Received January 22, 2020, accepted February 6, 2020, date of publication February 10, 2020, date of current version February 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973003

Sparse-to-Dense Multi-Encoder Shape Completion of Unstructured Point Cloud

YANJUN PENG[1], MING CHANG[1], QIONG WANG[2], YINLING QIAN[2], YINGKUI ZHANG[2], MINGQIANG WEI[3], AND XIANGYUN LIAO[2]

[1]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[2]Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
[3]School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

Corresponding authors: Qiong Wang (wangqiong@siat.ac.cn) and Yinling Qian (qianyinling@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61976126, Grant 61902386, and Grant 61802386, in part by the National Natural Science Foundation of China under Grant U1813204, in part by the Shenzhen Science and Technology Program under Grant JCYJ20180507182410327 and Grant JCYJ20180507182415428, and in part by the Natural Science Foundation of Shandong Province under Grant ZR2019MF003.

ABSTRACT Unstructured point clouds are a representative shape representation of real-world scenes in 3D vision and graphics. Incompletion inevitably arises, due to the way the set of unorganized points is captured, e.g., as fusion of depth images, merged laser scans, or structure-from-x. In this paper, an end-to-end sparse-to-dense multi-encoder neural network (termed an SDME-Net) is proposed for uniformly completing an unstructured point cloud with its shape details preserved. Unlike most existing learning-based shape completion methods that are enforced on the representations of 2D images and 3D voxelization of point clouds, and require priors of the underlying shape's structures, topologies and annotations, the SDME-Net is implemented on the incomplete and even noisy point cloud without any transformation, and makes no specific assumptions about the incompletion distribution and geometry features in the input. Specifically, the defective point cloud is completed and optimized in a sparse-to-dense manner of two-stages. In the first stage, we generate a sparse but complete point cloud based on a bistratal PointNet, and in the second stage, we yield a dense and high-fidelity point cloud by encoding and decoding the sparse result in the first stage using PointNet++. Meanwhile, we combine the distance loss and repulsion loss to generate more uniformly distributed output point clouds closer to the ground-truth counterparts. Qualitative and quantitative experiments on the public ShapeNet dataset illustrate that our approach outperforms the state-of-art learning-based point cloud shape completion methods in terms of real structure recovery, uniformity, and noise/partiality robustness.

INDEX TERMS Deep learning, incomplete point cloud, joint loss function, multiple encoders, PointNet, shape completion.

I. INTRODUCTION

Since early 1985s, point cloud has been recognized as a representative form of 3D objects which is widely used as the standard output of various sensors [1]. Recently, smart geometry processing of point clouds again entered into the spot of 3D vision and graphics, due to the rapid advances and applications of artificial intelligence (AI) in robotics, autonomous driving and mixed reality [2]–[4].

A large number of point clouds require completion to be transferred to downstream tasks from surface reconstruction and scene understanding to rendering. Shape completion of point cloud seeks to repair an incomplete point cloud to recover its real geometry. However, the raw point cloud contains incompletion due to imperfections of the capturing procedure (e.g., occlusion, motion, and multiple reflections), and it also varies largely in geometry and sampling [5]. These factors are causing the shape completion problem of point cloud to be ill-posed, since the underlying surface of an incomplete point cloud is commonly unknown. Therefore, existing shape completion algorithms cannot be effective for incomplete point clouds of all characteristics.

Recently, many effective learning-based techniques have been proposed to process point-based tasks.

The associate editor coordinating the review of this manuscript and approving it for publication was Yongping Pan.

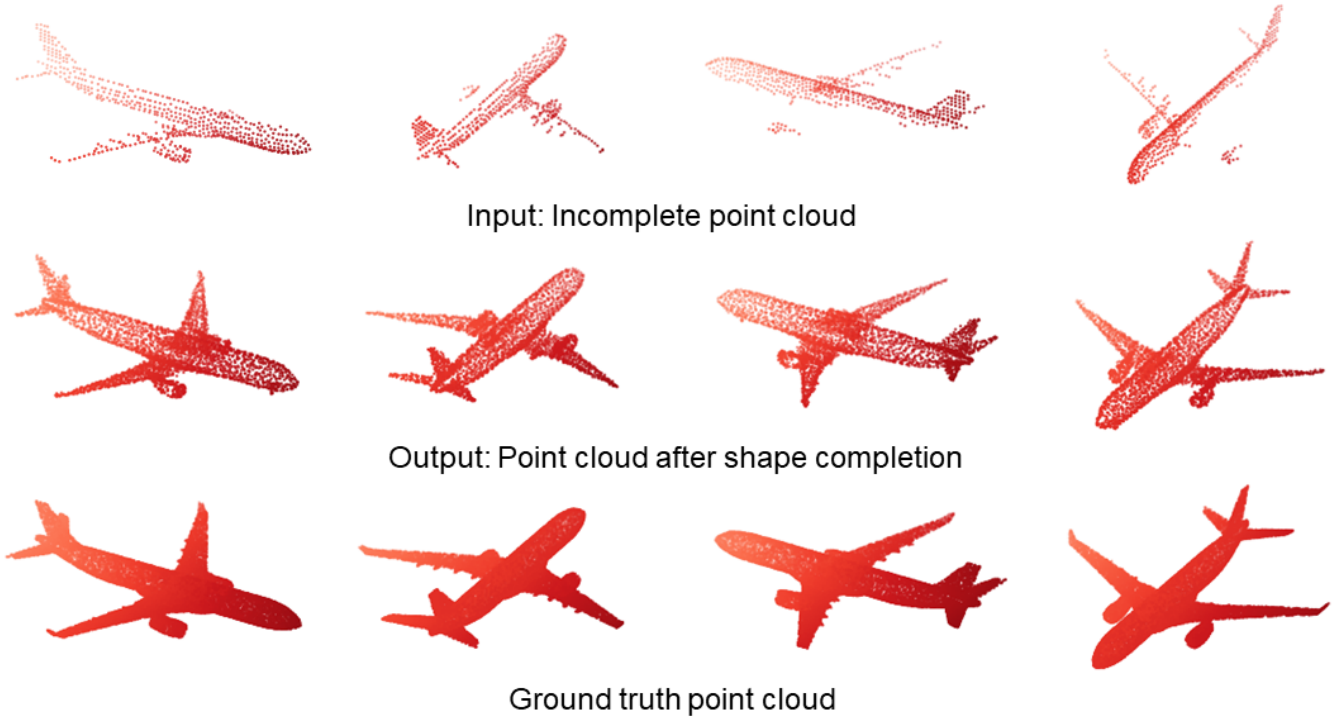


FIGURE 1. SDME-Net can consistently produce high-quality completion results which are faithful to the ground truths. Multi-view visualization of the input point cloud, our completion result, and the ground truth point.

