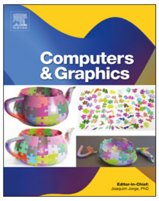
Computers & Graphics 115 (2023) 511–521

Contents lists available at ScienceDirect

Computers & Graphics

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



Special Section on CAD/Graphics 2023

Point2MM: Learning medial mesh from point clouds

Mengyuan Ge [a], Junfeng Yao [a,*], Baorong Yang [b], Ningna Wang [c], Zhonggui Chen [a], Xiaohu Guo [c]

[a] Xiamen University, China
[b] Jimei University, China
[c] The University of Texas at Dallas, United States of America

ARTICLE INFO

Article history:
Received 12 May 2023
Received in revised form 4 July 2023
Accepted 9 July 2023
Available online 18 July 2023

Keywords:
Medial mesh
Unsupervised learning
Medial axis transform
Point cloud
Shape analysis

ABSTRACT

Medial mesh is the most commonly used representation of medial axis transform (MAT) which is a high-fidelity compact representation of 3D shape. Existing learning methods of extracting medial mesh from point clouds either have strict requirements of supervising training data, or are incapable of learning the complete form of medial mesh, leading to large reconstruction error. We introduce *Point2MM*, an unsupervised method for learning a complete medial mesh from point clouds. Our key idea is to use the envelop geometry of medial primitives – spheres, cones, and slabs – to capture the intrinsic geometry of shape, and the connectivity of medial mesh to capture the topology of shape. We firstly predict initial medial spheres by learning the geometric transformation of point clouds, then construct an initial connectivity of the medial spheres by learning the probability of medial cones and medial slabs with a novel unsupervised formulation. Finally we propose an iterative strategy for fine-tuning medial primitives. Extensive evaluations and comparisons show our method has superior accuracy and robustness in learning medial mesh from point clouds. In addition, the excessive training time is also a concern for our research, and it is a limitation where we need to make improvements in our future work.

1. Introduction

Compact skeletal representations of 3D shapes could capture the underlying geometric and topological structures of the shape and have shown their advantages in shape recognition [1], segmentation [2], and animation [3,4].

Among all skeletal representations, medial axis transform (MAT) is a high-fidelity compact volumetric representation. Given a 3D shape, its MAT is defined as the set of maximally-inscribed spheres in the interior with at least two closest points on the boundary surface, as shown in Fig. 1(a). shape. Each 4D point (center and radius) on MAT, called *medial sphere*, denoted as $\mathbf{m} = (\mathbf{c}, r)$ with $\mathbf{c}$ and r the center and the radius respectively, can be associated with the local thickness, symmetry information, and part-structure of the object so that the original shape can be reconstructed from them [5]. A common representation of MAT is *medial mesh* [6], which is a non-manifold mesh structure consisting of medial vertices, *medial edges* and *medial faces*. The enveloping volume of a medial edge is a *cone* that is a linear interpolation of two spheres (see Fig. 1(b)), while the enveloping volume of a medial face is a *slab* (see Fig. 1(c)). The compactness of medial mesh makes it capable of precisely recognizing shape [1] and capturing the geometric details of shapes such as high-frequency local deformations in physics-based simulation [4].

It is known that computing MAT from surfaces of arbitrary representations is still a challenging problem. Recent works [5,7] show the possibility of learning the geometric transforms from point clouds with deep neural networks for computing the MAT of 3D shapes. These works, however, mostly obtain "incomplete" medial mesh representations and use post-processing to fill up the topological connectivity of medial cones and slabs. Specifically, P2MAT-NET [5] only predicts medial spheres, whereas Point2Skeleton [7] cannot learn medial slabs directly. As shown in Fig. 2, the post-processing of generating the connectivity of medial mesh in P2MAT-NET [5] could introduce reconstruction errors with large holes in the medial mesh, and the results of Point2Skeleton [7] indicate that planar surfaces could not be correctly reconstructed since it computes the medial slabs with post-processing instead of directly learning them from the input point clouds. These problems of previous methods motivate us to learn the complete medial mesh of 3D shapes from corresponding point clouds with neural networks for capturing the topology of shapes as well as their geometric details. We propose *Point2MM*,

* Corresponding author.
E-mail addresses: 24320190154490@stu.xmu.edu.cn (M. Ge), yao0010@xmu.edu.cn (J. Yao), yangbaorong@jmu.edu.cn (B. Yang), ningna.wang@utdallas.edu (N. Wang), chenzhonggui@xmu.edu.cn (Z. Chen), xguo@utdallas.edu (X. Guo).

https://doi.org/10.1016/j.cag.2023.07.020
0097-8493/© 2023 Elsevier Ltd. All rights reserved.

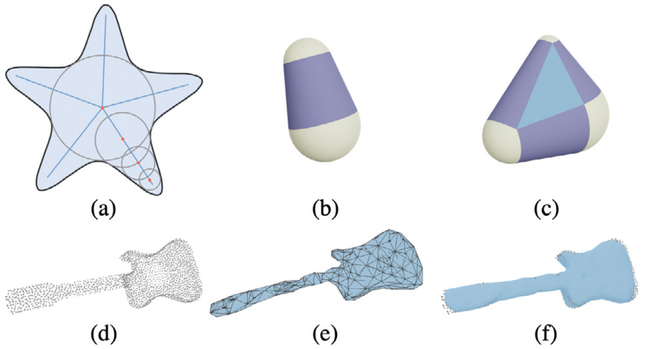


Fig. 1. Illustrations of (a) medial axis in 2D, (b) medial cone, (c) medial slab. We also show (d) an example 3D point cloud, (e) its medial mesh, and (f) the enveloping volume of the medial mesh.

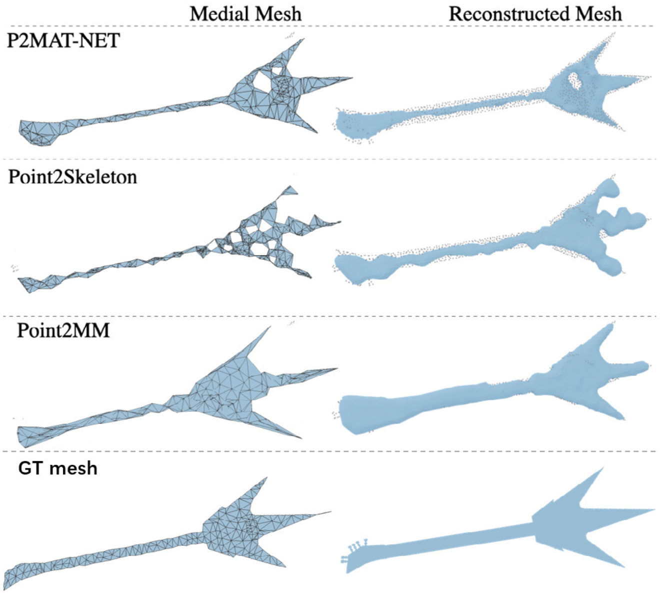


Fig. 2. An example of generated medial mesh with its enveloping volume (reconstructed mesh) given the input point clouds for three methods: P2MAT-NET [5], Point2Skeleton [7] and Point2MM.

an unsupervised method for learning complete medial mesh, i.e., directly learning medial spheres, medial cones, and medial slabs from 3D point clouds. Point2MM consists of two main steps, the first is to learn medial spheres from the input point cloud, while the second is to learn the connectivity of the predicted medial spheres. To be specific, the contributions of this paper are as follows:

- To the best of our knowledge, Point2MM is the first unsupervised method predicting medial mesh directly from point clouds, i.e., learning a complete medial mesh consisting of medial spheres, medial cones, and medial slabs.
- We propose a connectivity prediction network that integrates a graph convolutional network and graph attention network to precisely encode the local details of the shape.
- We present novel unsupervised formulations for geometric learning of 3D point clouds, i.e., learning geometric transformations of point clouds to medial spheres and predicting connectivity of medial mesh.

2. Related work

2.1. Medial axis transform

The concept of the medial axis was first introduced by Blum [8] as an effective tool to describe 2D shapes in image analysis, and it was subsequently generalized to represent geometric objects in higher dimensions and applied in a wide range of areas such as shape discrimination, object retrieval, object segmentation, shape deformation, and robotic path planning. In the last two decades, various methods have been proposed to extract medial axis transform, most of them worked on triangle meshes [9–12]. MATFP [12] propose a new framework based on a novel insight that the surface-restricted power diagram (RPD) can be utilized to compute the internal MAT of the triangle mesh. However, a small disturbance on the surface boundary will result in a significant change to the structure of the medial axis. MAT simplification methods [6,13] try to prune those insignificant branches and obtain a clean MAT for triangle meshes which adopted an error-bounded MAT simplification scheme. However, [6,13] can only calculate the manifold triangle mesh. MATFP [12] can handle the non-manifold mesh, but it cannot calculate a mesh with open boundary.

