learning

CNNs have brought remarkable breakthroughs in processing 2D images, which are represented as pixels in 2D uniform grids. Different from the 2D image dataset, there are various data representations for 3D data, which results in various 3D deep learning techniques.

Volume-CNNs: The most straightforward work in extending 2D CNNs to higher dimensional 3D voxels is 3D CNNs [12, 15, 23]. However, it's hard to bring the powerful feature extracting ability of CNNs from 2D to 3D, since the amount of both computation and memory inflates dramatically in higher dimensions. Though some works, like FPNN [10] and Vote3D [22], try to develop special methods to work around with the computation problem. However, their improvements are still limited, it's challenging for them to handle large-scale point clouds.

View-CNNs: [15,21] have tried to take advantage of the 2D CNNs by projecting 3D data into 2D images. This kind of methods can benefit from the well-engineered 2D CNNs and have achieved remarkable performance on shape classification and retrieval tasks. However, It's not suitable for other 3D tasks such as SSC.

Point-CNNs: The above methods learn features from regular domains, where the data are represented in regular grids. While point cloud is an unordered set of vectors. Recent works, such as PointNet [14, 16], PointCNN [9], PointSIFT [7], look into the unordered point set feature learning problem and archived remarkable performance. Point cloud feature learning methods show a super advantage over other methods.

2.2 Semantic Scene Completion

Following the SSCNet [20], who point out that the shape completion and semantic labeling are coupled tasks, a lot of work has been paid attention to SSC. According to the 3D feature learning skills they are used, these methods can be broadly divided into two categories.

Volume-SSC: One is the volumetric convolutional methods, such as SSCNet [20], this kind of methods take the 3D voxel data as input and apply 3D CNNs to process them. To relive the computation and memory problem brought by 3D voxel data, [24] tries to build efficient 3D CNN blocks for SSC. But the computation reduce is still limited, even with some loss of accuracy.

View-Volume-SSC: The other is View-Volume convolutional methods [3, 5, 8, 11], which takes 2D images or depth images as inputs and projects the 2D features into the 3D space in the middle of network, finally output the SSC results in the volume space. In this way, they can make trade-offs between the computation cost and result accuracy. This kind of work has the superiority of making use of powerful 2D CNNs and combining RGB color images to assist the SSC task. However, they can not fully explore the 3D structure in the 2D space and have to come back to the redundant 3D grid space.

Point-SSC: Comparing to those semantic scene completion method, our semantic point completion network falls into the point convolutional methods. Comparing to the first kind of work, our method provides a more natural way to exploit the intrinsic sparsity of 3D data. Comparing to the second kind of work, our method breaks away from the restriction of the 3D grid, while still can capture the 3D structure information.

3 METHODOLOGY

In this section, we first introduce the point cloud representation for SSC task and then introduce the SPCNet which contains an Observed Point Encoder (OP-Encoder) and an Observed to Occluded Point Decoder (O2OP-Decoder). Finally introduce the IPF-SPCNet for combining texture and geometry information.

3.1 Input Encoding

Given a single depth image, SSC requires to predict the occupancy and semantic labels of each voxel in a 3D grid space. Traditional methods usually encode the depth image into the target 3D grid space and all of the following procedures are carried out in this space. In this paper, we take input the point's coordinates instead of the 3D grid.

As shown in Figure 1, in the input 3D grid space, there are three kinds of points [20]. The first is observed points which are produced from the given depth images. The second is occluded points which are behind the observed points under the image view. The rest points are ignored points including the seen empty points and the points outside the view and room. In the output 3D grid space, both of the observed points and occluded points are assigned a semantic label (following [20], the empty point is treated as a kind of labels).

The 3D voxel data is memory expensive, especially for the room scene. Obviously, comparing to the traditional methods who take input the whole 3D grid, our method whihc only take input the observed points will save a large amount of memory and computation.

However, this encoding method results in another problem: the number of the observed and occluded points in each 3D grid sample is uneven, which is harmful to the training of the network. We note that in [24], they partition voxels uniformly into different groups, then conduct 3D sparse convolution on each group. This skill can also be applied to point cloud. We partition the point cloud into different groups. Each group point cloud contains the same number of observed and occluded points and combines all group's results to obtain the final prediction.

3.2 Semantic Point Completion Network

By jointly learning the scene completion and semantic labeling tasks, semantic and geometry information can be combined implicitly. Thus the two individual tasks can benefit from each other. Our SPCNet designing shares the same concept above, but addresses SSC in a more effective point cloud space.

Inspired by the recently developed point set deep learning techniques, we design the SPCNet which takes the observed points as input and output a label for each occluded and observed point. Note that previous point consuming networks cannot be directly applied to SSC task. Because the output structure of SSC is different from the input. So we introduce a semantic point completion network architecture for SSC which contains two main modules, namely OP-Encoder and O2OP-Decoder.

