Neurocomputing 430 (2021) 94–103

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

3D-RVP: A method for 3D object reconstruction from a single depth view using voxel and point

Meihua Zhao [a,b], Gang Xiong [c,d], MengChu Zhou [e,g], Zhen Shen [a,f,*], Fei-Yue Wang [a]

[a] State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[b] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
[c] Beijing Engineering Research Center of Intelligent Systems and Technology, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[d] Guangdong Engineering Research Center of 3D Printing and Intelligent Manufacturing, Cloud Computing Center, Chinese Academy of Sciences, Dongguan 523808, China
[e] Helen and John C. Hartmann Dept. of Electrical and Computer Engineering, New Jersey Institute of Technology, NJ 07102, USA
[f] Intelligent Manufacturing Center, Qingdao Academy of Intelligent Industries, Qingdao 266109, China
[g] Institute of Systems Engineering, Macau University of Science and Technology, Macau, China

ARTICLE INFO

Article history:
Received 23 August 2020
Revised 15 October 2020
Accepted 28 October 2020
Available online 12 November 2020
Communicated by Zidong Wang

Keywords:
3D object reconstruction
Encoder-decoder network
Machine learning
Point prediction network

ABSTRACT

Three-dimensional object reconstruction technology has a wide range of applications such as augment reality, virtual reality, industrial manufacturing and intelligent robotics. Although deep learning-based 3D object reconstruction technology has developed rapidly in recent years, there remain important problems to be solved. One of them is that the resolution of reconstructed 3D models is hard to improve because of the limitation of memory and computational efficiency when deployed on resource-limited devices. In this paper, we propose 3D-RVP to reconstruct a complete and accurate 3D geometry from a single depth view, where R, V and P represent Reconstruction, Voxel and Point, respectively. It is a novel two-stage method that combines a 3D encoder-decoder network with a point prediction network. In the first stage, we propose a 3D encoder-decoder network with residual learning to output coarse prediction results. In the second stage, we propose an iterative subdivision algorithm to predict the labels of adaptively selected points. The proposed method can output high-resolution 3D models by increasing a small number of parameters. Experiments are conducted on widely used benchmarks of a ShapeNet dataset in which four categories of models are selected to test the performance of neural networks. Experimental results show that our proposed method outperforms the state-of-the-arts, and achieves about 2.7% improvement in terms of the intersection-over-union metric.

1. Introduction

Three-dimensional (3D) object reconstruction is an important problem in computer vision and computer graphics. Reconstructing the complete and accurate 3D geometry of an object plays an important role in such fields as augment reality (AR), virtual reality (VR) [36], industrial manufacturing [31,47] and intelligent robotics [45,55,44]. Reconstructed 3D models can be used in AR or VR scenes directly. In industrial manufacturing, the popularity of 3D printing provides the possibility for personalized customization, in which obtaining 3D models is the first-step work. In intelligent robotics, 3D reconstruction can be used to model a surrounding environment, so as to achieve robot grasping and obstacle avoidance.

Nowadays, laser scanning equipment is used to acquire high-precision 3D models. However, the cost is unaffordable because of the expensive equipments and burdensome manpower. Structure from motion (SfM) [25,8,57] is a classical 3D reconstruction algorithm that estimates a 3D structure from multiple 2D images, and it can output sparse point clouds. Based on the images and sparse reconstruction results from SfM, dense surface reconstruction can be performed by using a multi-view stereo (MVS) algorithm. Sun et al. [41] propose an effective method for 3D urban scene reconstruction from high resolution oblique aerial images, and it can achieve large-scale scene reconstruction with high precision. However, this kind of method is difficult to process images that lack of texture information. In addition, the images need to cover all the surface of 3D models, and the occlusion problem remains to be solved. Learning has long been an important

* Corresponding author at: State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.
E-mail addresses: zhaomeihua2018@ia.ac.cn (M. Zhao), gang.xiong@ia.ac.cn (G. Xiong), zhou@njit.edu (M. Zhou), zhen.shen@ia.ac.cn (Z. Shen), feiyue.wang@ia.ac.cn (F.-Y. Wang).

https://doi.org/10.1016/j.neucom.2020.10.097
0925-2312/

technique and applied in many fields [11,26,3,12,18,20,35]. In recent years, deep learning-based 3D reconstruction algorithms have emerged. Deep neural networks can learn the shape of 3D models from a large number of samples, and reconstruct a complete 3D shape from a single image or multi-view images.

In deep learning-based 3D reconstruction methods, a single image, multi-view images, and depth images may act as input, and a complete 3D shape acts as the ground truth output of a deep neural network. Voxel is widely used to represent a generated 3D shape. It is short for a volume element, and a geometric 3D shape is represented as a probability distribution of binary variables on a 3D voxel grid [54]. Based on this representation, classical convolutional neural network (CNN) that widely used for 2D images processing [16,37,14] can be adopted directly. After a single image or multi-view images are preprocessed [39], researchers usually use a 2D CNN to extract their global features, and then use a 3D deconvolutional neural network (DCNN) to predict the most likely 3D shape from the global features. Depth images save the distance from the depth camera to the object, and reflect the geometry shape of 3D model surface. When depth images act as the input of deep neural networks [50], researchers usually convert depth image to voxel grid, use 3D CNN to extract global features, and then use 3D DCNN to predict the most likely 3D shape. In voxel-based 3D reconstruction methods, because the parameters of 3D CNN are much more than 2D CNN, 3D neural networks take up much more memory than 2D neural networks. Considering computational efficiency and memory, the design of neural network is limited and the resolution of generated 3D models is hard to improve.

We aim to reconstruct a complete 3D shape with fine-grained details and high resolution from a single depth view. Our work is motivated by PointRend that outputs crisp object boundaries in an image segmentation task [22]. Similarly, if we obtain the low-resolution output from a neural network and interpolate it to improve the resolution, the interior of the objects is accurate and the boundary is coarse. In this paper, we propose 3D-RVP as a simple yet effective model to reconstruct a complete and accurate 3D geometry from a single depth view. It is a two-stage method. First, we convert a depth image into voxel grid, and feed it into an encoder-decoder network. The network is used to encode voxel grid into a latent vector, and then decode it back to the most likely full 3D shape. The output has a resolution of $64 \times 64 \times 64$, and the value of each voxel represents the probability it being occupied. The probability close to 0.5 indicates that the uncertainty is high. Second, we sample points from the uniform distribution, and select those with high uncertainty. We use an interpolation algorithm to extract point-wise features of these selected points from features maps of the encoder-decoder network, and use a point head to predict the probability that these points are occupied. When doing inference, we iteratively predict the occupation of voxels in the boundaries. By combining a 3D encoder-decoder network with a point prediction network, we can improve the resolution to a higher level with high precision and low cost.

This work intends to make the following contributions:

1) It proposes a 3D encoder-decoder network with residual learning for reconstructing a complete 3D model from a single depth view. By adding a residual structure to the encoder and decoder, the network is easy to optimize; and
2) It proposes a point prediction network to achieve high-resolution prediction results. The proposed method alleviates the burden of memory, as well as achieves high accuracy.

In addition, this work conducts extensive experiments, and proves that the proposed method outperforms the state-of-the-arts and has a great generalization ability.

The rest of the paper is organized as follows. In Section 2, we briefly review the literature of deep learning-based 3D reconstruction methods, including voxel-based methods and point-based methods. In Section 3, we introduce the proposed 3D-RVP consisting of two components: an encoder-decoder network and a point prediction network. In Section 4, we perform two experiments, and the results prove the effectiveness of our proposed method. Finally, we conclude this paper and give future work in Section 5.

