P2MAT-NET: Learning Medial Axis Transform from Sparse Point Clouds

Baorong Yang[a], Junfeng Yao[a,*], Bin Wang[b], Jianwei Hu[b], Yiling Pan[b], Tianxiang Pan[b], Wenping Wang[c], Xiaohu Guo[d,*]

[a] *Xiamen University*
[b] *Tsinghua University*
[c] *University of Hong Kong*
[d] *The University of Texas at Dallas*

Abstract

The medial axis transform (MAT) of a 3D shape includes the set of centers and radii of the maximally inscribed spheres, and is a complete shape descriptor that can be used to reconstruct the original shape. It is a compact representation that jointly describes geometry, topology, and symmetry properties of a given shape. In this work, we present P2MAT-NET, a neural network which learns the pattern of sparse point clouds and transform them into spheres approximating MAT. The experimental results illustrate that P2MAT-NET demonstrates better performance than state-of-the-art methods in computing MAT from point clouds, in terms of MAT quality to approximate the 3D shapes. The computed MAT can be used as an intermediate descriptor for downstream applications such as 3D shape recognition from point clouds. Our results show that it can achieve competitive performance in recognition with state-of-the-art methods.

Keywords: Medial axis transform, Neural networks, Point clouds

1. Introduction

Medial axis transform (MAT) Blum (1967) of a 3D shape, consisting of the maximally inscribed spheres, is a compact and complete representation of the 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. Due to such compactness and expressiveness, MAT has been widely used in numerous applications Tagliasacchi et al. (2016) including shape recognition, shape editing, animation processing, etc. Recently, MAT has proved its excellent capability in shape recognition, even in the case of low resolution Hu et al. (2019). Researchers have been working on computing MAT from meshes Li et al. (2015); Pan et al. (2019); Rebain et al. (2019) or dense point clouds Amenta et al. (2001). However, it is still challenging to compute MAT on surfaces of arbitrary representations, especially sparse point clouds, limiting its broader adoption in various applications. Although by adding new points to sparse point clouds Yu et al. (2018b,a); Yifan et al. (2019), one can apply the existing methods to compute MAT by directly taking the upsampled point clouds as input or the reconstructed mesh from the upsampled point clouds, we are interested in a more direct way that taking the sparse point clouds as input. To the best of our knowledge, there are no prior works focusing on computing MAT from sparse point cloud.

As a fundamental shape representation, point cloud benefits from its easy access as the default output from most 3D shape acquisition devices. Recently, deep neural networks have been proposed to learn features of point clouds for shape classification and shape segmentation tasks Qi et al. (2017a,b); Li et al.

[*] Corresponding author
Email addresses: yangbr2011@gmail.com (Baorong Yang), yao0010@xmu.edu.cn (Junfeng Yao), wangbins@tsinghua.edu.cn (Bin Wang), hjw17@mails.tsinghua.edu.cn (Jianwei Hu), pyl16@mails.tsinghua.edu.cn (Yiling Pan), ptx9363@gmail.com (Tianxiang Pan), wenping@cs.hku.hk (Wenping Wang), xguo@utdallas.edu (Xiaohu Guo)

(2018b); Wang et al. (2019), shape analysis Guerrero et al. (2018); Ben-Shabat et al. (2019), as well as shape transformation tasks Berkiten et al. (2017); Tang et al. (2019). Specifically, P2P-NET Yin et al. (2018) transforms an input point set to an output point set with the same cardinality, e.g., skeletons, contours, cross sections, etc. This inspires us that neural network transformation is a potential way to compute a more compact representation of shape from point clouds, especially from sparse and noisy point cloud, for better approximating the shape.

In this paper, we are interested in exploring how deep neural network can benefit in transforming from sparse point clouds to MAT of the shape - a problem that has been considered as too challenging to be tackled. We propose a *Point-to-MAT displacement* network, called *P2MAT-NET*, which transforms an input sparse point cloud to an output set of spheres with the same cardinality by applying point-wise displacement vectors learned from the points to approximate the medial spheres. Furthermore, we construct the connectivity from the output spheres of P2MAT-NET, to form a piece-wise linear approximation of MAT, called *medial mesh* Sun et al. (2016). Evaluations of P2MAT-NET illustrate that it outperforms state-of-the-art methods for computing MAT from sparse point clouds, in terms of MAT quality. The computed MAT can be used as an intermediate descriptor for downstream applications such as 3D shape classification. Our results show that it can achieve competitive performance in shape classification with state-of-the-art methods.

2. Related Work

2.1. MAT Computation

There has been some existing techniques in computing MAT of 3D objects, as detailed in a recent survey Tagliasacchi et al. (2016). Here we focus on reviewing the methods that operate on 3D surface representations, especially point clouds and triangular meshes.

The Voronoi diagram can be used to compute MAT under the assumption that the input sample is sufficiently dense, and the medial axis is simply the vertices of the Voronoi diagram. Due to the existence of sliver tetrahedra, of which the centers of their circumspheres are far away from the medial axis, there is no guarantee that the medial axis could be approximated with a convergence guarantee Amenta and Bern (1999). Consequently, filtering the Voronoi diagram of the slivers is the most commonly-used method. The notable Voronoi poles Amenta and Bern (1999) are used to get a subset of Voronoi diagram by increasing the sampling density to remove as much slivers as possible. There are many methods following up this approach, such as the Power Crust Amenta et al. (2001), the Medial Meshes Sun et al. (2016) which extends the Sphere Meshes Thiery et al. (2013) to simplify the meshes of MAT, Q-MAT Li et al. (2015), as well as its extension Q-MAT+ Pan et al. (2019) which is feature-sensitive by adopting an error-bounded MAT simplification scheme. These methods suffer from two shortcomings: (1) the input sampling points should be sufficiently dense; (2) they always need some post-processing for removing the unstable branches of MAT.

