Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.031506
Article

Tech Science Press

Fast Mesh Reconstruction from Single View Based on GCN and Topology Modification

Xiaorui Zhang[1,2,3,*], Feng Xu[2], Wei Sun[3,4], Yan Jiang[2] and Yi Cao[5]

[1]Wuxi Research Institute, Nanjing University of Information Science & Technology, Wuxi, 214100, China
[2]Engineering Research Center of Digital Forensics, Ministry of Education, Jiangsu Engineering Center of Network Monitoring, School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[3]Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science & Technology, Nanjing, 210044, China
[4]School of Automation, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[5]Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, N9B 3P4, Canada
* Corresponding Author: Xiaorui Zhang. Email: zxr365@126.com
Received: 19 April 2022; Accepted: 16 June 2022

Abstract: 3D reconstruction based on single view aims to reconstruct the entire 3D shape of an object from one perspective. When existing methods reconstruct the mesh surface of complex objects, the surface details are difficult to predict and the reconstruction visual effect is poor because the mesh representation is not easily integrated into the deep learning framework; the 3D topology is easily limited by predefined templates and inflexible, and unnecessary mesh self-intersections and connections will be generated when reconstructing complex topology, thus destroying the surface details; the training of the reconstruction network is limited by the large amount of information attached to the mesh vertices, and the training time of the reconstructed network is too long. In this paper, we propose a method for fast mesh reconstruction from single view based on Graph Convolutional Network (GCN) and topology modification. We use GCN to ensure the generation of high-quality mesh surfaces and use topology modification to improve the flexibility of the topology. Meanwhile, a feature fusion method is proposed to make full use of the features of each stage of the image hierarchically. We use 3D open dataset ShapeNet to train our network and add a new weight parameter to speed up the training process. Extensive experiments demonstrate that our method can not only reconstruct object meshes on complex topological surfaces, but also has better qualitative and quantitative results.

Keywords: 3D surface reconstruction; deep learning; GCN; topology modification; end-to-end framework

1 Introduction

Image-based 3D reconstruction is the process of recovering 3D information from 2D image, with the aim of obtaining a 3D model that matches the 2D image. The advantages of single-view-based reconstruction

methods are that they require less input data and usually only need to reconstruct the image of a single perspective of the object as input, so as to restore the whole shape of the object. With the emergence of large-scale 3D data sets such as ShapeNet [1] and Pix3D [2], single-view reconstruction methods integrating deep learning have gradually become the main trend in recent years.

Early single-view 3D reconstruction based on learning represented 3D structures as voxels [3–6] or point clouds [7–11]. Voxels lack much detail due to the limitation of their resolution and expression. Each point in the point cloud is not connected, so it will lack the surface information of the object. In comparison, the representation method of mesh [12–16] has the advantages of light weight and rich shape details. Recent advances in single-view mesh reconstruction can recover more specific object surface details. However, they still have some limitations. For example, due to the fixed connection relationship between the vertices of the initial mesh, most of the current methods only perform well in the reconstruction of objects with approximately predefined templates. When reconstructing complex topologies, unnecessary mesh self-intersections and connections are generated, thus destroying surface details; the flexibility of the mesh topology and the final reconstruction effect cannot be achieved at the same time and increasing the model accuracy can easily reduce its generalization; the mesh representation contains more information, and longer training cycles are required to achieve sufficient accuracy, and the training speed of the model is slower.

To solve the above challenges, we propose a method for fast mesh reconstruction from single view based on Graph Convolutional Network (GCN) and Topology Modification. First, we use GCN with residual connection, namely G-ResNet [13,14], to predict the position offset of the mesh vertices and the surrounding shape features, so as to generate 3D meshes with good surface details; at the same time, the meshes are processed after each deformation. After each deformation, the mesh topology is modified, the error surface is trimmed, and the original fixed structure is broken, to improve the flexibility of the topology structure. Secondly, we propose a new feature fusion method to make full use of the features of different stages of the image in a hierarchical input manner, to meet the input requirements of different modules and improve the compatibility of data structures between modules. Finally, we use 3D supervision to constrain the formation of the meshes and add a weight parameter to the corresponding loss function, so that the entire network can pay more attention to the backbone during training, thereby increasing the training speed and reducing resource consumption.

The contributions of this paper are mainly as follows:

- An end-to-end learning framework is proposed, which combines GCN and topology modification for the first time. It is used to reconstruct the 3D mesh model from a single image, considering the details of the reconstructed mesh surface and the flexibility of the mesh topology, and it can be applied to the reconstruction of complex structures with good generalization ability.
- Different from the previous method that simply vectorizes the image, we propose a new feature fusion method that uses the features of the image at different stages multiple times to meet the input requirements of different modules and make each module compatible with each other. This module can be integrated into other learning frameworks.
- A weight parameter related to the 3D loss function is proposed, which can give priority to the location of key points during training process, so as to achieve the purpose of improving the training speed. We use this to optimize our training methods and improve the stability of the network.

2 Related Works

Human beings are good at using prior knowledge to make inferences and predictions [17]. They can infer the approximate size and approximate geometric shape of objects with just one eye. For neural

networks, the same is true for reconstructing objects in a 2D image from a single perspective. This chapter will introduce in detail the work of our algorithm in related fields from three aspects: single-view-based 3D reconstruction technology, GCN and topology modification.

2.1 Single Image 3D Reconstruction

According to different representations, the single-view-based 3D reconstruction technology is mainly divided into three directions: voxel, point cloud and mesh. Voxels discretize an object into a 3D voxel grid. Its advantage is that it is easy to integrate deep learning frameworks (such as 3D convolution and max pooling). Choy et al. proposed 3D-R2N2 based on the voxel representation and used the 3DLSTM network structure to establish a mapping from 2D graphics to 3D voxel models, which completed single-view or multi-view 3D reconstruction based on voxels [3]. Deep Marching Cubes [5] is an end-to-end trainable network, which can predict an explicit surface representation of any topology. OctNet [6] recursively subdivides the 3D voxel grid into eight quadrants based on an octree, thereby reducing the computational complexity of 3D convolution. However, 3D voxel representation lacks geometric details, and the increase in accuracy requires an increase in resolution. The increase in resolution will greatly increase the calculation time and occupy expensive memory resources. Only a few technologies can achieve subvoxel accuracy.