2. Related work

According to different representations of generated 3D models, 3D reconstruction methods can be divided into voxel, point and mesh-based methods. With the popularity of deep neural networks, voxel-based and point-based representations are widely adopted. In this section, we review deep learning-based 3D reconstruction methods based on these two representations.

2.1. Voxel-based methods

Voxel-based methods use voxel grid to represent the geometry of 3D objects. Wu et al. [54] propose to uses a probability distribution of binary variables on a 3D voxel grid to represent the geometric 3D shape. Given a voxel grid converted from the depth map of an object, shape completion is realized by using convolutional deep belief network. 3D recurrent reconstruction neural network (3D-R2N2) [6] reconstructs a complete 3D shape from multiple images. In this method, 2D-CNNs are first used to extract features from images, and then 3D long short-term memory (LSTM) networks are used to aggregate these features to obtain a global feature vector. Finally, 3D-DCNN is used to decode the feature vector into voxel grid. Generative adversarial networks (GANs) are widely used for 3D object reconstruction [50,49]. By combining a 3D encoder-decoder and a conditional adversarial network, an accurate 3D shape can be inferred from a single depth view. Semantic scene completion networks are proposed based on this representation [40,46]. Because of the large cost of obtaining large-scale supervised data, many methods of weakly supervised learning and unsupervised learning have been proposed [32,48,56,23].

Because of the limitation of their computational efficiency and memory, the resolution of 3D models is hard to improve. To achieve high-resolution results, 3D-EPN [9] correlates coarse prediction results with 3D geometry from a shape database and uses a patch-based 3D shape synthesis method. Han et al. [13] propose a local geometry refinement network for 3D shape completion. Considering the sparsity of 3D data, Riegler et al. propose OctNet [33]. It uses a hybrid structure of grid and unbalanced octrees to represent 3D models. This method can focus memory allocation and computation to the relevant dense regions, and can achieve high resolution. Based on this representation, a hierarchical surface prediction (HSP) framework that hierarchically predicts small blocks of voxels from coarse to fine is proposed [2]. Tatarchenko et al. [42] propose octree generating networks (OGN), which uses an octree representation to generate volumetric 3D outputs. However, the octree-based methods need a complex data structure to implement, which limits its development.

In voxel-based methods, CNN is widely used. Recently, some advancement of classical CNN has been shown. He et al. [16] propose ResNet to make deeper neural networks easier to optimize by adding an identity function. DenseNet [17] makes further improvement by introducing direct connections from any layer to all subsequent layers to further improve the information flow among layers. Chen et al. propose NeuralODE [5], which combines deep learning with an ODE solver. Compared with classical CNN, it has

advantages in memory efficiency, adaptive computation, scalable and invertible normalizing flows and continuous time-series models.

2.2. Point-based methods

Point-based methods generate point cloud directly. Point cloud is continuous in space; as a result, there is less information missing. Moreover, point cloud only saves the information of 3D object surface, which reduces the data redundancy. PointNet [29] is a pioneering study to use neural networks to process point cloud directly. It uses several shared multi-layer perceptrons (MLPs) to extract point-wise features, and then uses a symmetric function to aggregate all points' features to obtain global features. However, because each point learns the features independently, the PointNet network cannot capture neighborhood information of points. PointNet++ [30] extracts local features in multiple scales, and obtains deep features by using a multi-layer network structure. Recently, many neural networks based on point cloud have been designed [53,24,38], and used for 3D shape classification, 3D object detection and 3D segmentation [27,28,43].

Achlioptas et al. [1] propose the first generative model for 3D point clouds. Their approach consists of an AutoEncoder, which has an encoder following the design of PointNet, and a decoder that uses three fully connected layers to generate point cloud. Fan et al. [10] propose a point set generation network for 3D object reconstruction from a single image. It uses a fully-connected branch and a deconvolution branch to predict the point cloud. FoldingNet [51] introduces a folding-based decoder architecture. PCN [52] adopts an encoder-decoder network for 3D point cloud completion. The decoder combines the advantages of the fully-connected decoder and the folding-based decoder. PFNet [19] uses a multi-resolution encoder (MRE) and a point pyramid decoder (PPD) to realize point cloud completion. It only outputs the missing part of the point cloud instead of the whole object.

3. Proposed methodology

In this section, we introduce our proposed method for 3D reconstruction. In Fig. 1, we show the complete architecture of 3D-RVP. We describe its encoder-decoder network, and then its point prediction network.

3.1. Encoder-decoder network

We propose a 3D encoder-decoder network with residual learning as backbone. Voxel grid is used to represent a 3D shape, and it is defined as a probability distribution of binary variables [54]. By scanning a 3D model from a view angle, we can obtain a depth image and its corresponding complete 3D shape. We convert depth images into voxel grid, and feed it to a neural network. At the output of the neural network, we can obtain coarse prediction results. The input voxel grid and output coarse prediction results all have the resolution of $64 \times 64 \times 64$. As shown in Fig. 1, the encoder-decoder network consists of an encoder and a decoder. The encoder extracts features of voxel grid to get latent representation and the decoder predicts the most likely full 3D shape. The network adopts the design of U-Net [34,7], which concatenates the encoder and decoder to obtain a more accurate 3D shape.

The detailed architecture of the encoder-decoder network is shown in Fig. 2. The left part represents an encoder, and the right part a decoder. Each block represents one layer of neural networks. For the items in each block, the operation is in the left, and the output shape of feature maps after the operation is in the right. Taking "$3 \times 3 \times 3$, conv, 64 $64 \times 64 \times 64 \times 64$" as an example, the kernel size of the 3D convolutional layer is "$3 \times 3 \times 3$", "conv" represents convolutional operation, and the number of output channels is "64". After convolution, the output shape is "$64 \times 64 \times 64 \times 64$".

The encoder has 18 layers. Input voxel grid is with the shape of "$64 \times 64 \times 64 \times 1$", the first layer converts the number of channels to 64 by a convolution operation. The next 16 layers are a residual network with a total of four blocks, and each of blocks is marked in a different color. In each block, the number of channels is doubled, and the resolution of feature maps reduce by half to increase the receptive field. The last layer is a fully-connected layer with 2,000 nodes. The decoder also has 18 layers. The first fully-connected layer converts a latent vector of 2,000 dimensions to that of 32,768 dimensions. The next 16 layers are a residual network. A deconvolution operation is used to increase the resolution of feature maps. The last layer predicts the complete 3D shape by a convolution operation. A sigmoid function is used as activation in the last layer, while a rectified linear unit (ReLU) function is adopted in remaining layers. The encoder and the decoder are connected by a "concat" operation. The "concat" is short for concatenation, and the number of output feature maps' channels is the sum of those of two inputs after the operation. In the encoder, the feature maps have a small receptive field. As a result, it contains more local information. While in the decoder, the feature maps have a large receptive field, and it contains more global information. By concatenating the encoder and decoder, more fine-grained features can be added to the output.