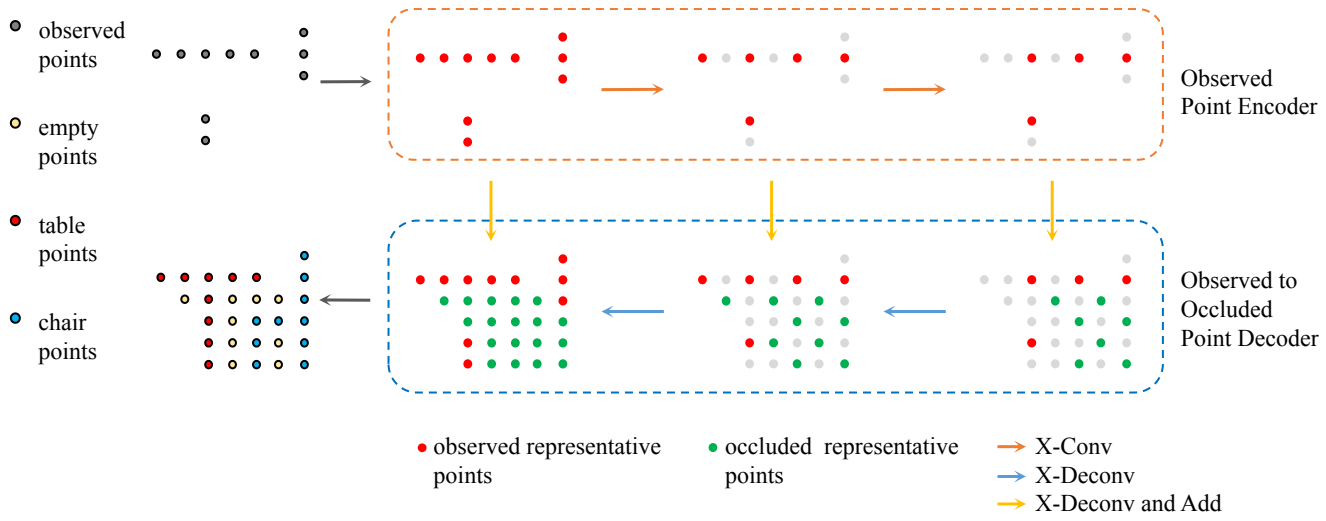

Figure 2: Illustration of the SPCNet. SPCNet contains two main modules, namely Observed Point Encoder (OP-Encoder) and Observed to Occluded Point Decoder (O2OP-Decoder). The OP-Encoder takes the observed points as input and projects the features of the points into less and less representative points (shown in red points). The O2OP-Decoder then projects the features to the occluded representation points (shown in green points) and finally predicts semantic label for each point.

In this work, we take the advantage of $\mathcal{X}$Conv operation [9] to design our network layers. The network architecture are build following the design of U-Net [18]. Before we describe the OP-Encoder and O2OP-Decoder, we briefly go through the $\mathcal{X}$-Conv operation which serves as the basic building block for SPCNet.

The input to $\mathcal{X}$Conv is a set of points $\mathcal{P}$, each associated with a feature $\mathcal{F}$. Before applying $\mathcal{X}$Conv on the input, we select a set of representation points $\mathcal{P}^{'}$ for carrying the output features. Via applying $\mathcal{X}Conv$ on $[\mathcal{P}, \mathcal{F}]$, we can get a higher level feature $\mathcal{F}^{'}$ associated to the representation points $\mathcal{P}^{'}$. Thus, $\mathcal{X}$Conv can be concisely be summarized as follows:

$$\mathcal{F}^{'} = \mathcal{X}Conv(\mathcal{P}, \mathcal{F}, \mathcal{P}^{'}). \tag{1}$$

For more detail about $\mathcal{X}$Conv, please refer to [9].

3.2.1 *Observed Point Encoder*

As illustrated in Figure 2, OP-Encoder takes the observed points as input and projects the points with features into less and less representative points (low point resolution). With OP-Encoder, we can capture the observed point structure features. Note that the occluded points do not participate in this feature encode process, because the occluded points include both empty and nonempty points. While the object structure is only decided by the nonempty points, so the empty points mixed in the nonempty points will hide the object structure. However, before we get the prediction of the semantic labels assigned to the occluded points, there is no way to separate the empty point from the nonempty points. So only observed points with known structure are feed into the OP-Encoder.

Formally, for encoder layer i, $(i = 1, 2....\mathcal{L})$, where $\mathcal{L}$ is the total number of encoder layers:

$$\begin{aligned} \mathcal{P}_b[i] &= Represent(\mathcal{P}_b[i-1]) \\ \mathcal{P}_c[i] &= Represent(\mathcal{P}_c[i-1]) \\ \mathcal{F}_b[i] &= Xconv(\mathcal{P}_b[i-1], \mathcal{F}_b[i-1], \mathcal{P}_b[i]). \end{aligned} \tag{2}$$

The inputs for encoder layer i are observed points $\mathcal{P}_b[i-1]$ and the corresponding features $\mathcal{F}_b[i-1]$ from the previous layer $i-1$. The outputs are $\mathcal{P}_b[i]$ and $\mathcal{F}_b[i]$, where $\mathcal{P}_b[i]$ is representation points sampled from $\mathcal{P}_b[i-1]$ via point sample operation $Represent(.)$, $\mathcal{F}_b[i]$ is the output features assigned to $\mathcal{P}_b[i-1]$. Note that each representation point is assigned with a point feature.