In comparison, the point cloud is a simple, unified and easy-to-learn structure. Since the connectivity between vertices does not need to be updated, the point cloud is easier to manipulate during geometric transformation and deformation. Point Set Generation Network (PSGN) proposed by Fan et al. solves the problem of loss when training a point cloud network [7]. Li et al. introduced the Generative Adversarial Network (GAN) to deal with the disturbance problem in the point cloud generation process [8]. DeformNet proposed by Kurenkov et al. can find the most similar shape template in advance, and then generate a 3D dense point cloud through the deformed template [11]. Algorithms based on point cloud representation can process 3D objects of arbitrary topology, but the point cloud reconstruction lacks connectivity, so the surface information of the object will be lacking, and the surface will be uneven after reconstruction.

Polygon mesh is composed of vertices and triangular faces. It has the characteristics of scalability and curved surface, as well as light weight and rich shape details. The most important parts of mesh are the connections between adjacent points. N3MR [12], as the first 3D reconstruction method using mesh representation, can reconstruct some low-precision models with a small number of vertices. Pixel2Mesh is a coarse-to-fine network architecture that adds mesh vertices through the graphical pool layer to refine the mesh surface details [13]. Pan et al. proposed a topology modification network, which is characterized by its ability to prune the mesh topology in multiple stages [15]. Image2Mesh combines a rough topology map structure according to the image characteristics, and then uses the Free Form Deformation (FFD) to restore the dense 3D mesh model according to the estimated deformation [16].

2.2 Graph Convolutional Network (GCN)

In reality, many important data are stored in the form of graphs, such as social network information, knowledge graphs, protein networks, the World Wide Web, and so on. Since it is difficult for CNN to choose a fixed convolution kernel to adapt to the irregularities of the entire graph [18–20], its translation invariance is often not applicable to graph topology. As a network structure designed for graphs [21,22], GCN's essential purpose is to extract the spatial characteristics of topological graphs. Its core idea is to use edge in formation to aggregate node information to generate new node representations. The extraction of graph features by GCN mainly relies on the adjacency matrix and Laplacian matrix [23]. Li et al. proposed an adaptive graph convolutional network (AGCN), which learns unknown hidden structural relationships through the adjacency matrix of the graph [24].

3D mesh can also be regarded as a graph topology, and there are corresponding connections between the vertices and edges of the mesh. Wang et al. proposed for the first time that GCN is applied to the grid 3D reconstruction of a single image [13], which has greatly improved the training speed and model accuracy compared with previous methods. We follow their ideas, and the purpose is to perform controllable operations on high-quality 3D representation.

However, the uncertainty in the design of the corresponding convolution parameters for the mesh is large, and the mesh representation is no suitable for conventional 3D convolution operations. Therefore, in this work, we use polygon mesh as the 3D format and introduce graph convolutional neural network (GCN) to control this structure and solve the problem of incompatibility between mesh representation and neural network.

2.3 Topology Modification

The graph topology is controlled by the connection of vertices and edges, so a structure containing a large number of vertices like a 3D mesh is not easy to update, and it consumes more resources when performing convolution operations. Topology modification, as a technology to update the topology in real time, has been used in the design of broadband antenna structures [25]. The most common method for designing compact broadband antennas is to make various topological modifications to the antenna radiator, feeder, or ground plane, and reduce the area occupied by the antenna by finding a balance point in structural flexibility and material adaptability.

Pan et al. applied topology modification to the 3D reconstruction of the mesh, which solved the problem that the deformation of the mesh is limited by the predefined shape template and can adapt to the reconstruction of the surface of complex objects [15]. We introduce topology modification to process the output results of GCN, trims the surface predicted by GCN that produces large errors, and strikes a balance between the high-quality mesh surface and the flexibility of the topology.

3 Methods

3.1 Systems Overview

As an end-to-end network structure, when we are given an input image, the system outputs a 3D mesh model. An overview of the framework of this article is shown in Fig. 1. The entire network architecture is based on the “Encoder-Decoder” structure. The “encoder” is the image coding layer used to extract the features of the input image, while the “decoder” consists of two identical subnets. Each subnet contains a mesh deform block, a topology modification module and a boundary optimization module.

The encoding layer is used to extract 2D image features hierarchically, convert the input image into feature maps and feature vectors, and input them to the deform block and the topology modification module, respectively. The deform block modifies the predefined sphere mesh. By manipulating the feature vector attached to the vertices, the vertices can be deformed, so that the sphere mesh gradually tends to the object described by the input image. The topology modification module dynamically trims the mesh surface after each deformation, so that the mesh topology is no longer limited to a predefined template. After each deformation of the vertices and modification of the topology, we use a boundary optimization loss function to trim the zigzag boundary and smooth the model surface. In order to make the network produce stable deformation and generate accurate meshes, we combine the commonly used 3D loss function with boundary refinement loss to train our network.

Next, we will introduce the encoding layer, deform block, topology modification module and the 3D loss function used one by one.

