Towards Efficient 3D Point Cloud Scene Completion via Novel Depth View Synthesis

Haiyan Wang, Liang Yang, Xuejian Rong, Yingli Tian
The City College of New York, New York, NY 10031
hwang005@citymail.cuny.edu
{lyang1,xrong,ytian}@ccny.cuny.edu

***Abstract*—3D point cloud completion has been a long-standing challenge at scale, and corresponding per-point supervised training strategies suffered from cumbersome annotations. 2D supervision has recently emerged as a promising alternative for 3D tasks, but specific approaches for 3D point cloud completion still remain to be explored. To overcome these limitations, we propose an end-to-end method that directly lifts a single depth map to a completed point cloud. With one depth map as input, a multi-way novel depth view synthesis network (NDVNet) is designed to infer coarsely completed depth maps under various viewpoints. Meanwhile, a geometric depth perspective rendering module is introduced to utilize the raw input depth map to generate a re-projected depth map for each view. Therefore, the two parallelly generated depth maps for each view are further concatenated and refined by a depth completion network (DCNet). The final completed point cloud is fused from all refined depth views. Experimental results demonstrate the effectiveness of our proposed approach composed of aforementioned components, to produce high-quality, state-of-the-art results on the popular SUNCG benchmark.**

I. Introduction

We live in a three-dimensional world, and a proper cognitive understanding of the 3D structures is crucial for acting and planning. The ability to anticipate under uncertainty is necessary for autonomous agents to perform various downstream tasks such as exploration, active grasping, and target navigation. Many recent deep learning methods have demonstrated effectiveness in solving these tasks, such as scene prediction [4], [6], [9], [15], [27], [32], [40] and 3D completion (on representations such as voxel and point cloud) [5], [8], [14], [16], [19], [34], [38]. Since grid-based data representations such as voxel or mesh need to divide the whole 3D space, it always leads to computational redundancy and inefficiency. However, compared to grid-based methods, point cloud is a better data representation and closer to the three-dimension world. While pioneered by PointNet [26], the point cloud direct processing methods have become more and more popular. These methods have been explored to apply on 3D completion and reconstruction tasks as well.

Most existing depth-based, 3D point cloud completion methods focus solely on single objects and surrounding viewpoints, which are not trivial to set up for natural scenes. Although scene point cloud completion can benefit various related applications, it is significantly more challenging than single object point cloud completion. The single objects can be easily reconstructed based on their inner geometric structure constraints. However, for scene point cloud, the learning process is limited by the large scale of points which are hard to annotate. The number of points can easily reach in to the millions for a 3D scene, leading to a heavy computation burden for deep convolution networks. On the other hand, complicated

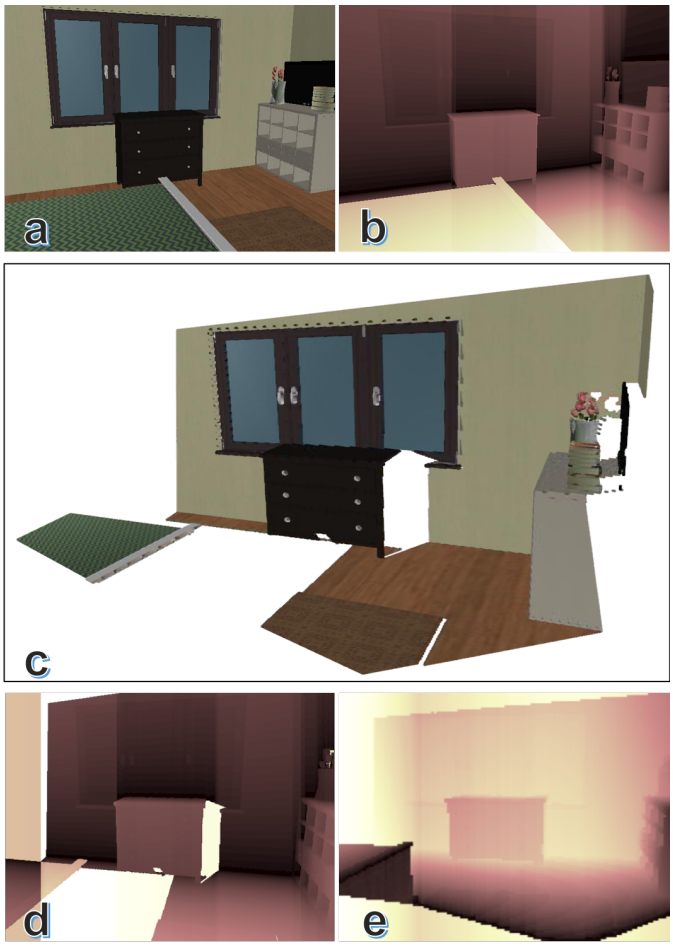

Fig. 1: Illustration of our scene completion via novel depth views: (a) RGB image (for visualization only, and not used in our framework); (b) input depth map; (c) back-projected point cloud visualized from another viewpoint, which shows holes under completion; (d) the incomplete depth map; (e) and the completed depth map with our proposed network. Other views are similar to this view whose holes are filled and thus the scene is completed.

spatial relationships between various 3D objects often lead to the challenge of directly inferring the complete 3D scene from a single depth view. There are many occluded situations across multiple objects from a single view. For instance, a desk might be occluded by chairs around it, or a computer on the desk.