Volumetric approaches [6]–[9] such as distance fields and binary voxel grids possess the advantage of easy application of convolution neural network, and have achieved some successes. However, the memory costs of these methods also cubically increase, which limited their achievable resolution and scalability of network architecture. Further, voxelization leads to the easy loss of detailed geometric information. As a sparse and disordered data form, point cloud can more concisely and efficiently represent the shape of 3D objects. On the application of convolutional neural networks to point clouds [10]–[13] has also made great progress. Neural network methods for processing point cloud data are increasingly used in 3D target recognition, scene reconstruction [13]–[16], target object completion [6], [7], [17], [18], and 3D shape representation learning [13], [16], [19].

Due to the lack of resolution of the sensor, occlusion and other issues, the 3D point cloud scanned in reality is often incomplete, and some geometric and semantic information are lost, making it very difficult to complete the tasks mentioned above. Therefore, 3D point cloud shape completion is a common subtask of these tasks. This paper proposes a sparse-to-dense multi-encoder shape completion approach for point clouds, a specific example is shown in Fig. 1.

Different from some existing point cloud generation methods, our network directly codes the raw input point cloud, avoiding the loss of geometric information caused by projection to 2D images or voxelization, and generates a more fine-grained complete object point cloud in two stages at a low memory cost. In addition, for input point clouds with less points and noise, our method can also complete the shape completion task and has good robustness.

More specifically, for the point cloud shape completion mission, our encoder embeds an incomplete input point cloud as a feature vector, the first stage (Section III-B) performs feature extraction on these feature vectors, and then uses a fully connected network [6], [13], [19] to generate the complete point cloud. Due to the low resolution of the output point cloud generated by the fully connected network, the huge computation and memory costs, we only generate a sparse but complete point cloud in this stage. In the second stage (Section III-C), a point cloud encoder is used to encode the output of the first stage. We use a parameter-sharing decoder to expand the number of points and concatenate features of 2D point-grid to optimize the representation of the point cloud shape edges. Finally, the network generates a dense high-resolution complete output point cloud.

The main contributions of this work are summarized as follows:

- We propose a learning-based shape completion approach, which works directly on the raw point cloud without transforming the data into other representations or any assumptions of specific structure.
- We design a two-stage sparse-to-dense multi-encoder network architecture that generates dense, high-resolution, complete point clouds.
- We combine the distance loss and repulsion loss to design a new joint loss function, which makes the output point clouds more uniformly distributed similar to the ground-truth counterparts.

Experimental results show that our approach is robust to sparse and noisy input point cloud, and the completed point

cloud is more accurate and uniform, and the performance is better than previous methods.

II. RELATED WORK

A. DEEP LEARNING ON 3D SHAPES

Volumetric representation of 3D shapes can be easily fed into to convolution neural network. As a result, more and more 3D shape learning and representation based on deep learning have spawned [6], [18], [20], [21], but the memory and computation costs of the 3D convolution are expensive, and the low accuracy caused by the 3D voxel grid has become an insurmountable limitation of such methods. Multi-view representation method [22], [23] is an alternative of volume representation, the convolutional neural network based on this kind of method has good performance in 3D shape recognition, 3D object classification and other tasks, but it is difficult to solve problems such as 3D shape reconstruction and completion due to the limitation of accuracy of 2D image projection to 3D space. PointNet [11] was the pioneer of applying neural network directly to point cloud, since then, point cloud has been widely used in 3d shape representation and learning.

B. 3D SHAPE COMPLETION

There are many methods focusing on shape completion of 3D objects. Some of the methods based on geometric [24]–[26] assumption that the incomplete part and one part of the input are geometrically symmetric, they could get the structure of local missing parts according to the observation area and generate smooth interpolation to the completion of incomplete input. The assumptions greatly limit the practical application of these methods. Methods based on database matching [27], [28] are trying to establish an appropriate standard model in a large 3D shape database to match the input shape, and then complete the missing part of the input shape according to the standard model, such methods require little noise in the input 3D shape, and the accuracy of matching and completion is limited by the size of the database. Some other methods are based on deep learning and most of them [6], [7], [29], [30] use voxels to represent 3D shapes, because this is beneficial for the application of 3D convolutional networks, but the volume method has high cost and low accuracy. A recent work [31] proposed an end-to-end point cloud completion network, and the network directly processes the incomplete input point cloud to generate complete point clouds, showing excellent performance and far superior to the methods based on volume representation in terms of cost and accuracy of 3D shape generation.

C. DEEP LEARNING ON POINT CLOUDS GENERATION

Fan *et al.* [15] first proposed a method for generating point clouds using neural networks. Their proposed point cloud encoder encodes the input 2D images into high-dimensional features and then decodes them to generate point clouds. They also introduced two loss functions, Chamfer Distance (CD) and Earth Mover's Distance (EMD) to represent the fitting degree of two point sets. However, the resolution and geometric details of the point cloud shape generated by the image are not good, and the point cloud generated by these methods is unstructured, there is no structured relationship between points with similar positions. There are other ways [13], [16], [31] to generate a manifold structure that is close to the real object point cloud by forcing a specific structured representation, but forcing a certain structure will also constrain the learning process of the method. In order to avoid the defect of the above two kinds of methods, we concatenate 2×2 point grid features to the point set features in the decoder part. In this way, the generated point cloud can have certain structure, thus improving the performance of our method in learning 3D shape.

D. POINT CLOUD UPSAMPLING

The upsampling problem of point clouds is essentially similar to the super-resolution problem of images, but simple interpolation between input points does not give satisfactory results, thus processing 3D points instead of 2D pixel grids brings new challenges. Early methods to solve this problem [33], [38] were based on optimization, which used various shape priors to constrain the generation of point clouds. Recently, deep neural networks have brought an alternative data-driven approach to this problem. PU-Net [35] shows the advantages of upsampling point clouds by learning.

