View-Volume Network for Semantic Scene Completion from a Single Depth Image

Yuxiao Guo[1,2], **Xin Tong**[2]
[1] University of Electronic Science and Technology of China
[2] Microsoft Research Asia
yuxiao.guo@outlook.com, xtong@microsoft.com

Abstract

We introduce a View-Volume convolutional neural network (VVNet) for inferring the occupancy and semantic labels of a volumetric 3D scene from a single depth image. The VVNet concatenates a 2D view CNN and a 3D volume CNN with a differentiable projection layer. Given a single RGBD image, our method extracts the detailed geometric features from the input depth image with a 2D view CNN and then projects the features into a 3D volume according to the input depth map via a projection layer. After that, we learn the 3D context information of the scene with a 3D volume CNN for computing the result volumetric occupancy and semantic labels. With combined 2D and 3D representations, the VVNet efficiently reduces the computational cost, enables feature extraction from multi-channel high resolution inputs, and thus significantly improves the result accuracy. We validate our method and demonstrate its efficiency and effectiveness on both synthetic SUNCG and real NYU dataset.

1 Introduction

Reconstructing and understanding a 3D scene from its partial observations is an important technique in many robotic and vision tasks, such as indoor navigation, object retrieval, and visual reasoning. Given a single depth image captured from a 3D scene, a set of methods recently have been developed [Song *et al.*, 2017; Yang *et al.*, 2017] for automatically predicting the semantic labels or completing 3D shapes of the objects in the scene using the convolutional neural networks (CNN). To achieve good performance, previous studies have revealed that both local geometric details and global 3D context of the scene need be learned in this task [Song *et al.*, 2017]. The former one helps the system to identify the small objects in the scene, and the later one is used for inferring occluded objects from the scene layout. However, designing a CNN that can efficiently learn both features is a non-trivial task.

2D CNN based methods [Gupta *et al.*, 2014; 2015] take the depth input as an additional channel of RGB image and apply 2D CNNs for scene segmentation and object detection. Although these methods can fully exploit the high resolution input to generate detailed segmentation results, they ignore the 3D context information of the scene and thus cannot infer the invisible part of the scene. 3D CNN based methods [Wu *et al.*, 2015; Nguyen *et al.*, 2016; Song and Xiao, 2016; Yang *et al.*, 2017] convert input depth maps or point clouds into a volumetric representation and design 3D CNNs for 3D scene segmentation or object completion. However, the high computational and memory cost of the 3D CNN limits their capability for recovering object details. Recently, Song et al.[2017] proposed SSCNet for semantic scene completion, where the system simultaneously predicts the object shapes and semantic labels from a single depth image. However, the 3D CNN used in their solution limits the input resolution and the depth of the neural networks, which leads to wrong labels and missing shape details in the results.

In this paper, we propose a cascaded convolutional neural network, named View-Volume Net (VVNet), for semantic scene completion from a single depth image. The key idea of our method is to handle the local geometric details and global 3D context with two convolutional neural networks: a 2D view CNN for extracting 2D geometric features, and a 3D volume CNN for learning 3D context of the scene. Given a single depth image captured from a 3D scene, our method first extracts a set of 2D feature maps from the input depth image with the 2D view CNN and then projects the feature maps into a 3D feature volume according to the input depth map via a projection layer. After that, the 3D volume CNN is applied to the 3D feature volume to learn the 3D context information and predicts the occupancy and semantic label of each voxel in the view frustum. Since the feature projection is differentiable, the cascaded VVNet can be trained end-to-end.

The VVNet efficiently learns both geometry features and 3D context from the training dataset. The 2D view CNN avoids the high computational cost and memory consumption of the 3D CNN, which not only enables us to extract geometric features from the high-resolution input depth map, but also allows us to exploit multiple signals computed from the input depth image for feature extraction. Meanwhile, the 3D volume CNN defined over the low resolution feature volume efficiently learns the global 3D context of the scene. Moreover, the VVNet provides a flexible framework for combining variant 2D and 3D CNNs. To this end, we design and eval-

uate a set of VVNet models with different configurations for semantic scene completion.

We train the VVNets on both synthetic SUNCG dataset and real NYU dataset and validate our design. We also compare their performance with previous methods. Experimental results demonstrate that our method outperforms the state-of-the-art methods on both datasets, with much better accuracy and three times speed up for training, as well as more than 7 times speed up for inference. With different configurations, we further offer VVNet models with different trade-offs between the training/inference cost and result accuracy. All these VVNet models achieve better accuracy than previous solutions.

2 Related Work

In this section, we discuss related work and focus on methods for analyzing and completing a 3D scene from depth images or 3D point clouds, as well as the deep learning approaches that are based on hybrid 2D and 3D representations. Please refer to [Ioannidou *et al.*, 2017] for a survey of deep learning techniques for 3D data processing.