The discreteness and irregularity of a point cloud make it difficult to extract its medial axis. Only several methods can be used for computing MATs from point clouds [5,7,14]. The input sampling points should be dense enough in the Power Crust method [14], while the number of the output medial spheres in P2MAT-NET [5] has to be the same as the number of the input points. But point clouds are relatively easier to acquire compared to 3D manifold meshes. Our paper focuses on the specific problem in extracting the MAT from point clouds.

2.2. Computing skeletal mesh from point cloud

There are many methods for extracting skeletal meshes. Tagliasacchi et al. [15] proposed a method ROSA (rotational symmetry axis), which is based on columnar point clouds, uses rotation invariance to extract skeletons, and solves the problem of accurately extracting skeletons with a large number of columnar points missing. But ROSA relies on the normals of point clouds to extract the skeleton and is sensitive to noises such as some sparse and scattered points which were far away from the target subject point cloud. Huang et al. [16] proposed a skeleton extraction method L_1-medial based on iterative shrinkage, which is insensitive to noises. However, L_1-medial skeleton extraction focuses on the accuracy of a single skeleton branch, ignoring the relationship between skeleton points. L_1-MST (L_1-minimum spanning tree) [17] solved the problem of skeleton line fracture based on L_1-medial. L_1-medial was used to extract the skeleton line of the initial fracture, and the minimum spanning tree was used to complete the structure. But after completion, the skeleton still has the problem that the position of the repaired endpoint is inaccurate, and the repaired skeleton does not conform to the shape of the original point cloud. Qin et al. [18] proposed a skeleton extraction algorithm, MdCS (mass-drive topology-aware curvelet), which is based on optimal transport theory to solve Wasserstein distance. It can extract a correct skeleton from the point cloud but is not suitable for sparse point clouds and has high time complexity. Deep learning methods [19–21] were proposed to extract curve skeletons, but they are mainly aimed at extracting skeletons from 2D images and are not suitable for skeleton extraction of unordered 3D point clouds.

Most of the previous works focus on extracting skeletal mesh, but not MAT. Point2Skeleton [7] was the first unsupervised method for generating medial meshes from 3D point clouds.

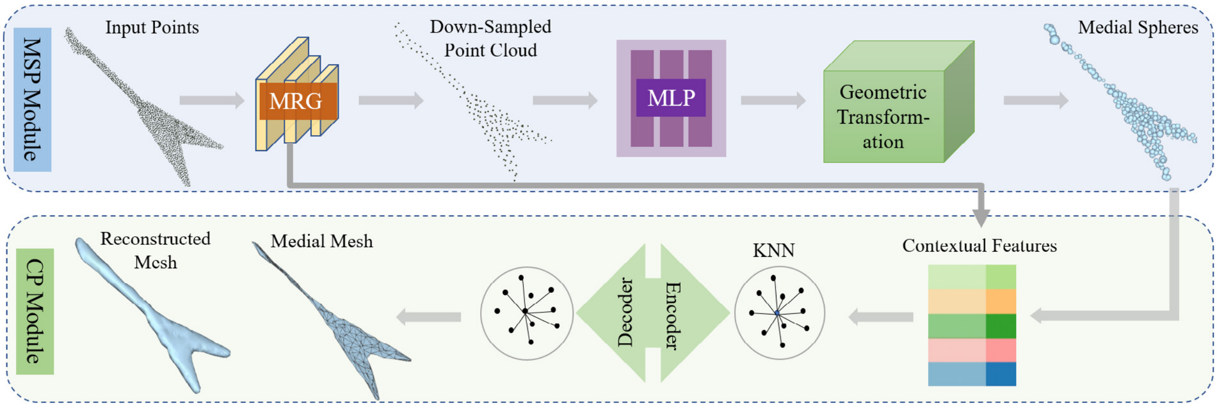


Fig. 3. An overview of Point2MM. The upper part represents the MSP module which predicts medial spheres from the input point cloud, while the bottom part represents the CP module for connectivity prediction from the medial spheres predicted by the MSP module.

However, it only predicted medial cones connecting spheres and medial slabs were generated from the medial cones. In this work, we present Point2MM, the first neural network that can learn a complete medial mesh, including medial spheres, cones, and slabs, from input 3D point clouds.

2.3. Mesh reconstruction from point clouds

Reconstruction from point cloud to mesh is a popular field of research in computer vision and computer graphics. Recently, deep learning-based approaches leverage neural networks to learn the mapping between point clouds and meshes. They often employ architectures such as convolutional neural networks (CNNs) or graph neural networks (GNNs) to extract meaningful features from the input point cloud and generate the corresponding mesh. DMC [22] is an end-to-end trainable network that can predict explicit surface representations with arbitrary topology. The authors utilized a differentiable representation to separate the mesh topology from geometry. However, the approach is limited to low-resolution meshes of size 32^3. Deeppano [23] reconstructed the 3D objects by projecting them onto 2D images, then applied a deep learning model for feature learning on the 2D images. MeshCNN [24] provides an effective framework for processing 3D mesh data, with a downsampling process conducted for tasks such as segmentation and classification. Due to its sequential pooling operation, MeshCNN exhibits slow computational speed and lower-resolution grids. Point2Mesh [25] learns from individual objects by optimizing the weights of a convolutional neural network (CNN) to deform an initial mesh and contract it to envelop the input point cloud, which restricts that it can only reconstruct shapes homeomorphic to the template mesh. Pixel2Mesh++ [26] generates 3D meshes based on image convolution. Although it does not directly generate meshes from point clouds, this approach stands out in the field of 3D reconstruction for its distinctive characteristics. DHSP [27] introduces a technique for reconstructing a textured mesh from a colored point cloud by utilizing self-prior in deep neural networks. Unlike these mesh reconstruction approaches, we aim to reconstruct the corresponding mesh from the input point cloud by learning the medial mesh of the object and approximating the shape with the envelope of the reconstructed medial mesh in this paper.

3. Method

Our approach is comprised of two modules, namely the MSP and CP modules, as illustrated in Fig. 3. Given a 3D shape's point cloud, the MSP module, which is the first module, predicts the medial spheres while incorporating geometric transformations (Section 3.1). The CP module predicts the connectivity between medial spheres (Section 3.2). Our training process involves four stages, namely pre-training, medial-spheres-training, connectivity training, and joint-training stages. During the pre-training stage, the initial centers of medial spheres are predicted using the MSP module. The medial-spheres-training stage optimizes the centers and radii of the medial spheres using the MSP module. The connectivity-training stage predicts the initial connectivity from the medial spheres, which are predicted in the medial-spheres-training stage, using the CP module. Finally, the joint-training stage optimizes both the medial spheres and the connectivity of MAT with the MSP module and the CP module, respectively.

Following the definition in Q-MAT [6], MAT of a shape is represented as medial mesh $\mathcal{M}$ that consists of vertices $\mathbf{M}$, edges $\mathbf{E}$, and triangles $\mathbf{F}$, namely medial spheres, medial cones, and medial slabs respectively (see Fig. 1). Each sphere $\mathbf{m} \in \mathbf{M}$ is denoted as $\mathbf{m} = (\mathbf{c}, r)$ with $\mathbf{c}$ and r being the center and the radius respectively. A medial cone $\mathbf{e}_{ij}$ is defined as the linear interpolation of two medial spheres $\mathbf{m}_i$ and $\mathbf{m}_j$, and a medial slab $\mathbf{f}_{ijk}$ is the linear interpolation of three medial spheres $\mathbf{m}_i$, $\mathbf{m}_j$ and $\mathbf{m}_k$.

3.1. Medial spheres prediction (MSP)

Given a point cloud $\mathcal{P} = \{(\mathbf{p}_i, \mathbf{n}_i)\}_{i=1}^{U}$ with U points, in which each point $(\mathbf{p}_i, \mathbf{n}_i) \in \mathcal{P}$ is represented as the 3D coordinates $\mathbf{p}_i$ and its normal $\mathbf{n}_i$, our goal is to predict N medial spheres $\{\mathbf{m}_j\}_{j=1}^{N} = \{(\mathbf{c}_j, r_j)\}_{j=1}^{N}$, denoted as $\mathbf{M} \in \mathbb{R}^{N\times 4}$ consisting of the centers $\mathbf{C} \in \mathbb{R}^{N\times 3}$ and their radii $\mathbf{R} \in \mathbb{R}^{N\times 1}$. Similar to Point2Skeleton [7], PointNet++[28] is employed to encode the input point cloud $\mathcal{P}$ by down-sampling it to $\bar{U}$ points, denotes as $\bar{\mathcal{P}}$, and extract the contextual features $\mathcal{F} \in \mathbb{R}^{\bar{U}\times D}$ of the down-sampled points $\bar{\mathcal{P}}$, where $\bar{U} < U$ and interactive dimension of the feature. Then the feature will be transformed to $W = \bar{U} \times N$ with 4 MLP layers. We take W as the weight matrix which represents the weight of down-sampled points for the medial spheres. The medial spheres are computed with the same method as Point2Skeleton [7], i.e. taking them as the convex combination of the sampled input point cloud $\bar{\mathcal{P}}$ with a geometric transformation. The centers $\mathbf{C}$ can be computed as follows:

$$\mathbf{C} = W^{\top}\bar{\mathcal{P}}, \tag{1}$$

and the radii $\mathbf{R}$ can be computed by the following equation:

$$\mathbf{R} = W^{\top}\mathbf{D}, \tag{2}$$

where $\mathbf{D} \in \mathbb{R}^{\bar{U}\times 1}$ summarizes the closest distances from down sampled points $\bar{\mathcal{P}}$ to the medial centers $\mathbf{C}$. The closest distance

