STD-Net: Structure-Preserving and Topology-Adaptive Deformation Network for Single-View 3D Reconstruction

Aihua Mao, Canglan Dai, Qing Liu, Jie Yang, Lin Gao, Ying He, and Yong-Jin Liu, *Senior Member, IEEE*

Abstract—3D reconstruction from single-view images is a long-standing research problem. There have been various methods based on point clouds and volumetric representations. In spite of success in 3D models generation, it is quite challenging for these approaches to deal with models with complex topology and fine geometric details. Thanks to the recent advance of deep shape representations, learning the structure and detail representation using deep neural networks is a promising direction. In this article, we propose a novel approach named STD-Net to reconstruct 3D models utilizing mesh representation that is well suited for characterizing complex structures and geometry details. Our method consists of (1) an auto-encoder network for recovering the structure of an object with bounding box representation from a single-view image; (2) a topology-adaptive GCN for updating vertex position for meshes of complex topology; and (3) a unified mesh deformation block that deforms the structural boxes into structure-aware meshes. Evaluation on ShapeNet and PartNet shows that STD-Net has better performance than state-of-the-art methods in reconstructing complex structures and fine geometric details.

Index Terms—Single-view reconstruction, deformation driven method, structure preservation, topology adaptivity

1 Introduction

3D reconstruction plays an essential role in various tasks in computer graphics and vision. Traditional approaches mainly utilize stereo correspondence based on multi-view geometry, hence are restricted to the coverage provided by the input views. Such limitation makes single-view reconstruction tricky due to large occlusion and lack of correspondence with other views. With the availability of a large-scale 3D shape database [1], deep neural networks can effectively encode shape information, enabling faithful 3D reconstruction from single-view images. Although point clouds and voxel-based representations have been utilized for 3D reconstruction, they are not able to express geometry details and may generate missing parts or broken structures. Furthermore, the voxel representation would cause high computational cost and memory storage.

Triangle meshes are widely used in computer graphics community thanks to its flexibility and effectiveness in modelling geometric details. Recently, mesh-based deep learning has received much attention [2], [3]. The triangle mesh can be represented by the graph-based neural network [4], [5]. Although these methods can reconstruct the surface of an object, the reconstruction results are limited to some categories of 3D models and miss structural information of the object. In literature, 3D structure recovery is mostly studied by traditional approaches, due to a lack of a structural representation of 3D shape that is suitable for deep neural networks. Thus, building up a deep neural network that could directly recover the 3D shape structure of an object from a single RGB image is important. Recent studies show that cuboid primitives are widely used for structure representation [6], [7], [8]. The cuboid structure representation can recover complete models with part relationship, whereas the results estimated by 3D-GAN [9] may lose some parts and lack the surface details of an object. To delicately express a shape's structural information by advancing the cuboid representation, we propose a deep neural network to reconstruct 3D objects by the mesh representation. Our method is able to express complex structure and fine-grained surface details of 3D objects.

- *Aihua Mao and Canglan Dai are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, Guangdong Province 510641, China. E-mail: {ahmao, 201821033708}@scut.edu.cn.*
- *Qing Liu is with the School of Software Engineering, South China University of Technology, Guangzhou, Guangdong Province 510641, China. E-mail: 202021045838@mail.scut.edu.cn.*
- *Jie Yang and Lin Gao are with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100049, China, and also with University of Chinese Academy of Sciences, Beijing 100049, China. E-mail: {yangjie01, gaolin}@ict.ac.cn.*
- *Ying He is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore. E-mail: yhe@ntu.edu.sg.*
- *Yong-Jin Liu is with BNRist, MOE-Key Laboratory of Pervasive Computing, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: liuyongjin@tsinghua.edu.cn.*

Manuscript received 19 Jan. 2021; revised 24 Nov. 2021; accepted 27 Nov. 2021. Date of publication 1 Dec. 2021; date of current version 31 Jan. 2023.
This work was supported in part by Tsinghua University Initiative Scientific Research Program, the Natural Science Foundation of Guangdong Province under Grant 2019A1515010833 and the Fundamental Research Funds for the Central Universities under Grant 2020ZYGXZR089, Singapore Ministry of Education under Grants RG20/20 and T2EP20220-0014, and the Natural Science Foundation of China under Grant 61725204.
(Corresponding author: Aihua Mao.)
Recommended for acceptance by E. Kalogerakis.
Digital Object Identifier no. 10.1109/TVCG.2021.3131712

1077-2626 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.

Mesh-based deep learning approaches mostly rely on Graph Convolution Network (GCN) [2], [3], [10]. The common practice of these methods is to use GCN to deform a pre-defined mesh that is generally a sphere, or deform a set of primitives (square planar) to form 3D shapes. GCN is effective in many classification and regression tasks. However, it is inadequate for reconstruction, generation and 3D structure analysis, because of over-smoothness when aggregating neighbour information at the vertex level. Furthermore, the existing GCN-based methods can only deal with fixed-topology meshes, whereas the cuboid structure is naturally suitable for representing variable topological mesh. Hence the existing GCNs are not suitable for the cuboid representation. In this paper, we aim to address these challenges and reconstruct 3D meshed models with adaptive topology and fine-grained geometric details from a single RGB image. The key idea for single view reconstruction is to obtain the structural representation of 3D objects with cuboids and then deform them into concrete meshes through an integrated deep neural network. The network, called STD-Net, allows a 3D structure-preserving object model to be reconstructed from a single RGB image. This paper makes three contributions:

- Our method conducts 3D reconstruction by a mesh-based learning framework that takes a single image as input and constructs 3D meshed models with complex structure and fine-grained geometric details.
- We represent 3D objects' structure by recovering cuboids bounding boxes from a single image that can delicately express rich structural information.
- To the best of our knowledge, we are the pioneers to investigate learning approaches for 3D shape reconstruction with meshes in non-fixed topology.

2 Related Work

2.1 3D Object Reconstruction

The voxel representation has been widely used for 3D shape generation and understanding with the neural network [11], [12], [13], [14], [15]. Although it is simple to be implemented and has good compatibility, voxel representation is limited by the resolution and computation cost of 3D convolution. Recently, octree is exploited to develop computationally efficient approaches [11], [16], [17], which solve the resolution issues to a certain extent but still rely on the subdivision of a bounding volume rather than a local geometric shape.

Another popular representation for 3D shapes is point cloud [18], [19], [20], which describes 3D shapes through a set of points in 3D space. Naturally, the point cloud can represent the surface information of 3D objects with no local connections between points. Hence the point cloud is flexible with sufficient degrees of freedom to match the 3D shape with arbitrary topology. However, the irregular structure of point clouds poses a great challenge for deep learning; it is only able to produce relatively coarse geometry and can not be used to directly recover a detailed 3D shape.

Recently, mesh models have been introduced in deep learning for 3D generation and reconstruction tasks. Image2mesh [21] proposes a learning framework to infer 3D mesh models from a single image using a compact mesh representation. Pixel2Mesh [2] generates 3D mesh models from an RGB image by deforming an initial ellipsoid mesh in a coarse to fine manner, but fails to consider the structural information in 3D representation, hence it can not reconstruct complex-structure objects. AtlasNet [3] deforms a set of primitives to generate parametric surface elements for 3D shapes. Similarly, Pix2Surf [22] represents the shape with a set of continuous parametric 3D surfaces while retains accurate 2D to 3D surface point correspondences via the unsupervised extraction of a learned *UV* parametrization. GEOMetrics [10] presents an approach for adaptive mesh reconstruction, which focuses on exploiting the geometric structure of graph encoded objects. Although it takes additional voxels into consideration to provide the global structure, it can not express the local structure. Some methods attempt to reconstruct category-specific mesh that is parameterized by a learned mean shape [23], [24]. Point2Mesh [25] reconstructs a watertight mesh by optimizing the weights of a CNN to iteratively deform an initial mesh to shrink-wrap the given input point cloud. Gao *et al.* [26] utilized a neural network to deform vertices of an initial tetrahedron mesh and to predict the occupancy for each tetrahedron, which is the first to produce tetrahedral meshes directly from single images. However, most of the mesh-based methods deform a generic pre-defined mesh to form 3D surfaces, which limits the types of objects they can handle. Furthermore, they require a fixed-topology mesh model, whereas our method allows input models with different topological types. As a result, our method is able to generate more delicate structure for 3D shape.