We add a residual structure to the encoder-decoder network. When the neural networks deepen, more deep features can be extracted and the accuracy can be improved. However, deeper neural networks are more difficult to optimize, known as a degradation problem. Following the method in [16], we adopt residual learning to each block of the encoder and decoder to make the network easy to optimize. As shown in Fig. 2, each residual block has two short cut connections. The output is calculated as

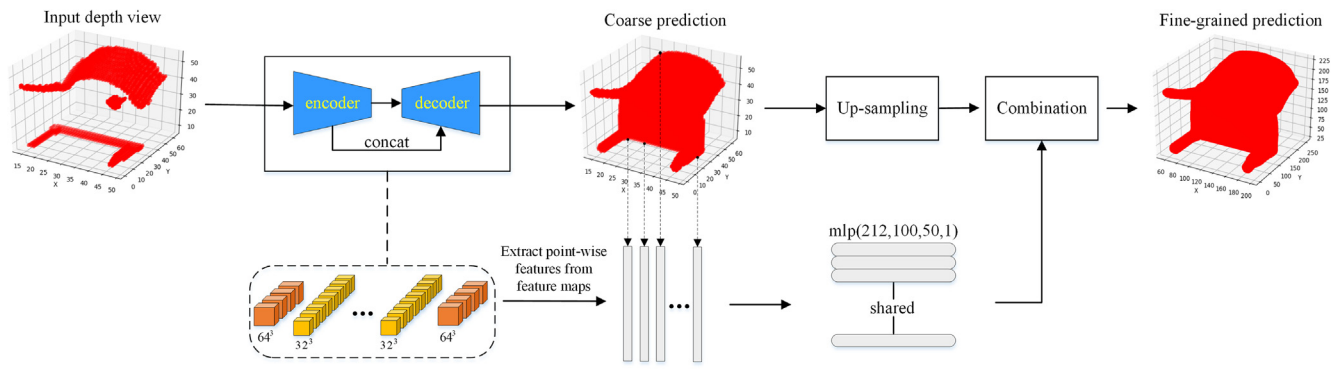

Fig. 1. Overview architecture of 3D-RVP.

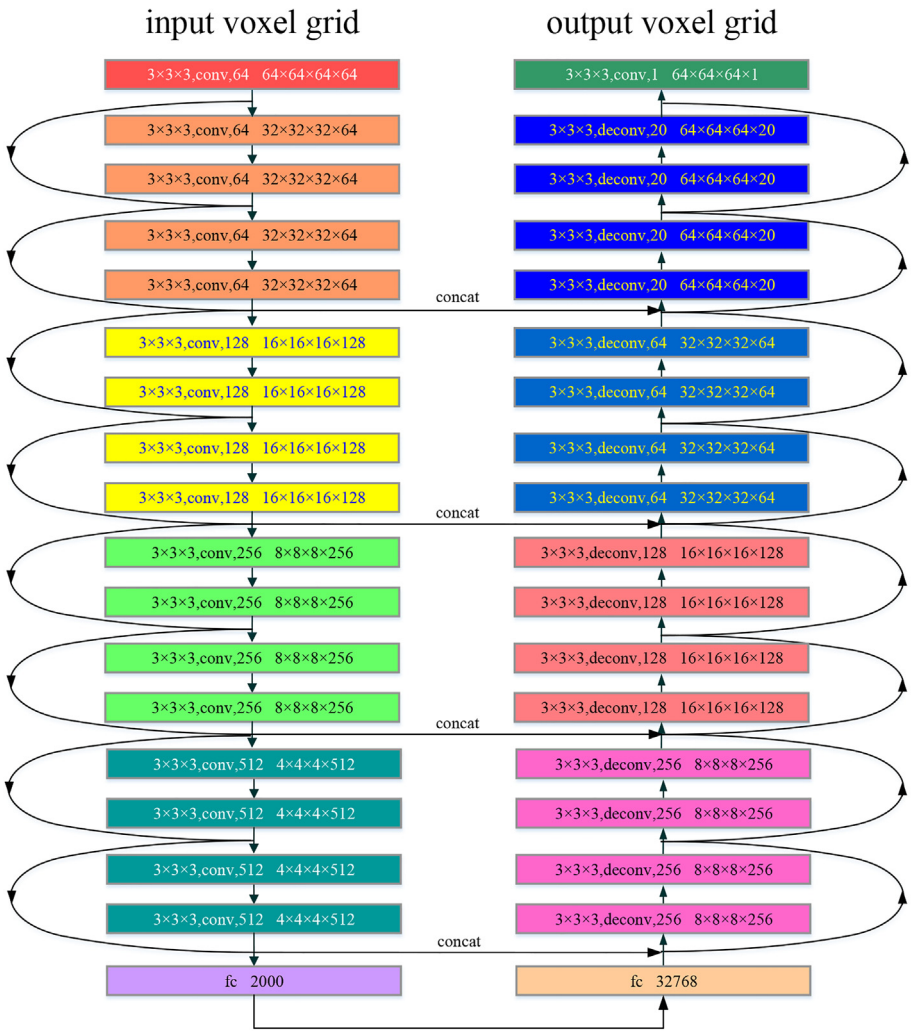

Fig. 2. Detailed architecture of the encoder-decoder network.

$$y = F(x, \{W_i\}) + W_s x, \tag{1}$$

where y is the output of a residual structure. x is the input of the residual structure. $F(x, \{W_i\})$ represents multiple convolution layers, and W_i represents its parameters. W_s is a linear projection to match the dimensions.

3.2. Point prediction network

In the interior of a generated 3D model, there is nearly no difference between the values of voxels and those of their neighbours. While at the boundary, the values of adjacent voxels vary greatly. Given the coarse prediction results, if we obtain higher resolution by interpolation directly, the interior of the 3D model is accurate and the boundary is coarse. A point prediction network selects points from the boundaries of coarse 3D models, and predicts the labels of these selected points. By combining coarse prediction results with the labels of these selected points, a more accurate 3D model with high resolution can be obtained. The point prediction network accepts feature maps $f \in R^{H\times W\times L\times C}$ from the encoder-decoder network, and outputs the prediction of probability that voxels are occupied $p \in R^{H'\times W'\times L'}$. In this experiment, the $H = W = L = 64$, and the $H' = W' = L' = 256$. The point prediction network consists of three parts: a point selection strategy, a point-wise feature extraction module and a point head. The detailed introduction is given next.

The point selection strategy aims at selecting points with high uncertainty. In coarse prediction results, the value of each voxel represents the probability it is occupied. The probability of points is closer to 0.5, their uncertainty is higher. Thus, the uncertainty can be calculated as

$$\tilde{V} = 1 - |V_p - 0.5|, \tag{2}$$

where $|\cdot|$ represents an absolute value. V_p is the prediction probability that a voxel is occupied, and $\tilde{V}$ represents its uncertainty. We use a different point selection strategy in training and inference. During inference, inspired by adaptive subdivision in computer graphics, we adopt an iterative method. Firstly, we interpolate the coarse prediction results to double the resolution. Then, we set a threshold γ, and select voxels which uncertainty larger than γ. In the experiment, we set γ to be 0.9. Finally, we calculate the center coordinates of selected voxels in the normalized coordinate system. These center coordinates are saved as selected points. After predicting the labels of these points, we repeat the above operation until the resolution meets the requirements.

This iterative method is not suitable for optimization using a back-propagation algorithm. Therefore, a non-iterative strategy based on random sampling is adopted during training. Our purpose

is to sample M points with large uncertainty. Firstly, in order to cover the entire 3D shape, we sample kM candidate points $(k > 1)$ from a uniform distribution. Then, we calculate the uncertainties of these sampled points by trilinearly interpolating the coarse prediction values, and select βM points ($\beta \in (0, 1]$) with largest uncertainties. Finally, we sample the remaining $(1 - \beta)M$ points from a uniform distribution. In the experiment, we set M to be 6400. The larger the value of k, the selection points is more inclined to the region with greater uncertainty. We set k to be 50, and β to be 0.75.