Marrying the maximally inscribed definition of medial spheres is an alternative way of the Voronoi method to compute MAT Ma et al. (2012). LSMAT Rebain et al. (2019) follows this idea by taking a densely sampled oriented point set as input, and computing the approximation of its signed distance function (SDF) in the near/far field to obtain an approximated MAT.

Our P2MAT-NET offers a method for computing MAT from sparse point clouds. With the output of the network taken as the approximated medial spheres, we can construct their connectivity to form a medial mesh, which can be used for downstream applications such as 3D shape recognition.

2.2. Neural Networks for Processing Point Clouds

With the rapid development of neural networks, there has been quite a few recent approaches in learning features from 3D point clouds. PointNet Qi et al. (2017a) is the pioneering network that exploits the order invariance of point permutations and aggregates the per-point features into a global descriptor vector. Its extensions PointNet++ Qi et al. (2017b) and SO-NET Li et al. (2018a) apply hierarchy to process multi-scale point features. To overcome the price of lossing information with these symmetric pooling-based approaches, CNNs Wang et al. (2019); Li et al. (2018b); Wang et al. (2018) and graph convolutions Monti et al. (2017); Wang et al. (2018) are applied for processing point clouds for the tasks of shape classification and segmentation via learning the features of point sets.

Besides, there are some prior works focusing on point cloud upsampling. EC-Net Yu et al. (2018a) is a following work of PU-Net Yu et al. (2018b) which is the pioneer network working on point cloud upsampling. 3PU is a detail-driven deep neural network for point set upsampling.

P2P-NET Yin et al. (2018) is the first deep neural network designed to learn geometric transformations between point-based representations. Inspired by P2P-NET, we adopt a single directional branch of P2P-NET to learn the transformation from point clouds to the approximated medial spheres, by proposing a geometric loss between the supervising spheres and the learned spheres.

3. P2MAT-NET

The transformation from point cloud to medial spheres is difficult to model explicitly. For this reason, we propose P2MAT-NET, a Point-to-MAT displacement network, to transform an input point cloud $\mathcal{S} = \{\mathbf{p}\}$ to an output set of medial spheres $\bar{\mathcal{M}} = \{\bar{\mathbf{m}}\}$, where $\bar{\mathbf{m}} = (\bar{\mathbf{c}}, \bar{r})$ includes the center $\bar{\mathbf{c}}$ and radius $\bar{r}$ of spheres. The input and output have the same cardinality $|\mathcal{S}| = |\bar{\mathcal{M}}|$, by applying point-wise displacement vectors learned from the points to obtain the sphere centers and radii.

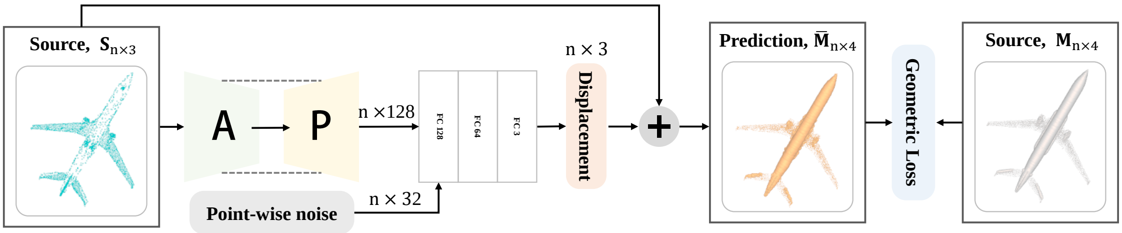

Figure 1: Network architecture of P2MAT-NET.

The architecture of P2MAT-NET is illustrated in Figure.1. The network first learns a multi-scale feature for each point, using layers of PointNet++ Qi et al. (2017b). As mentioned, MAT of a 3D shape consists of the maximally inscribed spheres. Local feature embedded in the k-nearest neighbors of each point aren't enough for capturing the maximal inscribed sphere, since at least the point on the other side of the shape and spanning the same sphere should also be captured for mapping to the displacement. The abstract operation of PointNet++ subsamples the input points and captures the point features at different scales arranged with ball query that finds all points within a radius to the query point with set abstraction layers (marked with A in Figure 1). Compared with kNN, ball query's local neighborhood guarantees a fixed region scale thus making local region feature more generalizable across space. A hierarchical propagation strategy is adopted to produce the point-wise multi-scale features with the feature propagation layers (marked with P in Figure 1). The propagation strategy randomly drop out input points with a randomized probability to prevent the network with training sets of various sparsity and varying uniformity, although there is no mathematical guarantee that the multi-scale local features captured are consistent with the definition of medial spheres. The multi-scale feature vectors are then concatenated with point-wise noise vectors. We keep the use of an independent Gaussian noise vector of length 32 for each noise vector and feed them to a set of fully connected layers that output displacement vectors $\mathcal{I}_{\mathbf{S}} = \{\mathbf{I}_{\mathbf{p}}\}$ for each input point in $\mathcal{S}$. The Gaussian noise vectors provide the system added degrees of freedoms to the displacements, to alleviate the problem that nearby points in the source shape tend to be mapped to similar displacement vectors due to the abstraction and propagation operations in PointNet++ Qi et al. (2017b). In the end, the network yields the centers of the predicted spheres $\bar{\mathbf{c}} = \mathbf{p} + \mathbf{I}_{\mathbf{p}}$, and the magnitudes of the displacements are the corresponding radii $\bar{r} = \|\mathbf{I}_{\mathbf{p}}\|$.