Specially, for $i = 0$, $\mathcal{P}_b[0]$ and $\mathcal{F}_b[0]$ is the input observed points and features respectively. $\mathcal{P}_c[0]$ is the input occluded points. But note that the selected occluded representative points $\mathcal{P}_c$ in each encoder layer are ready to be used in the O2OP-Decoder 3.2.2 and do not participate in the encoder feature extraction process.

3.2.2 *Observed to Occluded Point Decoder*

The decoder is responsible for propagating low-resolution information into high-resolution predictions. And there are two kinds of representative points in the decoder layers: one is the observed representative points and the other is sampled from occluded points. Both of them are from the corresponding encoder layers, namely $\mathcal{P}_b$ and $\mathcal{P}_c$. The decoder layers propagate the encoder features into the decoder representative points $[\mathcal{P}_b, \mathcal{P}_c]$, the previous decoder features are also propagated into the decoder representative points $[\mathcal{P}_b, \mathcal{P}_c]$, then this two kinds of features are added up.

Formally, for decoder layer j, $(j = 1, 2...\mathcal{L})$, the corresponding encoder layer is $(i = \mathcal{L} - j + 1)$. The decoder layer j is formalization

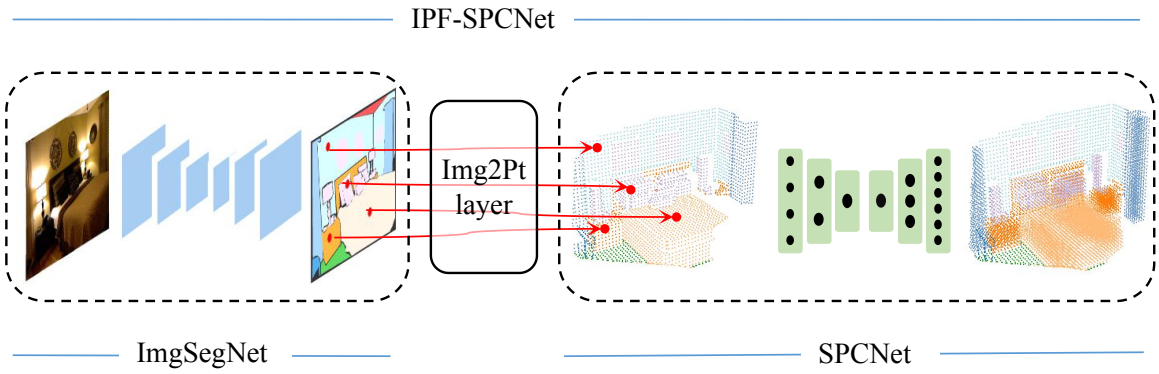

Figure 3: Illustration of the IPF-SPCNet. IPF-SPCNet mainly contains three modules: Image segmentation network (ImgSegNet) which is applied to extract the semantic features from the RGB images. 2D image to 3D point reprojection layer (Img2Pt) projects the semantic features to the corresponding points. Finally the SPCNet take the points associated with semantic features to infer the SSC results.

as following:

$$\begin{aligned}
\mathcal{P}_{bc}[j] &= [\mathcal{P}_b[i], \mathcal{P}_c[i])] \\
\mathcal{F}_e[j] &= Xconv(\mathcal{P}_b[i], \mathcal{F}_b[i], \mathcal{P}_{bc}[j]) \\
\mathcal{F}_d[j] &= Xconv(\mathcal{P}_{bc}[j-1], \mathcal{F}_{bc}[j-1], \mathcal{P}_{bc}[j]) \\
\mathcal{F}_{bc}[j] &= \mathcal{F}_e[j] + \mathcal{F}_d[j].
\end{aligned} \tag{3}$$

There are two input sources for decoder layer $j, 1 < j <= \mathcal{L}$. The first one is the observed points $\mathcal{P}_b[i]$ and the observed features $\mathcal{F}_b[i]$ from the encoder layer i. The second one is the output representation points $\mathcal{P}_{bc}[j-1]$ and features $\mathcal{F}_{bc}[j-1]$ from the previous decoder layer $j-1$. These inputs are operated by the $\mathcal{X}Conv$ functions and then added together to form the output feature $\mathcal{F}_{bc}[j]$ for decoder layer j.

Specially, $\mathcal{P}_{bc}[0] = \mathcal{P}_b[\mathcal{L}]$ and $\mathcal{F}_{bc}[0] = \mathcal{F}_b[\mathcal{L}]$ is the input points and features for the first layer of O2OP-Decoder, they are also the output of the last layer of OP-Encoder.

The final output feature of O2OP-Decoder layer are fed into several FC layers to produce a label for each observed and occluded points.