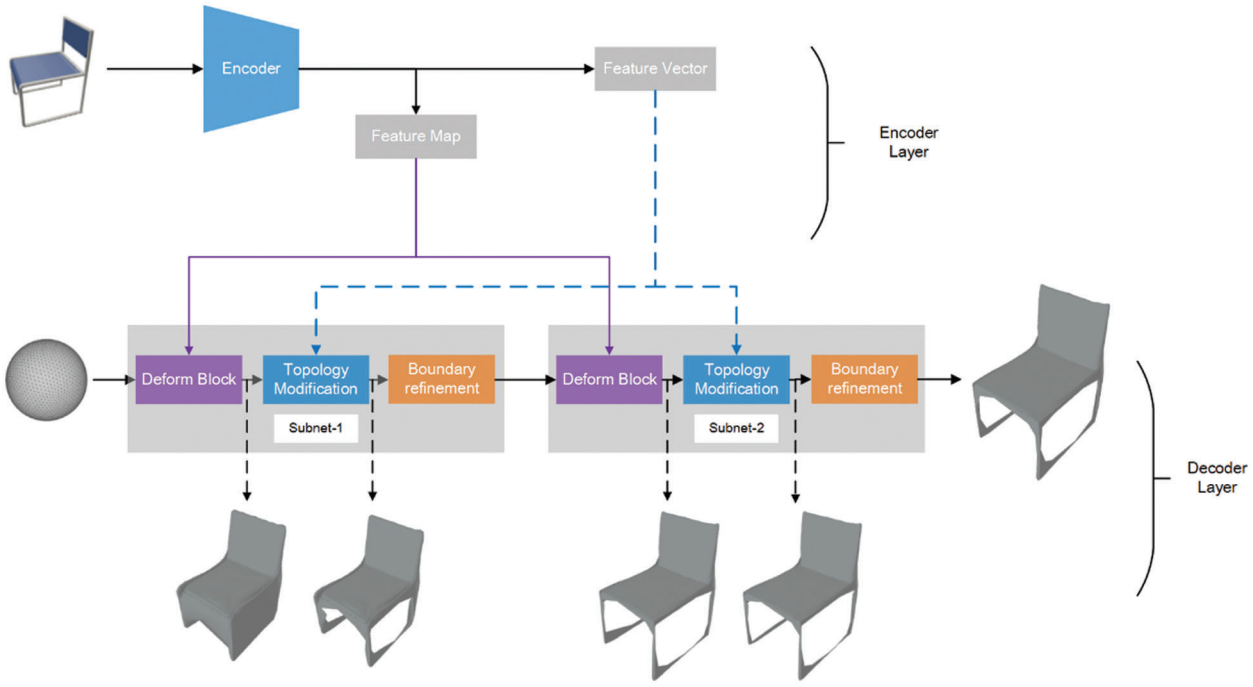

Figure 1: The overview of our method. Given a 2D image as input, we use an encoder to extract its features. Then the features are fed into the two subnets and the deformation of the initial sphere is calculated. Each subnet contains a deform block to predict vertex offsets, a topological modification module to trim the mesh, and a boundary refinement module to optimize surface details

3.2 Encoding Layer

The encoding layer uses VGG-19 as the main architecture of the network. First, input the image and extract it into feature maps of different layers and 1000-dimensional feature vectors, as shown in Fig. 2. Due to the use of the same size convolution kernel, the VGG-19 architecture has a small number of hyperparameters, which is simple and convenient in image encoding. Compared with the more commonly used two-dimensional image feature network VGG-16, VGG-19 has three additional convolutional layers, which can extract some deeper image features [26].

The whole process of image encoding can be reused. On the one hand, the three layers of feature maps, conv3_4, conv4_4 and conv5_4, are stitched together in series, and given any vertex in the 3D grid, its projection point on the input image is found according to the camera parameters, and the bilinear difference method finds and fuses the four pixels adjacent to the corresponding point of that vertex on the feature map as the feature vector for manipulating the deformation of that vertex in the grid deformation module. On the other hand, the 1024-dimensional vector formed by the whole VGG-19 network is an important benchmark to guide the topology modification. In this way, one feature extraction of the image satisfies the input requirements of both the mesh deformation and topology modification modules.

3.3 Deform Block

Define the mesh structure as $M = (V, E, F)$, where V is the set of mesh vertices, E is the set of edges connecting adjacent vertices, F is the set of triangular surfaces surrounded by edges. By modifying the feature vector f_p^l attached to the vertex, the deformation of the vertex is achieved. The feature vector f_p^l contains the coordinate information, shape feature, color feature and so on of the vertex. According to [13], the transformation process of the feature vector f_p^l can be expressed as:

$$f_p^{l+1} = \sigma\left(m_0 f_p^l + \sum_{q\in\phi(p)} m_1 f_q^l\right), \tag{1}$$

where f_p^l and f_p^{l+1} are the feature vectors of the vertex p in the prediction point set before and after convolution. q represents one of the neighboring vertices of p, $\phi(p)$ is the set of neighboring vertices of p, and bp is the feature vector of q. m_0 and m_1 are learnable parameter matrices applied to all vertices. We added a nonlinear activation function σ to the entire convolution operation to reduce the amount of calculation and memory consumption during backpropagation. Running convolutions updates the features, which is equivalent as applying a deformation. Mesh deformation is completed by a G-ResNet to predict the offset of the vertex. Input vertex coordinates, fused image perception features and shape features attached to the vertices, and G-ResNet outputs the moved vertex coordinates and features.