In this paper, as shown in Figure 1, we efficiently and effectively infer and complete a natural scene point cloud from a single incomplete depth map. All the supervision signals of our method come from various 2D view depth maps, which can reduce the effort to annotate 3D fully supervised labels. Taking a partial observation depth map as input, our model generates multi-way plausible depth views with different viewpoints through the proposed NDVNet (Novel Depth View Network). These predictions are then concatenated with the geometrically rendered depth views in the same viewpoint, which are generated by the proposed perspective rendering strategy. The purpose of utilizing the re-projected depth maps is to solve the occluded problem to some extent and further help refine the coarse-completed depth map. The concatenation is further jointly refined by a coarse-to-fine depth refinement module DCNet (Depth Completion Network) to obtain the completed depth map. All the refined outputs are fused to generate the final completed scene point cloud. Our key contributions are summarized below:

- We propose an end-to-end trainable network which is able to generate dense and complete 3D surface scene point clouds from a single shot depth map input.
- We introduce a coarse-to-fine point cloud completion schema. In conjunction with predicting novel view depth maps for completion, depth inpainting network is added to further complete the whole point cloud.
- To the best of our knowledge, this is the first work to use an end-to-end method to conduct the 3D indoor scene point cloud completion task. The experiments demonstrate the effectiveness of our proposed method which achieves comparable performance with state-of-the-arts on the SUNCG dataset.

II. Related Work

Various methods have been developed for 3D completion tasks. Beyond the conventional methods based on the geometry prior or template [13], [17], [18], [20], [23], [28], [30], [31], [33], [35], [36], [37], [44], recent deep learning-based methods show advantages. These methods can be divided into two categories: 1) direct prediction of 3D structures, or 2) 2D supervision, i.e., predicting 2D depth maps as a proxy and then obtaining the complete 3D objects and scenes with depth fusion.

A. 3D Completion by Direct Prediction

To recover incomplete single objects or scenes, one of the most intuitive methods for completion is to directly process 3D data (e.g. voxel or point cloud), and inpaint occluded or missing parts/regions in objects/scenes.

Volume-based Methods Essentially, 3D data can be represented as volumes/voxels in 3D space. Starting from SSCnet [34], the authors first performed the scene completion task by proposing a SSCnet that takes a single depth as input, then represented the depth as 3D volume and conducted 3D-CNN convolution on the volume. By predicting the occupancy and class label of each volume in the space, 3D semantic scene was completed. Following the work of SSCnet [34], other methods were proposed by taking advantages of volume representation. Schnabel et al. [30] introduced a fully convolutional network structure to handle large-scale volume data. Wang et al. [39] completed 3D scene using two encoders and one decoder from the adversarial perspective. Garbade et al. [11] voxelized a scene and predicted depth and semantic information from the input RGB images. Guo et al. [12] encoded the geometry information through a 2D CNN and then fed it to a 3D CNN to compute volumetric occupancy and semantic labels. However, volume-based 3D methods have to sacrifice representation accuracy and lead to huge redundancies.

Point Cloud-based Methods Recently, point cloud-based methods became more popular. Taking partial point cloud as input, networks directly predict the complete point cloud as output. Normally these networks are all made up of encoder and decoder networks. The encoder network is quite similar in different methods. Achlioptas et al. [1] just simply took the fully-connected layers as their decoder. Fan et al. [10] combined the fully-connected layers with deconvolution layers to obtain better point cloud prediction. Yang et al. [42] innovatively proposed the FoldingNet, which is an auto-encoder structure deforming a 2D canonical grid to 3D surface of objects through folding operations. In the PCN [43] paper, the authors proposed a coarse-to-fine completion idea. By combining the local and global features and taking advantage of FoldingNet's [42] decoder network, the network can output a dense completion point cloud for objects. However, these existing point cloud-based methods can only handle the completion task of single objects and cannot directly, end-to-end, accomplish the completion task for the 3D scene point cloud due to the complexity of 3D nature scene and large-scale number of points.

B. 3D Completion by 2D Supervision

Besides directly processing 3D data, there is also a branch of methods that complete 3D objects or scenes using 2D supervision signal such as 2D depth maps, binary masks, segmentation maps, etc.

For 3D object and scene completion, studies attempted to predict 2D depth maps of various viewpoints around the center and then fuse them to obtain complete point clouds. Lin et al. [21] proposed an efficient end-to-end network to generate a dense point cloud. They first took a single RGB image as input and then predicted the depth maps of sampled viewpoints. The loss is calculated with ground truth depth maps and optimized the whole reconstruction pipeline. In [24], the authors proposed a novel projection method named Capnet, which was used to predict the foreground masks and then supervise the whole reconstruction process. Chen et al. [3] introduced Point-MVSNet, which utilized coarse-to-fine reconstruction idea.