III. METHOD

A. NETWORK ARCHITECTURE

As shown in Fig. 2, the whole network architecture of the SDME-Net is two continuous point cloud coders, where *N* and *Ni* are the number of points, *Ci* are the number of feature channels, and the colored rounded rectangles represent certain baseline network architectures. It generates the final complete point cloud from the input incomplete point cloud in two stages, and the output point cloud is not explicitly forced to retain the position information of the input point. The input is the point cloud data obtained from an observation perspective of the object, and the point position in the point cloud is on the observation surface of the object. Due to occlusion, the point cloud obtained from a certain perspective is incomplete. We take the complete point cloud sampled uniformly from the entire surface of the object as the ground truth. In the first stage, we encode the input point cloud to get a high-dimensional feature vector, and then decode the feature vector to output a sparse point cloud with complete shape. In the second stage, we take the output point cloud of the first stage as the input, and get the final dense complete point cloud through encoding, feature expansion and decoding. We designed a joint loss function to train the whole network through backpropagation. The visual results of some examples are shown in Fig. 3.

B. STAGE I: SPARSE AND COMPLETE POINT CLOUD GENERATION

Among the current methods for point cloud completion in deep learning, PCN [31] is recognized as the best method. Part of the network that complements sparse point clouds

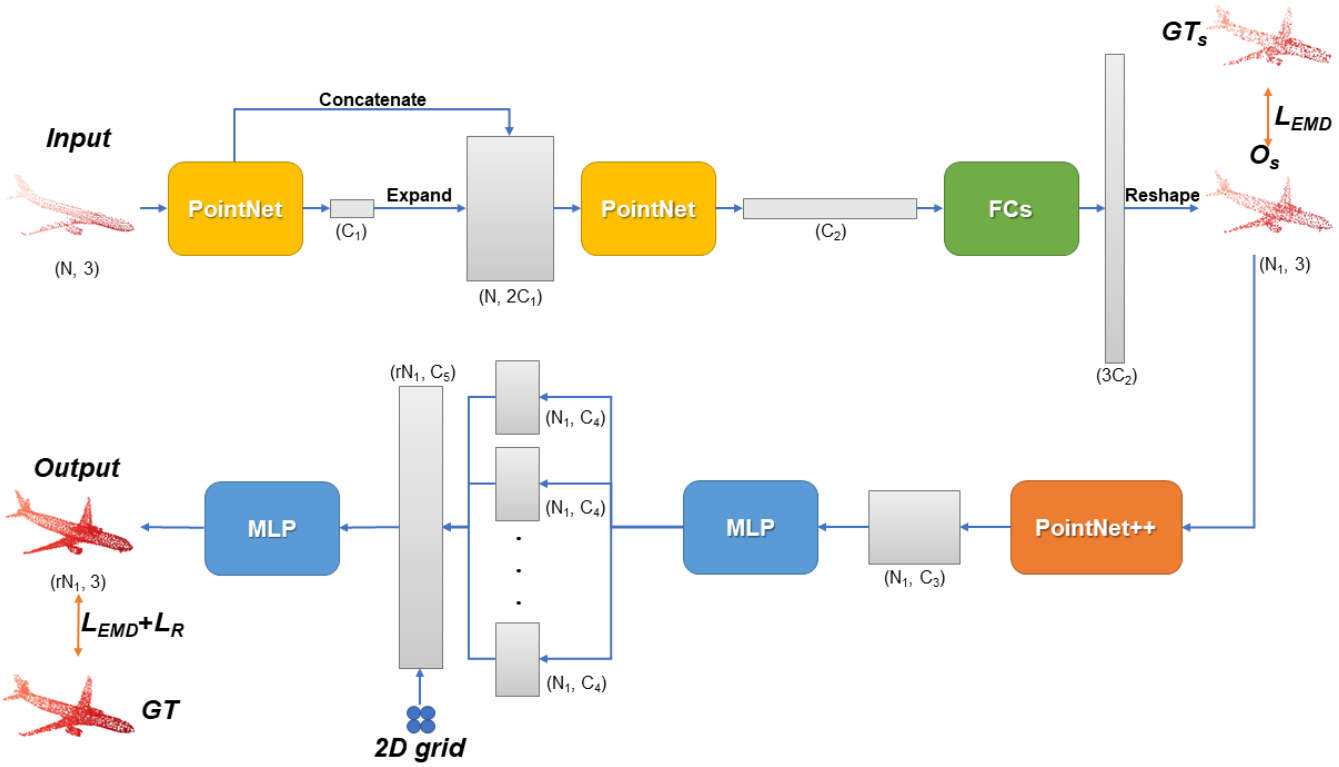


FIGURE 2. The architecture of SDME-Net. The upper and lower layers are two different point cloud coder-decoders, they are the two stages of the network. The specific details are described in Section III.

performs well. In this stage we refer to part of the structure of PCN. The experiment of PCN showed that PointNet++ was not better than PointNet in completing a sparse point cloud, and our experiment also verified this. Thus as shown in Fig. 2, we first use two consecutive PointNet layers to extract features of the input point cloud whose size is $N \times 3$ to get a 1024-dimensional feature vector, where each gray rectangle in the architecture denotes a matrix, N is the number of points of the input point cloud, this parameter can be changed according to the input point cloud of different data sets, the 3 represents three XYZ coordinate values of each point in the 3D space. Specifically, each of our PointNet network layers consists of a two-layer parameter sharing multi-layer perceptron (MLP) [11] and a maximum pooling layer. MLP uses linear layers and the ReLU activation function to raise the feature dimension of each point in the input point cloud to learn the features of the point, and then obtains a high-dimensional global feature of the input point cloud through a maximum pooling layer. Finally, a complete point cloud of 1024×3 is obtained by projecting the high-dimensional feature vector into 3D space using a three-layer fully connected network. Thanks to PointNet's invariance to permutation and robustness to noise, the point cloud encoder can learn the global feature of the input point cloud, and finally get a point cloud of which the missing parts have been completed.

C. STAGE I: DENSE HIGH-RESOLUTION AND COMPLETE POINT CLOUD GENERATION

In the first stage, we are committed to the global feature learning directly from the input point cloud, using PointNet.

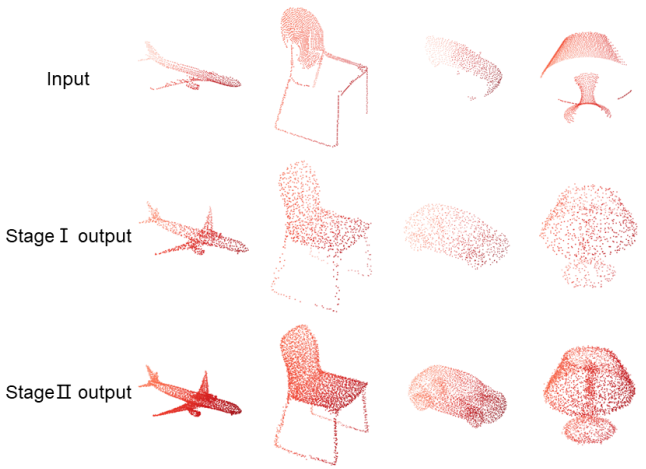


FIGURE 3. Examples of SDME-Net's output visualization. The first row is the input point cloud, the second row is the output of Stage *I*, and the third row is the output of Stage *II*.