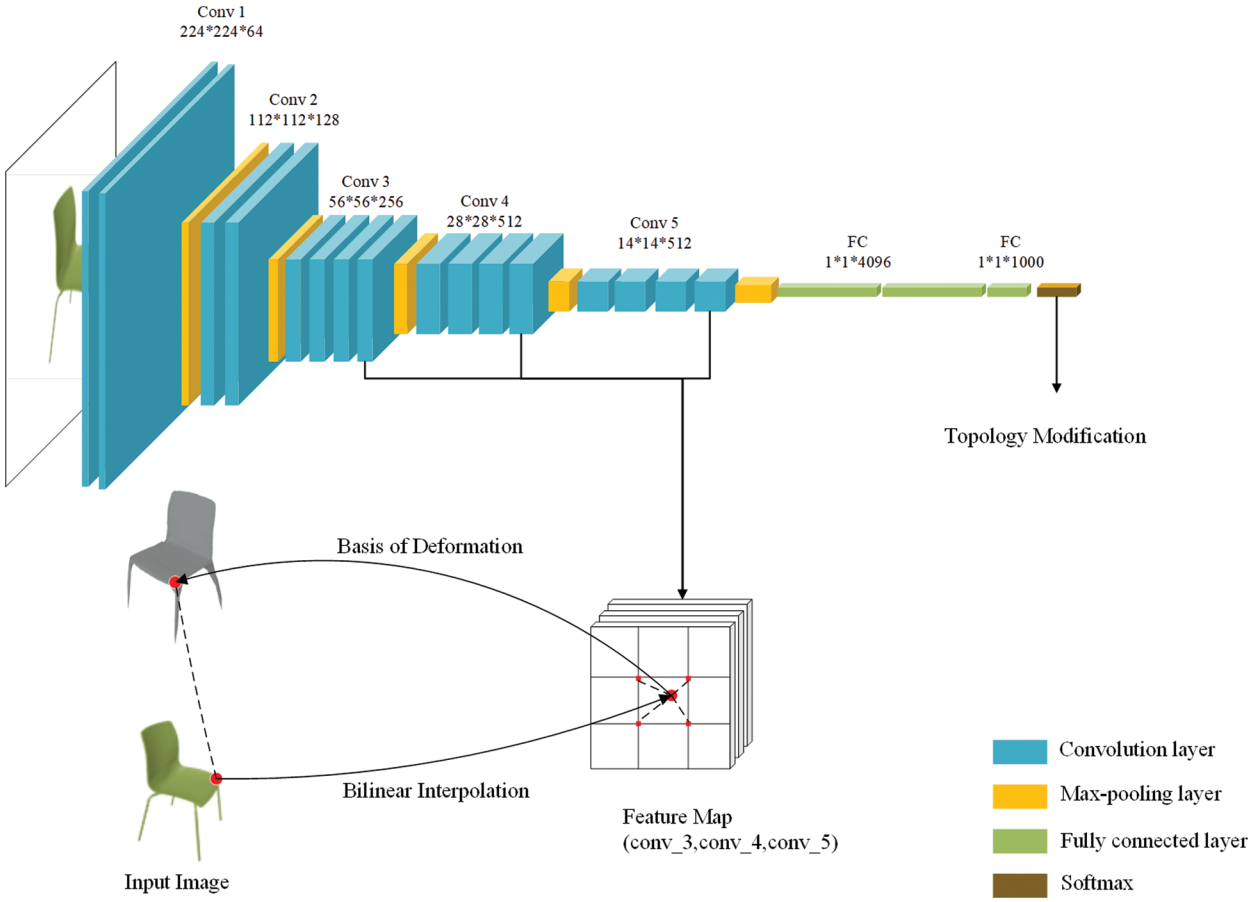

Figure 2: Reusable image feature network. We series three-layer feature map, input them to deform block, while apply the final feature vector to topology modification

However, mesh only predicted by G-ResNet is prone to obvious self-intersection, so it is necessary to trim the topology to achieve a suitable visual effect.

3.4 Topology Modification

In order to reduce the calculation of the deformation process and generate a more realistic 3D model, a topology modification process is added after each deform block to dynamically modify the topological relationship between the vertices and the surface of the mesh. A topology correction network is used to update the topological structure of the reconstructed grid by trimming the surfaces that clearly deviate from the ground truth.

Randomly sample points on the surface of the predicted grid topology M and connect the copied shape feature vector with the matrix containing all the sample points. Multilayer Perceptron (MLP) takes the spliced feature matrix as input and predicts the error distance of each vertex to the ground truth value. Calculate the average value of the prediction errors of all sampling points on the triangular surface of the grid and obtain the final error of each triangular surface.

We apply a threshold strategy to delete those faces whose errors exceed the predefined threshold, thereby updating the mesh topology. The threshold τ needs to be adjusted according to the actual situation to reach the most suitable grid structure for pruning. If the threshold τ is too high, it will reduce the trimming part and increase the reconstruction error; if the threshold τ is too low, it will delete too many triangular surfaces and destroy the topological structure of the mesh. Therefore, a coarse-to-fine method is adopted. First, a higher τ is given in the first module, and then τ is sequentially reduced in the subsequent modules to gradually refine the area to be trimmed.

Since the parameters of the network have not been trained at the initial stage, the 3D model after a round of mesh deformation and topology modification cannot achieve sufficient accuracy, so we use the corresponding 3D loss function to repeat the process many times until the generated model error is within the expected range.

3.5 Loss Functions

In this paper, the network is trained by 3D ground truth to constrain the deformation results of the mesh. The loss function is mainly based on Chamfer Distance $\mathcal{L}_{cd}$ and supplemented by Earth Mover's Distance $\mathcal{L}_{emd}$, which is used to constrain the position of the vertices of the mesh. At the same time, some regularization methods are used to optimize the results. Laplacian regularization $\mathcal{L}_{lap}$ [27] and edge length regularization $\mathcal{L}_{edge}$ [13] are used to constrain the generation of vertices and edges far away from the ground truth, while boundary regularization $\mathcal{L}_{bound}$ is used to smooth the jagged boundary generated by topology modification. Unless otherwise specified, in the following description, p is any point in the prediction point set, and q is any point in the ground truth point set. $\phi(p)$ represents the neighboring vertices of p, and k represents the neighboring pixels of p.