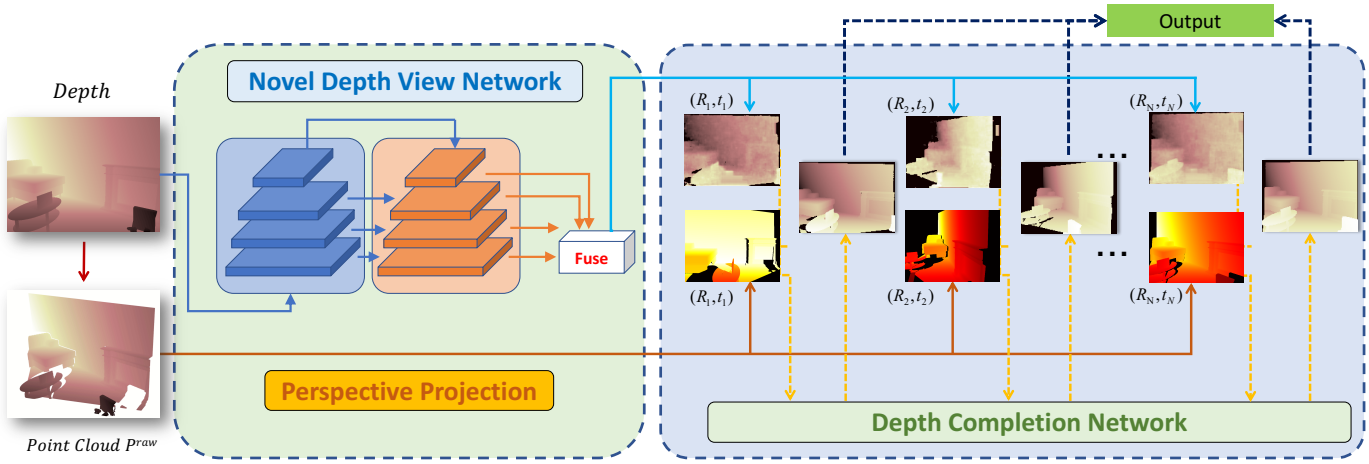

Fig. 2: Architecture of our proposed end-to-end scene point cloud completion network which takes a single depth map with original size as input. The whole structure has two subnetworks: **NDVNet** (Novel Depth View Network) and **DCNet** (Depth Completion Network). The NDVNet consists of skip-connected encoder and decoder networks. The pyramid fused feature map outputs are further utilized to predict novel depth maps from various viewpoints ((R_1, t_1), (R_1, t_1) ... (R_N, t_N)). Meanwhile, we re-project the point cloud (from input view) to N viewpoints to obtain the coarse depth maps, which can provide more detailed textures. A coarse-to-fine DCNet is then introduced to optimize each viewpoint depth prediction given a novel predicted depth map and a related re-projected coarse depth map. Finally all of the refined depth maps are fused together to form the final completed scene output.

They first predicted the coarse depth map from an RGB image and then refined the depth map by predicting the point flow from multiple view images. Combined with the depth residual and 3D geometry priors, the depth map is refined and output dense point cloud reconstructed. Han et al. [14] considered scene point cloud completion from a depth map inpainting perspective. They proposed DQN network to find the next best view to project a depth map and then complete the projected depth maps under the guidance of SSCNet [34]. However, this network depends on volume completion as guidance for inpainting by handling the occluded areas with off-the-shelf work (e.g. SSCNet). In addition, the network is composed of multiple sub-networks, which are too complicated and time-consuming.

III. Methodology

A. Architecture Overview

Given a depth map, the point cloud can be generated by back-projection following the rule: $P^{raw} = [R,t]^{-1}K^{-1}[u,v,1]^T \cdot D(u,v)$, where $D(u,v)$ denotes the depth measurement of pixel (u,v). However, due to occlusions of each single view, P^{raw} may miss occluded structure information. In this paper, we propose to complete scene point cloud from a single view depth map, taking a raw depth map as input and generating the complete point cloud as output. As shown in Fig. 2, our model comprises three main components: 1) a multi-way depth synthesis module, **NDVNet** (Novel Depth View Network), which aims to preliminarily lift a single depth input to multiple hallucinated novel depth views; 2) a geometric coarse depth rendering module which aims to render faithful coarse depth maps; and 3) a depth completion refinement module **DCNet** (Depth Completion Network) to jointly generate a high-quality, inpainted depth output. The proposed network takes a single depth map with original size (VGA) as input. To tackle the occlusion challenge, the NDVNet is designed to predict N novel viewpoint depth maps from well-selected N views, $(R_i, t_i), i = \{1, 2, ..., N\}$, and we further resize the output depth maps into 160×120. For each novel viewpoint v_i, a coarse depth map is re-projected from the input P^{raw} via perspective projection. Through DCNet, each predicted depth map and re-projected coarse depth map are paired to generate a complete depth map under v_i. Finally, the complete 3D scene point cloud can be obtained via fusing the complete depth map from all of the N viewpoints.

B. Novel Depth View Network and Perspective Rendering

The features of a single view depth map are extracted by the NDVNet as the coarse depth information from other novel views. Based on the latent representation, we propose to generate the coarse-completed depth ${}^pD$ from selected N views. Meanwhile, we take advantage of the input single view point-cloud P_{raw} and project it into each view to obtain the coarse depth maps ${}^cD$ without completion.