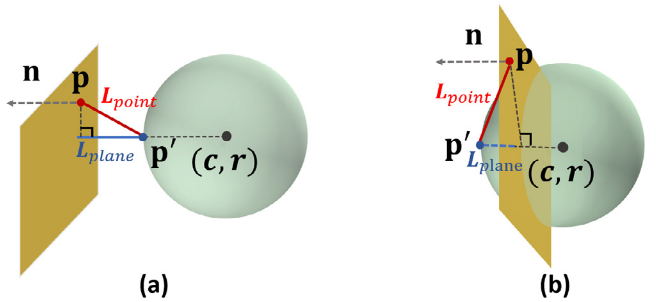


Fig. 4. Illustration of the definition of surface loss. (a) is the case when the plane is far away from the sphere, and (b) is the case when the plane is close to the sphere. Only when the sphere is tangent to the point, the two losses will be zero.

from an input point $\mathbf{p}$ to all the medial spheres is defined as follows,

$$d(\mathbf{p}, \{\mathbf{c}_i\}) = \min_{\mathbf{c}\in\{\mathbf{c}_i\}} \|\mathbf{p} - \mathbf{c}\|_2. \tag{3}$$

It is observed that the medial spheres might cluster together at the beginning of training. To precisely predict medial spheres from the input point cloud, it is crucial to evenly distribute the medial spheres in and around the shape. Furthermore, as the radius of a medial sphere is computed from its center, obtaining a good distribution of the centers of medial spheres in the pre-training stage, without considering the radii of medial spheres, is essential.

In the medial-spheres-training stage, both the centers and radii of medial spheres are optimized. To measure the reconstruction error from two different perspectives, sampling loss, and surface loss are introduced.

Sampling loss: We utilize the same sampling loss $\mathcal{L}_{sample}$ as in Point2Skeleton [7] to measure the reconstruction error between the input points $\mathcal{P}$ and the points $\mathcal{T}$ uniformly sampled from the predicted spheres. This loss function is equivalent to the Chamfer distance.

$$\mathcal{L}_{sample} = \sum_{\mathbf{p}\in\mathcal{P}} \min_{\mathbf{t}\in\mathcal{T}} \|\mathbf{p} - \mathbf{t}\|_2 + \sum_{\mathbf{t}\in\mathcal{T}} \min_{\mathbf{p}\in\mathcal{P}} \|\mathbf{t} - \mathbf{p}\|_2. \tag{4}$$

For a medial sphere $\mathbf{m} = (\mathbf{c}, r)$, a surface point $\mathbf{t}$ on it can be sampled by $\mathbf{t} = \mathbf{c} + r\mathbf{v}$, where $\mathbf{v}$ is a given unit direction vector. 8 points are sampled from one medial sphere by 8 given unit vectors $\mathcal{V} = \{(\pm\eta, \pm\eta, \pm\eta) | 3\eta^2 = 1\}$.

Surface loss. It is observed that the predicted medial spheres of Point2Skeleton [7] could not precisely approximate the shape, i.e., the associated radii of predicted spheres could not reveal local thickness. To overcome this limitation, we incorporate point normals during training and introduce surface loss which is similar to the energy function proposed by Wang et al. [12]. As shown in Fig. 4 and Eq. (5), for an input point $\mathbf{p}$ with normal $\mathbf{n}$ and a sphere $\mathbf{m} = (\mathbf{c}, r)$, $\boldsymbol{L}_{plane}$ in blue measures the distance from the sphere $\mathbf{m}$ to the tangent plane defined by $\mathbf{p}$ and $\mathbf{n}$, while $\boldsymbol{L}_{point}$ in red measures the distance from the sphere surface to the point $\mathbf{p}$. That is $\boldsymbol{L}_{plane}$ measures the perpendicular distance between the surface point $\mathbf{p}'$ on the predicted sphere pointing from the center along the direction of normal and projection of $\mathbf{p}'$ on the plane defined by point $\mathbf{p}$ and its normal $\mathbf{n}$, while $\boldsymbol{L}_{point}$ measures to the distance between $\mathbf{p}$ and $\mathbf{p}'$.

$$\begin{cases} \boldsymbol{L}_{plane} = ((\mathbf{p} - \mathbf{c})^\top \mathbf{n} - r)^2, \\ \boldsymbol{L}_{point} = \|\mathbf{c} + r\mathbf{n} - \mathbf{p}\|^2. \end{cases} \tag{5}$$

To measure the reconstruction error from the medial spheres $\mathbf{M}$ and the surface of the object which is represented as the tangent planes defined by the input points $\mathcal{P}$, we introduce surface loss $\mathcal{L}_{sur}$. $\mathcal{L}_{sur}$ consists of two parts: $\mathcal{L}_{plane}$ and $\mathcal{L}_{point}$.

Intuitively, we hope each sphere can be tangent to two planes defined by its two nearest input points, which is consistent with the definition of MAT, see the $\mathcal{L}_{plane}$ term in Eq. (6). However, the loss could be nearly zero if the sphere is tangent to both the planes defined by its two closest points, even when the points are not that close to the surface of the sphere. Therefore, we introduce the other $\mathcal{L}_{point}$ term which measures the distances between the point $\mathbf{p}$ and the surface point $\mathbf{p}'$. Only when the plane and the input point are both close enough to the sphere, the reconstruction error will be small. Our surface loss $\mathcal{L}_{sur}$ can be written as:

$$\begin{aligned} \mathcal{L}_{sur} &= \mathcal{L}_{plane} + \mathcal{L}_{point} \\ &= \sum_{i=1}^{N} \sum_{t=1}^{2} ((\mathbf{p}_i^t - \mathbf{c}_i)^\top \mathbf{n}_i^t - r_i)^2 \\ &+ \sum_{j=1}^{U} \|\mathbf{c}_j^{min} + r_j^{min} \mathbf{n}_j - \mathbf{p}_j\|^2, \end{aligned} \tag{6}$$

where $\mathbf{p}_i^t \in \mathcal{P}$, $t \in \{1, 2\}$ are the two closest points of sphere center $\mathbf{c}_i \in \mathbf{C}$ with normal $\mathbf{n}_i^t$, and $\mathbf{c}_j^{min} \in \mathbf{C}$ is the closest center of point $\mathbf{p}_j \in \mathcal{P}$ with the corresponding radius r_j^{min}. Note that the distances between points and the centers are measured in Euclidean space and the normal of point is only used in training.

In the medial-spheres-training stage, we use both the sampling loss and the sphere loss with $\beta_1 = 1$ and $\beta_2 = 0.3$:

$$\mathcal{L}_{skel} = \beta_1 \mathcal{L}_{sample} + \beta_2 \mathcal{L}_{sur}. \tag{7}$$

3.2. Connectivity prediction (CP)

The proposed CP module serves to predict the connectivity among medial spheres, i.e., to predict the cones and slabs. Given a set of predicted medial spheres, we connect each sphere with its K nearest medial spheres to initialize the connectivity of a medial mesh. Then a connectivity correction network is proposed to optimize the connectivity by using *Graph Auto Encoder* (GAE) [29] network. The encoder of the network is composed of several GCN-GAT layers and residual blocks, aiming to obtain the latent representations. Each GCN-GAT layer consists of a graph convolutional network (GCN) [30] and a graph attention network (GAT) [31], while the decoder decodes the latent code into the probability of cones and slabs.

3.2.1. Connectivity initialization

We use a graph $\mathbf{G} = (\mathbf{N}, \mathbf{E})$ to represent the medial mesh $\mathcal{M}$, where $\mathbf{N}$ is the set of medial spheres and $\mathbf{E}$ is the set of edges (cones) in the medial mesh. In our network, we use an adjacency matrix $\mathbf{A} \in (0, 1)^{N\times N}$ to represent the edges $\mathbf{E}$ in $\mathbf{G}$, where N is the number of medial spheres and $a_{ij} = \mathbf{A}[i][j]$ indicates whether the edge between medial sphere $\mathbf{m}_i$ and $\mathbf{m}_j$ exists or not.

To initialize the adjacency matrix $\mathbf{A}$, we use *K Nearest Neighbors* (KNN), which means that each medial sphere is initially connected to its K closest medial spheres. In our experiments, we set $K = 16$ to retain a sufficient number of edges and faces at the beginning of the connectivity prediction stage, and then propose a connectivity correction network to optimize the connectivity of the medial mesh.

