SeqXY2SeqZ: Structure Learning for 3D Shapes by Sequentially Predicting 1D Occupancy Segments from 2D Coordinates

Zhizhong Han[1,2], Guanhui Qiao[1], Yu-Shen Liu[1(✉)], and Matthias Zwicker[2]

[1] School of Software, BNRist, Tsinghua University, Beijing, People's Republic of China
qiaogh18@mails.tsinghua.edu.cn, liuyushen@tsinghua.edu.cn
[2] Department of Computer Science, University of Maryland, College Park, USA
h312h@umd.edu, zwicker@cs.umd.edu

Abstract. Structure learning for 3D shapes is vital for 3D computer vision. State-of-the-art methods show promising results by representing shapes using implicit functions in 3D that are learned using discriminative neural networks. However, learning implicit functions requires dense and irregular sampling in 3D space, which also makes the sampling methods affect the accuracy of shape reconstruction during test. To avoid dense and irregular sampling in 3D, we propose to represent shapes using 2D functions, where the output of the function at each 2D location is a sequence of line segments inside the shape. Our approach leverages the power of functional representations, but without the disadvantage of 3D sampling. Specifically, we use a voxel tubelization to represent a voxel grid as a set of tubes along any one of the X, Y, or Z axes. Each tube can be indexed by its 2D coordinates on the plane spanned by the other two axes. We further simplify each tube into a sequence of occupancy segments. Each occupancy segment consists of successive voxels occupied by the shape, which leads to a simple representation of its 1D start and end location. Given the 2D coordinates of the tube and a shape feature as condition, this representation enables us to learn 3D shape structures by sequentially predicting the start and end locations of each occupancy segment in the tube. We implement this approach using a Seq2Seq model with attention, called SeqXY2SeqZ, which learns the mapping from a sequence of 2D coordinates along two arbitrary axes to a sequence of 1D locations along the third axis. SeqXY2SeqZ not only benefits from the regularity of voxel grids in training and testing,

This work was supported by National Key R&D Program of China (2020YFF0304100, 2018YFB0505400), the National Natural Science Foundation of China (62072268), and NSF (award 1813583).

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58586-0_36) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12369, pp. 607–625, 2020.
https://doi.org/10.1007/978-3-030-58586-0_36

but also achieves high memory efficiency. Our experiments show that SeqXY2SeqZ outperforms the state-of-the-art methods under the widely used benchmarks.

Keywords: 3D reconstruction · Voxel grids · Implicit function · RNN · Attention

1 Introduction

3D voxel grids are an attractive representation for 3D structure learning because they can represent shapes with arbitrary topology and they are well suited to convolutional neural network architectures. However, these advantages are dramatically diminished by the disadvantage of cubic storage and computation complexity, which significantly affects the structure learning efficiency and accuracy of deep learning models.

Recently, implicit functions have been drawing research attention as a promising 3D representation to resolve this issue. By representing a 3D shape as a function, discriminative neural networks can be trained to learn the mapping from a 3D location to a label, which can either indicate the inside or outside of the shape [4,42,51] or a signed distance to the surface [47,60]. As a consequence, shape reconstruction requires sampling the function in 3D, where the 3D locations are required to be sampled near the 3D surface for training. Recent approaches based on implicit functions have shown superiority over point clouds in terms of geometry details, and advantages over meshes in terms of being able to represent arbitrary topologies. Although it is very memory efficient to learn implicit functions using discriminative models, these approaches require sampling dense 3D locations in a highly irregular manner during training, which also makes the sampling methods affect the accuracy of shape reconstruction during test.

To resolve this issue, we propose a method for 3D shape structure learning by leveraging the advantages of learning shape representations based on continuous functions without requiring sampling in 3D. Rather than regarding a voxel grid as a set of individual 3D voxels, which suffers from cubic complexity in learning, we represent voxel grids as functions over a 2D domain that map 2D locations to 1D voxel tubes. This voxel tubelization regards a voxel grid as a set of tubes along any one of three dimensions, for example Z, and indexes each tube by its 2D location on the plane spanned by the other two dimensions, i.e., X and Y. In addition, each tube is represented as a sequence of occupancy segments, where each segment consists of successive occupied voxels given by two 1D locations indicating the start and end points. Given a shape feature as a condition, this voxel tubelization enables us to propose a Seq2Seq model with attention as a discriminative model to predict each tube from its 2D location. Specifically, we leverage an RNN encoder to encode the 2D coordinates of a tube with a shape condition, and leverage an RNN decoder to sequentially predict the start and end locations of each occupancy segment in the tube. Because our approach essentially maps a coordinate sequence to another coordinate sequence, we call

our method *SeqXY2SeqZ*. Given the 2D coordinates of a tube, SeqXY2SeqZ produces the 1D coordinates of the occupancy segments along the third dimension. Not only can SeqXY2SeqZ be evaluated with a number of RNN steps that is quadratic in the grid resolution during test, but it is also memory efficient enough to learn high resolution shape representations. Experimental results show that SeqXY2SeqZ outperforms the state-of-the-art methods.

Our contributions are as follows. First, we propose a novel shape representation based on 2D functions that map 2D locations to sequences of 1D voxel tubes, avoiding the cubic complexity of voxel grids. Our representation enables 3D structure learning of voxel grids in a tube-by-tube manner via discriminative neural networks. Second, we propose SeqXY2SeqZ, an RNN-based Seq2Seq model with attention, to implement the mapping from 2D locations to 1D sequences. Given a 2D coordinate and a shape condition, SeqXY2SeqZ sequentially predicts occupancy segments in a 1D tube. It requires a number of RNN steps that grows only quadratically with resolution, and achieves high resolutions due to its memory efficiency. Third, SeqXY2SeqZ demonstrates the feasibility of generating 3D voxel grids using discriminative neural networks in a more efficient way, and achieves state-of-the-art results in shape reconstruction.

2 Related Work

Deep learning models have made big progress in 3D shape understanding tasks [13–20,22–25,27,28,40,41,49,62,63]. Recent 3D structure learning methods are also mainly based on deep learning models.

Voxel-Based Models. Because of their regularity, many previous studies learned 3D structures from voxel grids with 3D supervision [6,50] or 2D supervision with the help of differentiable renderers [8,9,56,57,66,67]. Due to the cubic complexity of voxel grids, these generative models are limited to relatively low resolution, such as 32^3. Recent studies [6,65,70] employed shallow 3D convolutional networks to reconstruct voxel grids in higher resolutions of 128^3, however, the computational cost is still very large. To remedy this issue, some methods employed a multi-resolution strategy [26,54]. However, these methods are very complicated to implement and additionally require multiple passes over the input. Another alternative was introduced to represent 3D shapes using multiple depth images [50]. However, it is hard to obtain consistency across multiple generated depth images during inference.