1) Novel Depth View Network: Our model predicts a depth map from each novel view, aiming to complete scene structure from the current view by providing more detailed information compared to coarse voxel completion [34]. Inspired by UPerNet [41], we employ the Feature Pyramid Network (FPN) to predict novel view depth and perform depth completion in an end-to-end manner. As shown in Fig. 2, NDVNet takes as

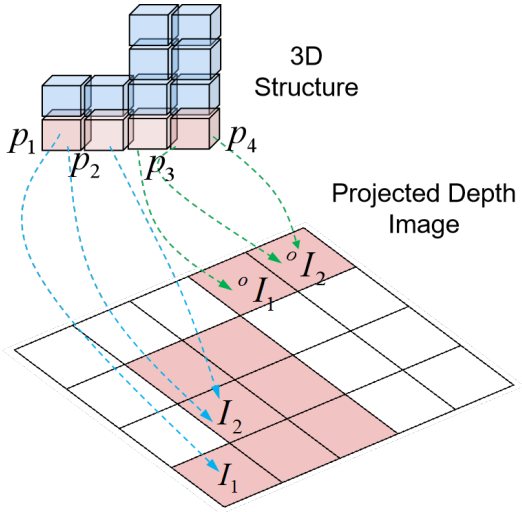

Fig. 3: Coarse depth rendering is performed through a ray-tracing and casting approach, Where the point collision is considered as a fusion to obtain the depth map. Since we use a cube to represent a point in 3D space, there may exist several cubes cast into one pixel or one cube cast into multi-pixels.

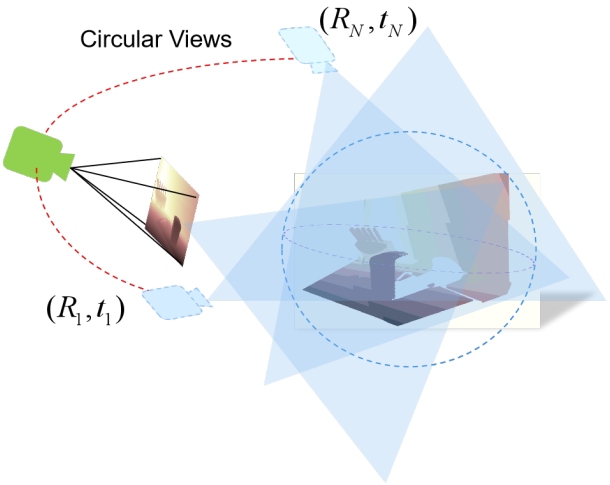

Fig. 4: Illustration of the novel views which are designed in a circular pattern to provide full-coverage of a target scene. The circle is formed by the camera center and the scene center with a fixed height Z.

input, a single depth map with the spatial size of 640×480. Through the pyramid feature extraction layers, the feature spatial size has been downgraded to 1/4, 1/8, 1/16, and 1/32 layer by layer, and then upsampled back to the original size. The skip connection links the low level and high level features together to ensure both local and global information are involved in the prediction. All decoder layers are fused together, followed by one convolution layer to generate novel depth maps with N viewpoints. The output size of each view is 160×120.

2) Coarse Depth Perspective Rendering: In our scene completion model, the input, single view point cloud, $P^{raw} = [R,t]^{-1}K^{-1}[u,v,1]^T$, is able to provide guiding information by projecting to each novel view. It is denser compared to depth prediction by the NDVNet. A joint learning by combing the predicted depth and the projected depth would allow a highly accurate completion of the target scene. As shown in Fig. 3, with the proposed perspective rendering approach, we re-project P^{raw} to each view $(R_i, t_i), i = \{1, 2, ..., N\}$, to obtain its depth map, called coarse depth map ${}^cD$. For the rendering process, several points may cast to the same pixel of the view leading to a collision problem. Thus, we propose the perspective rendering to solve this problem by fusing all the collided points' depth to obtain the depth value of a pixel.

To fuse the depth projected from multiple 3D points - inspired by [25] - we introduce a view angle-based weighted average method. For all collided points, $p_i = (X, Y, Z), i = 0, 1, ..., m$, depth is represented as $D(p_i)$. Then, we have the following fusion representation:

$$(u,v) = \frac{w_0 D(p_0) + w_1 D(p_1) + ... + w_m D(p_m)}{w_0 + w_1 + ... + w_m}, \quad (1)$$

where $w_i \propto cos(\theta)$ denotes the weight of each point, θ is the angle between the associated pixel ray direction and the ray direction from the camera center O^c to each point p_i. (u, v) is the coordinate location of the pixel on the re-projected depth map. We use such ray casting and depth fusion to obtain the projected coarse depth map from each view (R_i, t_i).

C. Depth Completion Network

To reconstruct completed 3D structure, we fuse 3D point cloud from each selected novel view in an iterative manner. Based on Section III-B1, we know the predicted depth map only has a resolution of 160×120, which is not able to provide high-quality dense reconstruction. To address this challenge, we further propose a depth refinement model which takes the predicted novel depth and coarse depth as input, and outputs a 640×480 refined depth map.

1) Novel View Generation: As illustrated in Fig. 4, we consider a circular view approach to enforce the coverage completeness to reconstruct the occluded scene. For each novel view, it starts from the input view (R^{raw}, t^{raw}) toward two sides. The moving trajectory is parallel to the circle formed by the initial view and the center of the scene. The circular trajectory shares the same height as the input view due to the purpose of keeping that view's similarity.