3.3 Image-point Fused SPCNet

In order to combine the texture with geometry, we design an IPF-SPCNet which mainly contains three modules: image segmentation network (ImgSegNet) serving as texture extractor, 2D-3D feature projection layer and the SPCNet fusing texture and geometry.

3.3.1 2D Texture Feature Extraction

Instead of concatenating the RGB vector to the point coordinate vector directly, we first apply an image segmentation network (denoted as ImgSegNet for convenience) on RGB images to extract the semantic feature $\mathcal{S}$. Then project the hight level semantic features to the corresponding points. There are three reasons for adopting this scheme. Firstly, the input point cloud is much sparser than the corresponding images (because we divide the whole point cloud into several groups by random sampling as mentioned in Section 3.1). So, projecting the low-level RGB features to the point cloud will lose the detailed texture information of the color images. Secondly, low-level RGB features may add some noise to the input point cloud. Because some objects with different semantics may have similar colors, such as wall and ceiling are both in white color. But the high-level semantic features can distinguish them easily. Thirdly, this scheme can take full advantage of the existing state-of-the-art 2D semantic segmentation techniques.

3.3.2 2D-3D Feature Projection and Fusion

The second module is Img2Pt layer, which is designed for the purpose of projecting the semantic feature $\mathcal{S}$ to the corresponding point $\mathcal{P}_b$. According to the camera projection equation, we can build a mapping function between 2D pixels and 3D points given the intrinsic camera matrix and the depth image. So we can assign the semantic feature vector $\mathcal{S}$ for each pixel to its corresponding 3D point $\mathcal{P}_b$ via this mapping function.

The Img2Pt layer plays a vital role in joining the powerful 2D semantic segmentation networks and 3D point cloud network together. Via Img2Pt layer, texture and geometry information can be fused effectively. Img2Pt layer seems simple yet demonstrated to be extremely useful in our experiments.

The third module is SPCNet introduce in Section 3.2. What is different is that the input of SPCNet is just points coordinate without other features, while the input here are points associated with the semantic feature $\mathcal{S}$. That is, $\mathcal{F}[0]$ in Equation 2 is none for SPCNet while is the semantic feature $\mathcal{S}$ here. Thus, the texture and geometry can be fused together inside the SPCNet.

4 EXPERIMENTS

4.1 Implementation Details

Datasets. In the following experiments, NYU [19] and NYU-CAD [4] datasets are used to train and evaluate our network. NYU [19] is a real indoor RGB-D dataset which contains 1449 RGB-D images (795 for training, 654 for test) captured via Kinect sensor. The corresponding volumetric ground truth is generated by voxelizing the CAD mesh annotations from Guo et.al. [4], object categories are mapped following [20]. Because the NYU RGB-D and their corresponding volumetric ground truth are not well aligned, so we also use the rendered RGB-D images from the CAD mesh annotations, which is denoted as NYUCAD dataset.

	scene completion			semantic scene completion											
Method	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
trained on **SUNCG + NYU** dataset															
SSCNet [20] (CVPR17)	55.6	91.9	53.2	5.8	81.8	19.6	5.4	12.9	34.4	26.0	13.6	6.1	9.4	7.4	20.2
SSCNet [20] (CVPR17)	59.3	92.9	56.6	15.1	94.6	24.7	10.8	17.3	53.2	45.9	15.9	13.9	31.1	12.6	30.5
VVNetR120* [5] (IJCAI18)	69.8	83.1	61.1	19.3	**94.8**	28.0	12.2	19.6	57.0	50.5	17.6	11.9	35.6	15.3	32.9
VVNetR60* [5] (IJCAI18)	68.3	85.1	60.9	21.6	94.5	28.6	12.9	19.7	56.3	51.0	17.2	10.4	35.2	15.6	33.0
SNet* [11] (NIPS18)	67.6	85.9	60.7	22.2	91.0	28.6	**18.2**	19.2	56.2	51.2	16.2	12.2	37.0	17.4	33.6
TNet* [11] (NIPS18)	67.3	85.8	60.6	17.3	92.1	28.0	16.6	19.3	**57.5**	**53.8**	17.7	**18.5**	**38.4**	18.9	34.4
trained on **NYU** dataset															
SSCNet [20] (CVPR17)	57.0	**94.5**	55.1	15.1	94.7	24.4	0.0	12.6	32.1	35.0	13.0	7.8	27.1	10.1	24.7
SGC [24] (ECCV18)	71.9	71.9	56.2	17.5	75.4	25.8	6.7	15.3	53.8	42.4	11.2	0.0	33.4	11.8	26.7
TS3D* [3] (18)	65.7	87.9	60.4	8.9	94.0	26.4	16.1	14.2	53.5	45.8	16.4	13.0	32.9	12.7	30.4
DDR* [8] (CVPR19)	71.5	80.8	61.0	21.1	92.2	33.5	6.8	14.8	48.3	42.3	13.2	13.9	35.3	13.2	30.4
SPCNet (Ours)	**72.1**	42.2	36.3	**33.8**	64.4	38.3	7.5	30.7	53.4	42.6	19.7	5.5	34.2	13.9	31.3
IPF-SPCNet* (Ours)	70.5	46.7	39.0	32.7	66.0	**41.2**	17.2	**34.7**	55.3	47.0	**21.7**	12.5	**38.4**	**19.2**	**35.1**