However, PointNet has certain limitations in learning local features of point cloud. As a result, the first stage only generates a sparse complete point cloud with a relatively accurate global geometry, but can't produce fine local details. The current optimal solution PCN combines a 4×4 2D point grid into each point of the sparse point cloud, the deformation grid is folded to represent the distribution of local point sets where these points are located, so as to improve the density and resolution of the output point cloud. However, the quality of the output highly depends on that of the input sparse point cloud, and the mandatory addition of a 2D point grid structure

will also constrain the network learning process of point cloud structure.

To overcome this limitation, we adopt PointNet++ [12] as the encoder to encode and decode the sparse point cloud generated in the first stage again. The hierarchical feature learning structure of PointNet++ has been proven to be able to learn the local and global features of point cloud simultaneously. Then our network decodes the features of different levels and concatenates them, and then gets low dimension features of point cloud through a three-layer MLP decoding, and repeats r times to achieve the result of expanding the number of features by r times like PU-Net [35], this can prevent the expanded points from being too close, and the MLP used for r times decoding is parameter sharing which can greatly improve the efficiency of the point set expansion. Finally, we use a three-layer MLP to decode the expanded feature and reduce the feature dimension to 3 to regress the 3D coordinates of the final point cloud.

D. LOSS FUNCTION

1) DISTANCE LOSS

In order to encourage the position of the generated output point cloud to be closer to the ground truth point cloud, we choose the Earth Mover's Distance (EMD) [15], [32] as the loss function to evaluate the approximate degree of the point cloud generated by the network prediction and the ground truth point cloud. In practical application, the Chamfer Distance (CD) [15] is another candidate method for evaluating the similarity of two point sets. However, the research results of Fan *et al.* [15] show that EMD can better capture the shape of objects compared with CD, thus we choose EMD as our distance loss function.

$$L_{EMD}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \frac{1}{|S_1|} \sum_{x \in S_1} \|x_i - \phi(x_i)\|_2 \quad (1)$$

EMD (1) finds a bijection mapping $\phi : S_1 arrow S_2$ to minimize the average distance between the predicted point cloud $S_1 \subseteq \mathbb{R}^3$ and the corresponding ground truth point cloud $S_2 \subseteq \mathbb{R}^3$, where x, x_i are points in S_1, and S_1, S_2 are required to have the same size.

$$L_{CD}(S_1, S_2) = \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2 + \frac{1}{|S_2|} \sum_{y \in S_2} \min_{x \in S_1} \|y - x\|_2 \quad (2)$$

CD (2) calculates the average closest point distance between the point set S_1 and the point set S_2, where x are points in S_1, y are points in S_2, and S_1, S_2 do not have to be the same size.

2) REPULSION LOSS

Although the positions of the generated points using the distance loss function training tend to be very close to the point of the ground. However, due to the limitations of the distance loss function and the point sampling algorithm, the positions of some points in the resulting point cloud tend to be very close or even coincident, so we incorporate the repulsion loss [35] to make the distribution of the resulting points set more uniform, expressed as:

$$L_R(S) = \sum_{i=0}^{N} \sum_{i' \in K(i)} \eta(\|x_{i'} - x_i\|)\, w(\|x_{i'} - x_i\|) \quad (3)$$

where N is the number of points in the generated point cloud $S \subseteq \mathbb{R}^3$, $K(i)$ is the index of the k-nearest neighbors of point x_i, and $\eta(r) = -r$ is a decreasing function used to punish x_i if x_i is too close to its neighbor points in $K(i)$. $w(r)$ is a fast-decaying weight function, in our experiments we refer to some methods [33]–[35] setting $w(r) = e^{-r^2/h^2}$ where h is a support radius defining the size of the influence neighborhood, and $\|\cdot\|$ is the L2-norm.

3) JOINT LOSS FUNCTION

We designed a new joint loss function to train our network in an end-to-end manner. This joint loss function consists of two parts, where O_s is the sparse point cloud of the first stage output, *Output* is the final output point cloud, *GT* is the ground truth point cloud, GT_s is the sub-point cloud of *GT* and its number of points is the same as O_s.

$$\begin{aligned} &L(O_s, GT_s, Output, GT) \\ &\quad = L_{EMD}(O_s, GT_s) \\ &\qquad + \alpha\left(L_{EMD}(Output, GT) + \beta L_R(Output)\right) \end{aligned} \quad (4)$$

The first part is the EMD loss of the sparse point cloud of the first stage output and the sparse ground truth point cloud. The second part is the EMD loss of the final dense output point cloud and the dense ground truth point cloud plus β times the repulsion loss of the final generated output point cloud, where the value of the β is 10^{-2}. These two parts are weighted by the hyperparameter α. During the training process, the weight α is a variable. Since the first part has not converged in the initial stage of the training process, the weight α will be set to be very small. As the training progresses, α will gradually increase to 1, so that the weights of the two parts are the same.

IV. EXPERIMENT

In this section, we first describe the employed dataset, as well as the specific parameter settings for our network training, and then quantitatively and qualitatively evaluate our method and other existing methods for shape completion tasks. Finally, we will use these methods to complete some of the actual examples of point clouds and show the visual results of their completions.

A. DATASET

We use point cloud data generated from a subset of the ShapeNet [36] dataset as our dataset. This dataset was created by Yuan and Held [31] and contains airplanes, cabinets, cars, chairs, lamps, sofas, tables and vessels in ShapeNet. There are 30,974 different CAD models in 8 categories.

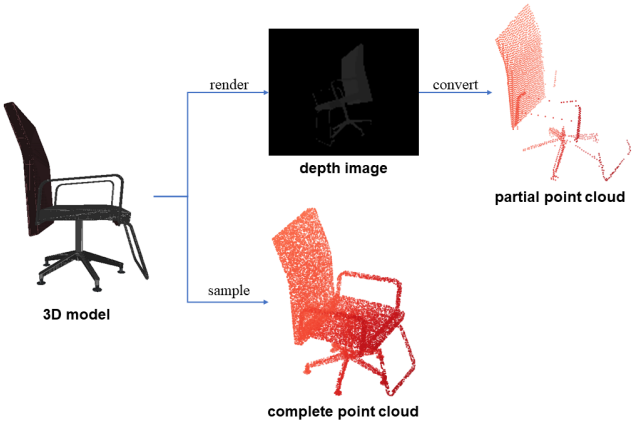


FIGURE 4. An instance of the dataset samples generation process.