2) Coarse to Fine Depth Completion Network: The DCNet refines the low-resolution prediction and generates a 640×480 output. The input is a concatenation of re-projected depth map ${}^cD$ and the coarse-completed low resolution depth map ${}^pD$ which results in a 2 channel image. It is worth noting that ${}^pD$ is up-sampled to 640×480 by using a bilateral filter before

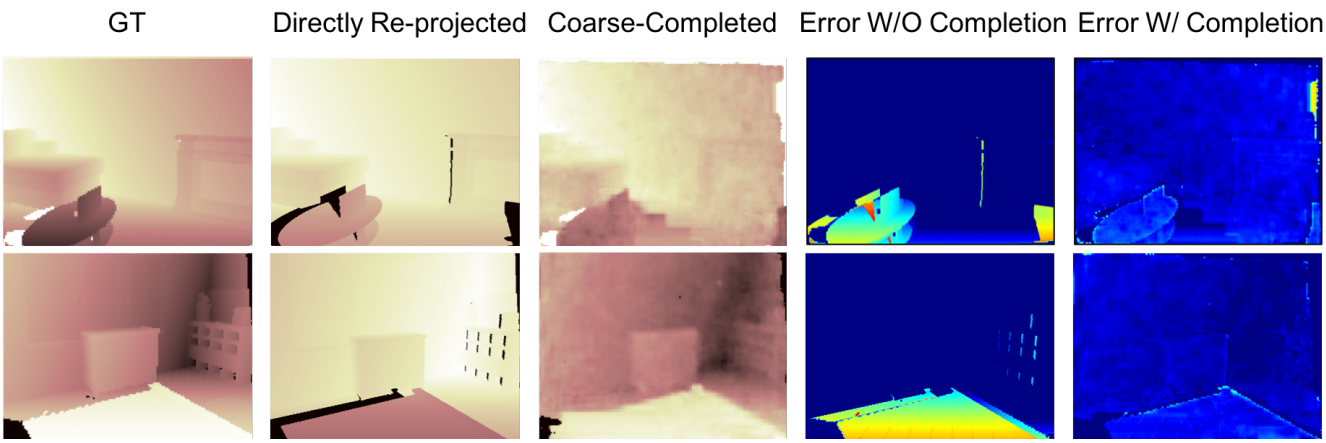

Fig. 5: Demonstration of the effectiveness for the proposed NDVNet using error maps which are obtained by computing the pixel-level difference of prediction over the ground truth. Here we compare the error maps of the re-projected depth map (2nd column) and coarse-completed depth maps (3rd column) against the ground truth (1st column). They are generated by the perspective rendering and the NDVNet, respectively. The last two columns show the error maps for the 2nd and 3rd columns

.

concatenation. The overall framework of the DCNet is just a simple encoder-decoder network which is very similar to UNet [29]. We just remove the skip connections, and the gated convolution [22] is added to the DCNet.

$$\begin{aligned} G_{y,x} &= \sum\sum W_g \cdot I, \\ F_{y,x} &= \sum\sum W_f \cdot I, \\ O_{y,x} &= \phi(F_{y,x}) \odot \sigma(G_{y,x}), \end{aligned} \tag{2}$$

where I and $F_{y,x}$ represent the input and output of the the gated convolution layer, respectively. The $G_{y,x}$ learns the soft mask from the coarse completed depth map and provides the guidance for further completion refinement. Here σ and ϕ are the non-linear function. W_g and W_f are different convolution kernels. We refer the readers to [22] for details of the architecture settings.

NDVNet and DCNet are trained independently first, and then we further jointly train the whole 3D scene point cloud completion model.

3) Multi-view Dense Fusion Approach: Our final goal is to reconstruct the 3D structure of the target view, and recover the occluded areas. Given N views - i.e. $(R_i, t_i), i = 1, .., N$ - and the corresponding completed depth map, we concatenate the partial point cloud obtained from each novel view into a global model following the pin-hole model back-projection,

$$P^r = \sum_{i=1}^{N} (R_i, t_i)^{-1} K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cdot D(u, v), \tag{3}$$

where P^r denotes the reconstructed global point cloud, which is obtained by concatenation, $\sum_{i=1}^{N}$, of points from each frame. The performance of the reconstruction is evaluated using Chamfer-Distance (CD) [2].

D. Loss Design and Learning

We adopt the $L1$ loss for the novel view depth map prediction during the first stage training process:

$$L_{novel} = \sum_{i=1}^{N} |D_{gt} - D_{predicted}|, \tag{4}$$

where D_{gt} is the ground truth depth map at view v_i, which is obtained by projecting the complete ground truth point cloud to the specific viewpoint, v_i. $D_{predicted}$ is the depth map generated by NDVNet at view v_i

And for the second stage, the loss is designed as follows:

$$L_{completion} = \sum_{i=1}^{N} |\zeta({}^{c}D||{}^{p}D) - D_{gt}|, \tag{5}$$

$$L = \lambda L_{novel} + L_{completion}, \tag{6}$$

where ${}^{p}D$ and ${}^{c}D$ are the aforementioned coarse-completed depth map and the re-projected depth map. ζ is the convolution function of DCNet. L is the total loss function which is combined with L_{novel} and $L_{completion}$. λ is the weighted parameter.