Chamfer loss. Chamfer Distance, as the most common constraint function in the field of 3D reconstruction, was originally used in the point cloud collection to represent the difference between the predicted vertex and the ground truth. Its main function is to limit the position of the vertex , gradually approaching the ground truth. If the loss is large, the difference between the two sets of vertices is large; if it is small, the reconstruction effect is better. The Chamfer loss can be defined as:

$$\mathcal{L}_{cd} = \sum_{q} \min_{p} \|p - q\|_2^2 + \sum_{p} \min_{q} \|p - q\|_2^2. \tag{2}$$

Earth Mover's loss. Earth Mover's Distance is defined as the minimum sum of the distances between a point in one set and a point in another set on all possible corresponding arrangements. Earth Mover's loss can be defined as:

$$\mathcal{L}_{emd} = \min_{p \to q} \sum_{p} \|p - \phi(p)\|_2^2. \tag{3}$$

Through Chamfer loss and Earth Mover's loss, the vertices can be gradually returned to the appropriate position, but it is not enough to produce a well-structured and stable mesh. Inspired by the work of Pixel2Mesh [13] and parameters compressing [28], we added a weight vector parameter $\Omega(q)$ to these two losses, expressed as:

$$\Omega(q) = w(q)_{q=\underset{q}{\operatorname{argmin}} \|p-q\|_2^2} \cdot \bar{w}[\phi(q)], \tag{4}$$

Here, $w(q)$ records the corresponding weight vector of vertex q in the ground truth, and q is the vertex closest to the predicted vertex p at this time. $\phi(q)$ is the neighboring vertex of vertex q. $\bar{w}[\phi(q)]$ represents the average of the weights of all adjacent vertices of q. The weight vector of the vertex p can be expressed as the weight relationship between the weight of the vertex q closest to the ground truth grid and its neighbor vertices [29]. This design is to make the predicted point set pay more attention to the vertices of key positions (such as high-weight vertices or vertices with a large number of adjacent vertices) in the process of returning to the ground truth. During training, the error of the key points is first minimized, which can stabilize the general reconstruction structure, thereby further improving the training speed.

Therefore, the Chamfer loss and Earth Mover's loss can be further defined as:

$$\mathcal{L}_{cd} = \sum_{q} \min_{p} \|p - q\|_2^2 \cdot \Omega(q) + \sum_{p} \min_{q} \|p - q\|_2^2 \cdot w(q), \tag{5}$$

$$\mathcal{L}_{emd} = \min_{p \to q} \sum_{p} \|p - \phi(p)\|_2^2 \cdot \Omega(q). \tag{6}$$

Boundary regularize. Sine the topological trimming of the mesh model will leave a jagged edge, which greatly destroys the visual appearance of the reconstructed mesh. In order to further improve the visual quality of the reconstructed mesh, we incorporate a boundary regularization term in the original loss, and penalize zigzag by forcing the boundary curve to remain smooth and consistent:

$$\mathcal{L}_{bound} = \sum_{x} \left\| \sum_{r \in \mathcal{N}(x)} \frac{(x - r)}{\|x - r\|} \right\|. \tag{7}$$

Here, x is the boundary point of the prediction mesh, $\mathcal{N}(x)$ represents the set of adjacent vertices of point x on the boundary, and r is any point in $\mathcal{N}(x)$.

Therefore, the final training goal of the model can be defined as:

$$\mathcal{L}_{all} = \mathcal{L}_{cd} + \lambda_1 \mathcal{L}_{emd} + \lambda_2 \mathcal{L}_{lap} + \lambda_3 \mathcal{L}_{edge} + \lambda_4 \mathcal{L}_{bound} \tag{8}$$

Here, $\lambda_{num} (num = 1, 2, 3, 4)$ are adjustable weight parameters. When the training produces the minimum value, the generated 3D mesh model is output.

4 Experiments

Figures and tables should be inserted in the text of the manuscript.

4.1 Experimental Setup

Next, we will introduce our experimental setup and details.

4.1.1 Dataset

The dataset ShapeNet is used for training, which contains 13 different object categories and corresponding 50,000 model images. We divide the dataset into a training set and a testing set. On the testing set, we can determine when to stop training by tracking the loss size of the method and all benchmarks.

4.1.2 Evaluation Metric

On the basis of following the standard 3D shape reconstruction evaluation method, we use two different numerical indicators to evaluate the performance of the model and compare with the existing advanced technology. The Chamfer Distance (CD) and Earth Mover's Distance (EMD) can be used both in training and testing. They are able to measure the error of the vertices between the predicted meshes and ground truth. When the two results are smaller, the experimental effect is better.

4.1.3 Baselines

We compare the proposed method with some existing 3D reconstruction techniques. Specifically, such as Deep Marching Cubes and PSGN, which are the more influential methods in volume reconstruction and point cloud reconstruction, respectively. In addition, we also compare Pixel2Mesh and TMNet in mesh reconstruction.

4.1.4 Training and Runtime

The input image size is set to 224*224. First, we pre-train the network structure shown in Fig. 1. In the pre-training, we only train the deform block, eliminating topology modification and boundary refinement. Then train Subnet-1 and Subnet-2 respectively. The training period of pre-training and Subnet-1 is set to 420, the training period of Subnet-2 is set to 120, and the learning rate is set to 0.001. The trained model uniformly contains 2562 vertices. The value of the hyperparameter mentioned in Eq. (8) is set as $\lambda_1 = 2e - 5$, $\lambda_2 = 0.3$, $\lambda_1 = 0.1$ and $\lambda_1 = 0.5$. The entire network structure is implemented in the pytorch framework. The number of vertices of the initial sphere is 10000, the number of sampling points in the topology modification module is 2500, and the number of vertices of the final training model is fixed to 2562. We put the system on a distributed server containing four GeForce RTX 3080 s for training.