Each model randomly takes 8 viewpoints, and then back projects the 2.5D depth image obtained from each viewpoint to generate an incomplete local point cloud as input, a simple example is shown in Fig. 4. Here, the incomplete point cloud can have different sizes, and finally, 16384 points are uniformly sampled from the surface model of the CAD model under this viewpoint as the ground truth complete point cloud corresponding to the incomplete point cloud. The back projection using the 2.5D depth image generates an incomplete point cloud in order to make the distribution of the input point cloud closer to the data collected by the real sensor.

B. IMPLEMENTATION DETAILS AND TRAINING SETUP

In our experiments, we used an input point cloud with 2048 points and a ground truth with 4096 points, they were obtained by random subsampling from the above dataset. In the point cloud encoder we used in the first stage, the two MLP output feature dimensions of the encoder are 256 and 1024, respectively, and the sparse complete point cloud size of the fully connected network output is 1024×3. In the second stage of the point cloud encoder, we used a four-layer PointNet++, the output feature dimensions are 64, 128, 256, 512, and the sub-point cloud space radii are 0.02, 0.04, 0.08, 0.12 (complete point cloud radius is 0.2), respectively. In the decoder, the point cloud expansion factor r is set to 4, the output feature dimension is 128, and then the 2×2 2D point grid feature is concatenated. Finally, after a three-layer MLP, the output feature dimension is 3, and the final completion point cloud is obtained, the size is 4096×3. In the loss function, the initial value of α is 0.01. After 8 epochs, α gradually increases to 1 and does not change. β is 0.01, and the parameters k and h in the repulsion loss function are set to 5 and 0.03, respectively. We implemented this work based on TensorFlow, using a single NVIDIA 1080Ti GPU with 11G graphics memory. We used the Adam [37] algorithm to optimize our model training. The training lasted 50 epochs with a batch size of 32. The initial learning rate was 0.0001, the learning rate is attenuated by 0.7 for every 50,000 iterations.

C. EVALUATION

We evaluate the performance of the model from the point cloud completion accuracy of the eight categories in the dataset. For each category, we calculate the mean EMD between all instance point clouds generated by the model prediction and their corresponding ground truth point clouds, then, the final metric is the mean EMD of all categories. The smaller the value of the mean EMD is, the more similar the complete point cloud generated by the model prediction and the ground truth point cloud is, the better the performance of the model is.

D. COMPARISONS WITH OTHER METHODS

We have chosen three existing strong baseline methods and one method for the current optimal performance of this task. These four methods can train a network model in an end-to-end manner, just like our method. We trained the models of the four methods with the same dataset and same training setup, and then compared the completion results.

FC: In the decoding part of the encoder, FC [19] has excellent performance in mapping high-level feature vectors to 3D space. In this network, the same encoder network as SDME-Net is used, and then the decoder network sets up a three-layer fully connected network to directly output the coordinates of the final point cloud.

Folding: In this network architecture, like the FC encoder mentioned above, the encoder part is also the same as SDME-Net. Its decoder part uses a 128×128 2D point grid, which is deformed by folding [13] and transformed into the output point cloud.

PN2: PointNet++ [12] is a baseline method improved by PointNet [11]. It has great improvement in point cloud local feature learning. In the PN2, the encoding part uses a three-layer PointNet++ instead of the continuous two PointNet layers of the SDME-Net. The decoding part uses the same network architecture as our SDME-Net.

PCN: This is the first method to do point cloud completion using a point cloud encoder. It has good results and is one of the best performance methods in the task. SDME-Net's first stage encoder uses some of its network architecture. Unlike our approach, PCN [31] does not use an encoder when generating a dense output point cloud in the second stage. Instead, it extends the first-stage high-dimensional features, then merge the sparse point cloud and the 4×4 2D point grid feature, and finally decode to generate the final output point cloud.

Our method uses EMD as the loss function, and finally output 4096 points, so we also use the same loss function and output size when training the models of the above methods. To quantitatively evaluate the merits of these methods, we calculate the mean EMD of the generated point cloud and the ground truth point cloud with the same test. The quantitative comparison results are shown in Table 1, where the values of Earth Mover's Distance are reported multiplied by 10^2. We also trained models of the methods using CD as the loss function. The results are shown in Table 2, and the values

TABLE 1. Quantitative comparison on ShapeNet.

Method	Mean Earth Mover's Distance								
	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Vessel	**Average**
FC	5.66	17.61	7.14	8.58	16.89	13.85	11.67	10.92	11.54
Folding	12.03	18.15	14.32	13.46	22.46	17.77	13.75	16.85	16.1
PN2	4.58	9.53	5.49	5	13.04	7.42	6.23	6.45	7.22
PCN	2.98	5.8	4.51	3.08	5.94	5.24	4.35	4.83	4.59
Ours	**2.93**	**5.64**	**4.35**	**3.07**	**5.18**	**5.09**	**3.64**	**4.7**	**4.33**