Different from these generative neural networks, we provide a novel perspective to benefit from the regularity of voxel grids but avoid their cubic complexity by leveraging discriminative neural networks in shape generation. Moreover, our representation is different from multi-layer depth maps [52] and scanline [64], since we do not require additional support, such as binary masks [52] or edge end point determination.

Point Cloud-Based Models. As pioneers, PointNet [48] and PointNet++ [49] enabled the learning of 3D structure from point clouds. Later, different variations were proposed to improve the learning of 3D structures from 3D point

clouds [7,24,40] or 2D images with various differentiable renderers [11,29,32,33,44,68]. Although point clouds are a compact and memory efficient 3D representation, they cannot express geometry details without additional non-trivial post-processing steps to generate meshes.

Mesh-Based Models. 3D meshes are also attractive in deep learning [3,10,31,34,35,38,58,61]. Supervised methods employed 3D meshes as supervision to train networks by minimizing the location error of vertices with geometry constraints [10,58,61], while unsupervised methods relied on differentiable renderers to reconstruct meshes from multiple views [3,31,34,35,38]. However, these methods cannot generate arbitrary vertex topology but inherit the connectivity of the template mesh.

Implicit Function-Based Models. Recently, implicit functions have become a promising 3D representation in deep learning models [4,42,43,46,47,51,60]. By representing a 3D shape as a 3D function, these methods employ discriminative neural networks to learn the function from a 3D location to an indicator labelling inside or outside of the shape [4,42,51] or a signed distance to the surface [47,60]. However, these methods required to sample points near 3D surfaces during training. To learn implicit functions without 3D supervision, different differentiable renderers were proposed to back propagate the loss calculated on 2D images [12,30,36,39,45,53,69]. Although it is very memory efficient to learn 3D implicit functions using discriminative models in a point-by-point manner, supervised method require sampling dense and irregular 3D locations during training, which also makes the sampling methods affect the accuracy of shape reconstruction during test.

Although our method is also a discriminative network for 3D structure learning, it can benefit from the regularity of voxel grids by learning a 2D function. It is memory efficient and avoids the dense and irregular sampling during training.

3 Overview

The core idea of SeqXY2SeqZ is to represent shapes as 2D functions that map each 2D location to a sequence of 1D occupancy segments. More specifically, we interpret each 3D shape $\boldsymbol{M}$ as a set of 1D tubes $\boldsymbol{t}_i$, where each tube $\boldsymbol{t}_i$ is indexed by its 2D coordinate $\boldsymbol{c}_i$. Tube $\boldsymbol{t}_i$ consists of a sequence of occupancy segments, where we represent each segment o_j by its 1D start and end locations s_j and e_j. To generate $\boldsymbol{M}$, SeqXY2SeqZ learns a 2D function to predict each tube $\boldsymbol{t}_i$ from its coordinate $\boldsymbol{c}_i$ and a shape condition by generating the start and end locations s_j and e_j of each occupancy segment o_j in $\boldsymbol{t}_i$.

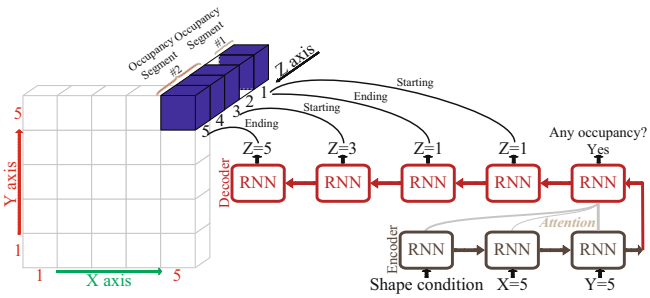

Fig. 1. The overview of SeqXY2SeqZ.

Figure 1 illustrates how SeqXY2SeqZ generates a tube along the Z axis from its 2D coordinates on the X-Y plane. Specifically, we input the 2D coordinate $X = 5$ and $Y = 5$ sequentially into an encoder, and a decoder sequentially predicts the start and end locations of two occupancy segments along the Z axis. In the figure, there is one occupancy segment with only one voxel starting at $Z = 1$ and ending at $Z = 1$, and a second segment starting at $Z = 3$ and ending at $Z = 5$. Therefore, the decoder sequentially predicts $Z = 1$, $Z = 1$, $Z = 3$, $Z = 5$ to reconstruct the tube at $X = 5$ and $Y = 5$. In addition, the decoder outputs a binary flag to indicate whether there is any occupancy segment in this tube at all. The encoder also requires a shape condition from an image or a learned feature as input to provide information about the reconstructed shape.

4 Voxel Tubelization

To train the SeqXY2SeqZ model, we first need to convert each 3D voxel grid into a tubelized representation consisting of sets of 1D voxel tubes over a 2D plane. For a 3D shape $\boldsymbol{M}$ represented by a grid with a resolution of R^3, voxel tubelization re-organizes these R^3 voxels into a set of $R \times R$ tubes $\boldsymbol{t}_i$ along one of the three axes. Each tube $\boldsymbol{t}_i$ can then be indexed by its location on the plane spanned by the other two dimensions using a 2D coordinate $\boldsymbol{c}_i$, such that $\boldsymbol{M} = \{(\boldsymbol{c}_i, \boldsymbol{t}_i)|i \in [1, R^2]\}$. We further represent each tube $\boldsymbol{t}_i$ using run-length encoding of its J_i occupancy segments o_j, where $j \in [1, J_i]$ and $J_i \in [1, R]$. An occupancy segment is a set of consecutive voxels that are occupied by the shape, which we encode as a sequence of start and end locations s_j and e_j. Note that s_j and e_j are discrete 1D indices, which we will predict using a discriminative approach. We denote the tubes consisting of occupancy segments as $\boldsymbol{t}_i = [s_1, e_1, ..., s_j, e_j, ..., s_{J_i}, e_{J_i}]$. In our experimental section we show that this representation is effective irrespective of the axis that is leveraged for the tubelization.

Our approach takes advantage of the following properties of voxel tubelization and run-length encoding of occupancy segments. First, run-length encoding of occupancy segments significantly reduces the memory complexity of 3D grids, since only two indices are needed to encode each segment, irrespective of its length. Second, our approach allows us to represent shapes as 2D functions that map 2D locations to sequences of 1D occupancy segments, which we will implement using discriminative neural networks. This is similar to shape

representations based on 3D implicit functions implemented by discriminative networks, but our approach requires only $\mathcal{O}(R^2)$ RNN evaluation steps during shape reconstruction. Third, networks that predict voxel occupancy using a scalar probability require an occupancy probability threshold as a hyperparameter, which can have a large influence on the reconstruction accuracy. In contrast, we predict start and end locations of occupancy segments and do not require such a parameter.