Our training set consists of spheres $\mathcal{M} = \{\mathbf{m}\}$ computed from the triangle meshes of ModelNet40 Wu et al. (2015) and the paired point set $\mathcal{S}$ sampled from the same meshes. The details for data set preparation will be discussed in a later subsection. The P2MAT-NET is trained under a supervised setting. In the training setting, we define the *footprint sphere* $\hat{\mathbf{m}} = (\hat{\mathbf{c}}, \hat{r}) \in \mathcal{M}$ of point $\mathbf{p} \in \mathcal{S}$ as:

$$\hat{\mathbf{m}} = \text{argmin}_{\mathbf{m} \in \mathcal{M}} \lambda(\mathbf{p}, \mathbf{m}), \tag{1}$$

where $\lambda(\mathbf{p}, \mathbf{m})$ is the signed distance from point $\mathbf{p}$ to the surface of sphere $\mathbf{m} = (\mathbf{c}, r)$, i.e., $\lambda(\mathbf{p}, \mathbf{m}) = \|\mathbf{p} - \mathbf{c}\| - r$.

To measure the geometric difference between the predicted spheres $\bar{\mathcal{M}}$ and the target spheres $\mathcal{M}$ from training set, the network is trained with a geometric loss to penalize the mismatch between them. The geometric loss is computed as the sum of differences by searching the footprint sphere $\hat{\mathbf{m}} \in \mathcal{M}$ for each input point:

$$L(\bar{\mathcal{M}}, \mathcal{M}) = \sum_{\bar{\mathbf{m}} \in \bar{\mathcal{M}}} \delta(\bar{\mathbf{m}}, \hat{\mathbf{m}}), \tag{2}$$

where $\delta(\bar{\mathbf{m}}, \hat{\mathbf{m}})$ measures the distance between the predicted sphere $\bar{\mathbf{m}}$ of point $\mathbf{p}$ and its footprint sphere $\hat{\mathbf{m}}$:

$$\delta(\bar{\mathbf{m}}, \hat{\mathbf{m}}) = \sqrt{\|\bar{\mathbf{c}} - \hat{\mathbf{c}}\|^2 + (\bar{r} - \hat{r})^2}. \tag{3}$$

Due to the limit of space, we will discuss the selection of the L2 norm for measuring the mismatch between $\bar{\mathbf{m}}$ and $\hat{\mathbf{m}}$ in the appendix.

The geometric loss $L(\bar{\mathcal{M}}, \mathcal{M})$ is minimized with an Adam optimizer, with the learning rate being set as 1e-3 and decays to 1e-4 at discrete intervals during training.

3.1. Connectivity between Medial Spheres

In this subsection, we propose a strategy for computing the connectivity of medial spheres. We use the CGAL package “Delaunay Triangulation 3” The CGAL Project (2019) to get the Delaunay Triangulation of the centers of spheres, and produce the initial connectivity of medial axis, then adaptively remove the extra edges using the following algorithm:

1. Remove all the invalid edges $\{e_{ij}\}$ that satisfy the following condition:

$$\|\mathbf{c}_i - \mathbf{c}_j\|^2 - (r_i - r_j)^2 < 0. \tag{4}$$

2. Compute the length l_{ij} of each valid edge $\mathbf{e}_{ij}$ connecting medial spheres $\mathbf{m}_i$ and $\mathbf{m}_j$, as the following tangential length between two spheres:

$$l_{ij} = \sqrt{\|\mathbf{c}_i - \mathbf{c}_j\|^2 - (r_i - r_j)^2}. \tag{5}$$

3. Compute the average length $\bar{l}$ of all remaining valid edges:

$$\bar{l} = \frac{\sum_{e_{ij} \in E} l_{ij}}{n_e}, \tag{6}$$

   where E is the remaining valid edge set, and n_e is the number of the remaining valid edges.
4. Remove all the remaining edges that satisfy $l_{ij} > \omega_1 \bar{l}$.
5. Repeat steps 3∼4 with N_1 number of iterations.
6. For each sphere $\mathbf{m}$, compute the average length $d_{\mathbf{m}}$ of the incidental edges $\mathcal{N}(\mathbf{m})$.
7. Remove all the remaining edges that satisfy $l_{ij} > \omega_2 d_{\mathbf{m}_i}$ or $l_{ij} > \omega_2 d_{\mathbf{m}_j}$.
8. Repeat steps 6∼7 with N_2 number of iterations.
9. If both the degrees of two medial spheres in edge $\mathbf{e}_{ij}$ are larger than a specified maximal degree ϕ, ascendingly sort the incidental edges of medial spheres $\mathbf{m}_i$ and $\mathbf{m}_j$ with the length of edge l_{ij}, and set the standard lengths s_i and s_j of $\mathbf{m}_i$ and $\mathbf{m}_j$ to be the lengths of their ϕ^{th} incidental edge, finally remove all the remaining edges with length larger than s_i and s_j.

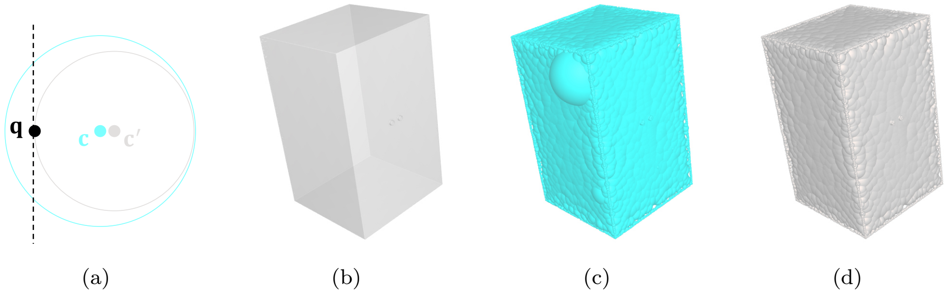

Figure 2: MAT reparation. (a) Illustration of MAT reparation. (b) The surface mesh and (c) the initial MAT spheres computed by Q-MAT+Pan et al. (2019) are fixed by pushing the extruded spheres into the mesh, and output (d) the repaired spheres for training.