Table 1: SSC results on the NYU test set. Methods with .* use the RGB images as additional information. SPCNet** are trained on both NYU and NYUCAD depth training set. Bold numbers represent the best scores.

	scene completion			semantic scene completion										
Method	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
trained on **SUNCG + NYUCAD** dataset															
SSCNet [20] (CVPR17)	75.4	**96.3**	73.2	32.5	92.6	40.2	8.9	33.9	57.0	59.5	28.3	8.1	44.8	25.1	40.0
trained on **NYUCAD** dataset															
TS3D* [3] (18)	80.2	91.0	74.2	33.8	**92.9**	46.8	**27.0**	27.9	61.6	51.6	27.6	**26.9**	44.5	22.0	42.1
DDR* [8] (CVPR19)	**88.7**	88.5	**79.4**	54.1	91.5	56.4	14.9	37.0	55.7	51.0	28.8	9.2	44.1	27.8	42.8
SPCNet (Ours)	81.4	70.9	61.0	58.1	91.6	53.7	13.0	52.1	68.9	57.7	31.9	6.4	50.5	28.1	46.6
IPF-SPCNet* (Ours)	83.3	72.7	63.5	**58.8**	91.9	**60.5**	25.2	**53.6**	**72.9**	**62.4**	**33.8**	12.4	**53.6**	**32.5**	**50.7**

Table 2: SSC results on the NYUCAD test set. Methods with .* use the color images as additional information. Bold numbers represent the best scores.

SUNCG is a manually created large-scale 3D Indoor scene dataset introduced by [20]. Nearly 150k pairs of depth map and complete ground truth volume for training, and totally 470 pairs for testing.

For the sake of fair comparison, we downsample the input and output volume into a size of $60 \times 36 \times 60$ followed the other SSC methods. While only the observed and occluded points are used as inputs. Each observed and occluded point is assigned a label as ground truth for training and evaluation.

Evaluation metric. Following the evaluation protocol of [20], we measure the IoU of each category and the mean IoU across all categories in SSC task. For scene completion task, IoU, as well as precision and recall, is computed. The IoU is point-level intersection over the union of predicted point labels compared to ground truth labels. For scene completion task, the IoU is computed on occluded points. For semantic scene completion task, the IoU is computed on both the observed and occluded points. Note that the computation of point-level IoU in this paper is equivalent to the voxel-level IoU used in other SSC works. Because the points and voxels are one to one corresponding.

Learning Policy. During training stages, we random sample 2048 observed points and 2048 occluded points as input to feed into the proposed model. Our model is trained using the Adam optimizer with a weight decay of 1e-8 and batch size is 4, the initial learning rate is 0.001 and decay by a factor of 0.8 every 5000 step. We use the point-wise softmax as the loss function as in [20] for network training. We do not apply the data balancing scheme in [20] in our training process. During testing stage, we split each point cloud samples into several groups by random sampling. Each group contains 2048 observed points and 2048 occluded points. We combine the predict results of each group to get the final prediction for the whole point cloud sample. The ImgSegNet used in IPF-SPCNet is based on DeeplabV2 [1] trained on NYU color images as in [11]. The IPF-SPCNet are initialized by the pre-trained SPCNet.

4.2 Compare SPCNet to State-of-the-art

NYU Dataset Table 1 shows the SPCNet results on NYU testing set with a comparison to state-of-the-art methods. These methods are trained on NYU or both NYU and SUNCG. Methods with .* also use NYU RGB images as additional information for training and testing. We compare SPCNet with methods trained on NYU for fairness.

Compares to the latest method (DDR), we achieve the best performance for semantic scene completion, even without using the color images as assistant information. Compares to the SGC [24] which aims to improve the computing efficiency of 3D CNN network, our methods with more efficient point cloud deep learning technique shows the best performance in overall performance. Our methods outperform the previous SSCNet by a notable margin, that are 6.6% gains in semantic scene completion. Note that the proposed SPCNet shows superior performance in some categories such as $ceil.$, $wall$, $win.$, $chair$, $sofa$, $table$, $objs.$. We believe this improvement is due to the powerful 3D structure capture ability of SPCNet, which can extract effective 3D features for semantic labeling.