3.2.2. Connectivity correction

The initial connectivity of the medial mesh, obtained by connecting each medial sphere with its K nearest neighbors, is both redundant and incorrect. To remove redundant edges and predict the missing links, we use a Graph Auto Encoder (GAE) network that analyzes the predicted connectivity of medial spheres obtained from the MSP module.

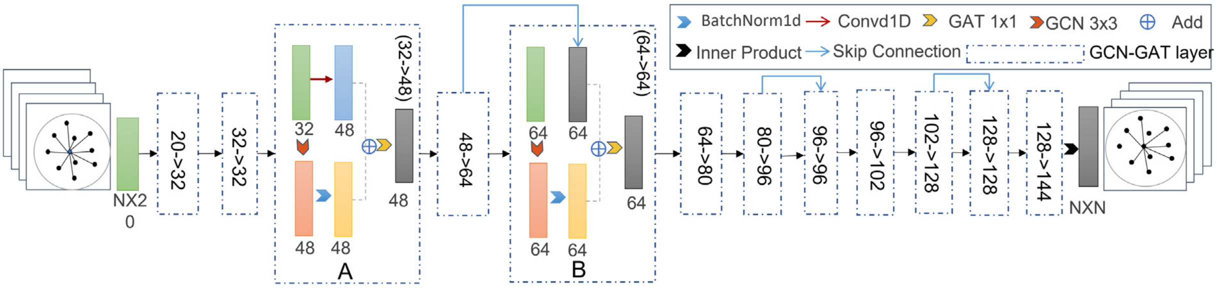


Fig. 5. The pipeline of CP module. The network takes the initial connectivity as input and the optimized connectivity of the medial mesh as output. It contains two parts: an encoder and a decoder. The encoder is composed of 12 GCN-GAT layers(see Fig. 5). A regular GCN-GAT layer (marked as 'A' in the figure) first applies a 1D convolution operation to the input features and adds the output features of the GAT layer to get the final features of the current layer. While a layer in which the dimensions of input features and output features are the same (marked as "B" in the figure) takes the output features of its prior GCN-GAT layer with a skip connection and adds the output features of the GAT layer to get the final features of this layer. The decoder applies an inner product operation on the features of the final GCN-GAT layer to get the adjacency matrix.

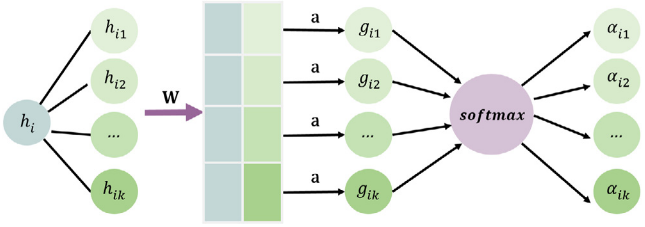


Fig. 6. Illustration of computing attention coefficients.

In addition to the adjacency matrix $\mathbf{A}$, we also input the feature of each medial sphere, which is a 4D vector $\mathbf{m} = (\mathbf{c}, r)$ concatenated with the corresponding contextual feature computed from the MSP module. The encoder is a deep GCN-GAT network consisting of 12 layers, which allows us to capture richer structures at various levels of abstraction. We also incorporated residual blocks to alleviate the problem of vanishing gradients due to the depth of the network.

To improve the performance of the GAE network in predicting the connectivity of the medial mesh, we introduced GCN-GAT layers instead of using only GCN layers as in traditional GAE networks. The GCN layers apply the same weight to all neighboring nodes. By concatenating a GAT [31] layer with each GCN layer, we can apply different weights (attention coefficients) to different neighboring nodes. This attention mechanism enables the network to better capture the contribution of each neighboring node to the current node in predicting connectivity.

Fig. 6 shows the process of obtaining the attention coefficient of neighbor nodes. To compute the attention coefficient α_{il} which represents the contribution of sphere $\mathbf{m}_l$ to sphere $\mathbf{m}_i$, the feature $\mathbf{h}_i$ of sphere $\mathbf{m}_i$, which is the output of the prior GCN layer, is firstly augmented to a higher dimension with a linear mapping $\mathbf{W}$ of shared parameters. The high-dimensional features $\mathbf{W}_{\mathbf{h}_i}$ and $\mathbf{W}_{\mathbf{h}_l}$ are concatenated and mapped to a real number g_{il} with the mapping operation $\mathbf{b}$. In our experiments, a single MLP layer is applied for the mapping, i.e.,

$$g_{il} = LeakyRelu(\mathbf{b}^T[\mathbf{W}_{\mathbf{h}_i} \parallel \mathbf{W}_{\mathbf{h}_l}]), \tag{8}$$

where $\cdot \parallel \cdot$ means the concatenation of the high-dimensional features.

At the beginning of training, the neighbors of each sphere are computed with the KNN algorithm, as described in Section 3.2.1. Later in training, the updated neighbors are used. Suppose $\mathbf{N}_i$ are the neighbors of medial sphere $\mathbf{m}_i$, the coefficients g_{il}, $l \in \mathbf{N}_i$ are then normalized with a softmax function. The normalized coefficients $\{\alpha_{il}\}$, $l \in \mathbf{N}_i$ are the attention coefficients of neighboring medial spheres to sphere $\mathbf{m}_i$. That is,

$$\alpha_{il} = \frac{\exp(g_{il})}{\sum_{l \in \mathbf{N}_i} \exp(g_{il})}. \tag{9}$$

After the attention coefficients are computed, the local information of each sphere should be integrated with its neighbors. We introduce a multi-head attention mechanism [31], which updates parameters for each head individually and finally averages the results of the G heads to obtain the output of the layer. G is set to be 3 in all our experiments. For each head of medial sphere $\mathbf{m}_i$, the initial features $\{\mathbf{h}_l\}$, $l \in \mathbf{N}_i$ of neighboring medial spheres are expanded by different linear mapping $\mathbf{W}^g_{\mathbf{h}_l}$. And the high-dimensional features of neighboring spheres are integrated with the corresponding attention coefficient $\{\alpha_{il}\}$, then activated with a sigmoid function. Thus the final feature $\mathbf{h}'_i$ of sphere $\mathbf{m}_i$ could be obtained as follows:

$$\mathbf{h}'_i = \frac{\sum_{g=1}^{G} \sigma(\sum_{l \in N_i} \alpha^g_{il} \mathbf{W}^g_{\mathbf{h}_l} \mathbf{h}_l)}{G}. \tag{10}$$

The decoder performs a simple inner product to produce a new adjacency matrix $\hat{\mathbf{A}}$ followed by a sigmoid function to output the probabilities. The probability $p(i,j)$ of medial cone $\mathbf{e}_{ij}$ is calculated as the average of $\hat{a}_{ij}$ and $\hat{a}_{ji}$. If it is greater than 0.5, we assume the medial cone exists. Similarly, for medial slab $\mathbf{f}_{ijk}$, the probability $p(i,j,k)$ is calculated as the average of $\hat{a}_{ij}$, $\hat{a}_{ji}$, $\hat{a}_{jk}$, $\hat{a}_{kj}$, $\hat{a}_{ik}$ and $\hat{a}_{ki}$. It exists when $p(i,j,k)$ is greater than 0.5.

3.2.3. Connectivity reconstruction loss

To measure the distance between the input point cloud and the mesh reconstructed using the enveloping volume of medial spheres, medial cones, and medial slabs, reconstruction error $\mathcal{L}_{recon}$ which measures on both cones and slabs is proposed, i.e.,

$$\mathcal{L}_{recon} = \alpha_1 \mathcal{L}_{cone} + \alpha_2 \mathcal{L}_{slab}. \tag{11}$$

In all experiments, we set the hyperparameters $\alpha_1 = 0.3$ and $\alpha_2 = 0.4$ respectively. To evaluate the accuracy of the predicted medial mesh, we introduce medial distance d^P_q (see Section 3.2.4) to measure the distance from an input point $\mathbf{q}$ to the medial primitive P. We only consider the inner points $\mathbf{Q}_P$ that are inside the scaled volume of the primitive computed by expanding the two (or three) medial spheres of the medial primitive by 1.1 times and taken as the AABB bounding box of the expanded medial primitive. Then we define the average medial distance D_P of all the inner points as follows,

$$D_P = \frac{\sum_{\mathbf{q} \in \mathbf{Q}_P} d^P_q}{\|\mathbf{Q}_P\|}, \tag{12}$$

where $\| \cdot \|$ represents the number of point set.