2.1 3D Scene Analysis

A set of methods have been proposed for scene segmentation, scene completion, and object detection from an input RGBD image or depth image. 2D image-based methods regard the depth as an additional channel of the 2D RGB image and leverage manually-crafted features [Gupta *et al.*, 2013; Atapour-Abarghouei and Breckon, 2017] or 2D deep neural networks [Gupta *et al.*, 2014; 2015] for these scene analysis tasks. 3D volume-based approaches convert the input depth map into a volumetric representation and exploit manually crafted 3D features [Ren and Sudderth, 2016] or 3D CNNs [Song and Xiao, 2016] for detecting 3D objects from the input RGBD image. Although these methods can successfully detect and segment visible 3D objects and scenes in the input RGBD images, they cannot infer the scenes that are totally occluded. Instead, our method predicts semantic labeling and 3D shapes for both visible and invisible objects in a 3D scene.

Liu et al. [2017] introduced 3DCNN-DQN-RNN for parsing 3D point cloud of a scene. PointNet [Qi *et al.*, 2016] and PointNet++ [Qi *et al.*, 2017] develop deep learning framework on 3D point cloud for scene semantic labeling and other 3D shape analysis tasks. These methods take the 3D point cloud of whole 3D scene as the input. On the contrary, our method takes a single depth image for semantic scene completion.

2.2 3D Scene Completion

Firman et al. [2016] inferred the occluded 3D object shapes from a single depth image via random forest. Zheng et al. [2013] completed the occluded scene in the input depth image with a set of pre-defined rules and refined the completion results by physical reasoning. These methods perform scene segmentation and completion in two separate steps. Recently, Song et al. [2017] proposed 3D SSCNet for simultaneously predicting the semantic labels and volumetric occupancy of the 3D objects from a single depth image. Although this method unifies segmentation and completion and significantly improves the result, the expensive 3D CNN limits the input volume resolution and network depth, and thus restrains its performance. By combining 2D CNN and 3D CNN, our method efficiently reduces the training and inference cost, enhances the network depth and thus significantly improves the result accuracy.

2.3 3D Object Completion

A set of methods reconstruct the 3D object shape from a single depth image using 3D shape retrieval [Rock *et al.*, 2015], Convolutional Deep Belief Network (CDBN) [Wu *et al.*, 2015; Nguyen *et al.*, 2016], or a 3D Generative Adversarial Networks (GAN) [Yang *et al.*, 2017]. All these methods model the input depth maps and resulting 3D shapes with a 3D volumetric representation. Although these methods can be combined with other scene segmentation methods for predicting 3D shapes of the visible object in the input depth map, they cannot be used for inferring objects that are totally occluded. Our method is designed for recovering complete 3D shapes of both visible and occluded 3D objects from a single depth image of a 3D scene.

2.4 Hybrid Representation for 3D Deep Learning

A set of methods [Choy *et al.*, 2016; Girdhar *et al.*, 2016; Jimenez Rezende *et al.*, 2016; Tulsiani *et al.*, 2017; Yan *et al.*, 2016; Häne *et al.*, 2017] cascade 2D encoder and 3D decoder for reconstructing 3D object shapes from a single-view color image. Because the feature vector extracted from the 2D color image has no 3D position information, it is mapped to a low resolution 3D volume (2^3 or 4^3) via fully connected (FC) neural networks. In our method, the 2D features extracted from the input depth map inherit the 3D positions of the input depth map and thus can be directly mapped to a 3D volume via projection. Wang et al. [2017] combined 3D GAN and 2D recurrent convolutional networks (RCN) for reconstructing high resolution 3D shapes from corrupted 3D models, where the 2D RCN takes 2D slices of the 3D volume generated by the 3D GAN as input for completing 3D shapes. Our method takes 2D depth map as the input and applies 3D volume CNN to the projected 2D features for 3D scene completion. Kalogerakis et al. [2017] fused semantic segmentation results generated by 2D fully convolutional networks (FCN) from different views into a conditional random field defined on 3D object surface for object segmentation. Instead of directly obtaining the segmentation results in each view, the 2D View CNN in VVNet only extracts intermediate geometric features for the 3D Volume CNN to learn the 3D context information in the whole volume for scene completion.