Recent works based on implicit fields use either binary occupancy [27], [28], [29], [30] or signed distance functions (SDF) [31], [32], [33] as shape representation for learning generative models. These methods predict binary occupancy or SDF values at continuous sampled point locations. Yariv *et al.* [34] represented the geometry with a zero level-set of a neural network. Atzmon *et al.* [35] constructed implicit neural representations from raw data by unsigned distance regression to introduce better local minima. They are not limited by resolution and are flexible to represent different topology. However, they require artificially closing shapes during post-processing, which often leads to loss of detailed geometry, artifacts, or inner structures.

2.2 3D Structure Learning

Many works based on learning methods generate 3D models using voxels, point cloud and meshes. However, their outputs are non-structured models, because they lack effective structural representation for 3D shapes. Some works attempt to address this issue through non deep-learning approaches. For example, Xu *et al.* [36] modeled 3D objects from photos by utilizing an available set of 3D candidate models. Huang *et al.* [37] jointly analyzed a collection of images and reconstructed the 3D shapes from existing 3D models collections.

Researchers have explored the possibility of expressing 3D structure through learning methods. Tulsiani *et al.* [12] proposed a deep architecture to map a 3D volume to a 3D cuboid primitive, which has potentials to automatically discover and exploit consistent structure in data. Niu *et al.* [38] developed a convolutional-recursive auto-encoder that

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.

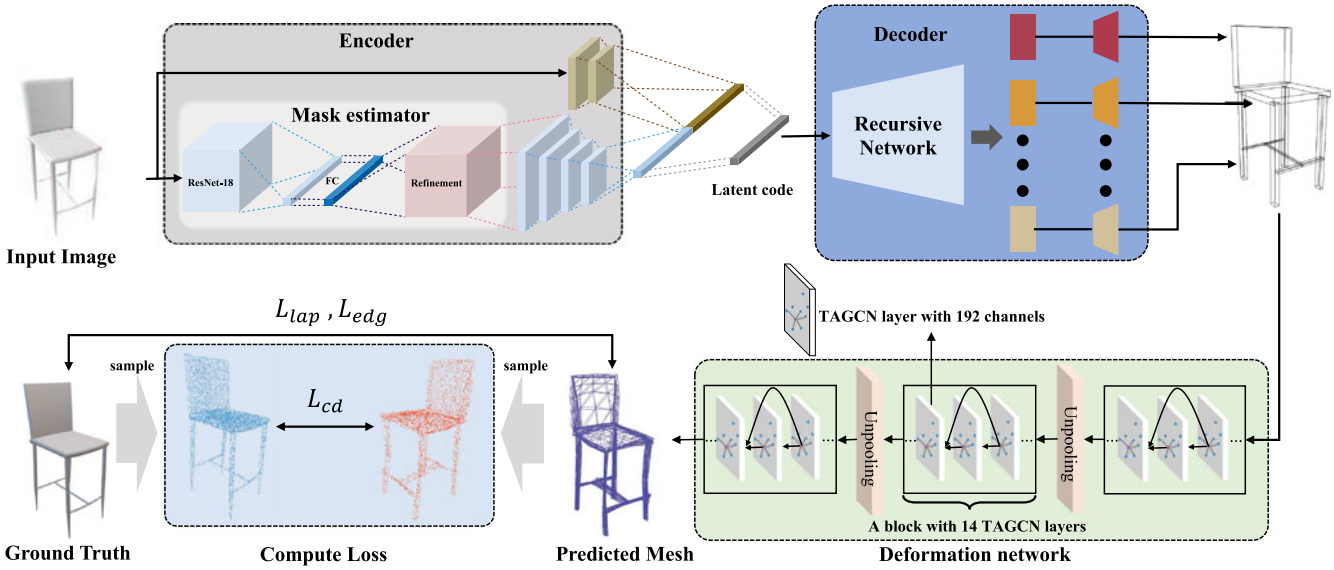


Fig. 1. Overview. Given a single-view input image, STD-Net first employs an auto-encoder to obtain bounding boxes covering the parts of the object. Then, it adopt multiple mesh deformation and unpooling operations to progressively deform the mesh vertices and update the topology to approximate the target 3D object surface.

entails structure parsing of a 2D image and structure recovering of a cuboid hierarchy. Gao *et al.* [39] reported SDM-Net to jointly encode shape structure and geometry for 3D mesh models by using a two-level variational auto-encoder, which encodes the part geometric deformation and part relationships together. Meanwhile, Mo *et al.* [6] proposed StructureNet, a hierarchical graph network, to produce a unified latent space to encode models with complex structure. Our method adopts the idea from StructureNet and expresses complex structure information by a hierarchy of cuboid bounding boxes, and thus can reconstruct the 3D shape with complex structure and arbitrary topology.

2.3 Graph Convolution Neural Network

Graph convolution neural network [40] is a popular choice in 3D shape analysis [41] and 3D reconstruction [2]. Unlike traditional CNN which is defined by 2D images and 3D voxels with regular grids, GCN regards the mesh as a graph and learns features by using graph convolution. The potential of graph convolution is to capture structural relations among node of data. However, the irregular structural attributes of graphs pose huge challenges to the convolutions on graphs. The issues in deep learning on graphs mainly come from the complex and diverse connectivity patterns in graph-structured data.

Defferrard *et al.* [42] proposed an approach to approximate the filters by applying Chebyshev polynomials applied on the Laplacian operator. This approximation was further simplified by [43], which proposed a filter in the spatial domain and achieved good performance on classification task. The assumption of symmetric adjacency matrix in the two spectral based methods above restricts the application of undirected graph-structured data. The issue of extending CNNs from grid-structured data to arbitrary graph-structured data remains unsolved. To adapt to the various topologies of graph data, Du *et al.* [44] implemented the graph convolution using a set of fixed-size learnable filters. We adopts the idea of topology adaptive graph convolutional network (TAGCN) in [44] to construct GCN to achieve deformation for meshed cuboid bounding boxes with different structural topology types.

3 Overview

Fig. 1 shows the network architecture of our method (STD-Net). The method is composed of two parts: *structure recovery network* and *mesh deformation network*. The structure recovery network is designed with an auto-encoder to predict the 3D structure of an object from a single RGB image. The recursive network architecture of decoder is designed in accordance with that of StructureNet [6]. It can generate an object's hierarchy cuboid bounding boxes, which delicately describe structural information in detail. These bounding boxes are further meshed and placed into the GCNs in the next phase.

The mesh deformation network aims to deform the input bounding box into a structure-preserving shape. It consists of three blocks intersected by two graph unpooling layers. Each block has the same network structure of 14 TAGCN layers with 192 channels that accept variable topologies of the graph. The TAGCN is constructed by the guidance of the work in [44]. The unpooling layer works to add the number of vertices to handle fine-grained geometric details. The following sections discuss in detail the two network parts.

Our method makes advances in 3D object reconstruction by building up a mesh-based deep neural network, which can directly recover the 3D shape structure of an object from a single RGB image. Thus, the simultaneous expression of complex structure and fine-grained surface details of 3D objects becomes feasible. Our method advances the cuboid representation to delicately express a shape's structural information, and address the challenges of existing GCN-based methods to deal with adaptive topology. To the best of our knowledge, we are the pioneers to investigate learning approaches at the mesh level to reconstruct 3D shapes with arbitrary topology from a single-view image. Most of current mesh-based methods deform a generic pre-

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.

defined mesh to form 3D surfaces, which limits the types of objects they can handle.

4 Structure Recovery Network

Recently, shape abstraction [6], [38] has been used to discover the high-level structure in un-annotated point clouds and images. These works inspire us to use the decoder in shape abstraction to recover the structure of an object. In our method, an encoder is first employed to map a shape (represented as a hierarchy of $n-$array graphs or cuboids) to a latent feature vector z. Then, a decoder transforms the feature vector z back into a shape (also represented as a hierarchy of graphs or cuboids). The structure of an object is represented by a hierarchical graph, and every node is represented by a bounding box. For the encoder part, the structural information of an image goes through a CNN network and is transferred into a latent code as features. For the decoder part, a recursively network unfolds features into a hierarchical organization of the bounding boxes, which are the recovered structure of the object.