4.2 Comparison to State of the Art

As shown in Fig. 3, we show the reconstructed visual effects of some chairs and compare the visual effects with the current state-of-the-art methods. Pixel2Mesh can construct the approximate outline and surface details of the object, but when dealing with the reconstruction of non-approximately spherical objects, it is easy to produce wrong surface connections. TMNet has made improvements on this point. It has independent topology modification capabilities, so it can reconstruct some complex topologies. For example, the back of a chair and its seat are clearly separated, and unnecessary connections are not generated at the legs of the chair. For example, there is a clear separation between the backrest of the chair and the seat of the chair, and there is no unnecessary connection at the legs of the chair. However, its reconstruction details are not good, and the reconstruction will take longer. Relatively speaking, our method combines the advantages of the two. Although there is still a certain gap with the ground truth, it can produce a shape topological structure with clear outlines and rich details.

We uniformly sample 1000 points on the surface of the generated model, and measure CD and EMD between them and the real point cloud of ground truth. Since PSG only generates the point cloud of the target, the ball-pivoting algorithm [30] is used to estimate the grid structure before sampling. The iterative closest point algorithm (ICP) [31] is used for the measurement results, so that it can be better compared with the ground truth value. The final results are recorded in Tabs. 1 and 2. It can be observed that most of the results of PSGN and Deep Marching Cubes are not ideal, because they use point cloud or voxel

representation methods, so the surface reconstruction results are not good. Since Pixel2Mesh uses mesh representation and a coarse-to-fine reconstruction process, its error rate is lower than the previous two. TMNet is close to the results of this method, but it uses MLP to perform affine transformation on objects to achieve the purpose of deformation, which consumes more resources in the training process, and the error rate in the initial training period will be higher. In comparison, our method is superior to the latest methods in most results, especially when reconstructing the surface of objects with complex topologies, such as tables and chairs with thin structures; at the same time, because we use the deformation module with the G-ResNet with residual connection and the corresponding weight parameter added to the loss function, our model consumes less resources during training and converges faster.

Figure 3: Qualitative results. (a) Input image; (b) Pixel2Mesh [11]; (c) TMNet [12]; (d) Ours; (e) Ground truth

4.3 Ablation Study

Now we conduct an ablation experiment to analyze the importance of each component in the entire model. Fig. 4 lists the reconstruction results of the 3D model when the corresponding components are missing, and each column corresponds to a reduction of one component. We found that after reducing the corresponding components, the quality of the reconstruction is reduced to varying degrees, and some of them cannot even generate suitable recognizable 3D shapes (for example, the column (b) in Fig. 4). After

removing the specific components, we perform the sampling point analysis experiment on the training results again and measure the error with the ground truth. The quantitative results of the ablation experiment are recorded in Tab. 3.

Table 1: Quantitative comparison of chamfer distance(CD) in units of 10^{-3}

Category	CD↓				
	PSG	Deep marching cubes	Pixel2Mesh	TMNet	Ours
Chair	6.647	5.415	4.932	4.850	**4.212**
Airplane	2.353	4.400	1.570	1.370	**1.125**
Lamp	2.740	3.292	2.828	**2.531**	3.295
Table	7.065	5.383	4.271	3.679	**3.180**
Firearm	2.186	4.907	1.790	1.754	**1.736**
Mean	4.198	4.679	3.078	2.836	**2.709**

Table 2: Quantitative comparison of earth mover's distance (EMD) in units of 10^{-2}

Category	EMD↓				
	PSG	Deep marching cubes	Pixel2Mesh	TMNet	Ours
Chair	13.809	13.266	12.106	11.256	**10.224**
Airplane	9.122	10.601	7.953	8.012	**7.560**
Lamp	12.174	11.630	10.457	**8.423**	8.637
Table	14.804	12.712	11.707	9.334	**8.221**
Firearm	7.696	9.412	7.590	7.769	**7.035**
Mean	11.521	11.524	9.962	8.958	**8.335**

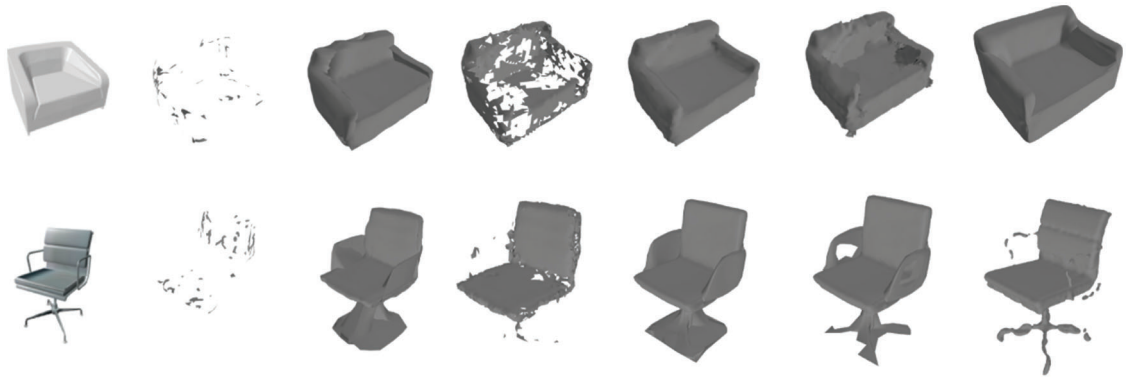

Figure 4: Qualitative results for ablation study. Each column shows the results of model training in the absence of the corresponding model component. (a) input image; (b) without both deform blocks; (c) without both topology modification; (d) without deform block in Subnet-2; (e) without topology modification in Subnet-2; (f) without boundary refinement; (g) full model

Table 3: Qualitative results for ablation study. The CD and EMD are in units of 10^{-2}. Each category shows the results from the model trained with the corresponding model component disabled

Category	CD↓	EMD↓
-Deform blocks (both)	N/A	N/A
-Topology modification (both)	5.071	12.698
-Deform block (Subnet-2)	6.249	15.463
-Topology modification (Subnet-2)	4.619	10.725
-Boundary refinement	4.087	11.311
Full model	4.212	10.224

We first remove the deform blocks in the two subnets, and directly perform topology modification and boundary refinement on the initial 3D sphere. It can be observed that the undeformed sphere lacks GCN's control over the topology, and a large number of error surfaces are predicted. Therefore, the topology modification trims most of the surfaces and destroys the original mesh topology, leaving only some Remaining grid fragments. Since the training result contains only a few vertices and mesh faces, we cannot perform sampling point analysis on them, as shown in Tab. 3. This is obviously inconsistent with the results we expected.