The recall and IoU of the SC task are low for NYU dataset. This is mainly because there is some misalignment between the input and output in NYU dataset. The "misalignment problem" has also been mentioned in the first work SSCNet on semantic scene completion [20]. The NYU dataset uses the depth from real-world as input,

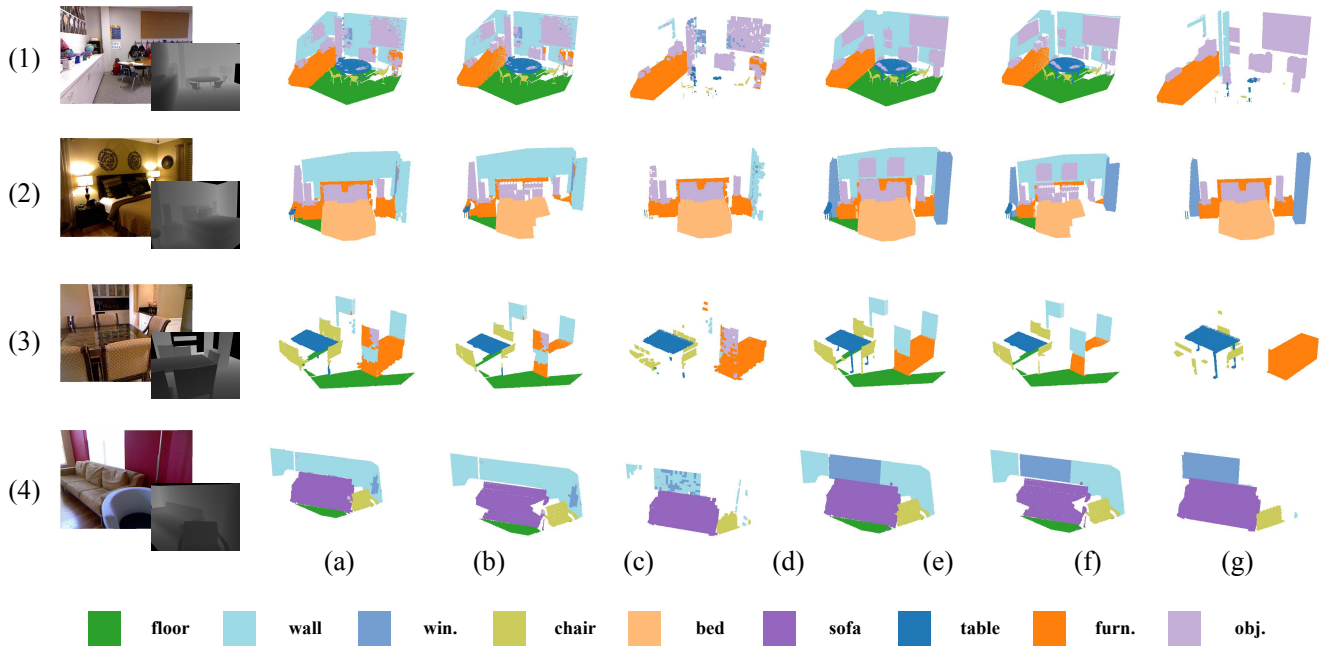

Figure 4: SPCNet results on NYUCAD. From left to right: Input Depth image (RGB images is just for better visualization), predicted result for both observed and occluded points, predicted result for observed point, predicted result for occluded points, ground truth for both observed and occluded points, ground truth for observed point, ground truth for occluded points. From top to bottom are different examples.

but uses the CAD model as output, which causes the misalignment between input and output. For the voxel-based approaches, the label of the input voxel will be shifted to another voxel when misalignment happens. But for our point-based approach, we will completely lose the label of the input point when misalignment happens. The point-based approach is much sensitive to the misalignment problem, so this dataset is very unfriendly to our point-based approach in scene completion (SC) task.

Fortunately, the NYUCAD dataset uses the depth from the CAD model as input, and use the CAD model as output. So we do not suffer the misalignment problem as in the NYU dataset.

NYUCAD Dataset Table 2 presents the SPCNet results on NYUCAD testing set with a comparison to other approaches. Note that SSCNet and SPCNet only use the depth images as input, while TS3D [3] and DDR use color images as additional information. We also achieve the best performance for semantic scene completion and comparable performance scene completion. The proposed SPCNet shows superior performance in some categories such as $ceil.$, $chair$, bed, $table$, $furn.$ $objs.$, which validate the robustness and generalization of the SPCNet.

Qualitative results Figure 4 shows visualized results of the scene segmentation generated by the SPCNet (b,c,d), ground truth (e,f,g) are also provided as a reference. All the results are acquired on the NYUCAD test set. (b) is the predicted results for both observed and occluded points. We take the observed result (c) and occluded result (d) apart for better visualization. As can be seen, the predicted result for the observed points is mostly correct compared to the corresponding ground truth (f). SPCNet can also complete the satisfactory occluded scene shapes (d) compared to the corresponding ground truth (g).

4.3 Compare IPF-SPCNet to State-of-the-art

NYU Dataset Table 1 shows the IPF-SPCNet results on NYU testing dataset with a comparison to state-of-the-art methods. Comparing to the latest methods (DDR) and (SNet, TNet), we achieve the best performance in terms of IoU for semantic scene completion, even without pre-training on big dataset (SUNCG). Note that the proposed IPF-SPCNet shows superior performance in some categories such as $wall$, $chair$, $table$, $furn.$, $objs.$. And compared to the SPCNet, IPF-SPCNet gains a great improvement in almost all categories. We believe this improvement is due to the combining of powerful 2D semantic segmentation network and powerful 3D SPCNet, which take both texture and geometry information into considering.