4.1 Encoder

The encoder takes a 2D RGB image as input and obtains a latent code containing object's structure. Inspired by the multi-scale network for detailed depth estimation [45], we design a two-streamed network shown in Fig. 1. Different from [45] whose two streams are both designed into a two-scale network, the top stream in our encoder simply consists of ResNet18 [46], which receives the original image as input and is followed by two convolutional layers. The down stream in our encoder includes a two-scale network similar to [45] and four convolutional layers. The first scale (ResNet18 instead of VGG-16 in [45]) captures the information of the whole image; and the second one produces a detailed mask map with a quarter of the input image, which estimates the object's contour providing strong cues for understanding shape structure of an object. The output of features from the two streams are then cancatenated and further encoded into $n-$D vector ($n = 256$) by two fully connected layers, thus capturing the object structure information from the input image.

4.2 Decoder

We adopt the recursive network architecture in StructureNet [6] as a hierarchy of graphs in the decoder that perform graph convolution and message-passing operations at each recursive level. In the whole structure recovery network, the latent code can be regarded as the root features of the structure tree. The decoder gradually decodes the node feature code into a hierarchy of features until it reaches the leaf nodes where each of them can be parsed into a vector of box parameters. The details of the decoder's architecture can be referred to [6]. We first use the graph decoder to transform a latent code into its child graph, and then use the box decoder to transform the resulting feature code of each child back into the bounding box parameter, which is a 10-d vector representing the center, scale and rotation. Following the graph decoder in [6], the box decoder is implemented as multi-layer perceptrons.

5 Mesh Deformation Network

Given a hierarchy of bounding boxes $\{B_i\}$ generated in Section 4 and ground truth $\{S_i\}$, our goal is to deform the bounding box to make it as close as possible to the ground truth shape. As depicted in Fig. 1, the mesh deformation module takes a meshed bounding box defined by a set of vertex positions as input and outputs predicted deformed meshes. Thus, the bounding boxes expressing different topology can be deformed to reconstruct the shape of variable objects through this deformation network module.

5.1 Network Structure

In the mesh deformation network, three deformation blocks are intersected by two graph unpooling layers to progressively achieve the mesh deformation. Each deformation block regards the graph as input. The graph represents the meshed bounding box and the shape of 3D object will be recovered by the vertices of the bounding box. The unpooling layer aims to increase the number of vertices, which can increase the capacity of handling fine-grained geometric details. We will investigate the influence of greater number of deformation block on the predicted results in Section 6.4.

The three deformation blocks with the same architecture contain 14 TAGCN layers with 192 channels. The first block takes the initial meshed bounding box as input, and the remaining two blocks take the output from the previous unpooling layer as input. For each TAGCN layer, the GCN is constructed in accordance with [44]. We only concern a small local region around the vertex. By defining a *path* of length m on a graph as sequence $v = (v_0, v_1, \ldots, v_m)$, $v_k \in V$, the small region can be determined by the node *path*. The convolution formula used in each convolution layer is

$$X_{l+1} = f(A^K X_l W_K^l + \ldots + A^1 X_l W_1^l + X_l W_0^l) \tag{1}$$

where A is the adjacency matrix, X_l is the input vector in the lth convolution layer, f is the nonlinear activation function, and W_k are the learnable weights. Experiments show that $k = 2$ can achieve good performance. The convolution defined in Eq. (1) is similar to the traditional convolution operations. In the convolution layer of traditional CNN, the receptive field is a small rectangle in the grid. In GCN, this field is also a small region around the vertex. The operation in Eq. (1) calculates a linear combination of the signals of nodes k which hop away from the start node. Moreover, we propose a deep network with several shortcut connections, which can alleviate the over-smoothing in GCN. In addition to the output of the network, a branch applies an extra TAGCN layer to the last layer and outputs the 3D coordinates. Similar to Pixel2Mesh [2], we add a new vertex at the center of each edge to form each unpooling layer, which can increase the number of vertices in GCN.

5.2 Training and Losses

The training procedure is divided into two stages. First, the mask network is trained to estimate the object mask of the input image. In the next stage, we jointly refine the mask network and train the structure recovery network to estimate the part-level boxes of object. The structure recovery network loss is computed as the sum of the box reconstruction error,

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.

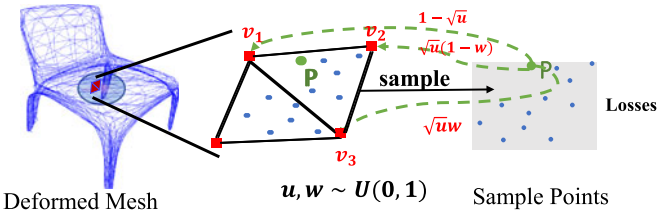


Fig. 2. Sampling strategy from predicted mesh.

which is calculated by squared differences between the input and reconstructed boxes.

$$\mathcal{L}_{recovery} = CHS(T(B_i)U, T(B_j)U) \tag{2}$$

where CHS is the squared version of Chamfer distance, T is a 4D transformation matrix that transforms the unit cube into the bounding box of a part, B_i and B_j are the predicted bounding box and ground truth bounding box of a part respectively, and U is a set of sampled vertices on the predefined unit cube.

After generating the object's part-level boxes, we use the deformation network to deform the box of a part into a 3D shape. The mesh deformation network is supervised by hybrid losses including the CD loss, normal loss, Laplacian loss and edge length loss. The CD loss measures the nearest neighbor distances between two point sets. We minimize the two directional distances between the deformed bounding boxes and the ground truth shape. The CD loss is defined as

$$\mathcal{L}_{cd}(P,G) = \sum_{x\in P} \min_{y\in G} \|x-y\|_2^2 + \sum_{y\in G} \min_{x\in P} \|x-y\|_2^2 \tag{3}$$

where P and G are the two point sets. Akin to 3dn [47] and GEOMetrics [10], which do not simply compute the CD loss between the predicted points and ground truth points, we uniformly sample the same number of vertices from the predicted mesh and ground truth mesh and then compute the CD loss between them. More precisely, for each triangular face, we first compute its area and store it in an array along with the cumulative area of triangles visited so far. Next, we select a triangle with a probability ratio between its area and the total cumulative area. It can be seen in Fig. 2. For each selected triangular face defined by vertices v_1, v_2 and v_3, a new point r can be sampled uniformly from the surface by the following formulation

$$r = (1-\sqrt{u})v_1 + \sqrt{u}(1-w)v_2 + \sqrt{u}wv_3 \tag{4}$$

where $u, w \sim U(0,1)$. Hence, the final CD loss (L_{pts}) can be written as

$$\mathcal{L}_{pts} = \mathcal{L}_{cd}(Sample(P), G) \tag{5}$$

Given that the CD loss does not concern the connectivity of mesh vertices, the predicted mesh could suffer from a few floating vertices and self-intersections. Thus, we add some geometric-aware regularization terms, including a normal loss, a Laplacian loss and an edge length loss. These terms prevent the vertices from moving in excessively long distance and can potentially avoid mesh self-intersection. The normal loss $\mathcal{L}_{normal}$ measures the normal consistency between the generated meshes and ground truth meshes. The laplacian loss $\mathcal{L}_{laplace}$ flattens the intersection angles of adjacent faces to make surface smooth. Lastly, the edge loss $\mathcal{L}_{edge}$ penalizes the flying vertices and overlong edges to guarantee the high-quality recovered 3D geometry. These terms of loss can be calculated by following the work of [2]: $\mathcal{L}_{normal} = \sum_p \sum_{q=argmin_q(||p-k||_2^2)} ||\langle p-k, n_q\rangle||_2^2$, where q is the closest vertex for p and k is the neighbor of vertex p; $\mathcal{L}_{laplace} = \sum_p ||\delta_p' - \delta_p||_2^2$, where δ_p is the laplace of the vertex p; $\mathcal{L}_{edge} = \sum_p \sum_{q\in N(p)} ||p-q||_2^2$, where $N(p)$ is the neighbor of the vertex p.

The total loss of deformation module thus is

$$\mathcal{L}_{deform} = \mathcal{L}_{recovery} + \mathcal{L}_{pts} + \lambda_1\mathcal{L}_{normal} + \lambda_2\mathcal{L}_{laplace} + \lambda_3\mathcal{L}_{edge} \tag{6}$$

where $\lambda_1, \lambda_2, \lambda_3$ are the hyper-parameters that weight the importance of each loss.

6 Experiments

In this section, we demonstrate the performance of STD-Net on structure-preserving reconstruction from single RGB images by taking the benefits of the mesh-based structure representation. We train the proposed model using the PyTorch and evaluated the results by a PC with NVIDIA GeForce GTX 1080 Ti and 32G RAM. We compare the results of our method and those of other relevant works, and also present an ablation study to demonstrate how individual component of our model contributes to its overall performance.