The point-wise feature extraction module extracts features for selected points from input feature maps. It combines two kinds of features: fine-grained and coarse features. The former contain more local information, and can resolve details. The latter contain more global information. Because a point is a real-value 3D coordinate, we compute a feature vector by trilinear interpolation on the feature maps. In the experiment, we extract the former from the output of the first layer and first residual module in the encoder, each of which has 64 channels. We extract the latter from the output of last two residual modules in decoder, which have 64 and 20 channels, respectively. The concatenation of these two features with 212 channels results in the point-wise features.

Point head is responsible for predicting labels of selected points given their point-wise features. In the experiment, an MLP is used. It shares parameters across different points. The MLP network has four layers. Its first layer has 212 nodes, and the last one has 1 node. Two hidden layers have 100, 50 nodes respectively. ReLU is used as an activation function in the network, except that sigmoid is used in the last layer to limit the output to the range (0, 1).

3.3. Loss function

The loss function consists of two parts: an encoder-decoder loss and a point loss. For the former, we use an improved cross-entropy loss function, aiming at solving the problem of unbalanced category labels in voxel grid. It adds a penalty parameter to punish more the situation that voxel with the value of 1 being recognized as 0. It is defined as

$$l_1 = -\frac{1}{N}\sum_{i=1}^{N}\left[\alpha y_i^* \log y_i + (1-\alpha)(1-y_i^*)\log(1-y_i)\right], \tag{3}$$

where l_1 is the loss for voxel grid. i is the index of a voxel, and N is the number of voxels in the voxel grid. y_i is the prediction probability that the ith voxel is occupied, and y_i^* is the corresponding ground truth. The value of y_i falls into $(0, 1)$, and the value of y_i^* is 0 or 1. α is a penalty parameter. If α is lager, more voxels are predicted to be occupied with greater probability. If it is too small, the probability of most voxels being occupied is lower than 0.5. We search for its proper value in the experiment and set it to be 0.85. For each voxel, we calculate a loss, and the loss for voxel grid is the mean of all voxels.

Point loss is only calculated on the selected M points. We use a cross-entropy loss function

$$l_2 = -\frac{1}{M}\sum_{i=1}^{M}[\alpha\tilde{z}_i \log z_i + (1-\alpha)(1-\tilde{z}_i)\log(1-z_i)], \tag{4}$$

where i is the index of selected points. z_i is the prediction probability that the ith point is occupied, and $\tilde{z}_i$ is the corresponding estimated label. Note that $\tilde{z}_i$ can be obtained by the nearest interpolation from the ground truth which has a resolution of $256 \times 256 \times 256$. For each point, we calculate a loss. The loss for all points in voxel grid is the mean of all points. The loss for the entire model is

$$loss = l_1 + \lambda l_2, \tag{5}$$

where parameter λ is used to balance l_1 and l_2. It is set to be 0.1 in the experiment. Both the encoder-decoder network and the point prediction network are jointly trained. The parameters of the neural networks are updated dynamically by using a back-propagation algorithm.

4. Experiments

In this section, we introduce the acquisition and preprocessing of a dataset. Then we show the implementation details of the network and the metrics that evaluate the quality of 3D reconstruction. Finally, we show the results of our proposed method.

4.1. Dataset

The training of a neural network needs a single depth view and its corresponding complete 3D model. We use the dataset provided in [49], which is generated from the ShapeNet database [4]. ShapeNet is a richly-annotated, large-scale 3D CAD models repository. It has more than 3,000,000 models, and 220,000 models of which are classified into 3,135 categories.

The training dataset is generated from four categories: bench, chair, couch, and table. For each category, 213 CAD models are randomly selected from ShapeNet. For each model, 5 different viewing angles are sampled from uniform distribution for each of roll, pitch and yaw space, and total 125 different viewing angles are obtained. Blender, the open source software, is used for rendering and voxelization from the 125 viewing angles to obtain 2.5D depth images and the corresponding voxel grid. There are 26,625 training pairs for each 3D object category.

The testing dataset is generated from four same categories as the training dataset. For each category, 37 CAD models are randomly selected from ShapeNet, and two groups of testing samples are generated. One group is scanned from 125 different viewing angles, which is the same as the training dataset. It is thus denoted as a same viewing angles testing dataset, called SV for short. Each category has 4,625 pairs of samples. The other group is scanned from 216 different viewing angles, which are sampled from 6 different viewing angles for each of roll, pitch and yaw space. This group of the testing dataset is denoted as cross viewing angles testing dataset, called CV for short. Each category has 7,992 pairs of samples.

4.2. Experimental setup

The experiment is conducted on a workstation with an Intel Xeon E5-2630v4 CPU and a Titan V Graphics Processing Unit. We use the Pytorch framework and Adam optimizer [21]. We do two experiments to verify the performance of the proposed neural networks. The first experiment is per-category training, and the second one is multi-category training. The initial learning rate is set to be 0.001. For per-category training, the learning rate is multiplied by 0.7 in the 16th, 24th, 32th, and 40th epochs, respectively. For multi-category training, the learning rate is multiplied by 0.7 in the 4th, 6th, 8th, and 10th epochs, respectively. Batch back propagation algorithm is used in the experiment, The batch size is set to be 6.

4.3. Metrics

We use two metrics to evaluate the quality of 3D reconstruction. The first metric is intersection-over-union (IoU). It is widely used in object detection. For 3D reconstruction, it is defined as the size of the intersection divided by the size of the union of the

prediction model and the ground truth. For a single voxel grid, it can be calculated as

$$IoU = \frac{\sum_i [I(y_i > \theta) \cdot I(y_i^*)]}{\sum_i [I(I(y_i > \theta) + I(y_i^*))]}, \quad (6)$$

where $I(x)$ is a function that equals 1 if x is greater than or equal to 1, and otherwise 0. i is the index of a voxel, y_i is the prediction value of the ith voxel, and y_i^* is the corresponding ground truth. θ is the threshold. In the experiment, it is set to be 0.5. If the prediction value is greater than 0.5, the output value is set to be 1, and otherwise 0. The value of IoU ranges from 0 to 1. The larger the value, the better the 3D reconstruction quality.

The second metric is standard cross-entropy (CE) loss. It is also used as loss function for training the neural networks in this experiment. For a single voxel grid, it can be calculated as

$$CE = -\frac{1}{N}\sum_i [y_i^* \log y_i + (1 - y_i^*) \log(1 - y_i)], \quad (7)$$

where the meanings of i, y_i and y_i^* are the same as those in IoU. N represents the number of voxels in a voxel grid. In this experiment, the resolution is $256 \times 256 \times 256$, and thus $N = 256^3$.

4.4. Competing methods

We compare our proposed 3D-RVP with four other deep learning based methods to verify its effectiveness. The results of the first four methods are from [49]. For fair comparison, the output models of competing methods are interpolated to achieve a resolution of $256 \times 256 \times 256$.