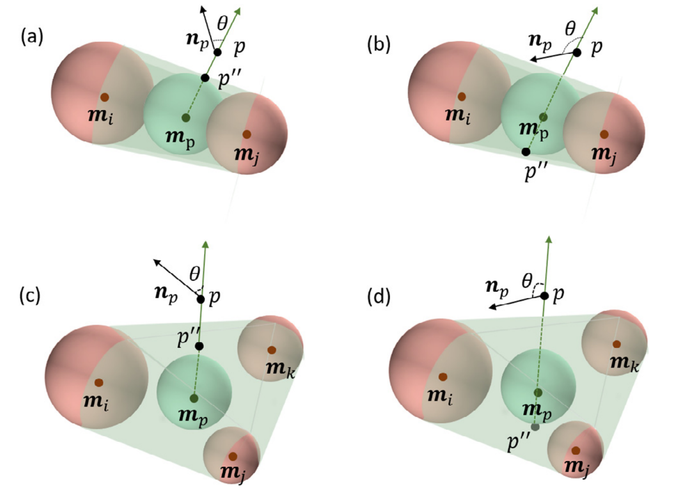


Fig. 7. Illustration of footprint sphere on a cone (a)(b) and a slab (c)(d). The angle between the corresponding outward normal and the sample point normal θ can be either (a)(c) $\theta \leq 90°$, or (b)(d) $\theta > 90°$.

For a medial primitive, when the average medial distance D_P is larger than a given threshold λ (0.02 in all experiments), it should have a small weight in reconstructing the medial primitive. On the contrary, when the medial distance is not larger than the threshold, the point should have a larger weight. Therefore, the reconstruction errors of medial cones $\mathcal{L}_{cone}$ and medial slabs $\mathcal{L}_{slab}$ are defined as follows:

$$\mathcal{L}_{cone} = \begin{cases} \sum_{i=1}^{N}\sum_{j=1}^{N} p(i,j)D_{\mathbf{e}_{ij}}, & \text{for } i \neq j \text{ and } D_{\mathbf{e}_{ij}} > \lambda, \\ \sum_{i=1}^{N}\sum_{j=1}^{N} (1-p(i,j))D_{\mathbf{e}_{ij}}, & \text{for } i \neq j \text{ and } D_{\mathbf{e}_{ij}} \leq \lambda, \end{cases}$$
$$\mathcal{L}_{slab} = \begin{cases} \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} p(i,j,k)D_{\mathbf{f}_{ijk}}, \\ \text{for } i \neq j \neq k \text{ and } D_{\mathbf{f}_{ijk}} > \lambda, \\ \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} (1-p(i,j,k))D_{\mathbf{f}_{ijk}}, \\ \text{for } i \neq j \neq k \text{ and } D_{\mathbf{f}_{ijk}} \leq \lambda. \end{cases} \tag{13}$$

3.2.4. Medial distance

To measure the distance from an input point to a medial primitive, denoted medial distance, we introduce the footprint sphere of a point on the medial primitive [32]. As shown in Fig. 7, for a medial cone $\mathbf{e}_{ij}$, the footprint sphere $\mathbf{m}_p = (\mathbf{c}_p, r_p)$ of sample point $\mathbf{p}$ can be defined by the two spheres $\mathbf{m}_p = \alpha\mathbf{m}_i + (1-\alpha)\mathbf{m}_j$, where $\alpha \in [0, 1]$. Then α can be calculated by minimizing the following equation:

$$\mathbf{E}_p(\mathbf{m}_p) = \|\mathbf{c}_p - \mathbf{p}\|^2 - r_p^2. \tag{14}$$

Similarly, the footprint sphere on a medial slab can be computed from the three medial spheres of the slab [32].

The next step is to measure the distance from input point $\mathbf{p}$ to medial primitive, namely medial distance, which is the Euclidean distance between $\mathbf{p}$ and its projection $\mathbf{p}''$ on the medial primitive. We denote the angle between the sample point normal $\mathbf{n}_p$ and the outgoing direction from the footprint center $\mathbf{c}_p$ to the sample point $\mathbf{p}$ as θ. The side pointing from the footprint center $\mathbf{c}_p$ to the point $\mathbf{p}$ is denoted as the near side while the other side is the rear side. For a medial cone as shown in Fig. 7(a) and (b), we consider two cases:

- $\theta \leq 90°$: $d_p^{ij} = |\|\mathbf{c}_p - \mathbf{p}\| - r_p|$. The projection $\mathbf{p}''$ is the projecting point on the projection sphere along the outgoing direction on the near side (see Fig. 7(a));
- $\theta > 90°$: $d_p^{ij} = \|\mathbf{c}_p - \mathbf{p}\| + r_p$. The projection $\mathbf{p}''$ is the projecting point on the projection sphere opposite to the outgoing direction on the rear side (see Fig. 7(b)).

Similarly, we can compute the medial distance d_p^{ijk} from sample point $\mathbf{p}$ to medial slab $\mathbf{f}_{ijk}$ by considering two cases, see Fig. 7(c) and (d).

4. Experimental results

Dataset. We collect 1086 models from 5 categories of ModelNet40 [33] and use the Poisson disk sampling method [34] to generate the input point cloud with 2000 random samples for each shape. We use 912 models for training and 174 models for testing. All models are normalized to the $[-0.5, 0.5]^3$ range.

Implementation. Our training consists of four stages: pre-training stage, medial-spheres-training stage, connectivity-training stage, and joint-training stage. Each stage is trained for 40, 40, 20, and 10 epochs respectively on a Nvidia Titian XP GPU with the Adam optimizer, taking about 2, 2, 26, and 13 h respectively. As for the test, each model will take 32 s per shape. For the first two stages, only the MSP module is trained, and the hyperparameters of the CP module are fixed. The hyperparameters of the MSP module are fixed to keep the medial spheres unchanged in the connectivity-training stage. In the joint-training stage, the hyperparameters of the MSP module and CP module are trained for reconstructing the whole medial mesh, including the medial spheres, medial cones, and medial slabs, to capture the fine details of shapes.

4.1. Results and comparisons

Evaluation metrics. We use two-sided Hausdorff distance (HD) and Chamfer distance (CD) to evaluate our method and the comparison methods. Note that the distances are normalized with respect to the diagonal length of the corresponding surface, and represented in percentage.

- To measure the reconstruction accuracy, we firstly use the method proposed by Wang et al. [12] to reconstruct the triangle mesh from the predicted medial mesh, then we use Hausdorff distance to measure the reconstruction error between the reconstructed triangle mesh and the corresponding ground truth mesh.
- To measure the quality of MAT, we compute the Chamfer distance between the predicted MAT to the ground truth MAT which is computed with Q-MAT+ [13]. To compute the Chamfer distance, spheres are densely sampled from the medial primitives from both the predicted MAT and the ground truth MAT.

Comparisons. We compare our method with a deep learning-based method Point2Skeleton [7] and a traditional geometric method DPC [10]. P2MAT-NET can extract medial mesh from sparse point clouds with the supervision of ground truth MATs. However, the number of predicted medial spheres should be the same as the number of input points. P2MAT-NET and Point2MM will predict medial mesh with a different number of spheres when feeding with a point cloud with the same number of points. To this end, it will not be able to conduct a fair comparison between P2MAT-NET and Point2MM. For Point2Skeleton, we retrain the network with our dataset for a fair comparison. DPC generates unstructured points for both surface-like and curve-like skeletons from an arbitrary 3D shape represented as a point cloud. For computing the CD loss of DPC [10], we directly use the densely-sampled points provided by the authors. And for