6.1 Dataset

Since StructureNet [6] is trained on PartNet [48], we use the intersected datasets of ShapeNet [1] and PartNet [48] as our main datasets for training and testing STD-Net. In the prepared datasets, each object is decomposed into a hierarchical graph of parts and every part is labeled. Each shape is rendered with 24 different views, and the rendered image resolution is 224×224. The rendered images are then set up as the inventory for input images. Each category is split into a training, a validation, and a test set with a ratio of 7:1:2 respectively, which is also similar to the works [2] [10].

6.2 Implementation Details

6.2.1 Structure Recovery network

For the encoder, the mask estimator consists of a two-scale network and four convolutional layers. As for the decoder, we use the pre-trained shape abstraction decoder from StructureNet [6]. Training the structure recovery network is divided into two stages. We first train the network for estimating an object mask for the input image. The first and second scale network are trained jointly. Second, we train the encode and decoder together, during which a gradually decreased learning rate for the encoder is used.

6.2.2 Mesh Deformation Network

Prior to training this network, we prepare the pairs of bounding box and mesh in PartNet. For each category, a bounding box(OBB)-to-mesh mapping is trained from the ground truth. These mappings are trained with Adam Optimizer ($\beta = 5e^{-4}$) and a learning rate of $3e^{-5}$. We conduct the

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
This Table Lists Out a Quantitative Performance Comparison of Our Approach and Other Approaches Including Pixel2Mesh[2], AtlasNet [3], GEOMetrics [10] and IM-Net [49]

		Chair	Table	SF	Display	Lamp
CD loss ↓	Pixel2Mesh	1.79	1.99	2.37	2.76	1.70
	AtlasNet	1.71	1.85	**2.09**	1.98	1.31
	GeoMetrics	1.69	1.85	2.41	**1.72**	1.85
	IM-Net	1.75	1.82	3.31	2.03	2.35
	Ours	**1.58**	**1.64**	2.34	1.75	**1.29**
EMD loss ↓	Pixel2Mesh	3.52	3.73	2.56	2.92	5.15
	AtlasNet	3.86	3.98	3.90	3.12	**3.23**
	GeoMetrics	3.63	2.81	3.51	3.41	4.12
	IM-Net	**3.05**	**2.61**	2.91	2.87	4.21
	Ours	3.37	2.71	**2.22**	**2.78**	3.99
IoU ↑	Pixel2Mesh	40.27	43.89	53.38	44.32	22.32
	AtlasNet	47.93	43.31	42.34	48.21	32.34
	GeoMetrics	48.91	49.15	57.14	45.23	24.17
	IM-Net	48.04	37.21	39.20	36.40	34.26
	Ours	**50.12**	**52.23**	**61.23**	**49.44**	**37.19**
F-score ↑	Pixel2Mesh	54.38	66.37	55.12	51.39	48.15
	AtlasNet	59.81	67.21	**71.15**	59.52	53.12
	GeoMetrics	56.61	66.33	59.52	61.09	58.65
	IM-Net	54.98	69.82	67.24	48.82	55.85
	Ours	**66.09**	**72.85**	70.76	**63.32**	**62.21**

training for 3000 iterations and empirically stop it, during which the best model is selected by evaluating the validation set every 10 iterations. The hyper-parameters setting is used, as described in Eq. (6), are $\lambda_1 = \lambda_2 = 0.3$, $\lambda_3 = 0.1$.

6.3 Performance Evaluation

6.3.1 Quantitative Evaluation

Though the method based on implicit fields reported in recent works [29], [31], [33], [49] can generate neat 3D models with variable topology, it uses either binary occupancy or signed distance functions for shape representation, and is thoroughly different from our work which is a method of mesh representation for learning generative models. In order to achieve an objective comparison, we evaluate the performance of our work by quantitatively comparing it with mesh-based approaches, including Pixel2Mesh [2], AtlasNet [3], GEOMetrics [10], and even with an implicit fields-based approach: IM-Net [49].

The comparison is conducted on five categories: Chair, Table, Storage Furniture, Display and Lamp. We show the comparisons on four metrics (CD, EMD, F1-score and 3D IoU) with alternative approaches. These metrics are computed between the ground truth point cloud and 10,000 points uniformly sampled from the generated meshes. Concerning that the output of Pixel2Mesh [2] is non-canonical, we align their predicted shapes to ground truth shapes by using the metadata available in the dataset. In accordance with IM-Net [49], we voxelize meshes in different resolution (16^3, 32^3, 64^3) and sample points in the volume and points near the surfaces to train IM-Net progressively and then extract the predicted mesh via Marching Cubes [51]. Table 1 lists out the performance comparison between our approach and other methods on these four metrics. The data shows that our approach outperforms the state-of-the-art methods on most of the five categories, in which most models have complex structure and topology, such as chairs, tables, storage furniture and lamps. Our approach obtains a maximum 3D IoU over all categories and a maximum F-Score over four categories.

TABLE 2
This Table Lists Out the Performance Data of Other Categories in PartNet [48]

		Bowl	Faucet	Kinfe	Scissors	Hat
CD ↓	IM-Net	**1.01**	**1.04**	**1.64**	**2.42**	1.76
	StructureNet1	1.45	3.80	5.04	3.74	5.42
	StructureNet2	7.81	1.74	3.22	4.36	2.28
	Ours	1.69	3.96	5.01	2.55	**1.42**
EMD ↓	IM-Net	2.80	**1.77**	2.40	3.27	5.78
	StructureNet1	4.78	11.37	10.35	11.39	7.15
	StructureNet2	5.95	7.95	7.38	4.42	5.81
	Ours	**0.15**	3.18	**3.18**	**1.87**	**0.94**
IoU ↑	IM-Net	38.61	8.38	30.23	20.30	16.45
	StructureNet1	26.33	9.61	4.16	2.88	15.29
	StructureNet2	12.75	12.39	27.18	13.73	2.19
	Ours	**77.64**	**40.73**	**41.43**	**48.81**	**73.57**
F-score ↑	IM-Net	41.14	29.35	38.97	35.62	15.50
	StructureNet1	18.84	14.39	26.62	19.59	36.12
	StructureNet2	36.72	46.05	29.97	34.16	44.44
	Ours	**84.15**	**51.55**	**47.33**	**59.52**	**82.08**

StructureNet1 means the mesh models by using Poisson reconstruction, StructureNet2 means the mesh models by using the algorithm [50].

In addition to the intersected datasets of ShapeNet and PartNet, we also extend the training work of our learning network to other categories in the rest of PartNet. It is found that the training could be well converged, and the performance data is collected and listed out in Table 2, which shows our method also has superiority in the metrics of EMD, F1-score and 3D IoU than IM-Net [49] and StructureNet [6] on these categories.

6.3.2 Qualitative Evaluation

Except for the quantitative evaluation in above, we also rendered the generated 3D models by our approach and other approaches to achieve a qualitative evaluation. Fig. 3 presents the comparison of rendered models of shape reconstruction by alternative approaches on the five categories. These results demonstrate that our approach can generate more accurate reconstruction of objects with variable topology from input RGB images and capture complex structure and fine-grained surface details effectively. By contrast, although Pixel2Mesh [2] can reconstruct the shapes, it fails to reconstruct some complex structures due to employing a deformation strategy on a zero-genus ellipsoid template mesh, which can be seen from the results of the chairs with structured backrest, storage furniture and tables in Fig. 3. AtlasNet [3] can generate meshes with more outlined shape, however, it also can not generate the fine structure of objects, as shown in the results of chairs, storage furniture and tables. Moreover, both Pixel2Mesh [2] and AtlasNet [3] require fixed topology in their approaches. GeoMetrics [10] can reflect the structure of objects in some categories, such as the tables, but the structure parts of tables are very

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.