(1) 3D-EPN [9] combines volumetric deep neural networks with 3D shape synthesis to complete partial 3D shapes. A low-resolution distance field is first predicted, and then high resolution details are generated from this coarse prediction. Following the comparison method of [49], we only focus on the performance of their neural network.
(2) Varley et al. [44] propose to complete 3D shape for robotic grasp (called 3D-RG for short). 3D convolutional layers are used to map a 2.5D shape to a latent vector, and fully-connected layers are used to obtain the complete 3D shape. The resolution of output is $40 \times 40 \times 40$.
(3) Han et al. [13] propose a new deep learning architecture for high-resolution shape completion (called 3D-HSC for short). This architecture first uses a global structure inference network to infer a global structure, and then uses a local geometry refinement network to produce a high-resolution, complete surface. The resolution of output is $256 \times 256 \times 256$.
(4) 3D-RecGAN++ [49] uses generative adversarial networks to reconstruct a complete 3D structure of a given object from a single arbitrary depth view. The generator adopts an encoder-decoder structure, and learns a correlation between partial and complete 3D structures. The discriminator aims to distinguish whether the estimated 3D shapes are plausible or not, which enables the output of 3D-RecGAN++ to be more robust and confident.
(5) We propose a 3D encoder-decoder network with residual learning. The encoder-decoder network consists of an encoder and a decoder, and it is the backbone of our proposed 3D-RVP. In the experiment, we compare the results to verify the effectiveness of our encoder-decoder network and point prediction network.

4.5. Results

In the first experiment, we train the neural networks on each category of a training dataset, respectively, including bench, chair, couch and table. All samples of the training dataset are scanned from 125 different viewing angles. In order to verify the generalization ability of neural networks, we select two groups of the testing dataset. The first group is SV, and Table 1 shows the corresponding IoU and CE loss results. In the following tables, we denote an encoder-decoder network as EDnet for short. It can be observed that our proposed 3D-RVP outperforms competing methods in all categories for IoU and 3 categories for CE loss. Compared with the state-of-the-art methods, 3D-RVP improves IoU by about 2.3% on average and reduces CE loss by about 1.7% on average. It proves the effectiveness of our proposed method.

Our purpose is to reconstruct a complete 3D shape from a single depth view scanned from any viewing angles. In the second group, we test the neural networks on CV. Different from the training dataset, all samples of the testing dataset are scanned from 216 different viewing angles. Table 2 shows the IoU and CE loss results. It can be observed that 3D-RVP outperforms 3D-RecGAN++ in 3 categories for IoU and 3 categories for CE loss. It also outperforms the encoder-decoder network in all categories for both IoU and CE loss. Specifically, it improves IoU by about 3.6% on average over 3D-RecGAN++, and improves it by about 2.3% on average over the encoder-decoder network. It proves that both our proposed encoder-decoder network and point prediction network have great generalization ability.

In the second experiment, we train the neural networks on a multi-category training dataset, including bench, chair, couch and table. All samples of the training dataset are scanned from 125 different viewing angles. Two groups of the testing dataset are used to test the performance of the neural networks.

(1) We test the multi-category IoU and CE loss on SV. The neural networks are tested on each category of bench, chair, couch and table respectively. The results of IoU and CE are shown in Table 3. It can be observed that the performance of the encoder-decoder network is comparable with that of 3D-RecGAN++. Compared with 3D-RecGAN++, 3D-RVP improves IoU by about 2.1% on average and reduces CE loss by about 4.3% on average.

Fig. 3 shows the visualizations of multi-category reconstruction on SV. From top to bottom, the figures show depth image, coarse prediction, fine-grained prediction and ground truth. Coarse prediction is the result of trilinear interpolation of the output of the encoder-decoder network to resolution $256 \times 256 \times 256$. Fine-grained prediction is the output of 3D-RVP. We can see that the encoder-decoder network can learn a complete 3D shape, while the boundaries of the generated models are coarse. The point prediction network can classify voxels over-smoothed by interpolation correctly, and as a result, the output of 3D-RVP has fine-grained details at the boundaries.

(2) We test the multi-category IoU and CE loss on CV. The neural networks are tested on each category of bench, chair, couch and table respectively. The results of IoU and CE are shown in Table 4. Compared with 3D-RecGAN++, 3D-RVP improves IoU by about 2.8% and reduces CE loss by about 5.3% on average. It further verifies the performance of our proposed method. Note that the obtained IoU for the couch category is higher than that obtained by per-category training. This proves that the features learned by the neural networks from other categories can supplement the features of the couch category effectively.

Fig. 4 shows the visualizations of multi-category reconstruction on CV. From top to bottom, the figures show depth image, coarse prediction, fine-grained prediction and ground truth. Similarly, 3D-RVP can output fine-grained results than the encoder-decoder

Table 1
Per-Category IoU and CE loss with same viewing angles.

Methods	IoU				CE			
	Bench	Chair	Couch	Table	Bench	Chair	Couch	Table
3D-EPN[9]	0.423	0.488	0.631	0.508	0.087	0.105	0.144	0.101
3D-RG[44]	0.227	0.317	0.544	0.233	0.111	0.157	0.195	0.191
3D-HSC [13]	0.441	0.426	0.446	0.499	0.045	0.081	0.165	0.058
3D-RecGAN++ [49]	0.580	0.647	0.753	0.679	0.034	0.060	**0.066**	0.040
EDnet (ours)	0.577	0.654	0.750	0.663	0.033	0.062	0.069	0.042
3D-RVP (ours)	**0.598**	**0.668**	**0.760**	**0.696**	**0.032**	**0.060**	0.067	**0.039**

Table 2
Per-Category IoU and CE loss with cross viewing angles.

Methods	IoU				CE			
	Bench	Chair	Couch	Table	Bench	Chair	Couch	Table
3D-EPN[9]	0.408	0.446	0.572	0.482	0.086	0.112	0.163	0.103
3D-RG[44]	0.185	0.278	0.475	0.187	0.108	0.171	0.210	0.186
3D-HSC [13]	0.439	0.426	0.455	0.482	0.047	0.090	0.163	0.060
3D-RecGAN++ [49]	0.531	0.594	**0.646**	0.618	0.041	0.074	**0.111**	0.053
EDnet (ours)	0.537	0.611	0.639	0.631	0.038	0.076	0.140	0.049
3D-RVP (ours)	**0.554**	**0.621**	0.643	**0.656**	**0.037**	**0.074**	0.138	**0.047**

Table 3
Multi-Category IoU and CE loss with same viewing angles.

Methods	IoU				CE			
	Bench	Chair	Couch	Table	Bench	Chair	Couch	Table
3D-EPN[9]	0.428	0.484	0.634	0.506	0.087	0.107	0.138	0.102
3D-RG[44]	0.234	0.317	0.543	0.236	0.103	0.132	0.197	0.170
3D-HSC [13]	0.425	0.454	0.440	0.470	0.045	0.087	0.172	0.065
3D-RecGAN++ [49]	0.581	0.640	0.745	0.667	0.030	0.051	0.063	0.039
EDnet (ours)	0.580	0.642	0.745	0.663	0.030	0.052	0.064	0.038
3D-RVP (ours)	**0.596**	**0.655**	**0.750**	**0.687**	**0.029**	**0.050**	**0.062**	**0.035**

Fig. 3. Qualitative results of multiple category reconstruction on testing datasets with same viewing angles. From top to bottom, the figures show depth image, coarse prediction, fine-grained prediction and ground truth.

Table 4
Multi-Category IoU and CE loss with cross viewing angles.