IV. Experiments

A. Dataset

SUNCG [34] is a large-scale synthetic scene dataset of an indoor scene, which contains 45,622 scenes with realistic room and furniture layouts. The 2D data, such as depth maps, RGB images, and segmentation maps, is rendered for each room. The rooms with less than 10 views are eliminated. We choose around 1,000 rooms for our experiments. For each rendered depth map in the room, we obtain the truncated complete point cloud through casting rays from the depth maps. Then we render the truncated complete point cloud to n fixed views

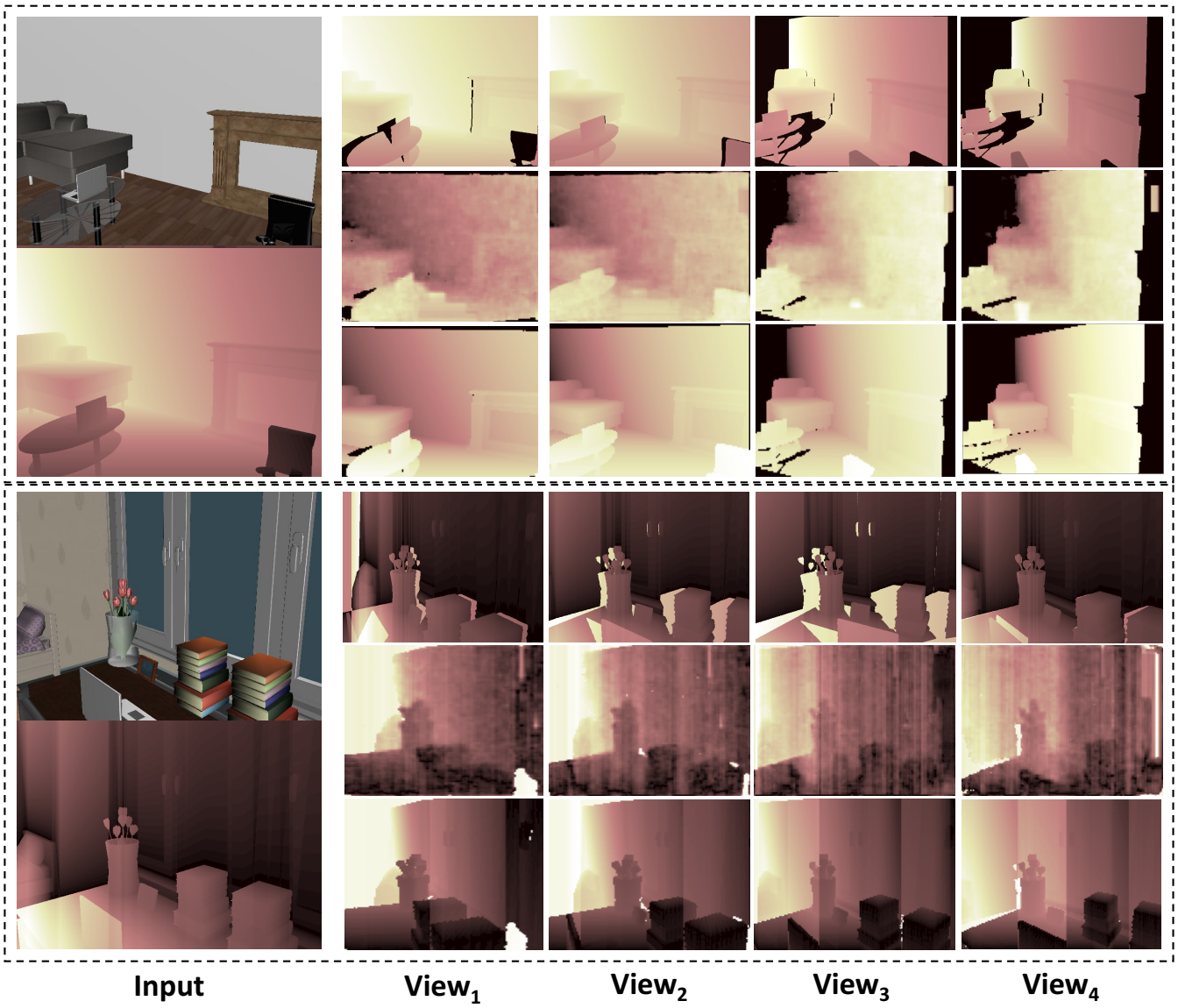

Fig. 6: Qualitative Results of 2 different indoor scenes (divided by the dashed line bounding boxes). For each scene, the first column is the RGB image and input depth map (RGB image is just for visualization and not used in our framework). The other columns are the depth maps of the experiment results under 4 novel views. The first row is the re-projected depth maps under the v_i, which have holes such as desks and floor. The second row is the coarse-completed depth maps generated by NDVNet. And the third row is the final completed depth maps generated by DCNet. Note that the holes are filled and the scenes are well completed through our proposed whole completion model and coarse-to-fine strategy.

TABLE I: Results comparison with other methods in CD distance and completeness.

	$SSCNet$ [34]	$ScanComplete$ [7]	$DQN_{w/o-hole}$ [14]	DQN [14]	$Ours$
CD	0.5162	0.2193	0.1495	0.1148	0.1221
$C_{r=0.002}(\%)$	14.61	34.46	79.22	79.26	80.01

around its center. In the experiment, we choose $n = 8$. In total, we have 20,000 depth maps, and choose 2,000 of them as a testing set, and the rest for training.

B. Implementation Details

The experiments are conducted with fixed viewpoints, which are selected by calculating the distance between depth map views and the center of the truncated complete point cloud as

radius. Then we evenly sample fixed angles around the circle. The training process has 3 stages: 1) the depth completion model NDVNet is trained from scratch with 200 epochs, adopting Adam optimizer, and a learning rate of 0.005. 2) The depth completion model, DCNet, is trained with the output of the completion model and the re-projected partial depth maps for another 100 epochs. 3) The end-to-end network is again fine-tuned with 100 epochs, learning rate at 0.0001, and Adam optimizer for training process optimization.