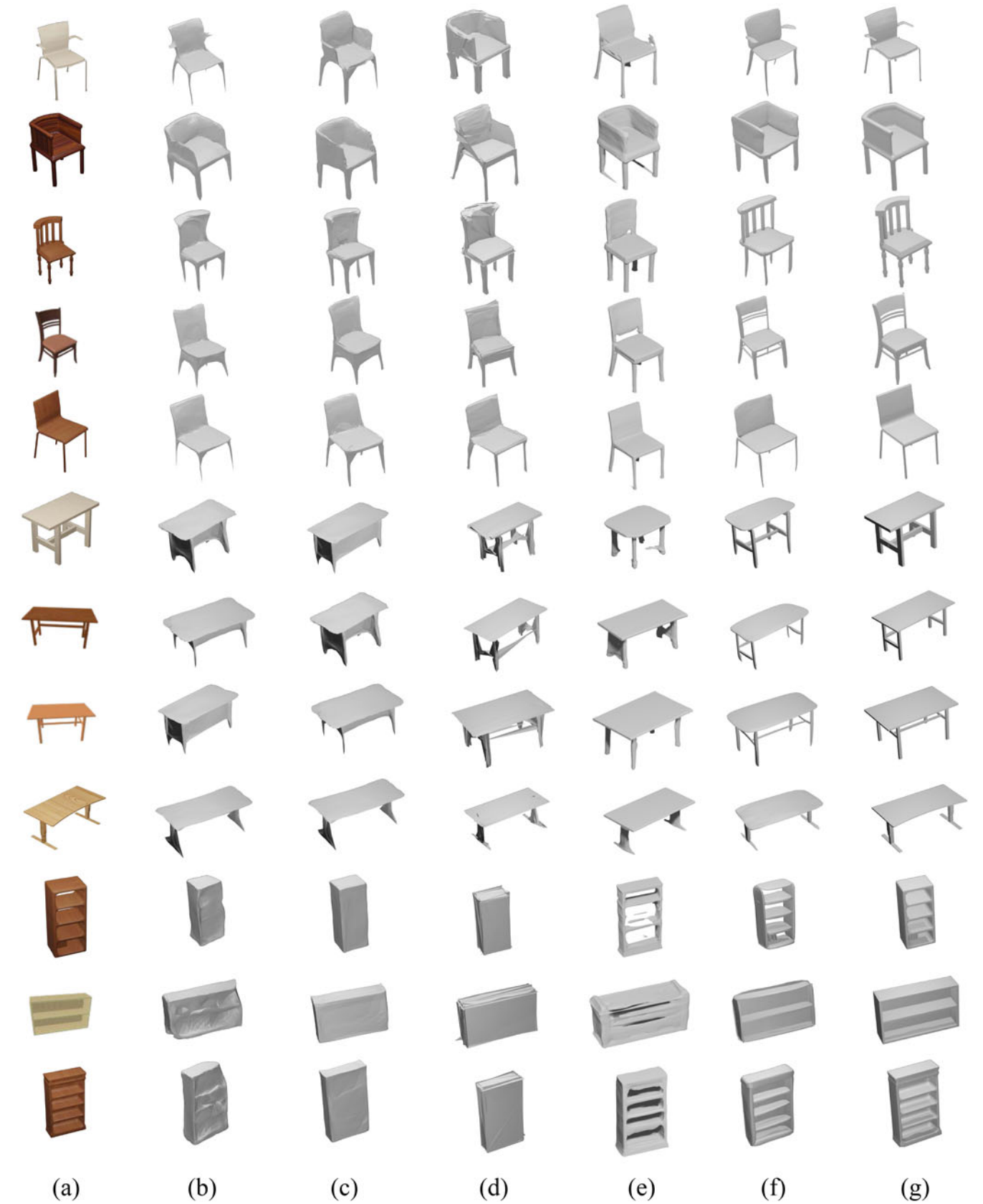


Fig. 3. A. The reconstruction results on the intersected datasets of ShapeNet [1] and PartNet [48]-part I. (a) Input images; (b) Pixel2Mesh [2]; (c) AtlasNet-25 [3]; (d) GEOMetrics[10]; (e) IM-Net [49]; (f) Ours; (g) Ground truth. (B)The reconstruction results on the intersected datasets of ShapeNet[1] and PartNet [48]-part II. (a) Input images; (b) Pixel2Mesh [2]; (c) AtlasNet-25 [3]; (d) GEOMetrics [10]; (e) IM-Net [49]; (f) ours; (g) Ground truth.

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.

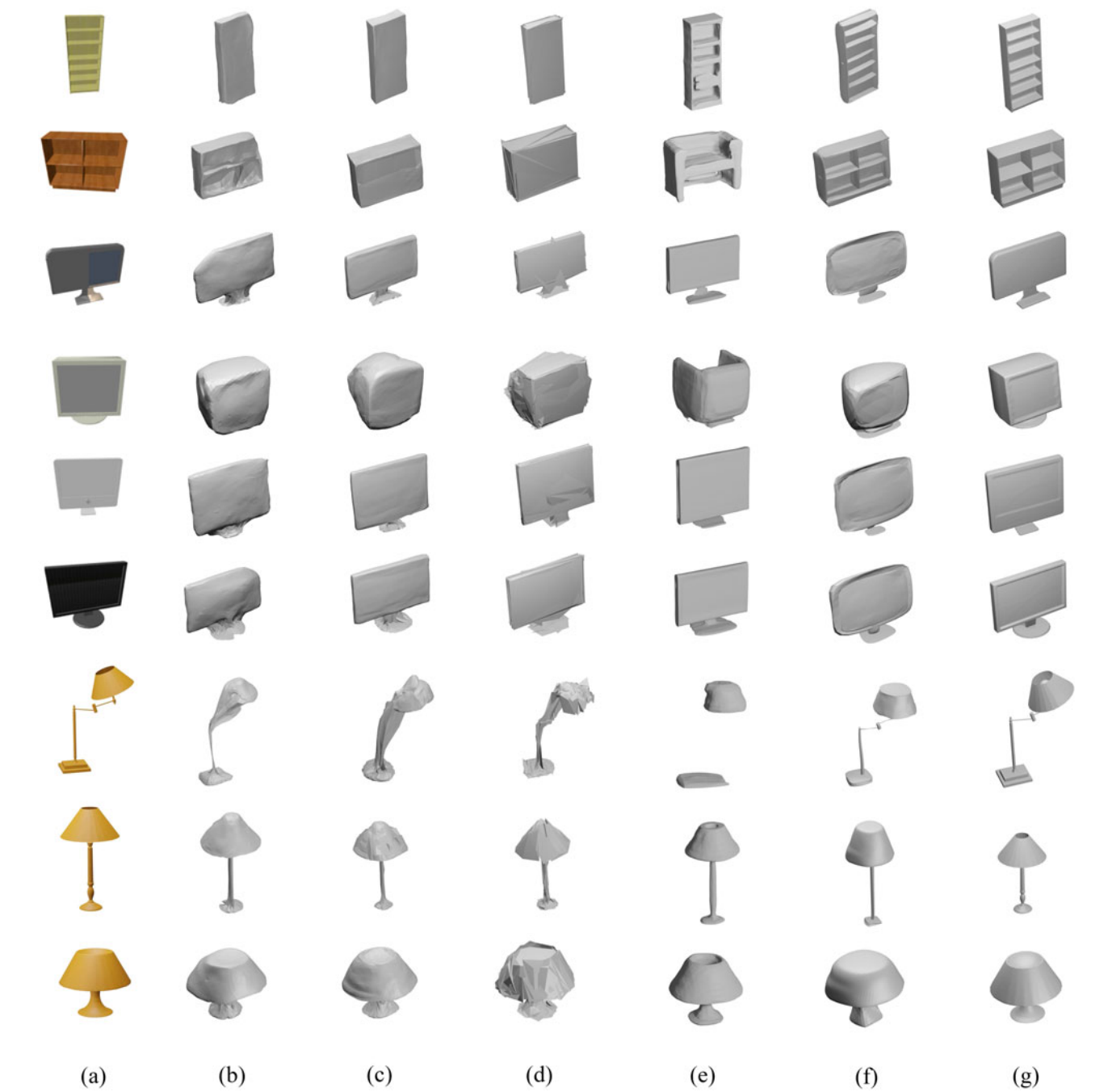


Fig. 3. B. The reconstruction results on the intersected datasets of ShapeNet [1] and PartNet [48]-part I. (a) Input images; (b) Pixel2Mesh [2]; (c) AtlasNet-25 [3]; (d) GEOMetrics [10]; (e) IM-Net [49]; (f) Ours; (g) Ground truth. (B)The reconstruction results on the intersected datasets of ShapeNet [1] and PartNet [48]-part II. (a) Input images; (b) Pixel2Mesh [2]; (c) AtlasNet-25 [3]; (d) GEOMetrics [10]; (e) IM-Net [49]; (f) ours; (g) Ground truth.

coarse. IM-Net [49] can generate more neat results on the modes with simple structure, but fails to recover the complex structure of models accurately, as shown in the results of storage furniture and tables.