Methods	IoU				CE			
	Bench	Chair	Couch	Table	Bench	Chair	Couch	Table
3D-EPN[9]	0.415	0.452	0.531	0.477	0.091	0.115	0.147	0.111
3D-RG[44]	0.201	0.283	0.480	0.199	0.105	0.143	0.207	0.174
3D-HSC [13]	0.429	0.444	0.447	0.474	0.045	0.089	0.172	0.063
3D-RecGAN++ [49]	0.540	0.594	0.643	0.621	0.038	0.061	**0.091**	0.048
EDnet (ours)	0.532	0.606	0.657	0.625	0.037	0.060	0.095	0.045
3D-RVP (ours)	**0.545**	**0.617**	**0.661**	**0.643**	**0.035**	**0.058**	0.093	**0.043**

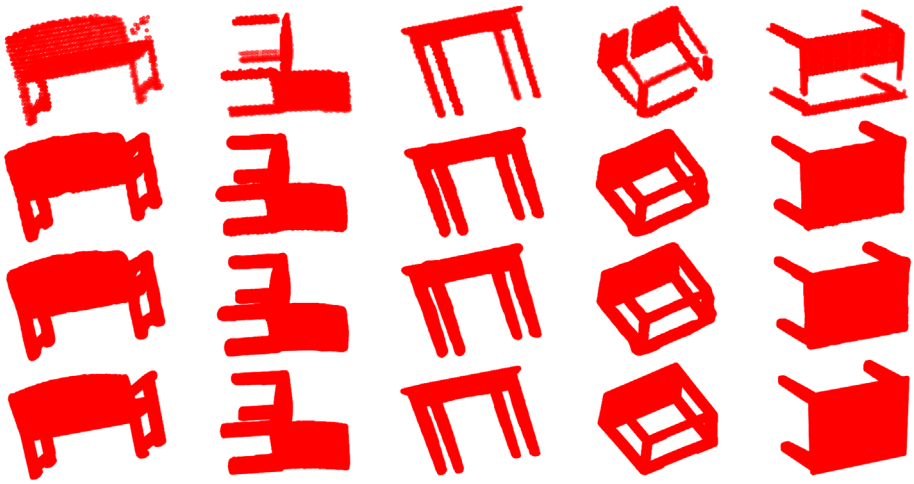

Fig. 4. Qualitative results of multiple category reconstruction on testing datasets with cross viewing angles. From top to bottom, the figures show depth image, coarse prediction, fine-grained prediction and ground truth.

network. It proves that our proposed method has great generalization ability.

4.6. Computational complexity analysis

Compared with 3D-RecGAN++, 3D-RVP uses about 8.3% more parameters. Specifically, the total number of parameters for 3D-RecGAN++ is 167.1 million. That for 3D-RVP is 181.0 million, among which the parameters of point prediction network are 0.02 million. The reason is that the encoder-decoder network is deepened to improve the accuracy and we add a point prediction network to 3D-RVP. Time complexity is critical for evaluating the performance of a method [15]. The time complexity of 3D-RVP is $O\left(\eta_1\sum_{i=1}^{l_1}A_i^3\cdot B_i^3\cdot C_{i-1}\cdot C_i+\eta_2\sum_{i=1}^{l_2}D_{i-1}\cdot D_i+\eta_3 M\sum_{i=1}^{l_3}E_{i-1}\cdot E_i\right)$, where l_1 represents the number of convolutional layers, A_i is the size of the output feature maps of the ith convolutional layer, B_i is the size of a convolution kernel, and C_i is the number of output channels of this layer. l_2 is the number of fully-connected layers in an encoder-decoder network, and D_i is the number of output nodes of the ith fully-connected layer in that network. M is the number of sampled points in a voxel grid, l_3 is the depth of the MLP network in a point prediction network, and E_i is the number of output nodes of the ith fully-connected layer in that network. We add constants η_1, η_2 and η_3 to denote the complexity of back propagation and number of iterations. In the experiment, $l_1 = 34$, $l_2 = 2$ and $l_3 = 3$. The advantages of 3D-RVP lie in two aspects: 1) it outputs a finer boundary than the encoder-decoder network by increasing 8.3% more parameters; and 2) it allows one to reconstruct complete 3D models at any desired resolution, while keeping the model memory footprint constant.

5. Conclusion

In this paper, we propose a novel method to reconstruct a complete 3D shape from a single depth view. We first use a 3D encoder-decoder network with residual learning to obtain coarse prediction results. By adding residual learning, our proposed network becomes easy to optimize. Given the coarse prediction results, we use a point prediction network to select points with high uncertainty, and then use a shared MLP network to predict the labels of these points. By combining coarse prediction results with the labels of these selected points, we obtain fine-grained results with much higher resolution. The proposed method can reach higher resolution by increasing a small portion of parameters. Experimental results show that our proposed method can outperform the state-of-the-arts.

IoU can be decreased if the depth image provides less information about the complete 3D shape, and the depth image can also be affected by illumination. As future work, we plan to consider the extreme situations, for example, when object surface is perpendicular to the camera, and objects are under different illumination. We plan to design a more powerful model for 3D object reconstruction. DenseNet may be a good alternative. We plan to explore ways to combine NeuralODE with our method. Also, because a single depth image contains very little information about a 3D shape, we attempt to use multiple images from different angles to reconstruct a complete 3D shape, and extend our proposed method to more experimental datasets. Because point cloud is continuous in 3D space and contains less redundant information, we can also explore point-based 3D reconstruction methods.

CRediT authorship contribution statement

Meihua Zhao: Methodology, Software, Validation, Formal analysis, Writing - original draft. **Gang Xiong:** Conceptualization, Writing - review & editing, Supervision, Funding acquisition. **MengChu Zhou:** Writing - review & editing, Supervision. **Zhen Shen:** Formal analysis, Writing - review & editing, Resources, Project administration, Funding acquisition. **Fei-Yue Wang:** Supervision, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 61773382, U1909218, U1909204, 61773381, U1811463, 61872365 & 61806198; CAS Key Technology Talent Program (Zhen Shen); Chinese Guangdong's S&T Project (2019B1515120030).

References

[1] P. Achlioptas, O. Diamanti, I. Mitliagkas, L.J. Guibas, Learning representations and generative models for 3D point clouds. arXiv: Computer Vision and Pattern Recognition, 2017.

[2] C. Bane, S. Tulsiani, J. Malik, Hierarchical surface prediction for 3D object reconstruction, in: International Conference on 3D Vision (3DV), 2017, pp. 412–420, https://doi.org/10.1109/3DV.2017.00054.

[3] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, W.A. Chaovalitwongse, Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions, IEEE Transactions on Evolutionary Computation 23 (2019) 718–731, https://doi.org/10.1109/TEVC.2018.2885075.

[4] A.X. Chang, T. Funkhouser, L.J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al., Shapenet: an information-rich 3D model repository. arXiv: Graphics, 2015.

[5] R.T.Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud, Neural ordinary differential equations, in: Advances in Neural Information Processing Systems (NIPS 2018), 2018.

[6] C.B. Choy, D. Xu, J.Y. Gwak, K. Chen, S. Savarese, 3D–R2N2: A unified approach for single and multi-view 3D object reconstruction, in: Computer Vision - ECCV 2016, PT VIII, 2016, pp. 628–644. DOI: 10.1007/978-3-319-46484-8_38.

[7] O. Cicek, A. Abdulkadir, S.S. Lienkamp, T. Brox, O. Ronneberger, 3D U-Net: Learning dense volumetric segmentation from sparse annotation, Medical Image Computing and Computer Assisted Intervention (2016) 424–432.

[8] H. Cui, S. Shen, W. Gao, Z. Wang, Progressive large-scale structure-from-motion with orthogonal MSTs, in: 2018 International Conference on 3D Vision (3DV), 2018, pp. 79–88.

[9] A. Dai, C.R. Qi, M. Niessner, Shape completion using 3D-encoder-predictor CNNs and shape synthesis, in: 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6545–6554, https://doi.org/10.1109/CVPR.2017.693.