NYUCAD Dataset Table 2 presents the IPF-SPCNet results on NYUCAD testing dataset with a comparison to state-of-the-art approaches. We also achieve the best performance in terms of IoU for semantic scene completion and comparable performance scene completion. The proposed IPF-SPCNet shows superior performance in some categories such as $ceil.$, $wall$, $chair$, bed, $sofa$, $table$, $furn.$ $objs.$, which validates the robustness and generalization of the IPF-SPCNet.

Qualitative results Figure 5 shows the visualized results of the scene segmentation generated by the SPCNet, IPF-SPCNet and ground truth. Comparing to the SPCNet, we can observe obvious improvements in IPF-SPCNet, mainly due to the addition of texture information. The additional texture information can help the SPCNet correct some wrongly predicted semantic labels.

5 CONCLUSION

In this paper, we introduce the SPCNet, a point ConvNet for SSC, which address this task in a more efficient point cloud space. We

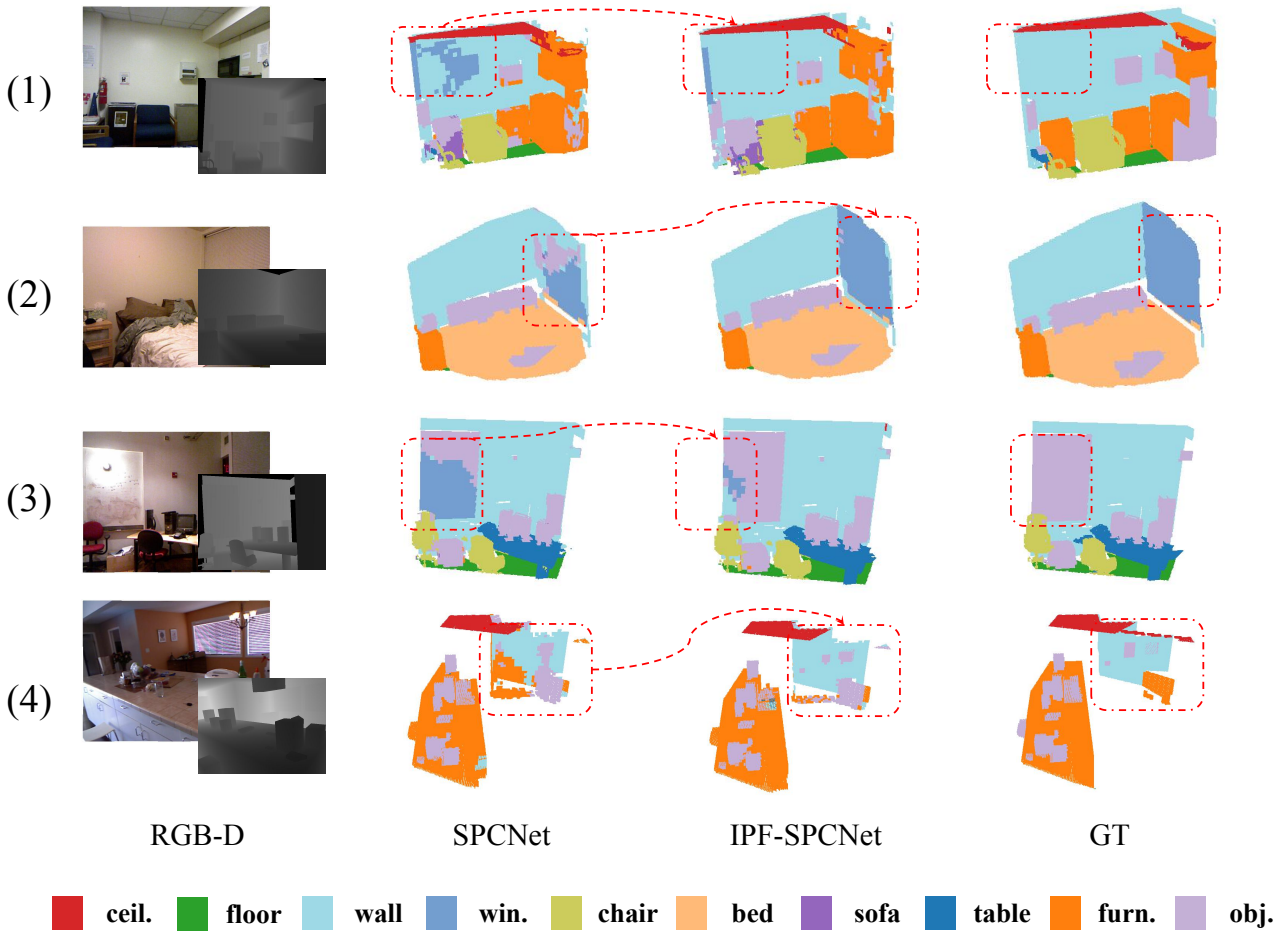

Figure 5: Compare SPCNet and IPF-SPCNet results on NYUCAD. From left to right: Input RGB-D images, SPCNet results, IPF-SPCNet results and ground truth. From top to bottom are different examples. There are obvious improvements from SPCNet to IPF-SPCNet.

also introduce an IPF-SPCNet to combine the texture with geometry information. Experimental results demonstrate that both SPCNet and IPF-SPCNet can can address the SC problem in the point cloud space, and achieve great SSC performance compared to the state-of-the-art methods.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (2017YFB1002601, 2016QY02D0304), National Natural Science Foundation of China (61375022, 61403005, 61632003), Beijing Advanced Innovation Center for Intelligent Robots and Systems (2018IRS11), and PEK-SenseTime Joint Laboratory of Machine Vision.