3 View-Volume Network

Given a single-view depth image I_d captured from a 3D scene, our VVNet outputs semantic label $C = c_0, ...c_{N+1}$ for each voxel of the scene inside the view frustum. Here we follow [Song *et al.*, 2017] to denote the number of object classes as N and mark all empty voxels by c_0. As illustrated in Fig. 1(b), the VVNet consists of three parts: a 2D view network for extracting 2D features from the input depth image,

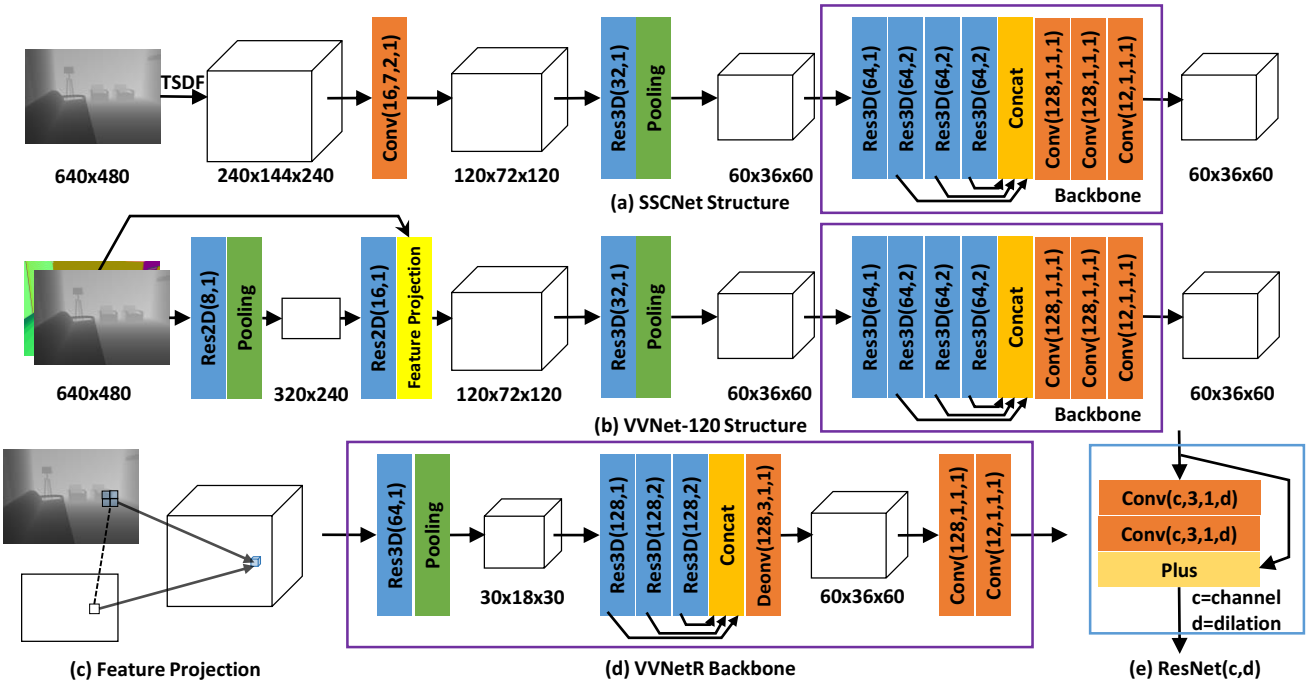

Figure 1: The network structures of VVNet and SSCNet for semantic scene completion.

a feature projection layer for converting the 2D features maps into a 3D feature volume, and a 3D volume network for inferring voxel labels from the projected feature volume. In the following part of this section, we discuss the design of each part and VVNet training.

3.1 2D View Network

The 2D view network extracts 2D geometry features from the input depth map. For this purpose, our method first computes the normal map from the input depth image and then feeds both normal and depth maps to the 2D view network. As illustrated in Fig. 1(b), we apply the residual neural network (ResNet) structure in our 2D view network design. Each ResNet block includes two convolution layers and a shortcut from input to output as shown in Figure 1(e). When a pooling layer is applied after a ResNet block, the feature map resolution is halved and the number of feature maps is doubled. For 2D ResNet used in the view network, a batch normalization layer is applied after each convolution layer. The total number of the layers in the 2D view network is determined by the resolution of the target feature maps (320×240 in Figure 1).

3.2 Feature Projection

As shown in Fig. 1(c), the projection layer projects the 2D feature maps constructed by the 2D view network into a 3D feature volume. The voxel size of the feature volume is set to be twice of the average distance of the neighboring depth pixels. Because the feature volume resolution is always lower than the feature map resolution, several neighboring features will be projected into the same voxel. This is equivalent to perform a pooling operation during the projection.