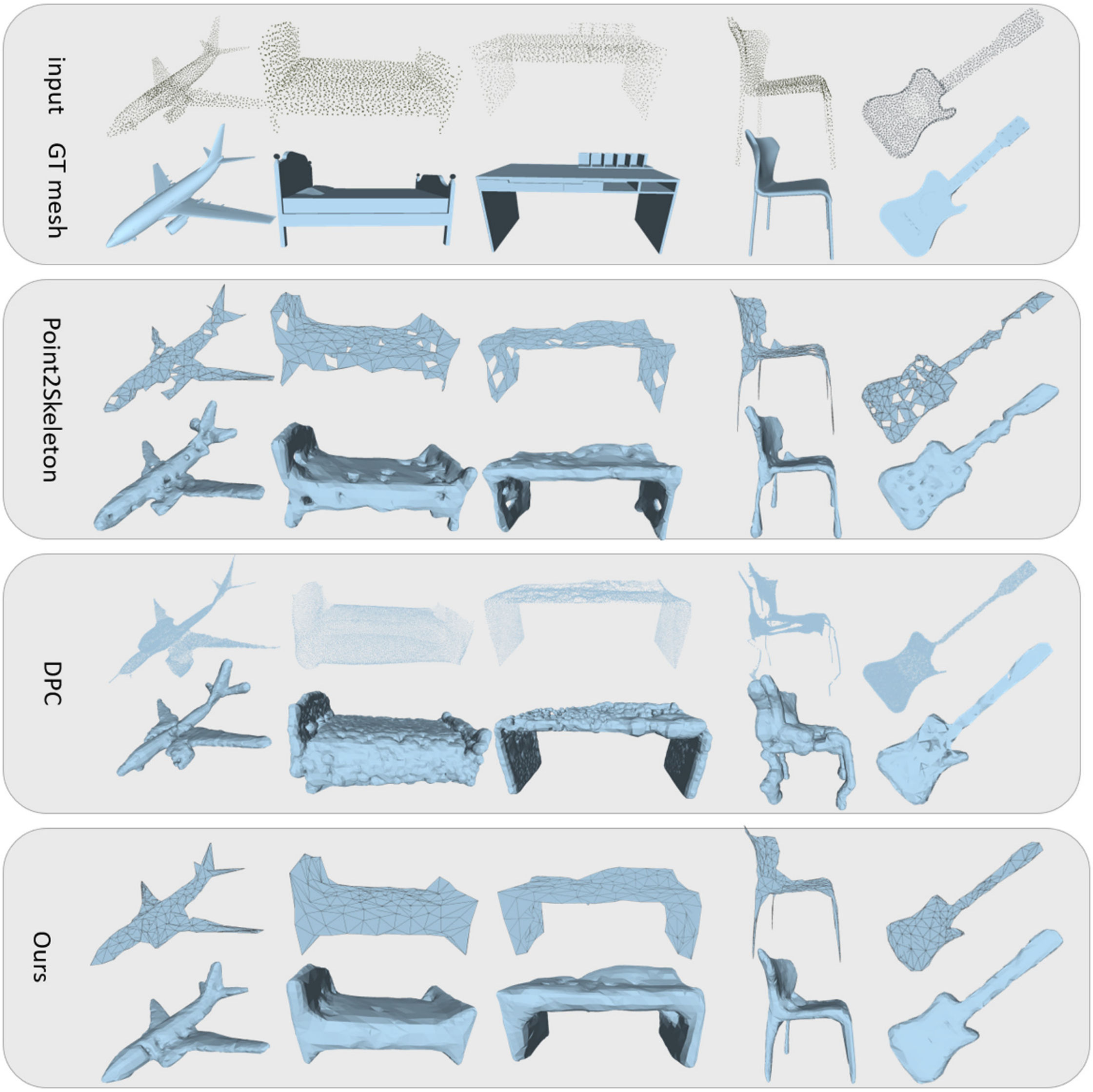


Fig. 8. Qualitative comparison with Point2Skeleton [7] and DPC [10]. The upper row shows the reconstructed MATs while the bottom row shows the reconstructed surface for each method. And the "GT mesh" represent the ground truth meshes.

Point2Skeleton and our method, we densely sample the spheres from the predicted or computed MATs. Fig. 8 shows the qualitative comparison with Point2Skeleton [7] and DPC [10], and Q-MAT+ [13]. The results show that our method could reconstruct the shape more accurately.

Besides, We compared our method with two approaches commonly used for computing the Medial Axis Transform (MAT) from point clouds. The first method is Power Crust [14], a sampling-based technique that takes point clouds as input and generates the MAT of the shape. The second method is to firstly reconstruct the surface from the point cloud, and compute the MAT of the reconstructed mesh. We use the Screened Poisson surface reconstruction (referred to as "SPR") [35] to reconstruct the mesh. To approximate the shape, we employed Q-MAT+[13] and MATFP method [12] to obtain a simplified MAT with 256 spheres.

However, the MATFP method [12] encountered difficulties in handling MAT calculations for the reconstructed models, as it struggled with numerous open boundaries. As shown in Fig. 9, the quality of the MAT calculated by Q-MAT+ depended on the result of the reconstructed mesh. Unfortunately, achieving a flawless manifold mesh with only 2000 points proved to be a challenging task. Additionally, Power Crust often computed a significant number of incorrect spheres, resulting in inaccurate connectivity.

We quantitatively compare our method with these two methods on Hausdorff distance for mesh reconstruction and Chamfer distance for MAT reconstruction on the same dataset. For Point2Skeleton, we conducted the post-processing while DPC generates only spheres without connectivity. The results in Table 1 and Fig. 8 show that our method could not only more accurately encode the information from the original input for reconstructing the shape with smaller Hausdorff error, but also produce more reasonable and accurate medial meshes except in the desk category. As shown in Fig. 8, although our method could compute more reasonable spheres than the other methods with the normal used for training. However, there are still some erroneous cones or slabs generated from the CP module, such as those between the legs and the desk surface, which results in large reconstruction errors and makes the reconstructed volume thicker than the ground truth.

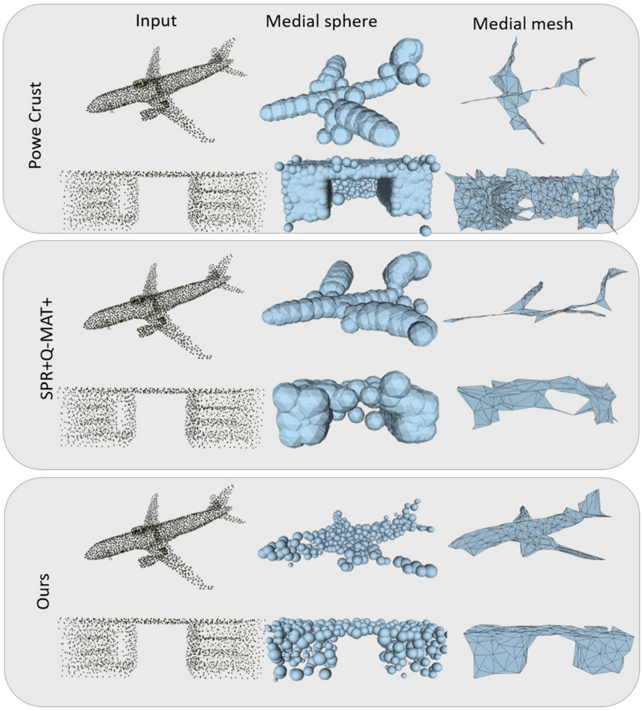


Fig. 9. Comparison with Power Crust on MAT computation with 2000 points as input and Q-MAT+ on MAT computation with SPR mesh.

Table 1
Comparison results with Point2Skeleton [7] and DPC [10] on reconstruction accuracy. For each method, the two rows report the CD loss and the HD loss respectively. The best results are in bold.

Methods	Chair	Airplane	Desk	Bed	Guitar
Point2Skeleton	1.12 1.13	1.30 1.32	**0.78** **0.83**	1.28 1.34	**0.86** 0.94
DPC	1.90 1.57	1.66 1.64	1.55 1.29	1.90 1.88	1.58 1.34
Ours	**1.01** **1.11**	**1.27** **1.22**	0.79 0.99	**1.13** **1.33**	0.89 **0.89**

4.2. Ablation study

We conduct a series of ablation studies to verify the various settings in our framework, and also some experiments to further explore the properties of the proposed method.

Medial sphere prediction. We use surface points with normals in the training. To validate the effectiveness of normal for MAT learning, we alternatively use the points without normal in training and analyze the quality of the predicted medial spheres. Accordingly, the surface loss in Eq. (6) is modified as follows:

$$\begin{aligned}\mathcal{L}_{sur} &= \mathcal{L}_{plane} + \mathcal{L}_{point}\\ &= \sum_{i=1}^{N}\sum_{t=1}^{2}(\|\mathbf{p}_i^t - \mathbf{c}_i\| - r_i)^2 \\ &+ \sum_{j=1}^{U}(\|\mathbf{p}_j - \mathbf{c}_j^{min}\| - r_j^{min})^2.\end{aligned} \tag{15}$$

Besides, the medial distance between input point $\mathbf{p}$ and medial primitive (medial cone $\mathbf{e}_{ij}$) for example, with its footprint sphere $(\mathbf{c}_p, r_p)$, is modified as follows:

$$d_{\mathbf{p}}^{ij} = |\|\mathbf{c}_p - \mathbf{p}\| - r_p| \tag{16}$$

Table 2
Quantitative comparison results on whether using normal for training or not. For each method, the two rows report the CD loss and the HD loss respectively. The best results are in bold.

Methods	Chair	Airplane	Desk	Bed	Guitar
w/o normal	1.12 1.14	1.30 1.36	**0.78** **0.89**	1.28 1.39	0.90 0.97
Ours	**1.01** **1.11**	**1.27** **1.22**	0.79 0.99	**1.13** **1.33**	**0.89** **0.89**

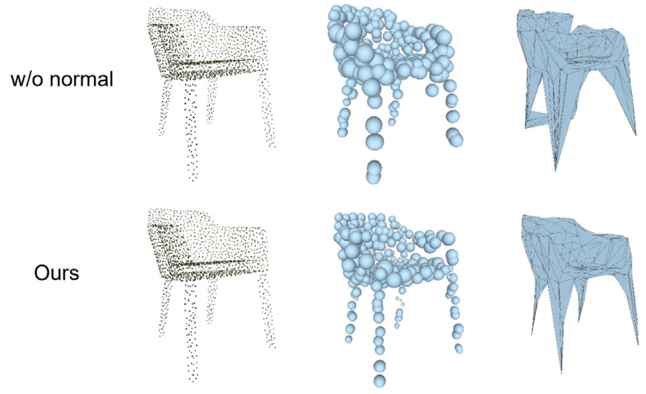


Fig. 10. Qualitative comparison results on whether using the normal for training or not. Results in the two rows are the predicted medial spheres and medial meshes without and with the normal vector used in training, respectively.