REFERENCES

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille, 'Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs', *TPAMI*, **40**(4), 834–848, (2018).

[2] Michael Firman, Oisin Mac Aodha, Simon J. Julier, and Gabriel J. Brostow, 'Structured prediction of unobserved voxels from a single depth image', in *CVPR*, pp. 5431–5440, (2016).

[3] Martin Garbade, Johann Sawatzky, Alexander Richard, and Juergen Gall, 'Two stream 3d semantic scene completion', *CoRR*, **abs/1804.03550**, (2018).

[4] Ruiqi Guo, Chuhang Zou, and Derek Hoiem, 'Predicting complete 3d models of indoor scenes', *arXiv*, (2015).

[5] Yuxiao Guo and Xin Tong, 'View-volume network for semantic scene completion from a single depth image', in *IJCAI*, pp. 726–732, (2018).

[6] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik, 'Perceptual organization and recognition of indoor scenes from RGB-D images', in *CVPR*, pp. 564–571, (2013).

[7] Mingyang Jiang, Yiran Wu, and Cewu Lu, 'Pointsift: A sift-like network module for 3d point cloud semantic segmentation', *CoRR*, **abs/1807.00652**, (2018).

[8] Jie Li, Yu Liu, Dong Gong, Qinfeng Shi, Xia Yuan, Chunxia Zhao, and Ian Reid, 'Rgbd based dimensional decomposition residual network for 3d semantic scene completion', *arXiv preprint arXiv:1903.00620*, (2019).

[9] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen, 'Pointcnn: Convolution on x-transformed points', in *NIPS*, pp. 828–838, (2018).

[10] Yangyan Li, Sören Pirk, Hao Su, Charles Ruizhongtai Qi, and Leonidas J. Guibas, 'FPNN: field probing neural networks for 3d data', in *NIPS*, pp. 307–315, (2016).

[11] Shice Liu, Yu Hu, Yiming Zeng, Qiankun Tang, Beibei Jin, Yinhe Han, and Xiaowei Li, 'See and think: Disentangling semantic scene completion', in *NIPS*, pp. 261–272, (2018).

[12] Daniel Maturana and Sebastian Scherer, 'Voxnet: A 3d convolutional

neural network for real-time object recognition', in *IROS*, pp. 922–928, (2015).

[13] Duc Thanh Nguyen, Binh-Son Hua, Minh-Khoi Tran, Quang-Hieu Pham, and Sai-Kit Yeung, 'A field model for repairing 3d shapes', in *CVPR*, pp. 5676–5684, (2016).

[14] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas, 'Pointnet: Deep learning on point sets for 3d classification and segmentation', in *CVPR*, pp. 77–85, (2017).

[15] Charles Ruizhongtai Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas, 'Volumetric and multi-view cnns for object classification on 3d data', in *CVPR*, pp. 5648–5656, (2016).

[16] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas, 'Pointnet++: Deep hierarchical feature learning on point sets in a metric space', in *NIPS*, pp. 5105–5114, (2017).

[17] Xiaofeng Ren, Liefeng Bo, and Dieter Fox, 'RGB-(D) scene labeling: Features and algorithms', in *CVPR*, pp. 2759–2766, (2012).

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, 'U-net: Convolutional networks for biomedical image segmentation', in *MICCAI*, pp. 234–241, (2015).

[19] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus, 'Indoor segmentation and support inference from RGBD images', in *ECCV*, pp. 746–760, (2012).

[20] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas A. Funkhouser, 'Semantic scene completion from a single depth image', in *CVPR*, pp. 190–198, (2017).

[21] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller, 'Multi-view convolutional neural networks for 3d shape recognition', in *ICCV*, pp. 945–953, (2015).

[22] Dominic Zeng Wang and Ingmar Posner, 'Voting for voting in online point cloud object detection', in *RSS*, (2015).

[23] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, '3d shapenets: A deep representation for volumetric shapes', in *CVPR*, pp. 1912–1920, (2015).

[24] Jiahui Zhang, Hao Zhao, Anbang Yao, Yurong Chen, Li Zhang, and Hongen Liao, 'Efficient semantic scene completion network with spatial group convolution', in *ECCV*, pp. 749–765, (2018).

[25] Liang Zhang, Le Wang, Xiangdong Zhang, Peiyi Shen, Mohammed Bennamoun, Guangming Zhu, Syed Afaq Ali Shah, and Juan Song, 'Semantic scene completion with dense CRF from a single depth image', *Neurocomputing*, **318**, 182–195, (2018).