In this paper, we fix all the parameters in constructing the connectivity of MAT by setting $\omega_1 = 2.5$, $N_1 = 1$, $\omega_2 = 1.8$, $N_2 = 5$, and $\phi = 8$. Note that in step 2, the tangential distance between two spheres is used for removing the extra edges. This is because MAT approximates the shape with the envelop of connected spheres. It is different from the use of geometric loss in Eq. 3, where the 4D Euclidean distance between a predicted sphere and the footprint sphere of the corresponding point is used to penalize the mismatch between them.

3.2. Dataset Preparation

We prepare the dataset from the meshes of ModelNet40 Wu et al. (2015), and will make them publicly available. The input of P2MAT-NET are the medial spheres of the MAT objects computed by Q-MAT+ Pan et al. (2019), with the tool provided by the authors. For each shape, the number of spheres includes 2048, 1024, 512 and 256. Note that Q-MAT+ needs densely sampled, closed manifold mesh as the input, but the majority of 3D models in ModelNet40 do not satisfy these requirements. So we use the repaired mesh provided by the authors of MAT-Net Hu et al. (2019), and successfully compute 9728 valid MAT objects in 40 categories, about 79.02% of the original ModelNet40. We use 7774 shapes for training and 1954 shapes for testing. For Q-MAT+ Pan et al. (2019), we use the default setting (0.5σ, where σ is the average SDF values) to initialize the most important parameter θ, which determines thin parts. The number of valid MAT objects in our dataset is less than the number of shapes in ModelNet40-MAT Hu et al. (2019), which is caused by the open mesh in the repaired set, and there could be some extruded spheres or external spheres in the computed medial meshes. The medial meshes with large spheres outside the surface are excluded from the dataset. According to the definition of MAT, a medial sphere should not bound any surface point in it. So we also repaired the MAT objects computed by Q-MAT+ by pushing the extruded spheres back into the surfaces. As shown in Figure 2(a), the dashed line represents the surface. When an inner most projecting point $\mathbf{q}$ of the sphere center $\mathbf{c}$ on the mesh is found, the cyan sphere is updated by reducing its radius and moving its center along the direction from $\mathbf{q}$ to $\mathbf{c}$ until the sphere is tangent to the surface mesh, illustrated as the gray sphere $\mathbf{c}'$. The strategy is conducted in an iterative way until the sphere becomes fully inside the surface, and the iteration number is set to be 2 for all objects. Figure 2(b)–2(d) shows an example of MAT reparation.

After the medial spheres of MAT objects are computed, the paired point set is sampled from the same mesh as follows: for each sphere of a shape, the surface vertex with the smallest signed distance to the sphere is sampled and included into the paired point set. All medial spheres and sampled points are normalized into a unit ball. Some subsets in ModelNet40 are augmented by random rotation of each paired data to increase their sizes by 3 to 10 times. To ensure a fair comparison, all compared methods in this paper are running on the same 79.02% objects of ModelNet40.

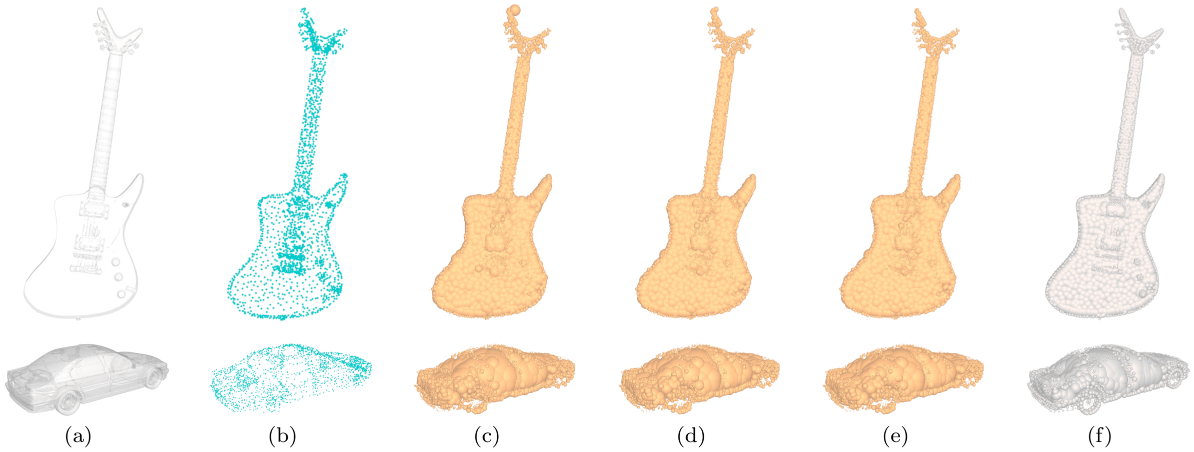

Figure 3: Testing results on different types of shapes with 2048 points as input. (a) The surface mesh, (b) the testing point cloud, (c) the predicted spheres of P2MAT-NET, (d) the refined predicted spheres with the sphere-bounding strategy, (e) the refined predicted spheres with both the normal-refinement strategy and the sphere-bounding strategy, (f) ground truth spheres.