To project the 2D feature maps into the feature volume, we first construct an axis-aligned volume in the viewing coordinate system as in [Song *et al.*, 2017] and then upsample the feature maps to input depth resolution with the nearest neighboring sampling. After that, we project the feature vectors of all depth pixels into the voxels of the 3D feature volume according to their viewing directions and depth values. Finally, we get the feature vector in each voxel by averaging the feature vectors projected into it. For the voxels that are not occupied by any depth image pixels, their feature vectors are zero. We found that other pooling operations (e.g. max pooling) can also be used for computing the feature vector in each voxel but have the similar affect to the result. Instead of downsampling the depth image to the feature map resolution, we apply this super-sampling scheme for feature projection so that we can avoid the holes caused by the perspective projection and large variations in the depth image. During training, we record the mapping between feature map pixels and voxels in a table for gradient back-propagation.

In our current implementation, we set the resolutions of our feature volumes to be same as the ones in the SSCNet [Song *et al.*, 2017] for a fair comparison. We set the resolution of the feature map that corresponds to the largest 3D volume ($240 \times 144 \times 240$) to be 640×480 because for the TSDF in this resoluton, the one constructed from half-resolution depth image (320×240) is almost same as the one constructed from 640×480 depth image. For the downsampled feature volumes, we scale down the resolutions of the feature maps accordingly.

3.3 3D Volume Network

After the feature projection layer, the view-dependent 2D feature maps extracted by the view CNN are converted to a view-independent 3D feature volume. In this step, we extract the 3D context of the scene and infer the semantic label of voxels

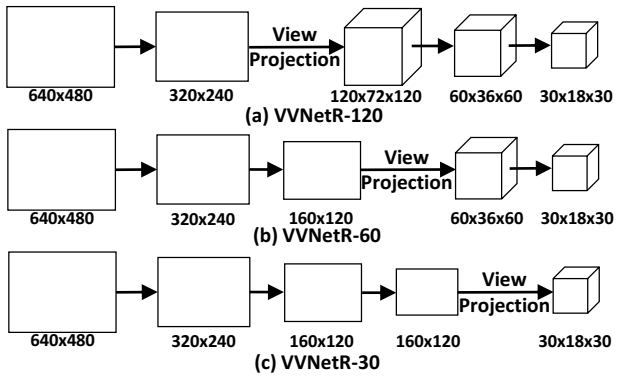

Figure 2: Different trade-offs between view and volume networks in VVNetR design.

from the 3D feature volume via a 3D volume CNN. To this end, we follow the SSCNet in our 3D feature volume CNN design(Figure 1(b)), where the 3D features are first extracted by the ResNet layers and then fed into a backbone network to generate the semantic labels for all voxels. We also design a new backbone network with enlarged reception field. As shown in Figure 1(d), the new backbone adds a new pooling layer to downsample the extracted 3D features and then applies the original backbone network to 3D features at a lower resolution. After that, we deconvolute the concatenated features to the output resolution and generate volumetric semantic labels using two convolution layers. Note that in 3D volume network, we do not apply batch normalization after each convolution layer. We denote the VVNet with the SSCNet backbone as *VVNet*, and the one with the new backbone as *VVNetR*. In Section 4, we demonstrate that our VVNetR not only reduces the computational cost for learning the 3D context, but also improves the result accuracy.

3.4 Trade-off between View and Volume Networks

The VVNet provides a flexible and general framework for combining 2D and 3D CNN for 3D scene analysis. By choosing different resolutions of the result feature maps of 2D view network, we can make different trade-offs between the depth of 2D view network and 3D volume network. Fig. 2 illustrates three VVNets, which are named as VVNetR-120, VVNetR-60, and VVNetR-30, where the numbers in the name indicate the resolution of the projected feature volume. On one side, as the resolution of the projected feature volume decreases, more layers in the 3D volume CNN are replaced by the corresponding 2D view network layers, which results in less computations and smaller memory footprints in both network training and inference. On the other side, as more 3D volume network layers are replaced by the 2D view network layers, more detailed 3D context information in the scene may be lost in the projected 3D feature volume and thus leads to degradation of the result accuracy. We evaluate this trade-off in details in the next section.

3.5 Network Training

Given the training data set (i.e the depth images and ground truth volumetric object labels of 3D scene), the VVNet can be trained end-to-end. For this purpose, we use the voxel-wise softmax as the loss function as in [Song *et al.*, 2017] in the network training. To compute the loss function, we remove all empty voxels in the visible free space, outside field of view and outside the room but include all non-empty voxels and occluded empty voxels. We do not apply the data balancing scheme in [Song *et al.*, 2017] in our training process.

4 Evaluation

We have implemented VVNet in TensorFlow under Ubuntu 16.04 on a workstation with an Intel 6700K CPU and two NVIDIA GTX 1080Ti GPUs. We use SGD optimization in VVNet training, where we set the momentum as 0.9, learning rate as 0.01, and weight decay as 0.0005. We found that the original SSCNet is not fully trained. For a fair comparison, we ported SSCNet in TensorFlow and trained it with more iterations (150K iteration with batch size 4). We denote our SSCNet implementation as SSCNet* in the following discussions.