[10] H. Fan, H. Su, L. Guibas, A point set generation network for 3D object reconstruction from a single image, in: 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), 2017, pp. 2463–2471. DOI: 10.1109/CVPR.2017.264.

[11] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, J. Wang, Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction, IEEE Transactions on Neural Networks and Learning Systems 30 (2019) 601–614, https://doi.org/10.1109/TNNLS.2018.2846646.

[12] H. Han, M. Zhou, Y. Zhang, Can virtual samples solve small sample size problem of KISSME in pedestrian re-identification of smart transportation?, IEEE Transactions on Intelligent Transportation Systems 21 (2020) 3766–3776

[13] X. Han, Z. Li, H. Huang, E. Kalogerakis, Y. Yu, High-resolution shape completion using deep neural networks for global structure and local geometry inference, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 85–93, https://doi.org/10.1109/ICCV.2017.19.

[14] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.

[15] K. He, J. Sun, Convolutional neural networks at constrained time cost, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5353–5360.

[16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[17] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), 2017, pp. 2261–2269. DOI: 10.1109/CVPR.2017.243.

[18] Z. Huang, X. Xu, H. Zhu, M. Zhou, An efficient group recommendation model with multiattention-based neural networks, IEEE Transactions on Neural Networks and Learning Systems (2020) 1–14, https://doi.org/10.1109/TNNLS.2019.2955567.

[19] Z. Huang, Y. Yu, J. Xu, F. Ni, X. Le, PF-Net: Point fractal network for 3D point cloud completion. arXiv: Computer Vision and Pattern Recognition, 2020.

[20] Q. Kang, S. Yao, M. Zhou, K. Zhang, A. Abusorrah, Enhanced subspace distribution matching for fast visual domain adaptation, IEEE Transactions on Computational Social Systems 7 (2020) 1047–1057, https://doi.org/10.1109/TCSS.2020.3001517.

[21] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 2015 International Conference on Learning Representations (ICLR), 2015.

[22] A. Kirillov, Y. Wu, K. He, R. Girshick, PointRend: Image segmentation as rendering. arXiv: Computer Vision and Pattern Recognition, 2019.

[23] J. Liu, F. Yu, T. Funkhouser, Interactive 3D modeling with a generative adversarial network, in: 2017 International Conference on 3D Vision (3DV), 2017, pp. 126–134, https://doi.org/10.1109/3DV.2017.00024.

[24] Y. Liu, B. Fan, S. Xiang, C. Pan, Relation-shape convolutional neural network for point cloud analysis, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 8887–8896.

[25] H.C. Longuethiggins, A computer algorithm for reconstructing a scene from two projections, Nature 293 (1987) 61–62.

[26] X. Luo, H. Wu, H. Yuan, M. Zhou, Temporal pattern-aware QoS prediction via biased non-negative latent factorization of tensors, IEEE Transactions on Cybernetics 50 (2020) 1798–1809.

[27] C.R. Qi, O. Litany, K. He, L. Guibas, Deep hough voting for 3D object detection in point clouds, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9276–9285.

[28] C.R. Qi, W. Liu, C. Wu, H. Su, L.J. Guibas, Frustum pointNets for 3D object detection from RGB-D data, in: 2018 IIEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 918–927.

[29] C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, in: 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), 2017, pp. 77–85. DOI: 10.1109/CVPR.2017.16.

[30] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet plus plus: Deep hierarchical feature learning on point sets in a metric space, in: Advances in Neural Information Processing Systems (NIPS 2017), 2017.

[31] X. Qi, G. Chen, Y. Li, X. Cheng, C. Li, Applying neural-network-based machine learning to additive manufacturing: current applications, challenges, and future perspectives, Engineering 5 (2019) 721–729, https://doi.org/10.1016/j.eng.2019.04.012.

[32] D.J. Rezende, S.M.A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, N. Heess, Unsupervised learning of 3D structure from images, in: Advances in Neural Information Processing Systems (NIPS 2016), 2016.

[33] G. Riegler, A.O. Ulusoy, A. Geiger, OctNet: Learning deep 3D representations at high resolutions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6620–6629, https://doi.org/10.1109/CVPR.2017.701.

[34] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, Medical Image Computing and Computer Assisted Intervention (2015) 234–241.

[35] M. Shang, X. Luo, Z. Liu, J. Chen, Y. Yuan, M. Zhou, Randomized latent factor model for high-dimensional and sparse matrices from industrial applications, IEEE/CAA Journal of Automatica Sinica 6 (2019) 131–141.

[36] A. Sharma, O. Grau, M. Fritz, VConv-DAE: Deep volumetric shape learning without object labels, in: Computer Vision - ECCV 2016 Workshops, PT III, 2016, pp. 236–250. DOI: 10.1007/978-3-319-49409-8_20.

[37] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2017) 640–651, https://doi.org/10.1109/TPAMI.2016.2572683.

[38] M. Simonovsky, N. Komodakis, Dynamic edge-conditioned filters in convolutional neural networks on graphs, in: 2017 Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 29–38, https://doi.org/10.1109/CVPR.2017.11.

[39] K.K. Singh, M.K. Bajpai, R.K. Pandey, A novel approach for enhancement of geometric and contrast resolution properties of low contrast images, IEEE-CAA Journal of Automatica Sinica 5 (2018) 628–638.

[40] S. Song, F. Yu, A. Zeng, A.X. Chang, M. Savva, T. Funkhouser, Semantic scene completion from a single depth image, in: 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 190–198, https://doi.org/10.1109/CVPR.2017.28.

[41] X. Sun, S. Shen, H. Cui, L. Hu, Z. Hu, Geographic, geometrical and semantic reconstruction of urban scene from high resolution oblique aerial images, IEEE-CAA Journal of Automatica Sinica 6 (2019) 118–130, https://doi.org/10.1109/JAS.2017.7510673.

[42] M. Tatarchenko, A. Dosovitskiy, T. Brox, Octree Generating Networks: Efficient convolutional architectures for high-resolution 3D outputs, IEEE International Conference on Computer Vision (ICCV) (2017) 2107–2115, https://doi.org/10.1109/ICCV.2017.230.

[43] L.P. Tchapmi, C.B. Choy, I. Armeni, J. Gwak, S. Savarese, SEGCloud: Semantic segmentation of 3D point clouds, in: International Conference on 3D Vision (3DV), 2017, pp. 537–547, https://doi.org/10.1109/3DV.2017.00067.

[44] J. Varley, C. DeChant, A. Richardson, J. Ruales, P. Allen, Shape completion enabled robotic grasping, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 2442–2447.

[45] L. Wang, H. Wei, Avoiding non-Manhattan obstacles based on projection of spatial corners in indoor environment, IEEE-CAA Journal of Automatica Sinica 7 (2020) 1190–1200, https://doi.org/10.1109/JAS.2020.1003117.

[46] Y. Wang, D.J. Tan, N. Navab, F. Tombari, Adversarial semantic scene completion from a single depth image, in: 2018 International Conference on 3D Vision (3DV), 2018, pp. 426–434, https://doi.org/10.1109/3DV.2018.00056.

[47] G. Xiong, F.Y. Wang, T.R. Nyberg, X. Shang, M. Zhou, Z. Shen, S. Li, C. Guo, From Mind to Products: Towards social manufacturing and service, IEEE-CAA Journal of Automatica Sinica 5 (2018) 47–57.