More results on other categories in the rest of PartNet are also rendered and depicted in Fig. 9, which shows that our approach can reconstruct 3D models of objects in variable categories with both concrete shapes and delicate structures. We also render the intermediate results of our approach including the predicted masks and boxes from the input images, which lay consistent basis to achieve high quality generated 3D shapes. Fig. 4 shows the results of mask prediction from our mask estimator. By referring to [52], we use depth mask to help to represent the object's 2.5D sketch. Our mask estimator is capable of capturing detailed structure of objects, which provides the structure information for the following network.

Fig. 4. Results of object mask prediction of our approach.

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.

Fig. 5. Comparison between our approach with StructureNet [6]. (a) Input image; (b) Generated bounding boxes by our approach; (c) Generated bounding boxes by StructureNet; (d) Predicted point cloud by StructureNet; (e) Mesh models of StructureNet by using Poisson reconstruction; (f) Mesh models of StructureNet by using the algorithm [50]; (g) Reconstructed mesh models of our approach; (h) Ground truth.The qualitative results compared with SDM-Net [39]. (a) Input image; (b) reconstructed by SDM-Net; (c) reconstructed by our method; (d) ground truth.

In particular, we compare our results with the results of StructureNet [6] in terms of bounding boxes and meshed model. For training StructureNet, an auto-encoder for reconstruction of bounding box or point cloud is trained, and a pre-trained ResNet18 [46] encoder in ImageNet [53] is refined with our training dataset. The training process of StructureNet is conducted respectively for the five categories. As shown in Fig. 5, our method (our encoder + StructureNet's decoder) can generate a better bounding box than that of StructureNet (both encoder + decoder from StrucutureNet). Furthermore, since StructureNet is not tailored to mesh representation, we reconstruct mesh models from the point clouds generated by StructureNet respectively using Poisson surface reconstruction [54] and the algorithm in [50], which are classic alternations for mesh generation. Fig. 5 shows that the mesh models by our method has higher quality than those by StructureNet both in shape and structure recovery. We also make quantitative comparison with StructureNet on the mesh models, as listed in Table. 3. The results indicate that our method outperforms

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.

TABLE 3
The Quantitative Results Compared With StructureNet [6]

		Chair	Table	Hat	Knife	Lamp
CD loss ↓	StructureNet1	4.66	6.51	5.42	5.04	3.69
	StructureNet2	**1.16**	2.90	2.28	**3.22**	3.21
	Ours	1.58	**1.64**	**1.42**	5.01	**1.29**
EMD loss ↓	StructureNet1	7.59	8.35	7.15	10.35	13.02
	StructureNet2	5.47	7.14	5.81	7.38	11.27
	Ours	**3.37**	**2.71**	**0.94**	**3.18**	**3.99**
IoU ↑	StructureNet1	11.84	5.63	15.29	4.16	10.75
	StructureNet2	18.62	7.07	2.19	27.18	9.64
	Ours	**50.12**	**52.23**	**73.57**	**41.43**	**37.19**
F-score ↑	StructureNet1	41.05	35.61	36.12	26.62	38.89
	StructureNet2	**67.91**	58.21	44.44	29.97	60.52
	Ours	66.09	**72.85**	**82.08**	**47.33**	**62.21**

StructureNet1 means the mesh models by using Poisson reconstruction, StructureNet2 means the mesh models by using the algorithm [50].

StructureNet on the majority of four metrics (CD, EMD, F1-score and 3D IoU).

In addition, we make comparison with SDM-Net [39]. Since SDM-Net is an auto-encoder for 3D mesh models, we use the pre-trained models in SDM-Net and a pre-trained 2D image encoder (ResNet18 [46]) to map the 2D image into the learned latent space in SDM-Net. Thus, 3D models can be reconstructed by SDM-Net from the single view images. As shown in Fig. 6, our method can reconstruct the models with complex structure, while it is challenging for SDM-Net to represent the complex structure models with composed parts. The purpose of SDM-Net and that of this work is different: SDM-Net is designed as a deep shape generative model for novel shape generation, while this method is designed as a shape reconstruction network given a single image. As demonstrated by StructureNet, the recursive GNN is more suitable than the fully-connected layer for modeling the complex relationships between a large of parts. Thus our method can handle much more flexible number of parts with complex structure, including handling holes in the structure, as opposed to SDM-Net.

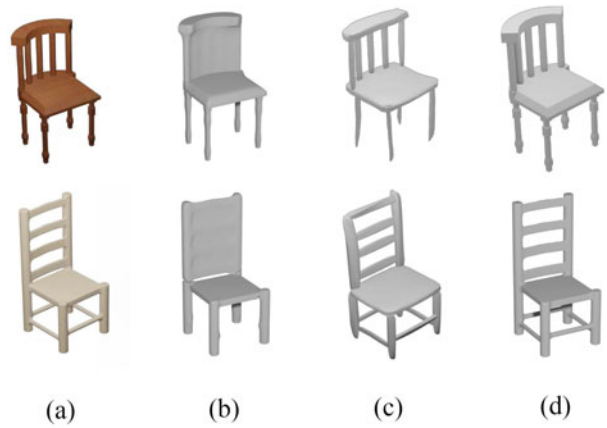


Fig. 6. The qualitative results compared with SDM-Net [39]. (a) Input image; (b) reconstructed by SDM-Net; (c) reconstructed by our method; (d) ground truth.

Regarding the box deformation, there is also a difference between SDM-Net and our method. SDM-Net learns the distribution of deformation feature of the training set with a low-dimensional space and uses the space to represent the box deformation, then recovers the deformation feature to the 3D coordinates. In contrast, the method we proposed directly uses a neural network to deform a box to 3D meshes under the guidance of an RGB image, which is designed for shape reconstruction.

6.4 Ablation Study

In this section, we investigate the influence of individual components of our learning network and demonstrate their importance through ablation studies.

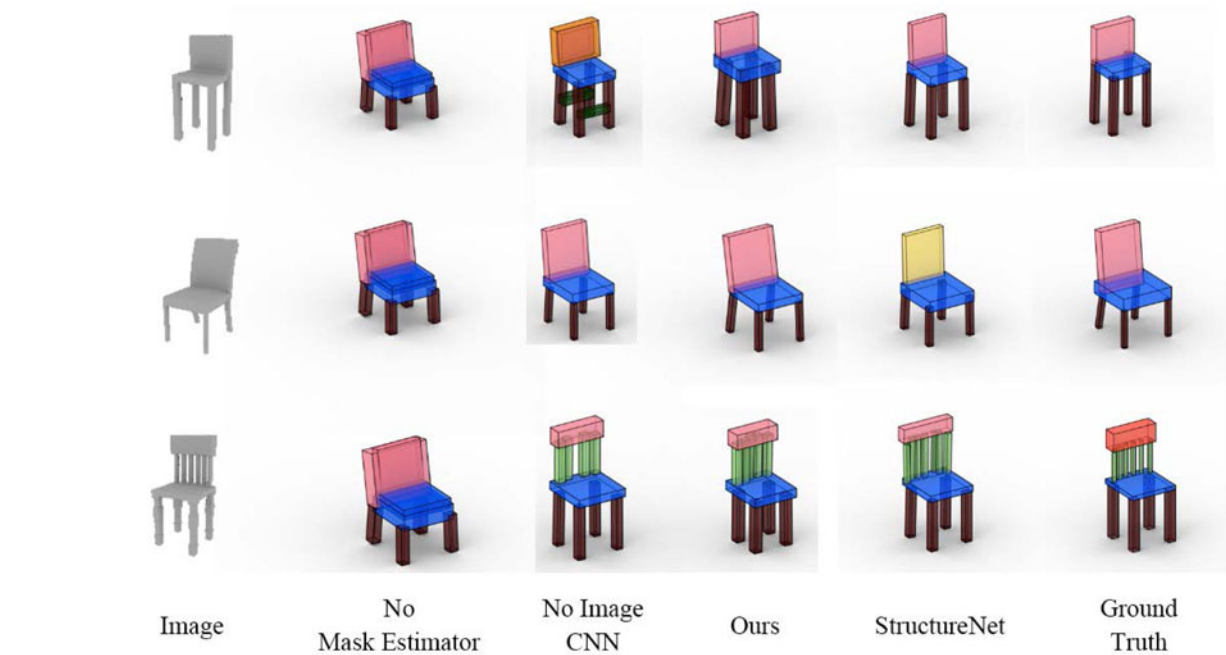


Fig. 7. The figure shows the ablation study of mask estimator and image CNN module in the encoder.