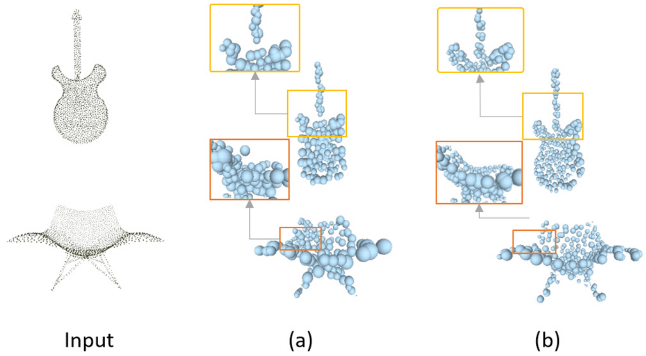


Fig. 11. Predicted medial spheres of (a) without and (b) with the pre-training in the medial spheres prediction stage(s).

The qualitative and quantitative results are shown in Fig. 10 and Table 2, respectively. The results in Fig. 10 show that with the normals used in training, the predicted spheres approximate the surface of the shape better and could lead to better connectivity. Besides, the Hausdorff distances on reconstructed meshes and the Chamfer distances on reconstructed MATs further validate the effectiveness of using the normals for training. Note that the normal vector is only used in training.

To validate the effectiveness of the pre-training stage, we conduct an ablation experiment that directly trains the MSP module by optimizing both the centers and the radii of medial spheres, with the pre-training stage. Results shown in Fig. 11 indicate that with the pre-training stage, the medial spheres could be evenly distributed in shape.

Connectivity prediction. In our method, KNN is used for generating the initial connectivity of medial mesh. We conduct an ablation study on the selection of the number K of neighboring spheres for connectivity initialization. Fig. 12 shows the average CD loss on all testing samples by selecting 8, 10, 12, 14, 16, 18, and 20 neighbors respectively, indicating that it has the smallest

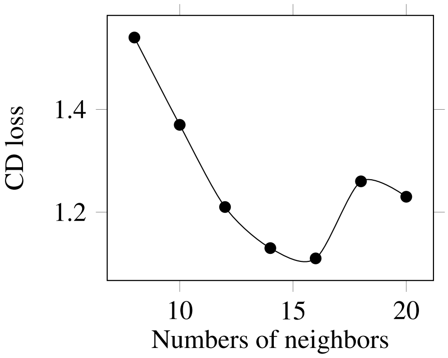


Fig. 12. Quantitative comparison results on CD loss with different numbers (8, 10, 12, 14, 16, 18, and 20) of neighbors used for initializing the connectivity.

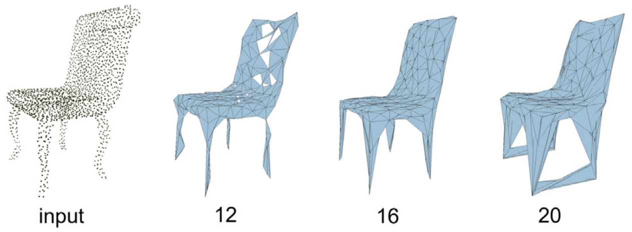


Fig. 13. Qualitative comparison results on using different numbers (12, 16, 20) of neighbors for initializing the connectivity.

Table 3
Quantitative comparisons with Delaunay triangulation and KNN for connectivity generation (the first 2 rows) and ablation study for the CP module (last 3 rows). For each method, the two rows report the CD loss and the HD loss respectively. The best results are in bold.

Methods	Chair	Airplane	Desk	Bed	Guitar
Delaunay Triangulation	1.08	1.29	0.92	**1.10**	0.97
	1.13	1.49	1.14	**1.14**	1.22
KNN	1.11	1.34	0.79	1.33	0.99
	1.33	1.65	1.37	1.58	1.35
w/o GAT	1.10	1.23	0.98	1.31	0.92
	1.12	1.33	0.88	1.42	0.94
w/o $\mathcal{L}_{slab}$	1.02	**1.22**	0.83	1.35	0.97
	1.11	1.34	0.93	1.33	0.89
Ours	**1.01**	1.27	**0.79**	1.13	**0.89**
	1.11	**1.22**	**0.99**	1.33	**0.89**

CD loss when using 16 neighbors. Results in Fig. 13 are consistent with Fig. 12 while using more neighbors will lead to the redundancy of medial primitives.

There are some other frequently used methods to generate mesh from a set of discrete points, for example, Delaunay triangulation (deleting the triangle faces with overlong edges using a threshold). We compare our method with Delaunay triangulation and KNN and report the errors in Table 3 (the first 2 rows). The results of the Delaunay triangulation show that selecting an appropriate threshold is vital to the reconstruction results. Our strategy for optimizing the initial connectivity computed with KNN, including introducing the GAT layers to capture the local graph of medial mesh and learning the complete connectivity, i.e., learning the slabs and the cones together by proposing the reconstruction error $\mathcal{L}_{recon}$ works better than Delaunay triangulation, especially for capturing the topology of shape.

To validate the effectiveness of the CP module, we present the errors without using the GAT layers (only the GCN layers) and without using the slab error term $\mathcal{L}_{slab}$ in Eq. (13), that is, using only the cone term $\mathcal{L}_{cone}$ for optimizing the connectivity in Table 3 (the last 3 rows) and Fig. 14. When comparing the results in Fig. 14, the strategies of using GAT layers to encode the local graph of the connectivity and taking the complete connectivity as output to better capture the local geometric details help to fill the holes and remove redundant medial cones and medial slabs. In addition, the results in Fig. 15 prove that the joint-training can optimize the radii and centers of the medial spheres and improve the reconstruction results.

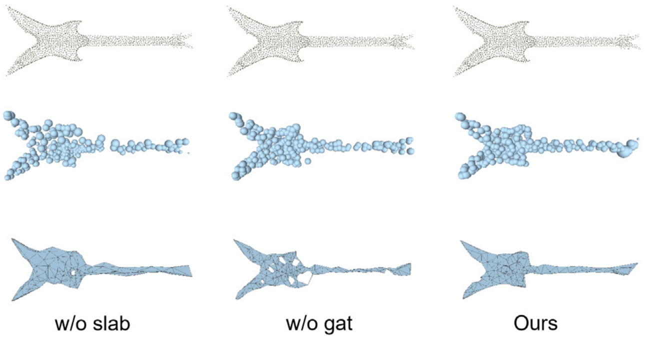


Fig. 14. Qualitative ablation study on different configurations of the CP module for connectivity prediction. Medial spheres and medial mesh are shown for each model.

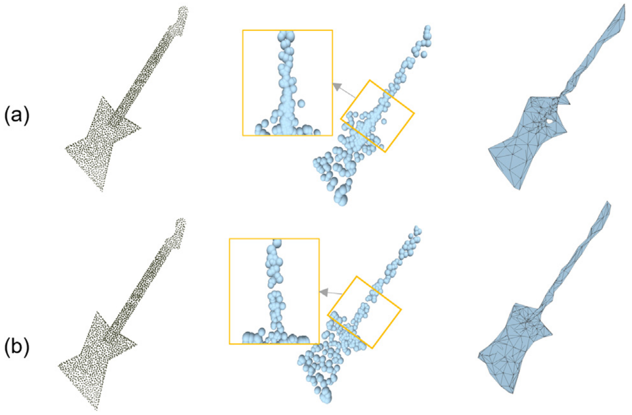


Fig. 15. Result of w/o jointly-training stage. (a) represents the predicted medial sphere and the predicted medial mesh before joint-training, while (b) shows the result after joint-training.

4.3. Shape classification with point clouds

The computed MAT can be used as an intermediate descriptor for downstream applications, such as 3D shape classification based on point clouds. In this section, we use our network to predict the MAT for each model in the training and testing sets of ModelNet40 (20 categories, 4879 models). We use MAT-NET to perform the shape classification task based on the our predicted MAT. We compare our method with PointNet++ [28]. We use the code provided by the authors on GitHub. For the PointNet++, we use the same point set as ours (2000 points). The grouping strategy of PointNet++ is the Multi-Scale Grouping (MSG) method (see Table 4).

5. Conclusion and future work

We propose *Point2MM*, a first unsupervised method for learning complete medial mesh from the point cloud. We first predict medial spheres by learning the geometric transformation of point clouds in the proposed MSP module, construct the connectivity

Table 4
The object classification accuracy on the same objects in the ModelNet40 dataset is reported as follows: "In" denotes the input data structure, and "Acc" represents the accuracy of point cloud classification.