[48] X. Yan, J. Yang, E. Yumer, Y. Guo, H. Lee, Perspective Transformer Nets: Learning single-view 3D object reconstruction without 3D supervision, in: Advances in Neural Information Processing Systems (NIPS 2016), 2016.

[49] B. Yang, S. Rosa, A. Markham, N. Trigoni, H. Wen, Dense 3D object reconstruction from a single depth view, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (2019) 2820–2834, https://doi.org/10.1109/TPAMI.2018.2868195.

[50] B. Yang, H. Wen, S. Wang, R. Clark, A. Markham, N. Trigoni, 3D object reconstruction from a single depth view with adversarial learning, in: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW 2017), 2017, pp. 679–688. DOI: 10.1109/ICCVW.2017.86.

[51] Y. Yang, C. Feng, Y. Shen, D. Tian, FoldingNet: Point cloud auto-encoder via deep grid deformation, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 206–215, https://doi.org/10.1109/CVPR.2018.00029.

[52] W. Yuan, T. Khot, D. Held, C. Mertz, M. Hebert, PCN: Point completion network, in: 2018 International Conference on 3D Vision (3DV), 2018, pp. 728–737, https://doi.org/10.1109/3DV.2018.00088.
[53] H. Zhao, L. Jiang, C. Fu, J. Jia, Pointweb: Enhancing local neighborhood features for point cloud processing, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5560–5568.
[54] Wu. Zhirong, S. Song, A. Khosla, Yu. Fisher, Linguang Zhang, Xiaoou Tang, J. Xiao, 3D ShapeNets: A deep representation for volumetric shapes, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1912–1920, https://doi.org/10.1109/CVPR.2015.7298801.
[55] H. Zhou, J. Jagadeesan, Real-time dense reconstruction of tissue surface from stereo optical video, IEEE Transactions on Medical Imaging 39 (2020) 400–412, https://doi.org/10.1109/TMI.2019.2927436.
[56] R. Zhu, H.K. Galoogahi, C. Wang, S. Lucey, Rethinking Reprojection: Closing the loop for pose-aware shape reconstruction from a single image, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 57–65, https://doi.org/10.1109/ICCV.2017.16.
[57] S. Zhu, R. Zhang, L. Zhou, T. Shen, T. Fang, P. Tan, L. Quan, Very large-scale global SfM by distributed motion averaging, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4568–4577.

Meihua Zhao received her B.E. degree in Electronic Information Engineering from China University of Geosciences, Wuhan, China, in 2018. She is currently working towards a Ph.D. degree at the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences as well as the School of Artificial Intelligence, University of Chinese Academy of Sciences. Her research interests include computer vision and 3D printing.

Gang Xiong (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees from Xi'an University of Science and Technology, Xi'an, China, in 1991 and 1994, respectively, and Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 1996. From 1996 to 1998, he was a Postdoctor and Associate Scientist with Zhejiang University, Hangzhou, China. From 1998 to 2001, he was a Senior Research Fellow with Tampere University of Technology, Tampere, Finland. From 2001 to 2007, he was a Specialist and Project Manager with Nokia Corporation, Finland. In 2007, he was a Senior Consultant and Team Leader with Accenture and Chevron, USA. In 2008, he was the Deputy Director of the Informatization Office, Chinese Academy of Science (CAS), Beijing, China. In 2009, he started his present position as a Research Scientist with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, CAS. In 2011, he became Deputy Director of Cloud Computing Center, CAS. From 2015 to 2017, he was a FinDiPro Professor with Aalto University, Finland. His research interests include parallel control and management, modeling and optimization of complex systems, cloud computing and big data, intelligent manufacturing, and intelligent transportation systems.

MengChu Zhou (Fellow, IEEE) received his B.S. degree in Control Engineering from Nanjing University of Science and Technology, Nanjing, China in 1983, M.S. degree in Automatic Control from Beijing Institute of Technology, Beijing, China in 1986, and Ph. D. degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY in 1990. He joined New Jersey Institute of Technology (NJIT), Newark, NJ in 1990, and is now Distinguished Professor in Electrical and Computer Engineering. His research interests are in Petri nets, intelligent automation, Internet of Things, big data, web services, and intelligent transportation. He has over 900 publications including 12 books, 600+ journal papers (450+ in IEEE transactions), 26 patents and 29 book-chapters. He is the founding Editor of IEEE Press Book Series on Systems Science and Engineering, Editor-in-Chief of IEEE/CAA Journal of Automatica Sinica, and Associate Editor of IEEE Internet of Things Journal, IEEE Transactions on Intelligent Transportation Systems, and IEEE Transactions on Systems, Man, and Cybernetics: Systems. He is a recipient of Humboldt Research Award for US Senior Scientists from Alexander von Humboldt Foundation, Franklin V. Taylor Memorial Award and the Norbert Wiener Award from IEEE Systems, Man and Cybernetics Society, and Excellence in Research Prize and Medal from NJIT. He is a life member of Chinese Association for Science and Technology-USA and served as its President in 1999. He is a Fellow of IEEE International Federation of Automatic Control (IFAC), American Association for the Advancement of Science (AAAS) and Chinese Association of Automation (CAA).

Zhen Shen (Member, IEEE) received the B.E. degree in automation and Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 2004 and 2009, respectively.
He is currently an Associate Professor with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, and also with the Intelligent Manufacturing Center, Qingdao Academy of Intelligent Industries, Qingdao, China. He has authored about 40 referred journal and conference papers, and holds 10 authorized patents of China and 3 PCTs. His current research interests include intelligent manufacturing and complex systems.
Dr. Shen was a recipient of the 2005 "Outstanding Achievement Award" from United Technology Research Center.

Fei-Yue Wang (Fellow, IEEE) received his Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He joined The University of Arizona in 1990 and became a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center at the Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Chinese Talents Program from the State Planning Council, and in 2002, was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory for Management and Control of Complex Systems.
His current research focuses on methods and applications for parallel intelligence, social computing, and knowledge automation. He is a fellow of INCOSE, IFAC, ASME, and AAAS. In 2007, he received the National Prize in Natural Sciences of China and became an Outstanding Scientist of ACM for his work in intelligent control and social computing. He received the IEEE ITS Outstanding Application and Research Awards in 2009 and 2011, respectively. In 2014, he received the IEEE SMC Society Norbert Wiener Award. Since 1997, he has been serving as the General or Program Chair of over 30 IEEE, INFORMS, IFAC, ACM, and ASME conferences. He was the President of the IEEE ITS Society from 2005 to 2007, the Chinese Association for Science and Technology, USA, in 2005, the American Zhu Kezhen Education Foundation from 2007 to 2008, the Vice President of the ACM China Council from 2010 to 2011, the Vice President and the Secretary General of the Chinese Association of Automation from 2008–2018. He was the Founding Editor-in-Chief (EiC) of the International Journal of Intelligent Control and Systems from 1995 to 2000, the IEEE ITS Magazine from 2006 to 2007, the IEEE/CAA JOURNAL OF AUTOMATICA SINICA from 2014–2017, and the China's Journal of Command and Control from 2015-2020. He was the EiC of the IEEE Intelligent Systems from 2009 to 2012, the IEEE TRANSACTIONS ON Intelligent Transportation Systems from 2009 to 2016, and is the EiC of the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS since 2017, and the Founding EiC of China's Journal of Intelligent Science and Technology since 2019. Currently, he is the President of CAA's Supervision Council, IEEE Council on RFID, and Vice President of IEEE Systems, Man, and Cybernetics Society.