5 SeqXY2SeqZ

SeqXY2SeqZ generates each tube $\boldsymbol{t}_i$ from its coordinate $\boldsymbol{c}_i$ and a shape condition. We use an RNN encoder to encode the coordinate $\boldsymbol{c}_i$ and the shape condition, while an RNN decoder produces the start and end locations of the occupancy segments o_j in $\boldsymbol{t}_i$.

RNN Encoder. We condition the RNN encoder on a global shape feature $\boldsymbol{f} \in \mathbb{R}^{1\times D}$ that represents the unique 3D structure of each object. For example, in 3D shape reconstruction from a single image, $\boldsymbol{f}$ could be a feature vector extracted from an image to guide the 3D shape reconstruction. In a 3D shape to 3D shape translation application, $\boldsymbol{f}$ could be a feature vector that can be jointly learned with other parameters in the networks, such as shape memories [21] or codes [47].

As shown in Fig. 2(a), the RNN encoder aggregates the shape condition $\boldsymbol{f}$ and a 2D coordinate $\boldsymbol{c}_i = [c_i^1, c_i^2]$ into a hidden state $\boldsymbol{h}_i$, which is subsequently leveraged by the RNN decoder to generate the corresponding tube $\boldsymbol{t}_i$. Rather than directly employing a location c_i^1 or c_i^2 as a discrete integer, we leverage the location as a location embedding $\boldsymbol{x}_i^1$ or $\boldsymbol{x}_i^2$, which makes locations meaningful in feature space. In this way, we have a location embedding matrix along each axis, i.e., $\mathbf{F}_X$, $\mathbf{F}_Y$ and $\mathbf{F}_Z$. Each matrix holds the location embedding of all R locations along an axis as R rows, i.e., $\mathbf{F}_X \in \mathbb{R}^{R\times D}$, $\mathbf{F}_Y \in \mathbb{R}^{R\times D}$ and $\mathbf{F}_Z \in \mathbb{R}^{R\times D}$, so that we can get an embedding for a specific location by looking up the location embedding matrix. In the case of tubelizing along the Z axis demonstrated in Fig. 1, the RNN encoder would employ the location embeddings along the X and Y axes, that is $\boldsymbol{x}_i^1 = \mathbf{F}_X(c_i^1)$ and $\boldsymbol{x}_i^2 = \mathbf{F}_Y(c_i^2)$.

We employ Gated Recurrent Units (GRU) [5] as the RNN cells in SeqXY2SeqZ. At each step, a hidden state is produced, and the hidden state $\boldsymbol{h}_i$ at the last step is leveraged by the RNN decoder to predict a tube $\boldsymbol{t}_i$ for the reconstruction of a shape conditioned on $\boldsymbol{f}$, where $\boldsymbol{h}_i \in \mathbb{R}^{1\times H}$. The encoding process is detailed in our supplemental material.

Location Embedding. Although we could employ three different location embedding matrices to hold embeddings for locations along the X, Y, and Z axes separately, we use $\mathbf{F}_X$, $\mathbf{F}_Y$ and $\mathbf{F}_Z$ in a shareable manner. For example, we can employ the same location embedding matrix on the plane used for indexing the 1D tubes, such as $\mathbf{F}_X = \mathbf{F}_Y$ in the case shown in Fig. 1. In our experiments, we justify that we can even employ only one location embedding matrix

along all three axes, that is $\mathbf{F}_X = \mathbf{F}_Y = \mathbf{F}_Z$. The shareable location embeddings significantly increase the memory efficiency of SeqXY2SeqZ.

RNN Decoder. With the hidden state $\boldsymbol{h}_i$ from the RNN encoder, the RNN decoder needs to generate a tube $\boldsymbol{t}_i$ for the shape indicated by condition $\boldsymbol{f}$ via sequentially predicting the start and end locations of each occupancy segment o_j. To interpret the prediction of tubes with no occupancy segments, we include an additional global occupancy indicator b that the decoder predicts first, where $b = 1$ indicates that there are occupancy segments in the current tube.

We denote $\boldsymbol{w}_i$ as the concatenation of b and $\boldsymbol{t}_i$, such that $\boldsymbol{w}_i = [b, s_1, e_1, ..., s_{J_i}, e_{J_i}]$, where each element in $\boldsymbol{w}_i$ is uniformly denoted as w_i^k and $k \in [1, 2 \times J_i + 1]$. Note that the start and end points s_j and e_j are discrete voxel locations, which we interpret as class labels. In each step, the RNN decoder selects a discrete label to determine either start or end location. Therefore, we leverage the following cross entropy classification loss to push the decoder to predict the correct label sequence $\boldsymbol{w}_i$ as accurately as possible under the training set,

$$L = -\sum_{k \in [1, 2 \times J_i + 1]} \log p(w_i^k | w_i^{<k}, \boldsymbol{h}_i), \tag{1}$$

where w_i^k is the k-th element in the sequence $\boldsymbol{w}_i$, $w_i^{<k}$ represents the elements in front of w_i^k, $p(w_i^k | w_i^{<k}, \boldsymbol{h}_i)$ is the probability of correctly predicting the k-th element according to the previous elements $w_i^{<k}$ and the hidden state $\boldsymbol{h}_i$ from the encoder. Finally, our objective function is given as

$$\mathbf{F}_X^*, \mathbf{F}_Y^*, \mathbf{F}_Z^*, \boldsymbol{\theta}^*, \boldsymbol{f}^* = \underset{\mathbf{F}_X, \mathbf{F}_Y, \mathbf{F}_Z, \boldsymbol{\theta}, \boldsymbol{f}}{\arg\min} L, \tag{2}$$

where $\boldsymbol{\theta}$ denotes the parameters of the RNN encoder and decoder, $\boldsymbol{f}$ is the shape condition, which is fixed or trainable depending on the application, and the location embedding matrices $\mathbf{F}_X, \mathbf{F}_Y, \mathbf{F}_Z$ could be shareable.

Training progress in a step by step manner is shown in Fig. 2(b). At the k-th step, element w_i^k in sequence $\boldsymbol{w}_i$ is predicted through a softmax layer. For example, w_i^1 is either true or false for the global occupancy indicator b, and w_i^2 and w_i^3 are the start and end locations s_1 and e_1 of the occupancy segment o_1 in the range of $[1, R]$, etc. In addition, for each w_i^k we look up its location embedding $\boldsymbol{y}_i^k$ from the location embedding matrix of the coordinate axis corresponding to the tube direction. The embedding $\boldsymbol{y}_i^k$ is then used in the prediction of w_i^{k+1} at the next step. For example, in the tubelization along the Z axis demonstrated in Fig. 1, $\boldsymbol{y}_i^k$ is looked up in $\boldsymbol{F}_Z$, such that $\boldsymbol{y}_i^k = \boldsymbol{F}_Z(w_i^k)$, where each row of $\boldsymbol{F}_Z$ represents an embedding for a location, and two additional rows for a true or false of b.