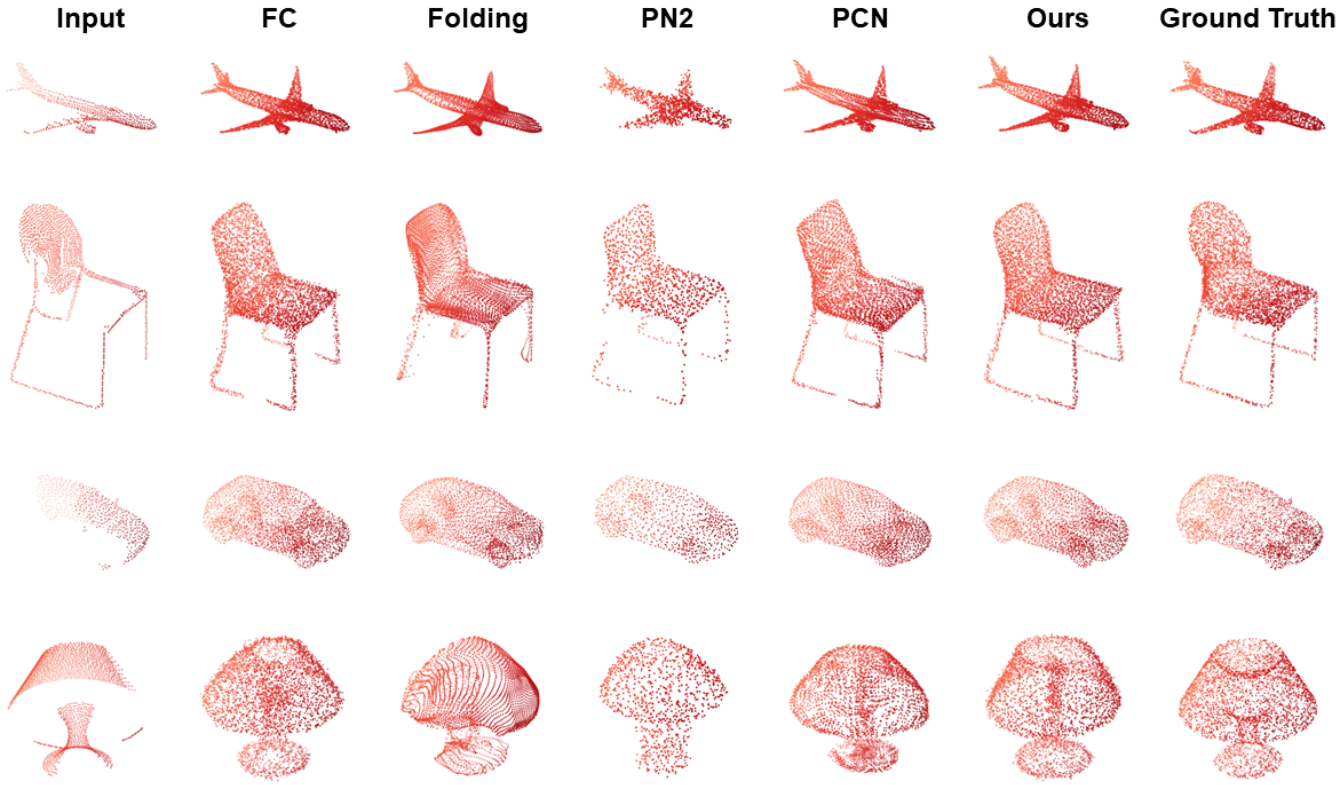


FIGURE 5. Qualitative completion results on ShapeNet. The methods generate complete point clouds from the input partial point cloud.

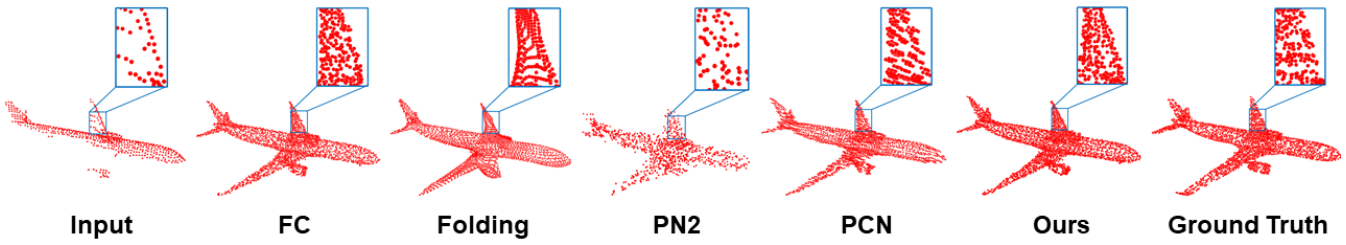


FIGURE 6. The comparison of partial enlarged visualization results. We can see the uniformity of the point cloud generated by these methods.

TABLE 2. Comparison of mean chamfer distance.

Method	FC	Folding	PN2	PCN	Ours
CD	1.81	1.87	2.59	1.78	**1.7**

of Chamfer Distance are reported multiplied by 10^2 too. The qualitative comparison of some specific examples is shown in Fig. 5. The comparison of partial enlarged visualization results of these methods are shown in Fig. 6. It can be seen that the point cloud generated by our method has better uniformity.

E. NUMBER OF PARAMETERS

As shown in Table 3, compared with PCN, whose overall architecture is similar to ours, our method adopted a dual encoder which can improve the performance while slightly increasing the parameter number.

F. ROBUSTNESS TO NOISE AND OCCLUSION

We tested the robustness of our shape completion method to sensor noise and large area occlusion which is the most common course of varying partiality. Specifically, we used

TABLE 3. Number of trainable model parameters.

Method	FC	Folding	PN2	PCN	Ours
#Params	46.55M	7.21M	20.29M	20.58M	20.73M

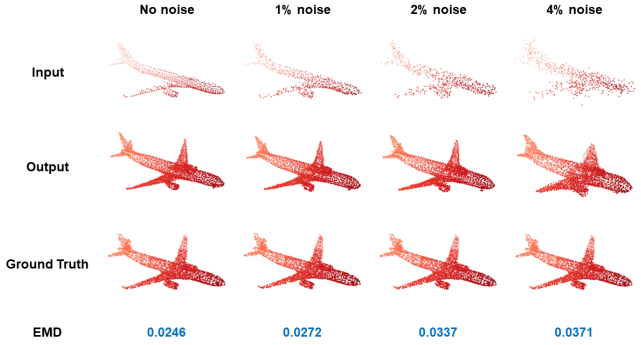


FIGURE 7. Point cloud completion results with different levels of noise.

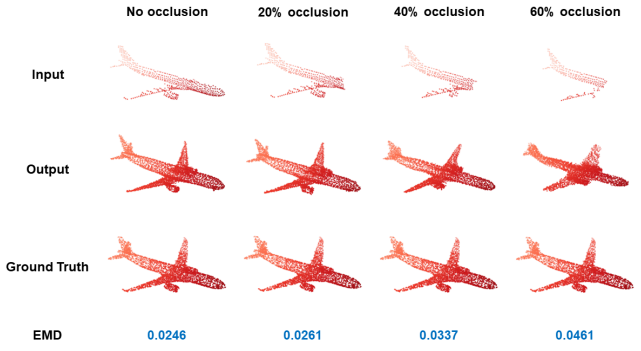


FIGURE 8. Point cloud completion results with different levels of occlusion.