Datasets We validate our method on the synthetic SUNCG [Song *et al.*, 2017] dataset and the real NYU [Silberman *et al.*, 2012] dataset. The SUNCG dataset consists of about 45K synthetic scenes. We select the same training/test dataset used in [Song *et al.*, 2017] for our network training and evaluation. Specifically, the training dataset includes nearly 150K depth images and corresponding ground truth volume, sampled from a 8K subset of the scenes. The test dataset is sampled from 170 scenes and consists of totally 470 pairs of depth image and ground truth volume. For SUNCG, we train VVNet with 150K iterations and change the learning rate to 0.001 after 100K iterations. We evaluate the results every 2000 steps after 130K iterations, and average them as the final results.

The real NYU dataset includes 1449 depth images (795 for training, 654 for test), captured by the Kinect depth sensor. The ground truth completion and segmentation notations are from [Guo *et al.*, 2015]. Because some manually labeled volumes and their corresponding depth images are not well aligned in the NYU dataset, we also use NYUCAD dataset in [Firman *et al.*, 2016] in our experiments, in which the depth map is rendered from the label volume. For both NYU and NYUCAD datasets, we fine tune the VVNet models trained from SUNCG dataset with 4K iterations. After that, we test the models at every 200 iterations and pick the best one as the final result.

Error Metric For each neural network, we measure the precision, recall, and IOU of all test results as in [Song *et al.*, 2017]. The IOU measures the overlapped ratio between intersection and union of the positive prediction volume and the ground truth volume. For scene completion (SC) task , the ground truth volume includes all the occluded voxels in the view frustum. For semantic scene completion (SSC) task, the ground truth volume includes both occluded voxels and visible surface voxels.

4.1 Ablation Test

We validate our VVNet design with a set of ablation tests on the SUNCG dataset.

	scene completion			semantic scene completion											
Network	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
SSCNet	76.3	**95.2**	73.5	96.3	84.9	56.8	28.2	21.3	56.0	52.7	33.7	10.9	44.3	25.4	46.4
SSCNet*	90.4	89.7	82.0	97.8	**88.2**	59.4	37.3	39.2	77.9	68.9	48.3	31.5	56.8	44.9	59.1
SSCNet*-half	90.5	89.5	81.9	97.8	88.0	60.8	34.8	39.8	77.5	69.5	47.8	29.8	56.0	44.8	58.8
VVNet-120-half	90.7	89.6	82.1	97.9	85.2	59.4	47.5	44.2	77.4	71.1	49.3	34.2	58.2	49.0	61.3
VVNet-120-depth	90.6	89.6	82.0	97.6	84.8	58.6	44.5	44.8	77.6	70.7	48.8	33.2	57.8	46.2	60.4
VVNet-120	**90.8**	90.0	82.5	97.9	85.4	58.6	49.2	45.3	79.2	71.8	50.3	37.3	62.0	50.9	62.5
VVNetR-120	**90.8**	91.7	84.0	**98.4**	87.0	**61.0**	**54.8**	**49.3**	**83.0**	**75.5**	**55.1**	**43.5**	**68.8**	**57.7**	**66.7**
VVNetR-60	90.6	92.5	**83.7**	97.6	86.7	60.2	54.4	47.2	80.7	75.0	53.8	39.4	66.9	56.1	65.3
VVNetR-30	88.8	90.2	81.0	98.0	86.4	55.6	**54.8**	41.8	78.0	72.1	48.7	31.6	63.2	51.8	62.0

Table 1: Performances of different variant VVNet design on the SUNCG dataset. **half** refers to the network that takes half-resolution image as input. **depth** refers to the network that use depth only as input.

	scene completion			semantic scene completion											
Method	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
[Lin *et al.*, 2013]	58.5	49.9	36.4	0.0	11.7	13.3	14.1	9.4	29.0	24.0	6.0	7.0	16.2	1.1	12.0
[Geiger and Wang, 2015]	65.7	58.0	44.4	10.2	62.5	19.1	5.8	8.5	40.6	27.7	7.0	6.0	22.6	5.9	19.6
SSCNet	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
SSCNet*	69.7	81.3	59.8	16.1	**94.8**	27.0	10.1	**20.6**	53.2	50.1	16.7	**14.3**	35.5	13.0	31.9
VVNet-120	68.4	83.2	60.0	19.2	94.4	27.2	**13.8**	19.1	54.0	49.3	17.1	11.2	35.3	12.4	32.1
VVNetR-120	**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
VVNetR-60	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**

Table 2: The performances of different scene completion methods on the NYU dataset.