C. Evaluation Metrics

Chamfer Distance [43] is adopted as the evaluation metric to calculate the average closest point distance between the generated point cloud and the ground truth complete point cloud:

$$\begin{aligned} CD(S_1, S_2) = & \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} ||x - y||_2 \\ & + \frac{1}{|S_2|} \sum_{y \in S_2} \min_{x \in S_1} ||y - x||_2. \end{aligned} \tag{7}$$

In order to compare the results with other papers [10], [14], we introduce the completeness as another evaluation metric:

$$C_r(S_1, S_2) = \frac{|\{d(x, S_1) < r | x \in S_2\}|}{|\{y | y \in S_2\}|}, \tag{8}$$

where S_1 is the output completed point cloud, S_2 is the ground truth point cloud, $d(x, S_1)$ represents the distance between a point to a point set P, and r is the distance threshold set as 0.02 in our experiments.

D. Effectiveness of NDVNet

The middle stage results are explored to show the effectiveness of the proposed NDVNet. We use error maps to make a clear comparison for the completeness. As shown in Fig. 5, the second column shows the depth maps from another novel view, which has clear incomplete holes due to the occlusion, while the third column shows the coarse-completed depth maps, which are generated by the proposed NDVNet. It is worth noting that those holes of the novel view are completed by NDVNet. For example, the scene in the first, raw part of the desk is occluded by the computer. However, with NDVNet, the scene is roughly completed and most of the holes are coarsely filled. In addition, the error maps with the pixel-level difference are computed for further comparison. We subtract the prediction with the ground truth depth map. From the error maps in the fourth column, there exist many error areas compared to the ground truth maps. However, after NDVNet, the errors are significantly reduced and the error maps in the fifth column are pretty smooth.

E. Ablation Study of Various Novel View Numbers

We conduct experiments on the different choices of novel view numbers. The numbers $3, 5, 8, and 10$ are tested for comparison. As shown in Table II, overall for the CD and $C_{r=0.02}$ evaluation metric, the performance will be boosted with the increase in number of viewpoints. The best performance is achieved at the number of 8, with the CD at 0.1221, and a $C_{r=0.02}$ of 80.01%. However, when the number goes to 10, the performance is becoming stable. Considering both the efficiency and effectiveness, we choose to adopt number of 8 here for the following experiments.

TABLE II: Results comparison with different numbers of viewpoints of 3, 5, 8, and 10.

	N_3	N_5	N_8	N_{10}
CD	0.2138	0.1628	0.1221	0.1220
$C_{r=0.002}(\%)$	31.35	68.20	80.01	80.00

F. Experiment Results

Quantitative Results As shown in Table I, quantitative results compared with other state-of-the-art methods show the effectiveness of our method. Our completed scene reduces the value of CD to 0.1221. It already outperforms most of the existing methods including 3D supervised methods or volume-based methods such as SSCNet [34], [7]. Although our performance is still a little lower than DQN [14], we did not apply the reinforcement learning strategy which is adopted in [14] to select the next best views. For the completeness metric $C_{r=0.02}$, the performance is even better than [14], which highlights the improved completeness results of our model.

Qualitative Results The qualitative experimental results of two scenes are demonstrated in Fig. 6. For each scene, we select several views (v_1 ... v_4) to show the final coarse-to-fine completion results. For each view of the scene, there is a set of 3 depth maps. The top one is the depth map directly re-projected for the novel view, v_i. The middle one is the coarse-completed depth map predicted by NDVNet, and the bottom one is the depth map refined by DCNet. For the output of NDVNet, which is the coarse completed depth map, even though the image is still sort of blurry and smooth, the scene is already completed to some extent. Followed by the output of DCNet, the scene is further refined based on the coarse-completed depth map, adopting the coarse-to-fine strategy, resulting in a depth map that is clearer and sharper. Most of the occluded areas are successfully completed. For example, part of the chair and the table in the first scene are all missed due to the novel view, whereas after the completion, most of them are recovered.

V. Conclusions

We have presented a versatile model for 3D scene point cloud completion, achieving state-of-the-art completion performance. We have demonstrated that novel depth view synthesis is capable of working as a proxy task for providing adequate 2D supervision signals to the 3D scene completion task. Our proposed model can generate high-quality plausible novel depth views for supervising various potential related tasks. One promising future research direction is to complete a semantic scene point cloud in a per-class fashion.

VI. Acknowledgement

This work was supported in part by NSF under awards IIS-1400802 and IIS-2041307.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. Learning representations and generative models for 3d point clouds. In *ICLR*, 2017.

[2] Gunilla Borgefors. Distance transformations in arbitrary dimensions. *Computer vision, graphics, and image processing*, 27(3):321–345, 1984.

[3] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. *ArXiv*, abs/1908.04422, 2019.

[4] Vincent S. Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Ré, and Fei-Fei Li. Scene graph prediction with limited labels. *ICCVW*, pages 1772–1782, 2019.

[5] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li. 3d sketch-aware semantic scene completion via semi-supervised structure prior. *ArXiv*, abs/2003.14052, 2020.