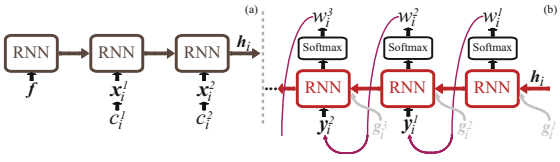

Fig. 2. The illustration of RNN encoder (a) and RNN decoder (b).

Attention. Finally, we leverage a state-of-the-art attention mechanism [1] to increase the prediction accuracy of the predicted locations. We employ a context vector $\boldsymbol{g}_i^k$ for the prediction of w_i^k, where $\boldsymbol{g}_i^k$ summarizes how well each step of the encoder matches the prediction of w_i^k. The decoding with attention is detailed in supplemental material.

6 Experiments and Analysis

We employ tubelization along the Y axis in all our experiments and learn only two location embedding matrices. We share the location embedding matrices along the X and Z axes providing the 2D coordinates of tubes, such that $\mathbf{F}_X = \mathbf{F}_Z$, while we use a separate matrix along the Y axis. The location embedding is $D = 512$-dimensional, and the hidden state of the RNNs is also $H = 512$-dimensional, where the RNN encoder is bidirectional.

We train SeqXY2SeqZ using the Adam optimizer with $\epsilon = 8 \times 10^{-6}$, with a batch size of 64 and a learning rate of 1×10^{-3} in all experiments. The maximum number of steps in the encoder and decoder are 4 and 30, respectively. We employ volumetric IoU to evaluate the accuracy of the reconstructed shapes, and all reported IoU values are multiplied by 10^2.

Table 1. Reconstruction (64^3) comparison in terms of IoU.

Methods	Plane	Car	Chair	Rifle	Table
IM-AE [4]	78.77	89.36	65.65	72.88	71.44
CNN-AE [4]	86.07	90.73	74.22	78.37	84.67
OccNet(Train) [42]	–	–	89.00	–	–
Our(512–512)	**90.35**	**91.18**	74.32	**84.46**	**86.21**
Our(1024–2048)	–	–	**93.10**	–	–

6.1 Representation Ability

Dataset. For fair comparison, we leverage five widely used categories from ShapeNetCore [2] in this subsection, including airplane, car, chair, rifle, and table, and keep the same train and test splitting as [4]. The ground truth shapes are also voxelized at a resolution of 64^3, such that $R = 64$.

Auto-Encoding. We evaluate the representation ability of SeqXY2SeqZ in an auto-encoding task. We leverage a learnable shape condition $\boldsymbol{f}$ to represent each shape. Specifically, shape features $\boldsymbol{f}$ are learned together with the other parameters in the RNN during training. During testing, we keep updating the shape features while fixing the parameters in the RNN including the location embedding matrices, which is similar as introduced by shape memories [21] or codes [47]. Note that $\boldsymbol{f}$ are also $D = 512$-dimensional vectors, similar as the location embeddings.

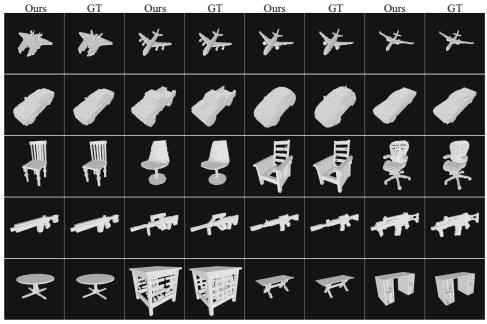

Fig. 3. Auto-decoded shapes by learned features.

In this task, we compare SeqXY2SeqZ with results from the implicit decoder (IM) [4] and occupancy network (OccNet) [42]. We show the comparison in Table 1, where the mean IoU over the first 100 shapes in the test set of each category is reported by IM while OccNet only reported its results on the training set of chair at a resolution of 256.

As shown by "Our(512-512)" in Table 1, our results with $D = 512$-dimensional location embeddings and $H = 512$-dimensional hidden states are the best among all compared methods under all shape categories. If we increase the learning ability of SeqXY2SeqZ by using location embeddings and hidden states with higher dimensions, such as $D = 2048$ and $H = 1024$ shown by "Our(2048-1024)", we achieve an even higher IoU of 93.10 under the challenging chair class.

Table 2. Tubelization direction comparison.

	Along Y	Along Z	Along X
IoU	**90.35**	89.96	90.21

In Fig. 3, we visualize the reconstructed shapes in the test set of each category with our best results in Table. 1. The reconstructed shapes with high fidelity demonstrate that SeqXY2SeqZ is capable of learning very complex structures of 3D shapes, such as the ones on chairs and tables.

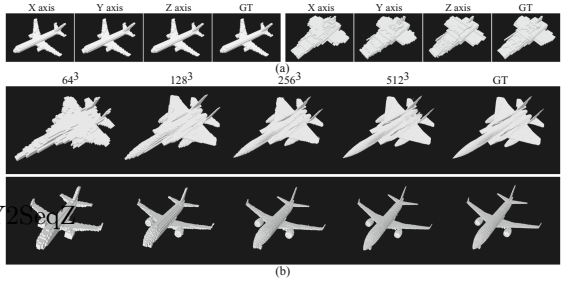

Fig. 4. (a) Qualitative comparison of reconstructions with tubelization along different axes. (b) Auto-encoded shapes in different resolutions.

Tubelization Direction. We can tubelize a voxel grid along any one of the X, Y or Z axes, which should be kept consistent in training and testing. Although the tubelization direction may lead to different ways of 3D structure learning, SeqXY2SeqZ does not exhibit any bias on the tubelization direction. We demonstrate this by training SeqXY2SeqZ using voxel grids tubelized under the X, Y and Z axis, respectively. Table 2 shows that we achieve comparable results along the three tubelization directions under the airplane class. Visual comparisons are shown in Fig. 4(a).

High Resolutions. Thanks to the 2D functions and the shareable location embedding matrices, SeqXY2SeqZ is memory efficient enough to reconstruct shapes in high resolutions. We show auto-encoded airplanes in different resolutions in Fig. 4(b). The high fidelity shapes justify our capabilities of high resolution reconstruction.

6.2 Single Image 3D Reconstruction

Dataset. We employ the dataset released from [6], which contains 3D shapes from 13 categories in the ShapeNetCore [2]. We also use the same train and test splitting, where each shape is represented as a voxel grid with a resolution of 32^3 accompanying 24 rendered images. While many 3D reconstruction techniques

(including ours, see Table 1 and Fig. 4) support higher resolutions, we follow previous works [37,50,55] and choose ground truth voxel grids in the benchmark to provide a comparison to a broad range of competing approaches.