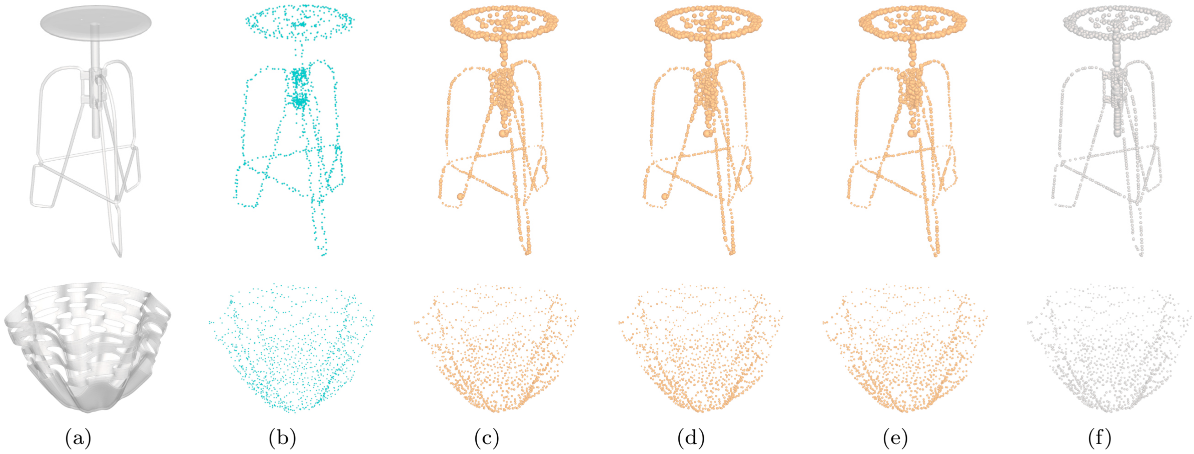

Figure 4: Testing results on different types of shapes with 1024 points as input. (a) the surface mesh, (b) the testing point cloud, (c) the predicted spheres of P2MAT-NET, (d) the refined predicted spheres with the sphere-bounding strategy, (e) the refined predicted spheres with both the normal-refinement strategy and the sphere-bounding strategy, (f) ground truth spheres.

4. Experiments and Applications

In this section, we show experiments to demonstrate the capability of P2MAT-NET in learning geometric transforms from point cloud to the approximated MAT, as well as to apply the predicted MAT for 3D object classification.

4.1. Evaluation of P2MAT-NET

For our prepared dataset, the network is trained for 200 epoches on a Nvidia Titian XP GPU and it takes approximately 3∼10 hours to finish the training process. The time for training is related to the number of predicted spheres as well as the number of training sets. During the testing phase, the surface point set (of sizes 2048, 1024, 512, 256 respectively) is fed to the network in one pass and the predicted sphere set of the same size is obtained. Note that the network is trained using data with each number of points separately.

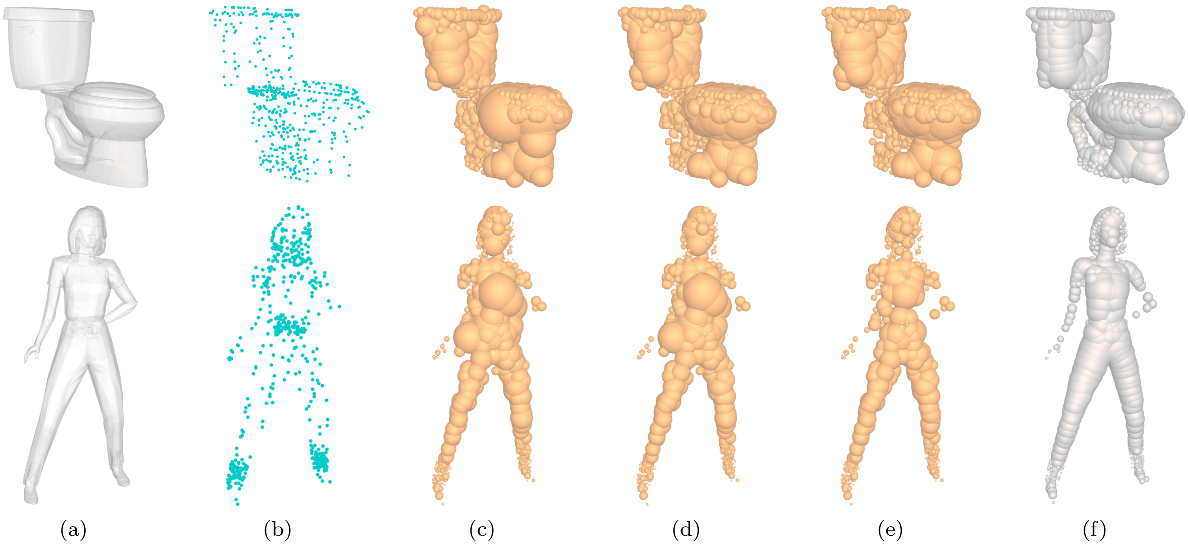

Figure 5: Testing results on different types of shapes with 512 points as input. (a) the surface mesh, (b) the testing point cloud, (c) the predicted spheres of P2MAT-NET, (d) the refined predicted spheres with the sphere-bounding strategy, (e) refined predicted spheres with both the normal-refinement strategy and the sphere-bounding strategy, (f) ground truth spheres.

We visualize the results obtained on different types of shapes in Figure 3–5. It can be observed that the transformation results can capture the principal characteristics of these shapes.

The predicted spheres which are the output of the network could be refined by a strategy similar to the MAT reparation in our data preparation. We assume that no surface point should be bounded inside any sphere. For a predicted sphere $\bar{\mathbf{m}}$, we iteratively find the point $\mathbf{q}$ which is bounded by the sphere and is the farthest point to the surface of $\bar{\mathbf{m}}$ to minify the sphere. We denote $\bar{\mathbf{m}}' = (\bar{\mathbf{c}}', \bar{r}')$ as the shrunk sphere. The sphere is shrunk by the following *sphere-bounding strategy*: firstly shrink the sphere by decreasing its radius, $\bar{r}' = \max(\bar{r} + \lambda(\mathbf{q}, \bar{\mathbf{m}})/2, 0)$, and use the direction of displacement $\mathbf{I_p}$ to compute the new center: $\bar{\mathbf{c}}' = \mathbf{p} + \frac{\mathbf{I_p}}{\|\mathbf{I_p}\|}\bar{r}'$, where point $\mathbf{p}$ is the corresponding point in the input point set. The testing results in the fourth columns of Figure 3–5 are the refined spheres using this sphere-bounding strategy. Result on the toilet object in Figure 5 shows that the sphere-bounding strategy could optimize the extruded spheres by pushing them back to the surface w.r.t. the sampled surface points, to approximate the medial spheres by the constraint that they should be inside the object.