[6] Bowen Cheng, Liang-Chieh Chen, Yunchao Wei, Yukun Zhu, Zilong Huang, Jinjun Xiong, Thomas Huang, Wen-Mei Hwu, and Honghui Shi. Spgnet: Semantic prediction guidance for scene parsing. *ICCV*, pages 5217–5227, 2019.

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, 2017.

[8] Angela Dai, Christian Diller, and Matthias Nießner. Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. *ArXiv*, abs/1912.00036, 2019.

[9] Marco Fraccaro et al. Generative temporal models with spatial memory for partially observed environments. In *ICML*, 2018.

[10] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2463–2471, 2016.

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

[12] Yu-Xiao Guo and Xin Tong. View-volume network for semantic scene completion from a single depth image. In *IJCAI*, 2018.

[13] Xiaoguang Han, Zhen Li, Haibin Huang, Evangelos Kalogerakis, and Yizhou Yu. High-resolution shape completion using deep neural networks for global structure and local geometry inference. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 85–93, 2017.

[14] Xiaoguang Han, Zhaoxuan Zhang, Dong Du, Mingdai Yang, Jingming Yu, Pan Pan, Xiaodong Yang, Ligang Liu, Zixiang Xiong, and Shuguang Cui. Deep reinforcement learning of volume-guided progressive view inpainting for 3d point scene completion from a single depth image. *CVPR*, abs/1903.04019, 2019.

[15] Yeping Hu, Wei Zhan, and Masayoshi Tomizuka. A framework for probabilistic generic traffic scene prediction. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2790–2796, 2018.

[16] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. *ArXiv*, abs/2003.00410, 2020.

[17] Michael M. Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32:29:1–29:13, 2013.

[18] Vladimir G. Kim, Wilmot Li, Niloy Jyoti Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Thomas A. Funkhouser. Learning part-based templates from large collections of 3d shapes. *ACM Trans. Graph.*, 32:70:1–70:12, 2013.

[19] Siqi Li, Changqing Zou, Yipeng Li, Xibin Zhao, and Yue Gao. Attention-based multi-modal fusion network for semantic scene completion. *ArXiv*, abs/2003.13910, 2020.

[20] Yangyan Li, Xiaokun Wu, Yiorgos Chrysanthou, Andrei Sharf, Daniel Cohen-Or, and Niloy Jyoti Mitra. Globfit: consistently fitting primitives by discovering global relations. *ACM Trans. Graph.*, 30:52, 2011.

[21] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[22] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018.

[23] Niloy Jyoti Mitra, Leonidas J. Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3d geometry. In *SIGGRAPH 2006*, 2006.

[24] L NavaneetK, Priyanka Mandikal, Mayank Agarwal, and R. Venkatesh Babu. Capnet: Continuous approximation projection for 3d point cloud reconstruction using 2d supervision. *CoRR*, abs/1811.11731, 2019.

[25] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136. IEEE, 2011.

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[27] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3308–3317, 2018.

[28] Jason Rock, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem. Completing 3d object shape from one depth image. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2484–2493, 2015.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[30] Ruwen Schnabel, Patrick Degener, and Reinhard Klein. Completion and reconstruction with primitive shapes. *Comput. Graph. Forum*, 28:503–512, 2009.

[31] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. Structure recovery by part assembly. *ACM Trans. Graph.*, 31:180:1–180:11, 2012.

[32] Yifei Shi, Angel Xuan Chang, Zhelun Wu, Manolis Savva, and Kai Xu. Hierarchy denoising recursive autoencoders for 3d scene layout prediction. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1771–1780, 2019.

[33] Ivan Sipiran, Robert Gregor, and Tobias Schreck. Approximate symmetry detection in partial 3d meshes. *Comput. Graph. Forum*, 33:131–140, 2014.

[34] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas A. Funkhouser. Semantic scene completion from a single depth image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 190–198, 2016.

[35] Olga Sorkine-Hornung and Daniel Cohen-Or. Least-squares meshes. *Proceedings Shape Modeling Applications, 2004.*, pages 191–199, 2004.

[36] Olga Sorkine-Hornung and Daniel Cohen-Or. Least-squares meshes. *Proceedings Shape Modeling Applications, 2004.*, pages 191–199, 2004.

[37] Minhyuk Sung, Vladimir G. Kim, Roland Angst, and Leonidas J. Guibas. Data-driven structural priors for shape completion. *ACM Trans. Graph.*, 34:175:1–175:11, 2015.

[38] Xiaogang Wang, Marcelo H. Ang, and Gim Hee Lee. Cascaded refinement network for point cloud completion. *ArXiv*, abs/2004.03327, 2020.

[39] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Adversarial semantic scene completion from a single depth image. *2018 International Conference on 3D Vision (3DV)*, pages 426–434, 2018.

[40] Henglai Wei, Xiaochuan Yin, and Penghong Lin. Novel video prediction for large-scale scene using optical flow. *ArXiv*, abs/1805.12243, 2018.

[41] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018.

[42] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Interpretable unsupervised learning on 3d point clouds. *ArXiv*, abs/1712.07262, 2017.

[43] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. *2018 International Conference on 3D Vision (3DV)*, pages 728–737, 2018.

[44] Wei Ke Zhao, Shuming Gao, and Hongwei Lin. A robust hole-filling algorithm for triangular mesh. In *CAD/Graphics*, 2007.