Single Image Reconstruction. We leverage a CNN encoder from [38] to extract a 512 dimensional feature from a rendered image as a shape condition in this experiment. We compare with the state-of-the-art supervised and unsupervised methods in Table 3. Among these methods, "DISN-V" is a network formed by a DISN [60] encoder and a 3D CNN decoder, "DISN-C" is DISN [60] working with the estimated camera poses which is required in the reconstruction, "PTN-R" is the result using retrieval from PTN [67]. Besides the voxel-based methods including R2N2 [6], PTN [67] and Matryoshka [50], all the other methods represent 3D shapes as triangle meshes, where IM [4], OccNet [42], and DISN [60] are based on learning 3D implicit functions. For fair comparison, all the results listed here are taken from the literature rather than being reproduced by us. For example, the results of NMR [31], SoftRas [37] and DIB-R [3] are all from DIB-R [3].

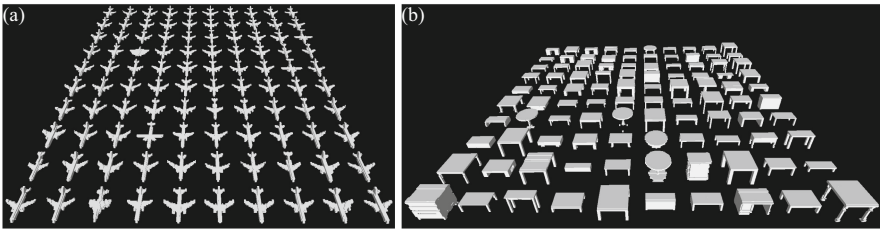

Fig. 5. Single image reconstruction for airplanes (a) and tables (b).

Table 3 demonstrates the performance of our method, showing that in terms of the mean IoU we improve by 6.3 over the best 3D implicit function based method (DISN) and by 2.1 over the best unsupervised method (DIB-R). We achieve the best IoU in 7 out of 13 categories among all supervised methods, and in 8 out of 13 categories among all unsupervised methods. Matryoshka [50] comes closest to our performance, but it employs non-standard augmentation on training images, which we omit. Figure 6 shows a visual comparison, where the shapes are reconstructed from the same input images using the trained network parameters released by different methods. Although we trained our method at a resolution of 32^3, the high accuracy enables us to reveal complex geometry that other methods cannot handle, which makes our results comparable to the meshes reconstructed by other methods. Figure 5 shows additional airplanes and tables reconstructed by our method.

Table 3. Quantitative comparison of single image 3D shape reconstruction in terms of IoU.

	Method	Modality	Plane	Bench	Cabinet	Car	Chair	Display	Lamp	Speaker	Rifle	Sofa	Table	Phone	Boat	Mean
Supervised	AtlasNet [10]	Mesh	39.2	34.2	20.7	22.0	25.7	36.4	21.3	23.2	45.3	27.9	23.3	42.5	28.1	30.0
	Pixel2mesh [59]		51.5	40.7	43.4	50.1	40.2	55.9	29.1	52.3	50.9	60.0	31.2	69.4	40.1	47.3
	3DN [60]		54.3	39.8	49.4	59.4	34.4	47.2	35.4	45.3	57.6	60.7	31.3	71.4	46.4	48.7
	R2N2 [6]	Voxel	51.3	42.1	71.6	79.8	46.6	46.8	38.1	66.2	54.4	62.8	51.3	66.1	51.3	56.0
	Matryoshka [51]		64.7	57.7	**77.6**	**85.0**	**54.7**	53.2	40.8	**70.1**	61.6	68.1	**57.3**	75.6	**59.1**	63.5
	IM [4]	3D Implicit	55.4	49.5	51.5	74.5	52.2	56.2	29.6	52.6	52.3	64.1	45.0	70.9	56.6	54.6
	OccNet [43]		54.7	45.2	73.2	73.1	50.2	47.9	37.0	65.3	45.8	67.1	50.6	70.9	52.1	56.4
	DISN-V [61]		50.6	44.3	52.3	76.9	52.6	51.5	36.2	58.0	50.5	67.2	50.3	70.9	57.4	55.3
	DISN-C [61]		57.5	52.9	52.3	74.3	54.3	56.4	34.7	54.9	59.2	65.9	47.9	72.9	55.9	57.0
	Ours	2D Implicit	**73.2**	**58.5**	71.0	78.1	50.3	**60.0**	**44.7**	62.2	**66.7**	**68.4**	55.0	**80.2**	58.4	**63.6**
Unsupervised	NMR [31]	Mesh	58.5	45.7	74.1	71.3	41.4	55.5	36.7	67.4	55.7	60.2	39.1	76.2	59.4	57.0
	SoftRas [37]		58.4	44.9	73.6	77.1	49.7	54.7	39.1	68.4	62.0	63.6	45.3	75.5	58.9	59.3
	DIB-R [3]		57.0	49.8	**76.3**	**78.8**	52.7	58.8	40.3	**72.6**	56.1	67.7	50.8	74.3	**60.9**	61.2
	PTN-R [68]	Voxel	55.6	48.8	57.1	65.2	35.1	39.6	29.1	46.0	51.3	53.1	31.0	67.0	40.8	47.7
	PTN [68]		55.6	49.2	68.2	71.2	44.9	54.0	42.2	58.7	59.9	62.2	49.4	75.0	55.1	57.4
	IMRender [40]	3D Implicit	65.1	53.6	-	78.2	**54.8**	-	-	-	-	-	51.5	-	60.8	60.7
	Ours	2D Implicit	**73.2**	**58.5**	71.0	78.1	50.3	**60.0**	**44.7**	62.2	**66.7**	**68.4**	**55.0**	**80.2**	58.4	**63.6**

6.3 Ablation Studies and Analysis

Ablation studies. We highlight some elements in our method by ablation studies in single image reconstruction under the chair class in Table 4. We compare our result with the ones without attention ("NoAtt"), the ones with LSTM RNN cells ("LSTM"), and the ones with single direction RNN encoder ("SingleDir"). We find that GRU performs better than LSTM, and both attention mechanism and bidirectional RNN encoder contribute to the performance.

Table 4. Ablation studies under chair class.

	NoAtt	LSTM	SingleDir	ShareableXYZ	Our(GRU)
IoU	47.5	49.8	48.8	49.1	**50.3**

Shareable Location Embedding Matrix. The memory efficiency is one advantage of SeqXY2SeqZ. We achieve this not only by avoiding the direct involvement of 3D voxel grids, but also by sharing the location embedding matrices. The above experiments have shown the effectiveness of shared location embedding matrices for the X and Z axes to define the plane indexing the tubes. In this experiment, we step further by employing only one location embedding matrix for all three axes. We also tubelize the voxel grids along the Y axis, and train SeqXY2SeqZ under the chair class in sigle image 3D reconstruction. In Table 4, "ShareableXYZ" still achieves the comparable result with "Our(GRU)".