Gaussian noise with 0.01 times, 0.02 times, and 0.04 times standard deviations the scale of the depth measurements to perturb the depth map. We occluded the depth map with a mask that covers p percent of points, where p ranges from 0% to 60%. It can be seen from Fig. 7 and Fig. 8 that as the noise or occlusion increases, the EMD will gradually increase. Note that our model is not trained using these noise and occlusion examples, but it is still robust to these examples. This shows, to some extent, its powerful generalizability to the data of real world.

G. ARCHITECTURE DESIGN ANALYSIS

1) MULTIPLE ENCODER ARCHITECTURE

SDME-Net's network architecture is similar to that of PCN. It is to complete the point cloud and then improve the resolution of the point cloud in two stages. Unlike PCN, we use a dual encoder architecture. In the second stage, we propose a PointNet++ encoder structure with strong point cloud local feature learning ability, it can avoid the excessive dependence on the low-resolution point cloud output of the first stage fully connected network decoder. We compare the color-code completion results of the single encoder SDME-Net (SDME-NetSE), PCN and SDME-Net in Fig. 9. We use the Chamfer Distance multiplied by 10^3 to show the deviations, where blue indicates low deviation and red indicates high deviation.

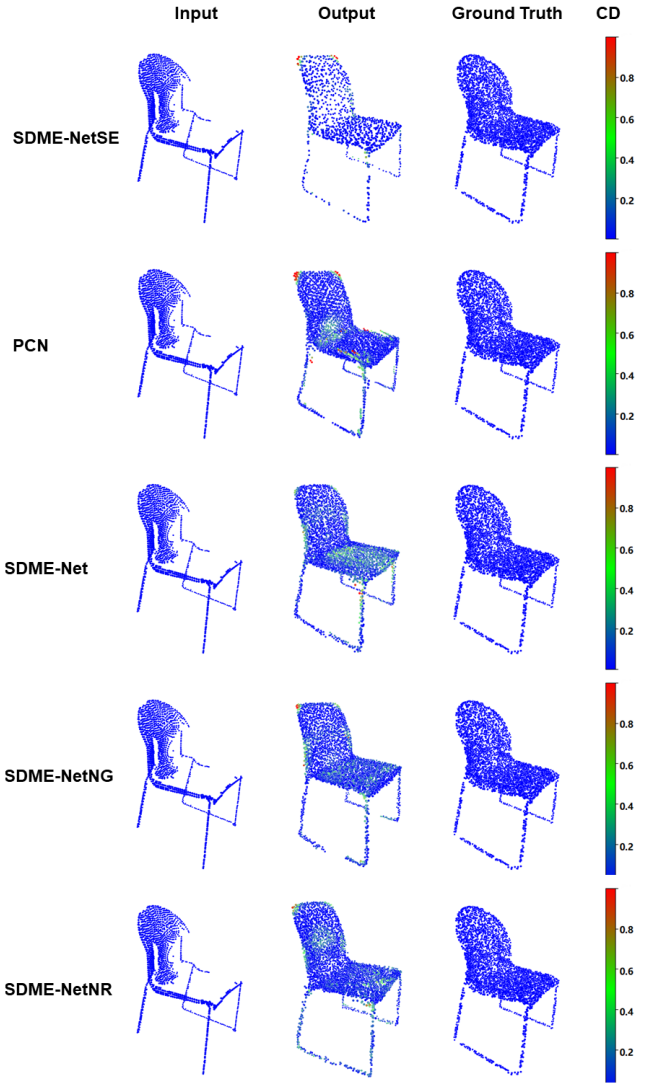


FIGURE 9. Visual comparison of completion results of several methods. We color-code all point clouds, and the colors on the points (see color map) reveal the deviation from the ground truth point cloud.

We can see that the model generated by our network has a significant improvement in local details. We refer to the PU-Net [35] in the second stage, using a two-layer parameter sharing MLP to expand the number of points, it allows our network having a four-layer PointNet++ encoder to have almost the same memory cost as the PCN with a single encoder architecture.

2) 2D POINT GRID ANALYSIS

Both Folding and PCN adopt a strategy of forcing the concatenation of 2D point grid features. By visualizing the experimental results, we can find that the edge of the generated point cloud geometry is very smooth using these methods, but because of the addition of a 2D grid structure to the point, the performance of the network learning point cloud structure is still greatly limited. We combine the advantages of the unstructured point cloud generation method and the structural assumption point cloud generation method, after the second

stage of the encoding part, we concatenate the features of the 2×2 2D point grid. Fig. 9 shows a comparison between SDME-Net without point grid features (SDME-NetNG) and SDME-Net completion results with point grid features, note the completeness of the completed models. Experimental results show that concatenating the feature of a 2D point grid can improve the quality of the completed point cloud.

3) REPULSION LOSS ANALYSIS

In the second stage of our shape completion network, we add the repulsion loss to the loss function, which makes the points generated after the point cloud upsampling will not be located too close to each other or even overlap, so that the final completed point clouds have good uniformity, and this is very useful for some point cloud applications, such as surface reconstruction, denoising and so on. In addition, our experimental results show that adding the repulsion loss will slightly reduce the mean Earth Mover's Distance between the generated point cloud and groud truth. Compared to the network without repulsion loss, the accuracy of point cloud shape completion is improved by 7.6%. Fig. 9 shows a comparison between SDME-Net without repulsion loss (SDME-NetNR) and SDME-Net with repulsion loss. The experimental results can be seen that the point cloud generated by SDME-Net is more uniform.

V. CONCLUSION

This paper proposes a new network to accomplish the task of point cloud shape completion. It directly processes the raw input point cloud with certain noise without any voxelization and any structural assumption. The point cloud generated by multi-layer encoder has better local details and better resolution than previous methods. The method is effective for the shape completion of a variety of object categories with different geometric features. This makes it convenient to be applied in the real scene, or it can be extended to the shape completion of the target object in the scene point cloud, improving the accuracy of the target object detection, recognition and other tasks. On the other hand, the sparse 3D representation of point cloud is also more suitable for real-time computing tasks, such as autonomous driving and augmented reality.

REFERENCES

[1] H. Chen, M. Wei, Y. Sun, X. Xie, and J. Wang, "Multi-patch collaborative point cloud denoising via low-rank recovery with graph constraint," *IEEE Trans. Visual. Comput. Graph.*, to be published.

[2] C. Yi, D. Lu, Q. Xie, S. Liu, H. Li, M. Wei, and J. Wang, "Hierarchical tunnel modeling from 3D raw LiDAR point cloud," *Comput.-Aided Des.*, vol. 114, pp. 143–154, Sep. 2019.