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.

Fig. 8. The figure shows the ablation study of the number of TAGCN block, which shows the greater number of TAGCN block has little influence on the predicted results.

First, we make an ablation study to demonstrate the effectiveness of the mask estimator in the encoder. As show in Fig. 7, the hierarchy of cuboid bounding boxes of three chairs with different structures look very similar if the mask estimator is removed. With the help of the mask estimator, we can obtain the bounding boxes with finer details. By comparison, the bounding boxes recovered by StructureNet for the chair backs of the three chairs have either inferior shape and angle or incorrect holder compared to those of ours and ground truth. It proves that the mask estimator importantly helps to extract the complex topological structure information of the models. Also, we evaluate the impact of image CNN module. From the results we can see that the generated bounding boxes may lose some structural expression and cause shattered topology, such as the chairs in the first and third rows when the image CNN module is removed.

Then, we make another ablation study on the number of TAGCN blocks in the deformation network. In this study the number of TAGCN block is changed as 3, 5 and 7. From Fig. 8, we can see that the greater number of TAGCN block has little influence on the predicted results. However, we also find in the experiment that when the number of block is over 7, the network is difficult to train due to the memory expense. Meanwhile, we evaluate the impact of the TAGCN by replacing them with naive GCN. Table 4 shows that the TAGCN has a crucial improvement on the evaluation metrics (CD, F1-score and IoU), which is because that TAGCN has better performance in deforming the bounding boxes to corresponding shapes.

Furthermore, we make one more ablation study for the contribution of different terms in the loss function on the three Metrics (CD, F1-score and IoU). As shown in Table 5, we can see that every term has great improvement on the performance metrics. The recovery loss $\mathcal{L}_{recovery}$ helps to produce more accurate bounding boxes representing the structure. Sampling loss $\mathcal{L}_{pts}$ helps the deformation to produce plausible shapes. Laplacian loss $\mathcal{L}_{laplacian}$ helps to produce smoothness in the deformation field across 3D space and along the mesh surface. The normal loss $\mathcal{L}_{normal}$ and edge loss $\mathcal{L}_{edge}$ help to guarantee a better topology and construct a better mesh in the deformation process.

TABLE 4
This Table Shows the Ablation Study of TAGCN and Naive GCN Layers

	CD loss	F1-score	IoU
Naive GCN	3.550	30.84	12.76
TAGCN	**1.390**	**72.98**	**52.6**

TABLE 5
This Table Shows the Result of Ablation Study for Different Terms in the Loss Function

	CD loss	F1-score	IoU
STD-net	**1.390**	**72.98**	**52.60**
w/o $\mathcal{L}_{recovery}$	3.467	30.99	10.56
w/o $\mathcal{L}_{pts}$	2.390	50.78	32.41
w/o $\mathcal{L}_{normal}$	3.357	31.28	10.68
w/o $\mathcal{L}_{laplacian}$	3.550	30.84	20.54
w/o $\mathcal{L}_{edge}$	3.812	36.46	10.41

6.5 Limitation Discussion

The main limitations of our work are: (i) we use the decoder in StructureNet [6] to recover the structure of objects, which makes our final results also influenced by the performance of StructureNet on recovering the object's structure, such as suffering from the errors of missing parts, duplicate parts, detached parts in StructureNet; (ii) the training data of images in our dataset are prepared by the rendered images of 3D shapes in PartNet and ShapeNet, thus using the real world images as input tends to generate incorrect bounding boxes, which is also the same as that of StructureNet, as shown in Fig. 10; (iii) the structural details of reconstructed models still have defect, such as the corners of the table tops or monitor screens, which are rounded instead of forming perpendicular angles (e.g., see tables and displays models in Fig. 3); (iv) large part deformations are often not captured, such as the hat tops and lamp covers, which are reconstructed with deformed shape compared to the ground truth (e.g., see hats and lamps models shown in Fig. 5).

7 Conclusion

In this paper, we propose a structure-preserving and topological-adaptive network for 3D objects reconstruction from single images. Our method provides a graph representation of 3D models with cuboid bounding boxes, which can delicately describe the structure information of an object. Thus, our method can reconstruct 3D models with complex structure directly from a single image. Our learning framework consists of a structure recovery network and a mesh deformation network. The former is designed with an auto-encoder that generates cuboid bounding boxes for an object, and the latter consists of three deformation blocks intersected with two graph unpooling layers, which progressively deform the input meshed bounding boxes into the meshed models. The most significant feature of the mesh deformation network is that it accepts a hierarchy of bounding boxes with different topology types, which enables the reconstruction of 3D models with various complex structures. Compared with previous methods, our method could achieve better performance in 3D shape structure-preserving reconstruction. The future work of this paper would focus on improving

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.

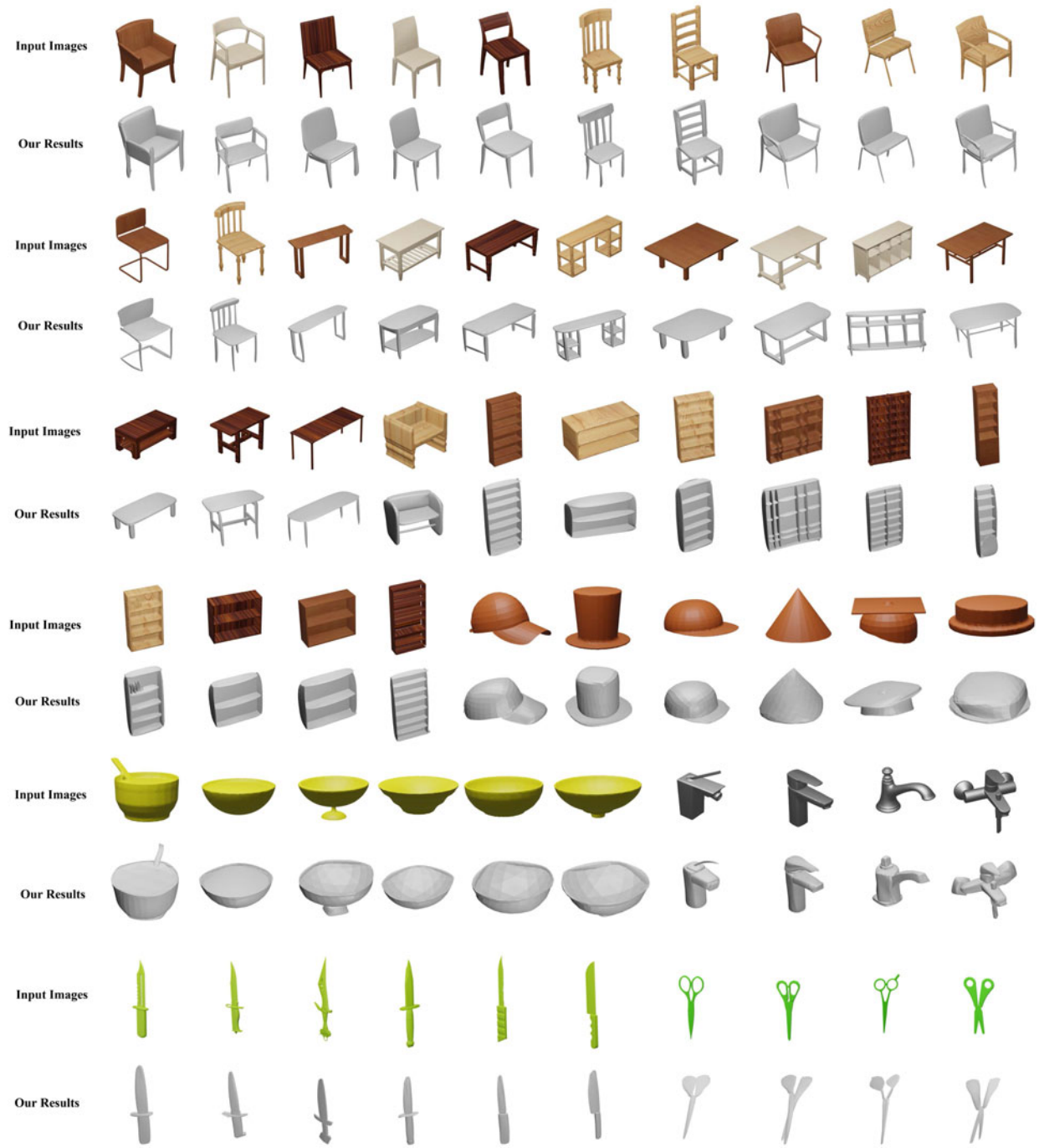


Fig. 9. More results of generated 3D models of our approach.