If the normal of the point cloud is known a priori, we can refine the predicted spheres by the following *normal-refinement strategy*: firstly mark the predicted sphere $\bar{\mathbf{m}}$ when the angle between the normal of corresponding surface point $\mathbf{p}$ and the corresponding displacement $\mathbf{I_p}$ is smaller than $\pi/2$, denoted as the *normal-displacement condition*. Next, find the closest surface point $\mathbf{q}$, which is in the k nearest neighbor set of $\mathbf{p}$ and whose corresponding predicted sphere does not violate the normal-displacement condition, and use the displacement of $\mathbf{q}$ to update $\bar{\mathbf{m}}$ as long as the updated sphere does not violate the normal-displacement condition. The testing results in the fifth columns of Figure 3 − 5 are the refined spheres using both the sphere-bounding strategy as well as the normal-refinement strategy. The result on the person object in Figure 5 demonstrates the effectiveness of the normal-refinement strategy, considering the result of Figure 5(d) and Figure 5(e) on the person object.

According to the definition of MAT, the sphere-bounding strategy should always be used for the post-processing, while the normal-refinement strategy is only adopted when the normal of the point cloud is a priori.

Results in Figure 6(a)–6(c) demonstrate that our method could learn MAT from point clouds with different sizes: 256 / 512 / 1024 / 2048. The training pair of (points, spheres) is sampled from the same

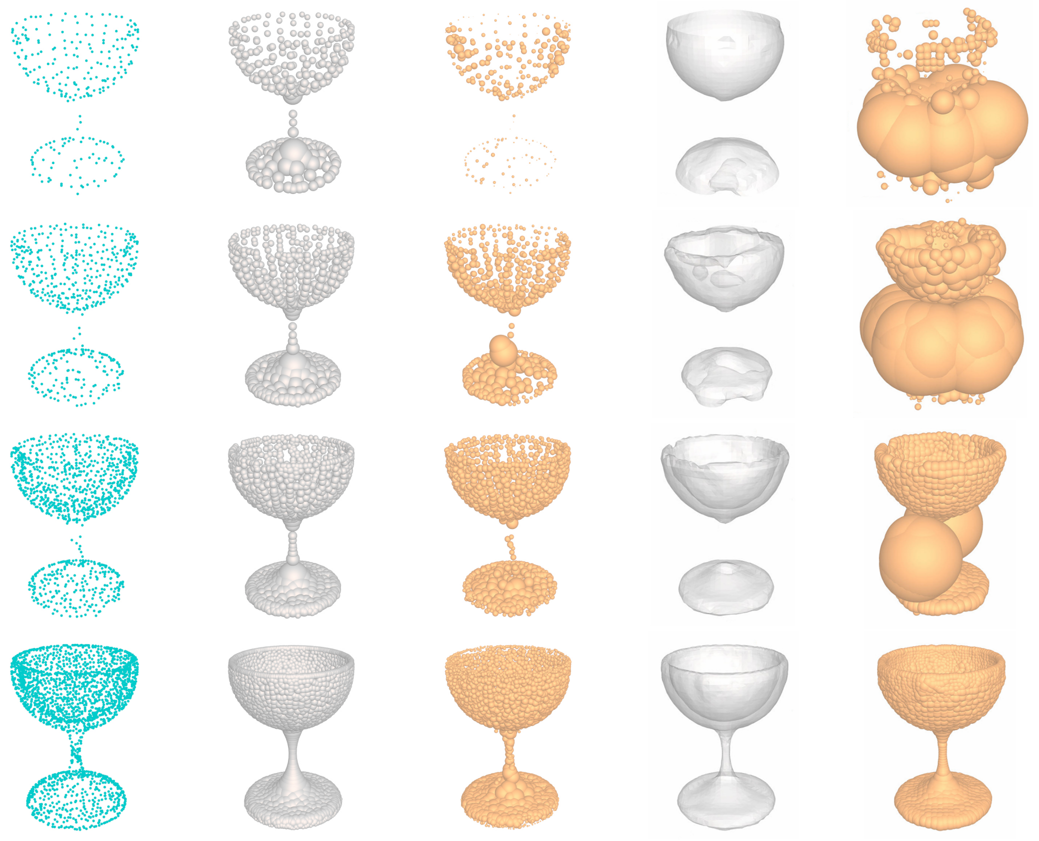

Figure 6: Compute MAT from point clouds with sizes of 256, 512, 1024 and 2048 (from top to bottom). (a) the point clouds, (b) the ground truth spheres, (c) the refined spheres with sphere-bounding and normal-refinement strategies, (d) the SPR reconstructed mesh with (a), (e) the result of Q-MAT+ from the surface of (d).