Fig. 6. Qualitative comparison with the state-of-the-art supervised and unsupervised methods.

Location Embedding Visualization. We visualize the location embeddings learned in auto-encoding of Table 1 in Fig. 7(a), where each class leverages two sets of location embeddings including one shared by the X and Z axes, and the other along the Y axis. We visualize each set of location embeddings using a cosine distance matrix whose element is the pairwise cosine distance between arbitrary two location embeddings. The structure of a shape category is demonstrated by the distinctive patterns on the cosine distance matrix in different shape categories, which demonstrates the effectiveness of the learned location embeddings. In each similarity matrix, blue means more similar between two location embeddings while yellow means more different. The similarity indicates whether the two corresponding locations show similar occupancy surrounding. For a class containing shapes with similar structures, like cars, the patterns are more obvious, while a class containing shapes with large structure variations, like chairs, the patterns are less obvious. In addition, we visualize the location embeddings learned in single image reconstruction of Table 3 in Fig. 7(b), where we also observe the different patterns on the cosine distance matrix in different shape categories. Note that we show the 64^2 dimensional distance matrix in Fig. 7(a) and the 32^2 dimensional distance matrix in Fig. 7(b) in the same size.

Attention Visualization. We further visualize the attention learned in auto-encoding of Table 1. At each 2D coordinate, an attention vector $\boldsymbol{a}$ is learned at each decoder step for all encoder steps. For each decoder step, we leverage entropy $(-\boldsymbol{a}.*log_2\boldsymbol{a})$ to visualize $\boldsymbol{a}$ at all 2D coordinates (if there is no output at this decoder step, we encode -1 at this 2D coordinate) into an attention image, and we normalize the whole attention image using the maximal entropy. We show five attention images at the first five decoder steps for each shape in Fig. 8(a). In each image, the higher entropy (above 0, the lighter color) indicates this decoder step is paying attention more equally on all encoder steps to generate more complex structure, such as chairs, while the lower entropy (above 0, the darker color) indicates this decoder step is focusing on a specific encoder step to generate relatively simple structure, such as cars. Similarly, we visualize the attention learned in single image reconstruction of Table 3 in Fig. 8(b), where the chair can be reconstructed by only one occupancy segment at all 2D coordinates, which makes the attention much simpler than the one for the chair in Fig. 8(a). Note that we show the 64^2 dimensional attention images in Fig. 8(a) and the 32^2 dimensional attention images in Fig. 8(b) in the same size.

Memory and Computation Time. We compare the memory and computation time requirements with methods based on learning 3D implicit functions in Table 5, including OccNet [42] and DISN [60]. To reconstruct a 3D shape at a resolution of R^3 from a single image during test, OccNet [42] requires to get occupancy values for about $3.8 * R^3$ sampled points with sub additional steps of subdivision, while DISN requires to get SDF values for R^3 sampled points, both of which are higher complexity than our $\mathcal{O}(R^2)$ RNN steps. Since DISN cannot run on a single GPU as OccNet and SeqXY2SeqZ, we report a fair comparison in terms of the CPU run time and RAM space with $R = 64$ and $sub = 2$

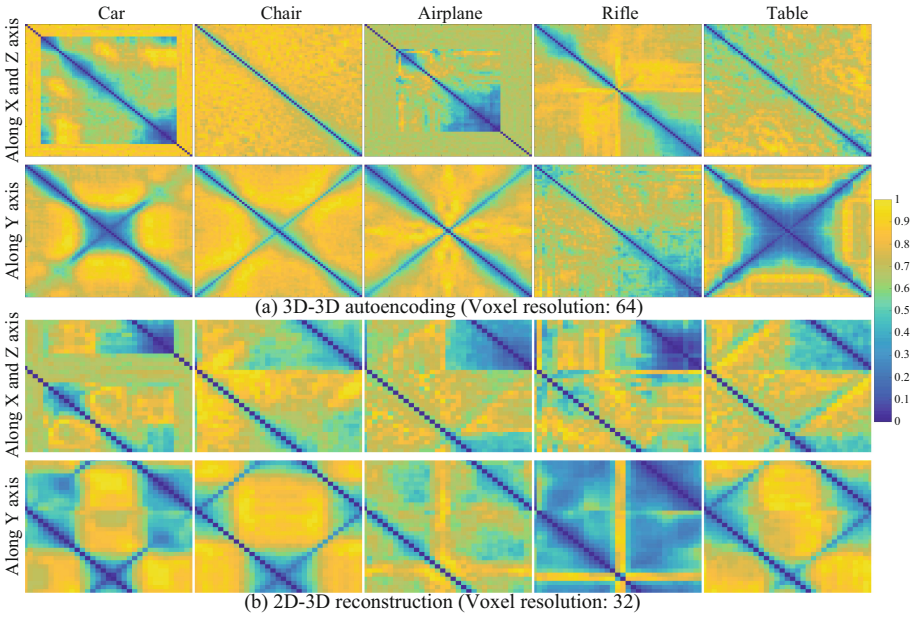

Fig. 7. Pairwise cosine distances of location embeddings learned in auto-encoding (a) and single image reconstruction (b). In each similarity matrix, blue means two locations indicated by their embeddings show similar occupancy surrounding while yellow means more different.

Table 5. Complexity comparison with 3D implicit functions.

	OccNet [42]	DISN [60]	Ours
Network evaluations	$\mathcal{O}(3.8 * R^2)$	$\mathcal{O}(R^3)$	$\mathcal{O}(R^2)$
Time (CPU)	55.80 s	14.68 s	**8.79** s
Space	1175 MB	>11 GB	**286** MB

for reconstructing one shape from a single image. Benefiting from learning 2D functions that predict sparse representations of 1D voxel tubes, SeqXY2SeqZ achieves both the lowest time and memory requirements by a large margin.

More Analysis. More analysis on the efficiency of our voxel tubelization and the feature space learned by our method can be found in our supplemental material.