Methods	In.	Acc
PointNet++	xyz	95.6
PointNet++	xyz+normal	96.3
Ours w/MAT-Net (sphere)	xyzr	97.1
Ours w/MAT-Net (sphere and connectivity)	xyzr+connectivity	**97.9**

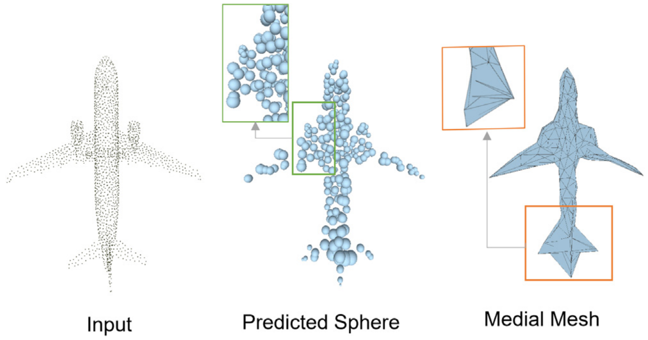


Fig. 16. Illustration of failure cases. Inside the green box: the connection of medial mesh is redundant. Inside the orange box: complex structure misleads the network to predict inappropriate spheres which results in an incorrect medial mesh.

of medial spheres by learning the probability of medial cones and medial slabs with the proposed CP module, and iteratively fine-tune the medial primitives with a joint-training of the MSP module and the CP module. Experiments show that our method effectively captures the underlying structures, as well as the fine details of shapes. We believe the predicted medial mesh can be used as an intermediate descriptor for downstream applications.

However, there is still space for improving the reconstruction accuracy. As shown in Fig. 16, there is still redundancy in the connectivity, e.g., the tail and wing of the plane. Besides, an incorrectly predicted sphere will lead to inaccurate connectivity. In the future, how to extracting medial mesh from a dynamic object or scene for 3D learning tasks, would be a promising direction. In addition, the excessive training time is also a concern for our research, and it is a limitation where we need to make improvements in our future work.

CRediT authorship contribution statement

Mengyuan Ge: Software, Conceptualization. **Junfeng Yao:** Funding acquisition, Project administration. **Baorong Yang:** Validation, Investigation, Writing – original draft. **Ningna Wang:** Writing – original draft, Visualization. **Zhonggui Chen:** Validation, Investigation, Writing – original draft. **Xiaohu Guo:** Validation, Investigation, Writing – original draft.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

Junfeng Yao and Mengyuan Ge were supported by the Natural Science Foundation of China (No. 62072388), the industry guidance project foundation of science technology bureau of Fujian province in 2020 (No. 2020H0047), and Fujian Sunshine Charity Foundation. The research of Zhonggui Chen was supported by the National Natural Science Foundation of China (No. 61972327) and Natural Science Foundation of Fujian Province (No. 2022J01001). Baorong Yang was supported by Natural Science Foundation of Fujian Province (No. 2021J05170), General Program Funded by Fujian Provincial Department of Education (No. JAT200286), Scientific Research Start-up Fund Project of Jimei University (No. ZQ2021002). Ningna Wang and Xiaohu Guo were partially supported by National Science Foundation (No. OAC-2007661).

References

[1] Hu J, Wang B, Qian L, Pan Y, Guo X, Liu L, et al. MAT-Net: Medial axis transform network for 3D object recognition. In: Proceedings of the twenty-eighth international joint conference on artificial intelligence. 2019, p. 774–81.

[2] Lin C, Liu L, Li C, Kobbelt L, Wang B, Xin S, et al. SEG-MAT: 3D shape segmentation using medial axis transform. IEEE Trans Vis Comput Graphics 2020.

[3] Yang B, Yao J, Guo X. DMAT: Deformable medial axis transform for animated mesh approximation. Comput Graph Forum 2018;37(7):301–11.

[4] Lan L, Luo R, Fratarcangeli M, Xu W, Wang H, Guo X, et al. Medial Elastics: Efficient and collision-ready deformation via medial axis transform. ACM Trans Graph 2020;39(3):1–17.

[5] Yang B, Yao J, Wang B, Hu J, Pan Y, Pan T, et al. P2MAT-NET: Learning medial axis transform from sparse point clouds. Comput Aided Geom Design 2020;80:101874.

[6] Li P, Wang B, Sun F, Guo X, Zhang C, Wang W. Q-MAT: Computing medial axis transform by quadratic error minimization. ACM Trans Graph 2015;35(1):1–16.

[7] Lin C, Li C, Liu Y, Chen N, Choi Y-K, Wang W. Point2Skeleton: Learning skeletal representations from point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, p. 4277–86.

[8] Blum H. A transformation for extracting new descriptions of shape. In: Models for the perception of speech and visual form. MIT Press; 1967, p. 362–80.

[9] Chazal F, Lieutier A. The "Lambda-Medial Axis". Graph Models 2005;67(4):304–31.

[10] Sun F, Choi Y-K, Yu Y, Wang W. Medial meshes – a compact and accurate representation of medial axis transform. IEEE Trans Vis Comput Graphics 2015;22(3):1278–90.

[11] Foskey M, Lin MC, Manocha D. Efficient computation of a simplified medial axis. J Comput Inf Sci Eng 2003;3(4):274.

[12] Wang N, Wang B, Wang W, Guo X. Computing medial axis transform with feature preservation via restricted power diagram. ACM Trans Graph 2022;41(6).

[13] Pan Y, Wang B, Guo X, Zeng H, Ma Y, Wang W. Q-MAT+: An error-controllable and feature-sensitive simplification algorithm for medial axis transform. Comput Aided Geom Design 2019;71:16–29.

[14] Amenta N, Choi S, Kolluri RK. The power crust. In: Proceedings of the sixth ACM symposium on solid modeling and applications. 2001, p. 249–66.

[15] Tagliasacchi A, Zhang H, Cohen-Or D. Curve skeleton extraction from incomplete point cloud. ACM Trans Graph 2009;28(3):1–9.

[16] Huang H, Wu S, Cohen-Or D, Gong M, Li G, Chen B. L1-medial skeleton of point cloud. ACM Trans Graph 2013;32.

[17] Mei J, Zhang L, Wu S, Wang Z, Zhang L. 3D tree modeling from incomplete point clouds via optimization and L1-MST. Int J Geogr Inf Sci 2017;31(5–6):999–1021.

[18] Qin H, Han J, Li N, Huang H, Chen B. Mass-driven topology-aware curve skeleton extraction from incomplete point clouds. IEEE Trans Vis Comput Graphics 2020;26(9):2805–17.

[19] Atienza R. Pyramid U-network for skeleton extraction from shape points. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops. 2019.

[20] Demir I, Hahn C, Leonard K, Morin G, Rahbani D, Panotopoulou A, et al. Skeleton 2019: Dataset and challenge on deep learning for geometric shape understanding. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2019.

[21] Yang L, Oyen D, Wohlberg B. A novel algorithm for skeleton extraction from images using topological graph analysis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2019.

[22] Liao Y, Donné S, Geiger A. Deep marching cubes: Learning explicit surface representations. In: IEEE/CVF conference on computer vision and pattern recognition. 2018.

[23] Shi B, Bai S, Zhou Z, Bai X. DeepPano: Deep panoramic representation for 3-D shape recognition. IEEE Signal Process Lett 2015;22(12):2339–43.

[24] Hanocka R, Hertz A, Fish N, Giryes R, Fleishman S, Cohen-Or D. MeshCNN: A network with an edge. ACM Trans Graph 2019;38(4):1–12.

[25] Hanocka R, Metzer G, Giryes R, Cohen-Or D. Point2Mesh: a self-prior for deformable meshes. ACM Trans Graph 2020;39(4).

[26] Wen C, Zhang Y, Li Z, Fu Y. Pixel2Mesh++: Multi-view 3D mesh generation via deformation. In: IEEE/CVF international conference on computer vision. 2019.

[27] Wei X, Chen Z, Fu Y, Cui Z, Zhang Y. Deep hybrid self-prior for full 3D mesh generation. In: IEEE/CVF international conference on computer vision. 2021, p. 5805–14.

[28] Qi CR, Yi L, Su H, Guibas LJ. PointNet++: Deep hierarchical feature learning on point sets in a metric space. Adv Neural Inf Process Syst 2017;30.

[29] Kipf TN, Welling M. Variational graph auto-encoders. In: NIPS workshop on Bayesian deep learning. 2016.

[30] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: International conference on learning representations. 2017.

[31] Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. In: International conference on learning representations. 2018.

[32] Lan L, Yao J, Huang P, Guo X. Medial-axis-driven shape deformation with volume preservation. Vis Comput 2017;33(6–8):1–12.

[33] Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, et al. 3D ShapeNets: A deep representation for volumetric shapes. In: IEEE/CVF conference on computer vision and pattern recognition. 2015, p. 1912–20.

[34] Cignoni P, Callieri M, Corsini M, Dellepiane M, Ganovelli F, Ranzuglia G, et al. MeshLab: an open-source mesh processing tool. In: Eurographics Italian chapter conference, vol. 2008. Italy: Salerno; 2008, p. 129–36.

[35] Kazhdan M, Hoppe H. Screened poisson surface reconstruction. ACM Trans Graph 2013;32(3):1–13.