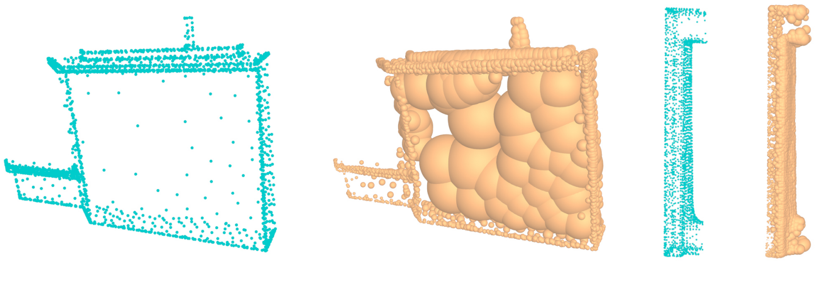

Figure 7: Testing results on incomplete point clouds of 2048 points sampled from sub-domains of meshes. (a) and (c) testing point clouds, (b) and (d) predicted spheres corresponding to (a) and (c) respectively.

shape. To further demonstrate that the network has learned a proper transformation, we test the trained model on incomplete point clouds (Figure 7), point clouds sampled from simulated scan data (Figure 8), as well as noisy point clouds (Figure 9). Figure 7 shows the testing results on incomplete and non-uniformly sampled point clouds sampled from two different meshes in Figure 7(a) and Figure 7(c) and the corresponding predicted spheres in Figure 7(b) and Figure 7(d) respectively. The incomplete point cloud is sampled by

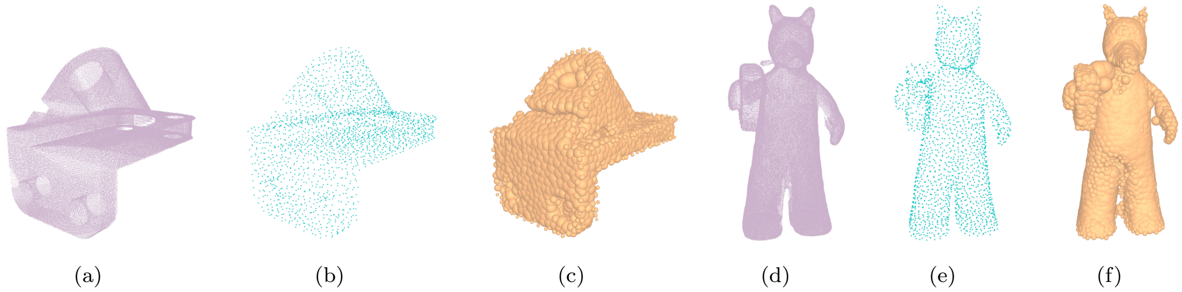

Figure 8: Testing results on point clouds with 2048 points sampled from BSR Berger et al. (2013). (a) and (d) the simulated scan data in BSR, (b) and (e) the testing point clouds sampled from (a) and (d) respectively, (c) and (f) refined predicted spheres with the sphere-bounding strategy.

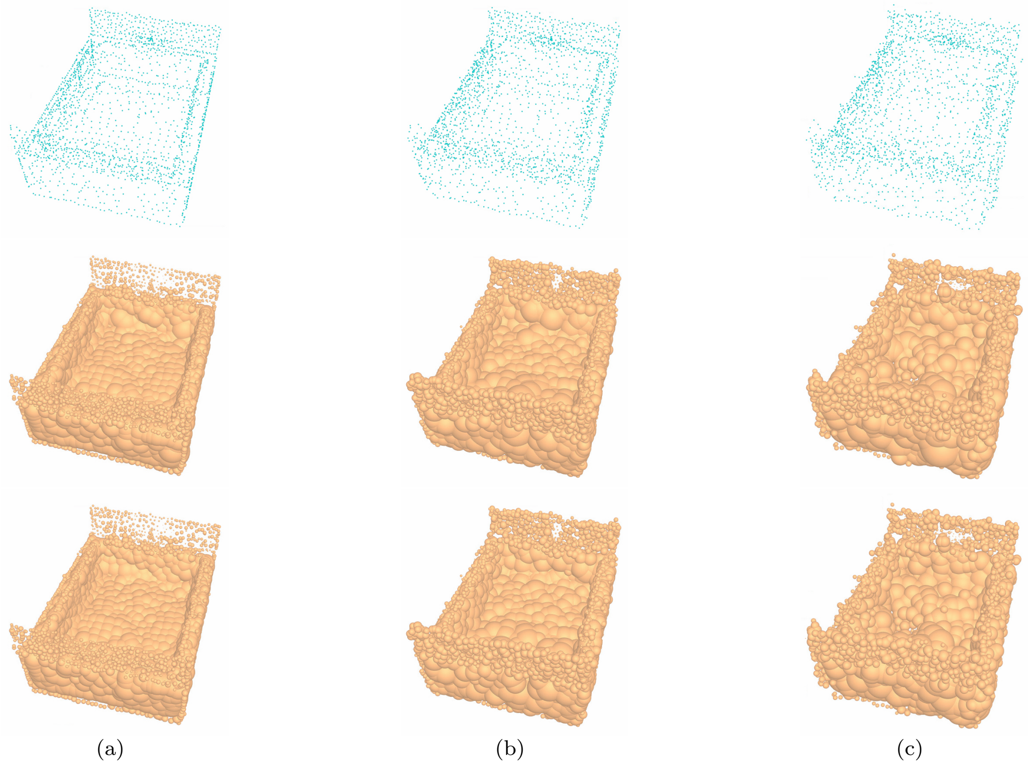

Figure 9: Testing results on noisy point clouds with 2048 points. From left to right, the results (a) without noise, (b) with low noise level points, (c) with high noise level points. From top to bottom are the testing point clouds, the predicted spheres and the refined predicted spheres with the sphere-bounding strategy.

setting the boundary of point clouds and randomly sample 2048 points. Scan data could be noisy and biased with larger error pointing toward the sensor. To demonstrate that the network could also work on scan data, we test the trained model on point clouds sampled from the simulated scan data in the benchmark for surface reconstruction (denoted as "BSR" in the following section) Berger et al. (2013)), as well as noisy point clouds. Figure 8 shows the results on the 2048 points sampled from BSR. It is noticed that the simulated scan data could be incomplete, for example the holes in the anchor model as shown in Figure 8(a) and Figure 8(d). For the noisy point clouds, we first sample the paired point set of the 2048 medial spheres, and move each point with a random perturbation with a random scale in the interval of [0,0.03] (Figure

9(b)) or [0,0.06] (Figure 9(c)) along its normal direction. The result shows that our P2MAT-NET is stable for handling even incomplete or noisy point clouds.