Does Higher Image Resolution Help? We downsample the depth input images to half resolution (320×240) and use them for training a SSCNet*-half model and a VVNet-120-half model. As shown in Table. 1, the limited TSDF resolution in SSCNet cannot preserve the geometric details in high resolution input image and thus leads to very similar results for both SSCNet* and SSCNet*-half. Note that the TSDF resolution in SSCNet cannot be increased anymore due to the large memory and computational cost of 3D CNN. On the contrary, our method can fully exploit the high resolution input. Compared to VVNet-120-half, the IOUs of the VVNet-120 for SC and SSC tasks improve 0.4% and 1.2% respectively.

Does Multi-channel Input Help? To validate the contribution of the normal map to the VVNet result, we train VVNet-120-depth with the depth image only as input. Compared to VVNet-120 that takes both depth and normal as the input, the IOUs of the VVNet-120-depth decreases 0.5% and 2.1% for SC and SSC tasks respectively, which demonstrates that the normal input helps VVNet to learn the local geometric features. Note that these extra non-depth features (e.g. normal, RGB, etc.) are difficult to be used in 3D CNN training and inference as discussed in several previous methods [Song *et al.*, 2017; Dai *et al.*, 2017; Guedes *et al.*, 2018].

Does Larger Reception Field Help? Table. 1 compares the performances of VVNet-120 and VVNetR-120. For SC and SSC tasks, our new backbone with larger reception field in VVNetR-120 provides 1.9% and 5.4% IOU improvements compared to VVNet-120.

	training		inference
Network	memory	speed	speed
SSCNet*	852M	912ms	578ms
VVNet-120	846M	386ms	75ms
VVNetR-120,	712M	375ms	74ms
VVNetR-60,	336M	194ms	51ms
VVNetR-30,	246M	156ms	45ms

Table 3: Memory footprints and computational times of different networks for model training and inference.

Trade-offs between View and Volume Networks We compare the performances of VVNet models with different combinations of 2D view and 3D volume CNNs. As shown in Table. 1, the IOUs of the VVNetR-60 is slightly worse than the IOUs of the VVNetR-120 (0.3% and 1.4% decreasing for SC and SSC respectively). However, the VVNetR-60 only requires half of memory footprint and 70% computational time that the VVNetR-120 needs in the training. For VVNetR-30, we observe a relatively large performance drop from the VVNetR-120. A possible reason is that the low resolution feature volume projected from the 2D features lose too much detailed 3D context information.

4.2 Evaluation

we test the performance of VVNet network for semantic scene completion task on all three datasets and compare our method with other existing approaches.

SUNCG For SUNCG dataset, we compare the IOUs of VVNetR-120, SSCNet, and SSCNet* for both SC and SSC tasks. As shown in Table. 1, the VVNetR-120 achieves the best performance in both SC and SSC tasks. Compared to