Fig. 10. Failure cases of predicted bounding boxes using the real world images from internet as input.

the capability to express the delicate structures of more wide categories of objects and further improve the structural details of the reconstructed results.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments that help us to improve the article. Canglan Dai Dai and Qing Liu contributed equally to this work.

REFERENCES

[1] A. X. Chang *et al.*, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.

[2] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y. G. Jiang, "Pixel2Mesh: Generating 3D mesh models from single RGB images," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 55–71.

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

[4] Q. Tan, L. Gao, Y.-K. Lai, J. Yang, and S. Xia, "Mesh-based autoencoders for localized deformation component analysis," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018.

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

[6] K. Mo, P. Guerrero, L. Yi, H. Su, P. Wonka, N. Mitra, and L. Guibas, "StructureNet: Hierarchical graph networks for 3D shape generation," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–19, 2019.

[7] J. Li, K. Xu, S. Chaudhuri, E. Yumer, H. Zhang, and L. Guibas, "Grass: Generative recursive autoencoders for shape structures," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–14, 2017.

[8] C. Sun, Q. Zou, X. Tong, and Y. Liu, "Learning adaptive hierarchical cuboid abstractions of 3D shape collections," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–13, 2019.

[9] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proc. Advances Neural Inf. Process. Syst.*, 2016, pp. 82–90.

[10] E. Smith, S. Fujimoto, A. Romero, and D. Meger, "GEOMetrics: Exploiting geometric structure for graph-encoded objects," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 5866–5876.

[11] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2088–2096.

[12] S. Tulsiani, H. Su, L. J. Guibas, A. A. Efros, and J. Malik, "Learning shape abstractions by assembling volumetric primitives," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2635–2643.

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

[14] Z. Huang *et al.*, "Deep volumetric video from very sparse multi-view performance capture," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 336–354.

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

[16] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6620–6629.

[17] C. Hane, S. Tulsiani, and J. Malik, "Hierarchical surface prediction for 3D object reconstruction," in *Proc. Int. Conf. 3D Vis.*, 2018, pp. 412–420.

[18] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.

[19] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 40–49.

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

[21] J. K. Pontes, C. Kong, S. Sridharan, S. Lucey, A. Eriksson, and C. Fookes, "Image2Mesh: A learning framework for single image 3D reconstruction," in *Proc. Asian Conf. Comput. Vis.*, 2017, pp. 365–381.

[22] J. Lei, S. Sridhar, P. Guerrero, M. Sung, N. Mitra, and L. J. Guibas, "Pix2Surf: Learning parametric 3D surface models of objects from images," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 121–138.

[23] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning category-specific mesh reconstruction from image collections," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 371–386.

[24] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry, "3D-CODED: 3D correspondences by deep deformation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 235–251.

[25] R. Hanocka, G. Metzer, R. Giryes, and D. Cohen-Or, "Point2Mesh: A self-prior for deformable meshes," *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 126–1, 2020.

[26] J. Gao *et al.*, "Learning deformable tetrahedral meshes for 3D reconstruction," 2020, *arXiv:2011.01437*.

[27] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4455–4465.

[28] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser, "Local deep implicit functions for 3D shape," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4857–4866.

[29] J. Chibane, T. Alldieck, and G. Pons-Moll, "Implicit functions in feature space for 3D shape reconstruction and completion," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6968–6979.

[30] M. Michalkiewicz, J. K. Pontes, D. Jack, M. Baktashmotlagh, and A. Eriksson, "Deep level sets: Implicit surface representations for 3D shape inference," 2019, *arXiv:1901.06802*.

[31] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 165–174.

[32] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5939–5948.

[33] W. Wang, Q. Xu, D. Ceylan, R. Mech, and U. Neumann, "DISN: Deep implicit surface network for high-quality single-view 3D reconstruction," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 492–502.

[34] L. Yariv *et al.*, "Multiview neural surface reconstruction by disentangling geometry and appearance," *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020.

[35] M. Atzmon and Y. Lipman, "SAL: Sign agnostic learning of shapes from raw data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2565–2574.

[36] K. Xu, H. Zheng, H. Zhang, D. Cohen-Or , L. Liu, and Y. Xiong, "Photo-inspired model-driven 3D object modeling," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 80:1–80:10, 2011.

[37] Q. Huang, H. Wang, and V. Koltun, "Single-view reconstruction via joint analysis of image and shape collections," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 1–10, 2015.

[38] C. Niu, J. Li, and K. Xu, "Im2Struct: Recovering 3D shape structure from a single RGB image," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4521–4529.

[39] L. Gao *et al.*, "SDM-NET: Deep generative network for structured deformable mesh," *ACM Trans. Graph.*, vol. 38, pp. 243:1–243:15, 2019.

[40] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[41] L. Yi, H. Su, X. Guo, and L. Guibas, "SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6584–6592.

[42] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[43] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[44] J. Du, S. Zhang, G. Wu, J. M. Moura, and S. Kar, "Topology adaptive graph convolutional networks," 2017, *arXiv:1710.10370*.

[45] J. Li, R. Klein, and A. Yao, "A two-streamed network for estimating fine-scaled depth maps from single RGB images," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 3392–3400.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[47] W. Wang, D. Ceylan, R. Mech, and U. Neumann, "3DN: 3D deformation network," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1038–1046.

[48] K. Mo *et al.*, "PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 909–918.

[49] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5932–5941.

[50] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proc. 19th Annu. Conf. Comput. Graph. Interactive Techn.*, 1992, pp. 71–78.

[51] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, p. 163–169, 1987.

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.

[52] X. Sun *et al.*, "Pix3D: Dataset and methods for single-image 3D shape modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2974–2983.
[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[54] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. 4th Eurographics Symp. Geometry Process.*, 2006, pp. 61–70.

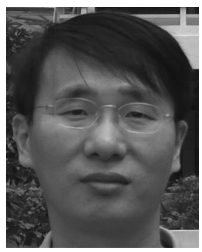
Aihua Mao received the PhD degree from Hong Kong Polytechnic University, Hong Kong, China in 2009, the MSc degree from Sun Yat-Sen University, Guangzhou, China, in 2005, and the BEng degree from Hunan University, Changsha, China, in 2002. He is currently a professor with the School of Computer Science and Engineering, South China University of Technology (SCUT), China. His research interests include 3D vision and computer graphics.

Canglan Dai He received the BEng degree from Zhongnan University of Economics and Law, Wuhan, China, in 2018. He is currently working toward the MSc degree with the School of Computer Science and Engineering, South China University of Technology (SCUT), Guangzhou, China. His research interests include computer graphics.

Qing Liu received the BS degree from Nanchang University, Nanchang, China, in 2020. He is currently working toward the MSc degree with the school of Software Engineering, South China University of Technology (SCUT), Guangzhou, China. His research interests include computer graphics.

Jie Yang received the bachelor's degree in mathematics from Sichuan University, Chengdu, China, in 2016. He is currently working toward the PhD degree with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His research interests include computer graphics and geometric processing.

Lin Gao received the bachelor's degree in mathematics from Sichuan University, Chengdu, China and the PhD degree in computer science from Tsinghua University, Beijing, China. He is currently an associate professor at the Institute of Computing Technology, Chinese Academy of Sciences. He has been awarded the Newton Advanced Fellowship from the Royal Society and the AG young researcher award. His research interests include computer graphics and geometric processing.



Ying He received the bachelor's and master's degrees in electrical engineering from Tsinghua University, Beijing, China, in 1997 and 2000, respectively and the PhD degree in computer science from Stony Brook University, Stony Brook, NY, in 2006. He is an currently associate professor with the School of Computer Engineering, Nanyang Technological University, Singapore. He is interested in the problems that require geometric computing and analysis.

Yong-Jin Liu (Senior Member, IEEE) received the BEng degree from Tianjin University, Tianjin, China, in 1998, and the PhD degree from the Hong Kong University of Science and Technology, Hong Kong, China, in 2004. He is currently a professor with the Department of Computer Science and Technology, Tsinghua University, China. His research interests include computer graphics and computer-aided design.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Authorized licensed use limited to: Tsinghua University. Downloaded on February 01,2023 at 09:43:17 UTC from IEEE Xplore. Restrictions apply.