Second, we remove the topology modification modules in the two subnets and re-train the network. The generated model has a specific 3D shape, but there are some self-intersecting connections between the error surface and the grid. In particular, unnecessary connections exist in the thinner parts such as the chair legs and armrests. The reason is the lack of error prediction and surface trimming for topology modification, which only maintains the basic posture of the reconstructed object; at the same time, GCN will not break the constraints of spherical topology to form such a "hollow" surface.

After clarifying the indispensability of these two modules to the model, we also conduct ablation experiments and analysis on the number of modules. After training Subnet-1, we remove the deform blocks in both subnets and topology modification modules in Subnet-2. As shown in the detailed results in Fig. 5, the lack of deform blocks (Fig. 5d) can generate a specific shape, but the necessary part of the chair "legs" is over-trimmed, thus destroying the original shape. The error is quite different from the ground truth. We speculate that this is caused by the smaller threshold τ when predicting the error surface in Subnet-2. The lack of one topological modification (as shown in Fig. 5e) does not seem to have a significant impact on the surface details, and its error is relatively small, but there is still a certain gap between the final reconstruction effect. In addition, the verification based on the necessity of each regularization has been completed in Pixel2Mesh, so we only conduct ablation experiments and analysis for boundary regularization. As shown in Fig. 5f, although the lack of boundary regularization is not much different from the complete model in terms of error, its qualitative results have obvious jagged surfaces, especially near the trimmed thin structures.

Finally, we find that the discontinuous surface with more complex structure (such as the office chair in Fig. 4) is more affected in the ablation experiment analysis than the continuous surface approaching a sphere (the sofa in Fig. 4). It further illustrates that the method in this paper has good adaptability to the reconstruction of complex structures. At the same time, we conclude that among all other possible module combinations, the result after the complete subnet training of the two components is relatively the most appropriate and optimal.

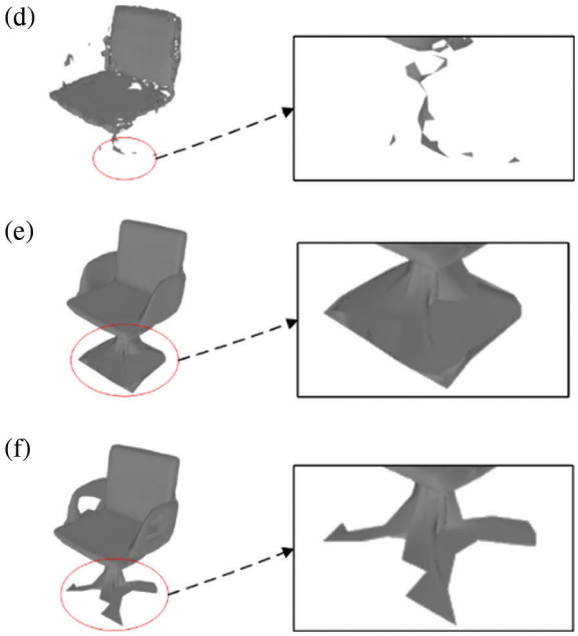

Figure 5: Detailed view of the columns (d), (e) and (f) in Fig. 4. For complex topology parts such as chair legs, deleting any block of the network will lead to a significant

5 Conclusion

Based on GCN and topology modification technology, we propose an improved end-to-end network architecture that can quickly generate 3D mesh models with complex topologies from a single perspective. Through the iterative use of GCN and topology modification, the problem that the high-quality surface reconstruction effect and the high flexibility of the topological structure cannot be achieved is solved. At the same time, the feature fusion method we propose uses hierarchical input to make full use of the various stages of the image and solve the problem of data input incompatibility between modules; in addition, the proposed weight parameters can help the network pay attention to the backbone position during training and reduce training consumption. A large number of experiments and measurement results show that the method in this paper can have a good reconstruction effect on common categories (especially categories with complex topological structures).

For future work, we will test our algorithm on other 3D data sets, such as Pix3D with pixel-level 2D-3D correspondence and Pascal3D + [32] with 3D graphics annotations. We will adjust our network structure based on the test results to improve the generalization ability of the model. At the same time, we will integrate the work done by existing reconstruction methods on surface detail optimization, such as color reconstruction, background detection [33–35], digital watermarking reconstruction [36] and texture fitting [37], to further improve the accuracy and authenticity of target reconstruction.

Acknowledgement: Thanks to the supervisor for writing guidance and other colleagues in the laboratory for their help.

Funding Statement: This work was supported, in part, by the Natural Science Foundation of Jiangsu Province under Grant Numbers BK20201136, BK20191401; in part, by the National Nature Science Foundation of China under Grant Numbers 61502240, 61502096, 61304205, 61773219; in part, by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang *et al.,* "3D shapenets: A deep representation for volumetric shapes," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 1912–1920, 2015.

[2] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang *et al.,* "Pix3d: Dataset and methods for single-image 3D shape modeling," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 2974–2983, 2018.

[3] C. B. Choy, D. Xu, J. Gwak, K. Chen and S. Savarese, "3d-R2n2: A unified approach for single and multi-view 3D object reconstruction," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 628–644, 2016.

[4] J. Wu, Y. Wang, T. Xue, X. Sun, B. Freeman *et al.,* "MarrNet: 3D shape reconstruction via 2.5d sketches," *Advances in Neural Information Processing Systems*, Barcelona, Spain, pp. 540–550, 2017.

[5] Y. Liao, S. Donn´e and A. Geiger, "Deep marching cubes: Learning explicit surface representations," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 2916–2925, 2018.