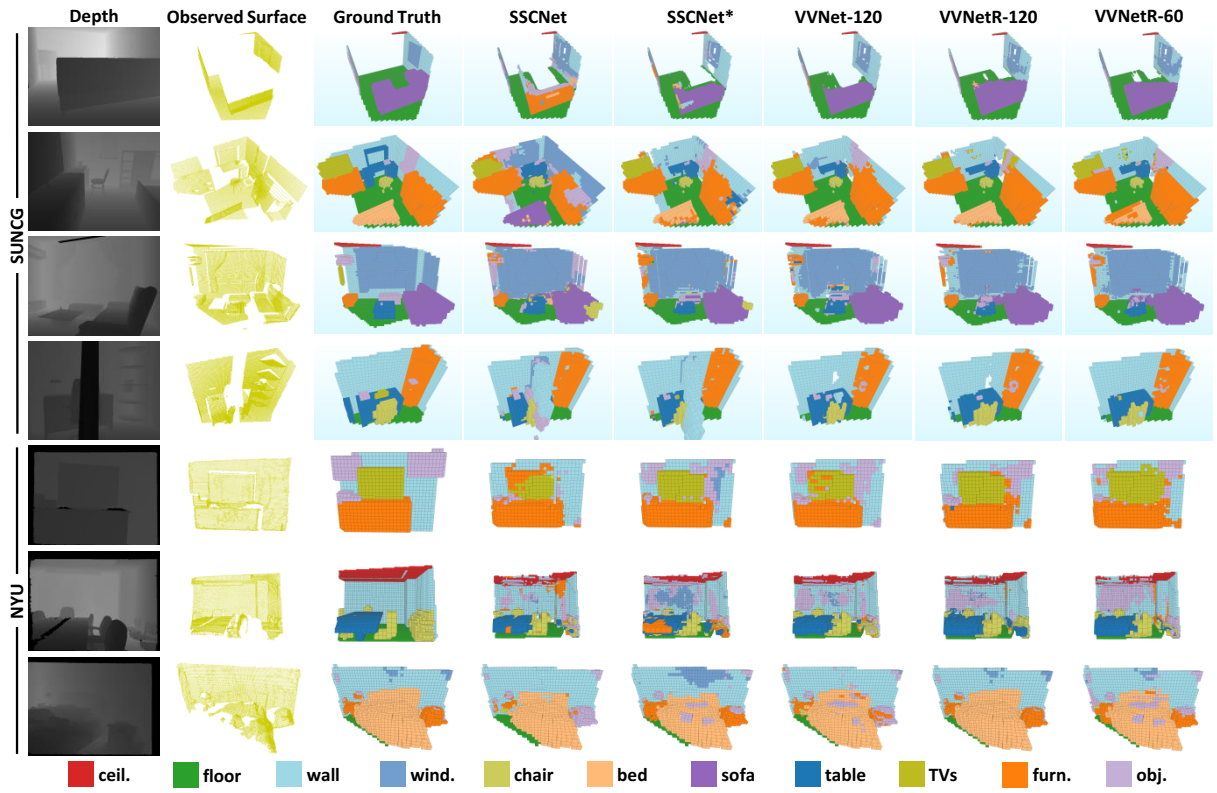

Figure 3: Semantic scene completion results generated by different methods for SUNCG and NYU datasets.

Method	prec.	recall	IoU
[Zheng *et al.*, 2013]	60.1	46.7	34.6
[Firman *et al.*, 2016]	66.5	69.7	50.8
SSCNet	75.0	96.0	73.0
SSCNet*	83.2	92.7	78.0
VVNet-120	83.3	93.1	78.5
VVNetR-120	86.4	92.0	80.3
VVNetR-60	85.6	91.5	79.2

Table 4: Performances of different methods on NYUCAD dataset.

SSCNet*, the IOUs of VVNetR-120 increase 1.4% and 5.9% for SC and SSC tasks respectively. Table. 1 also lists the IOU for each object class. Note that for all object class except floor, the VVNetR-120 achieves the best results. Fig. 3 illustrates the semantic scene completion results generated by different methods.

NYU & NYUCAD We compare our method with SSCNet* and other existing methods on both NYU and NYUCAD dataset. Table 2 and 4 list the performances of these methods on both datasets. For both NYU and NYUCAD datasets, our VVNet achieves the best performance among all the methods [Lin *et al.*, 2013; Geiger and Wang, 2015]. We believe that the small performance gap between our VVNetR models and SSCNet* on the NYU dataset is caused by the misalignment between the input and output.

Memory Footprint & Computational Cost Compared to the SSCNet that is based on 3D CNN, our VVNet combines the 2D CNN and 3D CNN and thus significantly reduces both memory footprint and computational cost in training and inference. Table.3 compares the memory footprints and computational costs of different models for training and inference. Compared to SSCNet, the VVNetR-120 provides much better accuracy and three-times speed up for training, as well as more than 7 times speed up for inference. With 40% memory footprint of SSCNet in training, the VVNetR-60 offers 4.7 times speed up for training and more than 10 times speed up for inference at the cost of slightly degraded accuracy compared to VVNetR-120. Note that the accuracy of VVNetR-60 is still better than the SSCNet.

5 Conclusion

We introduce a view-volume CNN for semantic scene completion from a single depth image. Our method concatenates a 2D view CNN and a 3D volume CNN with a projection layer and can be trained end-to-end. The VVNet provides a general and flexible framework for fusing 2D and 3D CNNs for efficient 3D learning. We validate our method on both synthetic and real datasets. Results shown that our method significantly improves the result accuracy, reduces the computational cost in training and inference, and offers variant trade-offs between the cost and accuracy.

For the future work, it is interesting to explore the trade-offs between 2D CNNs and 3D CNNs and find solutions to improve the performance of VVNet with less 3D volume

layers. Another interesting direction is to extend our view-volume framework for multiple view CNN solutions.

References

[Atapour-Abarghouei and Breckon, 2017] A. Atapour-Abarghouei and T.P. Breckon. Depthcomp: Real-time depth image completion based on prior semantic scene segmentation. In *BMVC*, pages 1–13, 2017.

[Choy *et al.*, 2016] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, pages 628–644, 2016.

[Dai *et al.*, 2017] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. *arXiv preprint arXiv:1712.10215*, 2017.