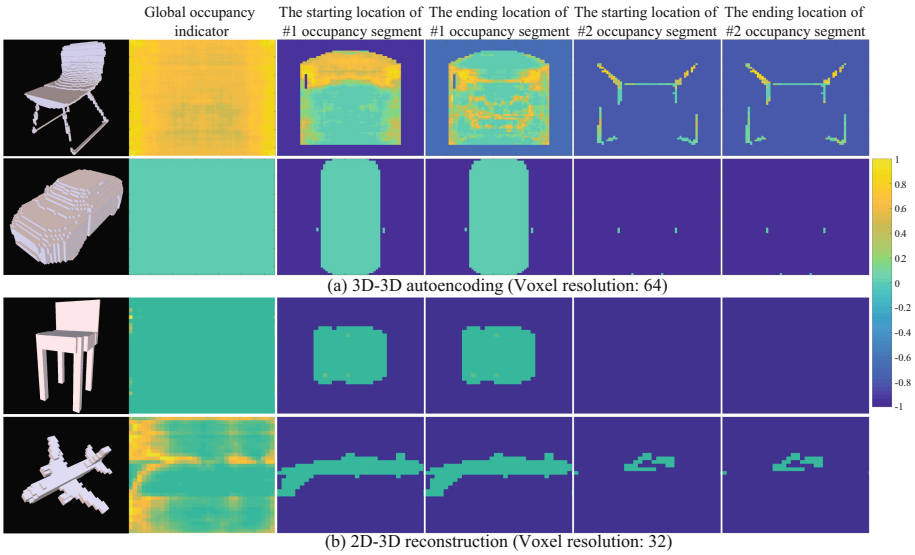

Fig. 8. The visualization of attention learned in auto-encoding. We visualize the attention weights learned at the first five steps of the decoder. The attention at each step for all 2D coordinates is shown as an image, where attention weights on the encoder at each 2D coordinate are encoded as entropy shown by color.

7 Conclusion

We propose SeqXY2SeqZ to learn the structure of 3D shapes using a discriminative neural network not only benefiting from the regularity inherent in voxel grids during both training and testing, but also avoiding cubic complexity for high memory efficiency. SeqXY2SeqZ successfully resolves the issue of dense and irregular sampling during structure learning or inference required by 3D implicit function-based methods, which leads to higher inference times compared to our approach. This is achieved based on the encoding of voxel grids by our 1D voxel tubelization, which effectively represents a voxel grid as a mapping from discrete 2D coordinates to sequences of discrete 1D locations. This mapping further enables SeqXY2SeqZ to effectively learn the 3D structures as 2D functions. We demonstrate that SeqXY2SeqZ outperforms the state-of-the-art methods under widely used benchmarks.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473 (2014)
2. Chang, A.X., et al.: ShapeNet: an information-rich 3D model repository. CoRR abs/1512.03012 (2015)

3. Chen, W., et al.: Learning to predict 3D objects with an interpolation-based differentiable renderer. CoRR abs/1908.01210 (2019)
4. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
5. Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: encoder-decoder approaches. In: SSST@EMNLP, pp. 103–111 (2014)
6. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3D–R2N2: a unified approach for single and multi-view 3D object reconstruction. In: Proceedings of European Conference on Computer Vision, pp. 628–644 (2016)
7. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3D object reconstruction from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2463–2471 (2017)
8. Gadelha, M., Maji, S., Wang, R.: 3D shape induction from 2D views of multiple objects. In: International Conference on 3D Vision, pp. 402–411 (2017)
9. Gadelha, M., Wang, R., Maji, S.: Shape reconstruction using differentiable projections and deep priors. In: International Conference on Computer Vision (2019)
10. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3D surface generation. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
11. Han, Z., Chen, C., Liu, Y.S., Zwicker, M.: DRWR: a differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images. In: ICML (2020)
12. Han, Z., Chen, C., Liu, Y.S., Zwicker, M.: DRWR: a differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images. In: International Conference on Machine Learning (2020)
13. Han, Z., Chen, C., Liu, Y.S., Zwicker, M.: ShapeCaptioner: generative caption network for 3D shapes by learning a mapping from parts detected in multiple views to sentences. In: ACM International Conference on Multimedia (2020)
14. Han, Z., Liu, X., Liu, Y.S., Zwicker, M.: Parts4Feature: learning 3D global features from generally semantic parts in multiple views. In: IJCAI (2019)
15. Han, Z., Liu, Z., Han, J., Vong, C.M., Bu, S., Chen, C.: Unsupervised learning of 3D local features from raw voxels based on a novel permutation voxelization strategy. IEEE Trans. Cybern. **49**(2), 481–494 (2019)
16. Han, Z., Liu, Z., Han, J., Vong, C.M., Bu, S., Chen, C.: Mesh convolutional restricted Boltzmann machines for unsupervised learning of features with structure preservation on 3D meshes. IEEE Trans. Neural Netw. Learn. Syst. **28**(10), 2268–2281 (2017)
17. Han, Z., Liu, Z., Han, J., Vong, C.M., Bu, S., Li, X.: Unsupervised 3D local feature learning by circle convolutional restricted Boltzmann machine. IEEE Trans. Image Process. **25**(11), 5331–5344 (2016)
18. Han, Z., et al.: Deep spatiality: unsupervised learning of spatially-enhanced global and local 3D features by deep neural network with coupled softmax. IEEE Trans. Image Process. **27**(6), 3049–3063 (2018)
19. Han, Z., et al.: BoSCC: bag of spatial context correlations for spatially enhanced 3D shape representation. IEEE Trans. Image Process. **26**(8), 3707–3720 (2017)
20. Han, Z., et al.: 3D2SeqViews: aggregating sequential views for 3D global feature learning by CNN with hierarchical attention aggregation. IEEE Trans. Image Process. **28**(8), 3986–3999 (2019)