[3] M. Wei, J. Huang, X. Xie, L. Liu, J. Wang, and J. Qin, "Mesh denoising guided by patch normal co-filtering via kernel low-rank recovery," *IEEE Trans. Visual. Comput. Graph.*, vol. 25, no. 10, pp. 2910–2926, Oct. 2019.

[4] J. Wang, J. Dai, K.-S. Li, J. Wang, M. Wei, and M. Pang, "Cost-effective printing of 3D objects with self-supporting property," *Vis. Comput.*, vol. 35, no. 5, pp. 639–651, May 2019.

[5] C. Yi, Y. Zhang, Q. Wu, Y. Xu, O. Remil, M. Wei, and J. Wang, "Urban building reconstruction from raw LiDAR point data," *Comput.-Aided Des.*, vol. 93, pp. 1–14, Dec. 2017.

[6] A. Dai, C. R. Qi, and M. Niebner, "Shape completion using 3D-encoder-predictor CNNs and shape synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5868–5877.

[7] X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu, "High-resolution shape completion using deep neural networks for global structure and local geometry inference," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 85–93.

[8] O. Litany, A. Bronstein, M. Bronstein, and A. Makadia, "Deformable shape completion with graph convolutional autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1886–1895.

[9] A. Sinha, A. Unmesh, Q. Huang, and K. Ramani, "Surfnet: Generating 3D shape surfaces using deep residual networks," in *Proc. CVPR*, 2017, pp. 791–800.

[10] R. Klokov and V. Lempitsky, "Escape from cells: Deep Kd-Networks for the recognition of 3D point cloud models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 863–872.

[11] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

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

[13] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 206–215.

[14] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-R2N2: A unified approach for single and multi-view 3D object reconstruction," in *Proc. ECCV*, 2016, pp. 628–644.

[15] H. Fan, H. Su, and L. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2463–2471.

[16] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A Papier-Mâché approach to learning 3D surface generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 216–224.

[17] D. Li, T. Shao, H. Wu, and K. Zhou, "Shape completion from a single RGBD image," *IEEE Trans. Visual. Comput. Graph.*, vol. 23, no. 7, pp. 1809–1822, Jul. 2017.

[18] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[19] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. J. Guibas, "Learning representations and generative models for 3d point clouds," in *Proc. ICML*, 2018, pp. 40–49.

[20] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[21] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3577–3586.

[22] C.-H. Lin, C. Kong, and S. Lucey, "Learning efficient point cloud generation for dense 3D object reconstruction," in *Proc. AAAI*, 2018, pp. 7114–7121.

[23] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.

[24] N. J. Mitra, M. Pauly, M. Wand, and D. Ceylan, "Symmetry in 3D geometry: Extraction and applications," *Comput. Graph. Forum*, vol. 32, no. 6, pp. 1–23, Sep. 2013.

[25] I. Sipiran, R. Gregor, and T. Schreck, "Approximate symmetry detection in partial 3D meshes," *Comput. Graph. Forum*, vol. 33, no. 7, pp. 131–140, Oct. 2014.

[26] M. Sung, V. G. Kim, R. Angst, and L. Guibas, "Data-driven structural priors for shape completion," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–11, Oct. 2015.

[27] F. Han and S.-C. Zhu, "Bottom-up/top-down image parsing with attribute grammar," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 1, pp. 59–73, Jan. 2009.

[28] E. Kalogerakis, S. Chaudhuri, D. Koller, and V. Koltun, "A probabilistic model for component-based shape synthesis," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–11, Jul. 2012.

[29] A. Sharma, O. Grau, and M. Fritz, "VConv-DAE: Deep, volumetric shape learning without object labels," in *Proc. ECCV*, Oct. 2016, pp. 236–250.

[30] D. Stutz and A. Geiger, "Learning 3D shape completion from laser scan data with weak supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1955–1964.
[31] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 728–737.
[32] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, Nov. 2000.
[33] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or, "Consolidation of unorganized point clouds for surface reconstruction," *TOGACM Trans. Graph.*, vol. 28, no. 5, pp. 1–8, Dec. 2009.
[34] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer, "Parameterization-free projection for geometry reconstruction," *ACM Trans. Graph.*, vol. 26, no. 3, p. 22, Jul. 2007.
[35] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-Net: Point cloud upsampling network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2790–2799.
[36] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q.-X. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3D model repository," *CoRR*, vol. abs/1512.03012, 2015.
[37] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, Dec. 2014, pp. 1–15.
[38] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. Zhang, "Edge-aware point set resampling," *ACM Trans. Graph.*, vol. 32, no. 1, pp. 1–12, Jan. 2013.

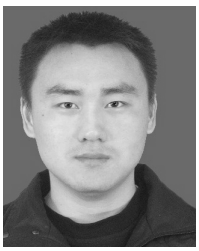
YANJUN PENG received the Ph.D. degree in March 2004. He joined the Department of Computer Science, Shandong University of Science and Technology, Qingdao, China, in 1996, where he was promoted to Professor, in 2010. His main research interests include medicine visualization, virtual reality, and image processing.

MING CHANG was born in Shandong, China, in 1993. He received the B.S. degree from the Shandong University of Science and Technology, China, in 2016, where he is currently pursuing the M.S. degree. His research interests include deep learning, computer vision, and 3D geometry processing.

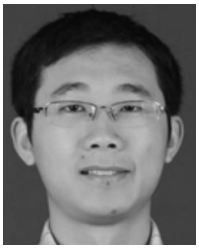
QIONG WANG received the Ph.D. degree from The Chinese University of Hong Kong, China, in 2012. She is currently an Associate Researcher with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. Her research interests include VR applications in medicine, visualization, medical imaging, human–computer interaction, and computer graphics.

YINLING QIAN currently holds a postdoctoral position at the Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His research interests include virtual reality, augmented reality, and physics-based modeling.

YINGKUI ZHANG was born in Shanxi, China, in 1994. He received the B.E. degree from the North University of China, China, in 2016. He is currently pursuing the M.S. degree with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His research interests include computer graphics, 3D vision, and point cloud processing.



MINGQIANG WEI received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong (CUHK), in 2014. He is currently an Associate Professor with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA). Before joining NUAA, he served as an Assistant Professor for the Hefei University of Technology, and a Postdoctoral Fellow with CUHK. His research interest is computer graphics with an emphasis on smart.



XIANGYUN LIAO is currently an Associate Researcher with the Guangdong Provincial Key Laboratory of Machine Vision and Virtual Reality Technology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His research interests include virtual reality, physically-based simulation, and medical imaging.