[6] G. Riegler, A. O. Ulusoy and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 3577–3586, 2017.

[7] H. Fan, H. Su and L. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 605–613, 2017.

[8] C. Li, M. Zaheer, Y. Zhang, B. Poczos and R. Salakhutdinov, "Point cloud GAN," arXiv preprint arXiv:1810.05795, 2019.

[9] X. R. Zhang, W. F. Zhang, W. Sun, X. M. Sun and S. K. Jha, "A robust 3-D medical watermarking based on wavelet transform for data protection," *Computer Systems Science & Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.

[10] X. R. Zhang, H. L. Wu, W. Sun, A. G. Song and S. K. Jha, "A fast and accurate vascular tissue simulation model based on point primitive method," *Intelligent Automation & Soft Computing*, vol. 27, no. 3, pp. 873–889, 2021.

[11] A. Kurenkov, J. Ji, A. Garg, V. Mehta, J. Gwak *et al.,* "DeformNet: Free-form deformation network for 3D shape reconstruction from a single image," in *2018 IEEE Winter Conf. on Applications of Computer Vision*, Nevada, USA, pp. 858–866, 2018.

[12] H. Kato, Y. Ushiku and T. Harada, "Neural 3D mesh renderer," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 3907–3916, 2018.

[13] N. Wang, Y. Zhang, Z. Li, Y. Fu, H. Yu *et al.,* "Pixel2mesh: 3D mesh model generation via image guided deformation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3600–3613, 2020.

[14] J. Zhang, J. Sun, J. Wang and X. G. Yue, "Visual object tracking based on residual network and cascaded correlation filters," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 8, pp. 8427–8440, 2021.

[15] J. Pan, X. Han, W. Chen, J. Tang and K. Jia, "Deep mesh reconstruction from single rgb images via topology modification networks," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, Seoul, Korea (South), pp. 9964–9973, 2019.

[16] J. K. Pontes, C. Kong, S. Sridharan, S. Lucey, A. Eriksson *et al.,* "Image2mesh: A learning framework for single image 3D reconstruction," *Asian Conference on Computer Vision*, Perth, Australia, pp. 365–381, 2018.

[17] F. Bi, X. Ma, W. Chen, W. Fang, H. Chen *et al.,* "Review on video object tracking based on deep learning," *Journal of New Media*, vol. 1, no. 2, pp. 63–74, 2019.

[18] X. R. Zhang, J. Zhou, W. Sun and S. K. Jha, "A lightweight CNN based on transfer learning for COVID-19 diagnosis," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 1123–1137, 2022.

[19] X. R. Zhang, X. Sun, W. Sun, T. Xu and P. P. Wang, "Deformation expression of soft tissue based on BP neural network," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 1041–1053, 2022.

[20] J. Chen, Z. Zhou, Z. Pan and C. Yang, "Instance retrieval using region of interest based CNN features," *Journal of New Media*, vol. 1, no. 2, pp. 87–99, 2019.

[21] T. Li, H. Li, S. Zhong, Y. Kang, Y. Zhang *et al.,* "Knowledge graph representation reasoning for recommendation system," *Journal of New Media*, vol. 2, no. 1, pp. 21–30, 2020.

[22] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam and P. Van-dergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[23] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, Barcelona, Spain, pp. 3844–3852, 2016.

[24] R. Li, S. Wang, F. Zhu and J. Huang, "Adaptive graph convolutional neural networks," in *Proc. of the AAAI Conf. on Artificial Intelligence*, New Orleans, LA, USA, vol.32, no.1, 2018.

[25] M. A. ul Haq and S. Koziel, "Comparison of topology modification for size-reduction-oriented wideband antenna design," in *2018 IEEE Int. Symp. on Antennas and Propagation & USNC/URSI National Radio Science Meeting*, Boston, USA, pp. 2161–2162, 2018.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556, 2014.

[27] J. H. Pang and G. Cheung, "Graph laplacian regularization for image denoising: Analysis in the continuous domain." *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1770–1785, 2017.

[28] S. He, Z. Li, Y. Tang, Z. Liao, F. Li *et al.,* "Parameters compressing in deep learning," *Computers Materials & Continua*, vol. 62, no. 1, pp. 321–336, 2020.

[29] W. Sun, X. Chen, X. R. Zhang, G. Z. Dai, P. S. Chang *et al.,* "A multi-feature learning model with enhanced local attention for vehicle re-identification," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 3549–3560, 2021.

[30] F. Bernardini, J. Mittleman, H. E. Rushmeier, C. T. Silva and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349–359, 1999.

[31] P. J. Besl and N. D. McKay, "A method for registration of 3D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[32] Y. Xiang, R. Mottaghi and S. Savarese, "Beyond pascal: A bench-mark for 3D object detection in the wild," in *IEEE Winter Conf. on Applications of Computer Vision*, Colorado, CO, USA, pp. 75–82, 2014.

[33] X. R. Zhang, X. Chen, W. Sun and X. Z. He, "Vehicle re-identification model based on optimized densenet121 with joint loss," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3933–3948, 2021.

[34] W. Sun, G. Z. Dai, X. R. Zhang, X. Z. Hea and X. Chen, "TBE-Net: A three-branch embedding network with part-aware ability and feature complementary learning for vehicle re-identification," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021.

[35] W. Sun, L. Dai, X. R. Zhang, P. S. Chang and X. Z. He, "RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring," *Applied Intelligence*, vol. 52, pp. 1–16, 2021.

[36] X. R. Zhang, X. Sun, X. M. Sun, W. Sun and S. K. Jha, "Robust reversible audio watermarking scheme for telemedicine and privacy protection," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3035–3050, 2022.

[37] A. Kanazawa, S. Tulsiani, A. A. Efros and J. Malik, "Learning category-specific mesh reconstruction from image collections," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 371–386, 2018.