21. Han, Z., Shang, M., Liu, Y.S., Zwicker, M.: View inter-prediction GAN: unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. In: AAAI, pp. 8376–8384 (2019)
22. Han, Z., et al.: SeqViews2SeqLabels: learning 3D global features via aggregating sequential views by rnn with attention. IEEE Trans. Image Process. **28**(2), 658–672 (2019)
23. Han, Z., Shang, M., Wang, X., Liu, Y.S., Zwicker, M.: Y2Seq2Seq: cross-modal representation learning for 3D shape and text by joint reconstruction and prediction of view and word sequences. In: AAAI, pp. 126–133 (2019)
24. Han, Z., Wang, X., Liu, Y.S., Zwicker, M.: Multi-angle point cloud-vae: unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. In: ICCV (2019)
25. Han, Z., Wang, X., Vong, C.M., Liu, Y.S., Zwicker, M., Chen, C.P.: 3DViewGraph: learning global features for 3D shapes from a graph of unordered views with attention. In: IJCAI (2019)
26. Hane, C., Tulsiani, S., Malik, J.: Hierarchical surface prediction for 3D object reconstruction. In: International Conference on 3D Vision, pp. 412–420 (2017)
27. Hu, T., Han, Z., Shrivastava, A., Zwicker, M.: Render4Completion: synthesizing multi-view depth maps for 3D shape completion. ArXiv abs/1904.08366 (2019)
28. Hu, T., Han, Z., Zwicker, M.: 3D shape completion with multi-view consistent inference. In: AAAI (2020)
29. Insafutdinov, E., Dosovitskiy, A.: Unsupervised learning of shape and pose with differentiable point clouds. In: Advances in Neural Information Processing Systems, pp. 2807–2817 (2018)
30. Jiang, Y., Ji, D., Han, Z., Zwicker, M.: SDFDiff: differentiable rendering of signed distance fields for 3D shape optimization. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
31. Kato, H., Ushiku, Y., Harada, T.: Neural 3D mesh renderer. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3907–3916 (2018)
32. L., N.K., Mandikal, P., Agarwal, M., Babu, R.V.: Capnet: continuous approximation projection for 3D point cloud reconstruction using 2D supervision. In: AAAI (2019)
33. Lin, C.H., Kong, C., Lucey, S.: Learning efficient point cloud generation for dense 3D object reconstruction. In: AAAI Conference on Artificial Intelligence (2018)
34. Liu, H.T.D., Tao, M., Jacobson, A.: Paparazzi: surface editing by way of multi-view image processing. ACM Trans. Graph. **37**, 221 (2018)
35. Liu, H.T.D., Tao, M., Li, C.L., Nowrouzezahrai, D., Jacobson, A.: Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. In: International Conference on Learning Representations (2019)
36. Liu, S., Zhang, Y., Peng, S., Shi, B., Pollefeys, M., Cui, Z.: DIST: rendering deep implicit signed distance function with differentiable sphere tracing. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
37. Liu, S., Chen, W., Li, T., Li, H.: Soft rasterizer: differentiable rendering for unsupervised single-view mesh reconstruction. CoRR abs/1901.05567 (2019)
38. Liu, S., Li, T., Chen, W., Li, H.: Soft rasterizer: a differentiable renderer for image-based 3D reasoning. In: The IEEE International Conference on Computer Vision (2019)
39. Liu, S., Saito, S., Chen, W., Li, H.: Learning to infer implicit surfaces without 3D supervision. In: Advances in Neural Information Processing Systems (2019)

40. Liu, X., Han, Z., Liu, Y.S., Zwicker, M.: Point2Sequence: learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. In: AAAI, pp. 8778–8785 (2019)
41. Liu, X., Han, Z., Xin, W., Liu, Y.S., Zwicker, M.: L2G auto-encoder: understanding point clouds by local-to-global reconstruction with hierarchical self-attention. In: ACMMM (2019)
42. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: learning 3D reconstruction in function space. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
43. Michalkiewicz, M., Pontes, J.K., Jack, D., Baktashmotlagh, M., Eriksson, A.P.: Deep level sets: implicit surface representations for 3D shape inference. CoRR abs/1901.06802 (2019)
44. Navaneet, K.L., Mandikal, P., Jampani, V., Babu, R.V.: DIFFER: moving beyond 3D reconstruction with differentiable feature rendering. In: CVPR Workshops (2019)
45. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: learning implicit 3D representations without 3D supervision. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
46. Oechsle, M., Mescheder, L., Niemeyer, M., Strauss, T., Geiger, A.: Texture fields: learning texture representations in function space (2019)
47. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: learning continuous signed distance functions for shape representation. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
48. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: deep learning on point sets for 3D classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
49. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems, pp. 5105–5114 (2017)
50. Richter, S.R., Roth, S.: Matryoshka networks: predicting 3D geometry via nested shape layers. In: CVPR, pp. 1936–1944 (2018)
51. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: PIFu: pixel-aligned implicit function for high-resolution clothed human digitization. In: IEEE International Conference on Computer Vision (2019)
52. Shin, D., Ren, Z., Sudderth, E.B., Fowlkes, C.C.: 3D scene reconstruction with multi-layer depth and epipolar transformers. In: IEEE International Conference on Computer Vision (2019)
53. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: continuous 3D-structure-aware neural scene representations. In: Advances in Neural Information Processing Systems (2019)
54. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: efficient convolutional architectures for high-resolution 3D outputs. In: IEEE International Conference on Computer Vision, pp. 2107–2115 (2017)
55. Tatarchenko, M., Richter, S.R., Ranftl, R., Li, Z., Koltun, V., Brox, T.: What do single-view 3D reconstruction networks learn? In: The IEEE Conference on Computer Vision and Pattern Recognition (2019)
56. Tulsiani, S., Efros, A.A., Malik, J.: Multi-view consistency as supervisory signal for learning shape and pose prediction. In: Computer Vision and Pattern Recognition (2018)

57. Tulsiani, S., Zhou, T., Efros, A.A., Malik, J.: Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 209–217 (2017)
58. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.: Pixel2mesh: generating 3D mesh models from single RGB images. In: European Conference on Computer Vision, pp. 55–71 (2018)
59. Wang, W., Ceylan, D., Mech, R., Neumann, U.: 3DN: 3D deformation network. In: CVPR (2019)
60. Wang, W., Xu, Q., Ceylan, D., Mech, R., Neumann, U.: DISN: deep implicit surface network for high-quality single-view 3D reconstruction. In: NeurIPS (2019)
61. Wen, C., Zhang, Y., Li, Z., Fu, Y.: Pixel2Mesh++: multi-view 3D mesh generation via deformation. In: IEEE International Conference on Computer Vision (2019)
62. Wen, X., Han, Z., Youk, G., Liu, Y.S.: CF-SIS: semantic-instance segmentation of 3D point clouds by context fusion with self-attention. In: ACM International Conference on Multimedia (2020)
63. Wen, X., Li, T., Han, Z., Liu, Y.S.: Point cloud completion by skip-attention network with hierarchical folding. In: The IEEE Conference on Computer Vision and Pattern Recognition (2020)
64. Whitted, T.: A scan line algorithm for computer display of curved surfaces. In: The 5th Annual Conference on Computer Graphics and Interactive Techniques SIGGRAPH, p. 26 (1978)
65. Wu, J., Wang, Y., Xue, T., Sun, X., Freeman, B., Tenenbaum, J.: MarrNet: 3D shape reconstruction via 2.5D sketches. In: Advances in Neural Information Processing Systems, pp. 540–550 (2017)
66. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In: Advances in Neural Information Processing Systems, pp. 82–90 (2016)
67. Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H.: Perspective transformer nets: learning single-view 3D object reconstruction without 3D supervision. In: Advances in Neural Information Processing Systems, pp. 1696–1704 (2016)
68. Yifan, W., Serena, F., Wu, S., Öztireli, C., Sorkine-Hornung, O.: Differentiable surface splatting for point-based geometry processing. ACM Trans. Graph. **38**(6), 1–14 (2019)
69. Zakharov, S., Kehl, W., Bhargava, A., Gaidon, A.: Autolabeling 3D objects with differentiable rendering of SDF shape priors. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
70. Zhang, X., Zhang, Z., Zhang, C., Tenenbaum, J., Freeman, B., Wu, J.: Learning to reconstruct shapes from unseen classes. In: Advances in Neural Information Processing Systems, pp. 2257–2268 (2018)