[Firman *et al.*, 2016] Michael Firman, Oisin Mac Aodha, Simon Julier, and Gabriel J. Brostow. Structured prediction of unobserved voxels from a single depth image. In *CVPR*, pages 5431–5440, 2016.

[Geiger and Wang, 2015] Andreas Geiger and Chaohui Wang. Joint 3d object and layout inference from a single rgb-d image. In *GCPR*, pages 183–195, 2015.

[Girdhar *et al.*, 2016] R. Girdhar, D.F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, pages 484–499, 2016.

[Guedes *et al.*, 2018] Andre Bernardes Soares Guedes, Teofilo Emidio de Campos, and Adrian Hilton. Semantic scene completion combining colour and depth: preliminary experiments. *arXiv preprint arXiv:1802.04735*, 2018.

[Guo *et al.*, 2015] Ruiqi Guo, Chuhang Zou, and Derek Hoiem. Predicting complete 3d models of indoor scenes. *arXiv preprint arXiv:1504.02437*, 2015.

[Gupta *et al.*, 2013] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from RGB-D images. In *CVPR*. 2013.

[Gupta *et al.*, 2014] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*, pages 345–360, 2014.

[Gupta *et al.*, 2015] Saurabh Gupta, Pablo Andrés Arbeláez, Ross B. Girshick, and Jitendra Malik. Aligning 3D models to RGB-D images of cluttered scenes. In *CVPR*, pages 4731–4740, 2015.

[Häne *et al.*, 2017] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *arXiv preprint arXiv:1704.00710*. 2017.

[Ioannidou *et al.*, 2017] Anastasia Ioannidou, Elisavet Chatzilari, Spiros Nikolopoulos, and Ioannis Kompatsiaris. Deep learning advances in computer vision with 3d data: A survey. *ACM Computing Surveys*, 50(2):20:1–20:38, 2017.

[Jimenez Rezende *et al.*, 2016] Danilo Jimenez Rezende, S. M. Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *NIPS*, pages 4996–5004. 2016.

[Kalogerakis *et al.*, 2017] Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. 3D shape segmentation with projective convolutional networks. In *CVPR*, pages 3779–3788, 2017.

[Lin *et al.*, 2013] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *ICCV*, pages 1417–1424, 2013.

[Liu *et al.*, 2017] Fangyu Liu, Shuaipeng Li, Liqiang Zhang, Chenghu Zhou, Rongtian Ye, Yuebin Wang, and Jiwen Lu. 3dcnn-dqn-rnn: A deep reinforcement learning framework for semantic parsing of large-scale 3d point clouds. In *ICCV*, pages 5678–5687, 2017.

[Nguyen *et al.*, 2016] D. T. Nguyen, B. S. Hua, M. K. Tran, Q. H. Pham, and S. K. Yeung. A field model for repairing 3d shapes. In *CVPR*, pages 5676–5684, 2016.

[Qi *et al.*, 2016] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2016.

[Qi *et al.*, 2017] 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*, pages 5105–5114, 2017.

[Ren and Sudderth, 2016] Zhile Ren and Erik B. Sudderth. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *CVPR*, pages 1525–1533, 2016.

[Rock *et al.*, 2015] Jason Rock, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem. Completing 3d object shape from one depth image. In *CVPR*, pages 2484–2493, 2015.

[Silberman *et al.*, 2012] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012.

[Song and Xiao, 2016] Shuran Song and Jianxiong Xiao. Deep Sliding Shapes for amodal 3D object detection in RGB-D images. In *CVPR*, pages 808–816, 2016.

[Song *et al.*, 2017] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, pages 1746–1754, 2017.

[Tulsiani *et al.*, 2017] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, pages 2626–2634, 2017.

[Wang *et al.*, 2017] Weiyue Wang, Qiangui Huang, Suya You, Chao Yang, and Ulrich Neumann. Shape inpainting using 3d generative adversarial network and recurrent convolutional networks. In *ICCV*, pages 2298–2306, 2017.

[Wu *et al.*, 2015] Zhirong Wu, Shuran Song, Aditya Khosla, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shape modeling. In *CVPR*, pages 1912–1920, 2015.

[Yan *et al.*, 2016] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *NIPS 2016*, pages 1696–1704, 2016.

[Yang *et al.*, 2017] Bo Yang, Hongkai Wen, Sen Wang, Ronald Clark, Andrew Markham, and Niki Trignoi. 3d object reconstruction from a single depth view with adversarial learning. In *ICCV*, pages 679–688, 2017.

[Zheng *et al.*, 2013] Bo Zheng, Yibiao Zhao, Joey C. Yu, Katsushi Ikeuchi, and Song-Chun Zhu. Beyond point clouds: Scene understanding by reasoning geometry and physics. In *CVPR*, pages 3127–3134, 2013.