4.2. Computing MAT from Point Clouds

We compare our method with the following two state-of-the-art methods which have been used to compute MAT from point clouds. One is Power Crust Amenta et al. (2001) which is a sampling-based method and can take point clouds as input and output 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 Poisson surface reconstruction method Kazhdan et al. (2006), Screened Poisson surface reconstruction (denoted as "SPR") Kazhdan and Hoppe (2013), GR method Lu et al. (2018) which is based on the modified Gauss formula to get a watertight and smooth surface from the point cloud equipped with normal information and the VIPSS method Huang et al. (2019) which reconstructs an implicit surface from an un-oriented point set. When the mesh is reconstructed from the point cloud, we use Q-MAT+ Pan et al. (2019) to obtain the simplified MAT with 2048 spheres to approximate the shape. However, the VIPSS method Huang et al. (2019) fails at reconstructing meshes for all the paired point clouds with medial spheres computed with Q-MAT+ Pan et al. (2019). As demonstrated by the authors, the approach for initializing the optimization isn't guaranteed to work for all inputs (particular those exhibiting a varying level of noise and complexity) Huang et al. (2019). We denote the methods using the Poisson method Kazhdan et al. (2006) and the SPR method Kazhdan and Hoppe (2013) for mesh reconstruction and then adopting Q-MAT+ Pan et al. (2019) for MAT computation as "Poisson & Q-MAT+" method and "SPR & Q-MAT+" method in the following comparison and take point clouds with 2048 points as the input.

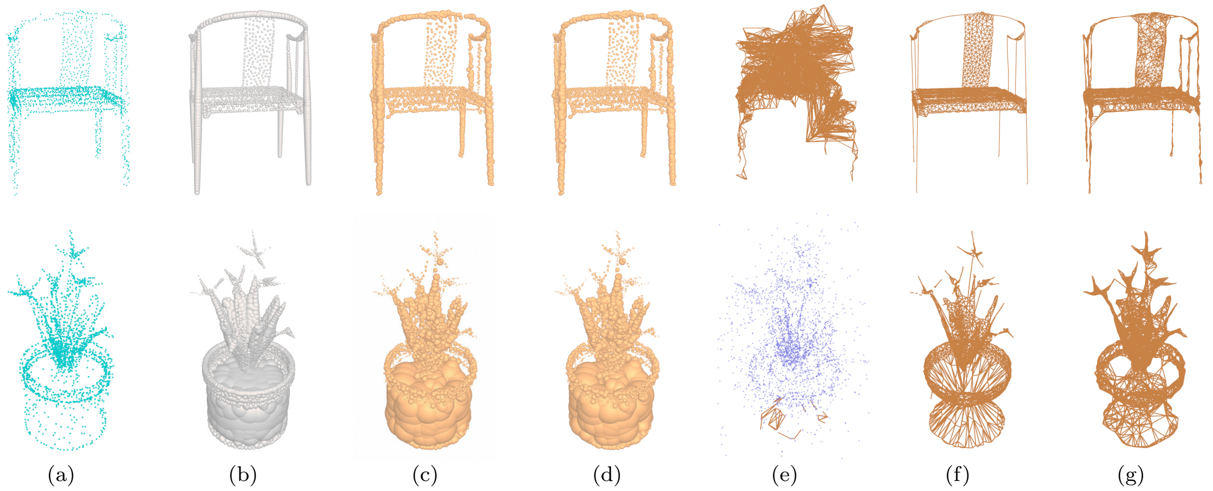

Figure 10: Comparison with Power Crust on MAT computation with 2048 points as input. (a) The testing point cloud, (b) the ground truth medial spheres, (c) our predicted spheres, (d) our refined predicted spheres with the sphere-bounding strategy, (e) the computed medial edges of Power Crust, (f) our computed medial edges from spheres in (d), and (g) the ground truth medial edges.

To evaluate the approximation accuracy, we use the two-sided Hausdorff distance, denoted as ϵ. For the ground truth MAT and the other methods with computed MAT for mesh approximation, we use the two-sided Hausdorff distance in DMAT Yang et al. (2018), between the ground truth mesh and the enveloping surface of MAT. For fair comparison, only the objects that both the Poisson & Q-MAT+ method and the SPR & Q-MAT+ method succeed in computing the MAT in the testing set are measured. For the GR method Lu et al. (2018), we measure the error between the ground truth mesh and the reconstructed mesh.

For the Power Crust method, we set the sampling density constant as 0.6, the parameter deciding whether to propagate the same label ("inside" or "out-side") to neighboring poles as 0.4, and multiplier as

100000. We use the implementation of Power Crust in VTK environment to compute the medial axis from input point clouds. In Figure 10, the sphere-bounding strategy is applied to refine the output spheres of P2MAT-NET, and our results are compared with the Power Crust method. It is noticed that there are some isolated spheres with their centers outside the object in the output medial axis of the Power Crust due to the sparseness of the point cloud, so we only show their computed medial edges. Besides, the computed edges in Figure 10(e) have large error with the ground truth. For example, the MAT of the flower pot computed by the Power Crust method has only a few valid medial edges. The centers of computed medial spheres in purple are also shown for a clear illustration.

For the Poisson & Q-MAT+ method and the SPR & Q-MAT+ method, we use MeshLab v1.3.2 and v2016.12 respectively to reconstruct the surface with the default parameters. Since the reconstructed mesh may not satisfy the strict input requirement of Q-MAT+ Pan et al. (2019), the Poisson & Q-MAT+ method and the SPR & Q-MAT+ may fail in computing MAT for some objects. The output medial axis with no edges or faces is marked as invalid, and the Poisson method finally succeed in about 85% of the objects we used, of which 6549 objects in the training set and 1693 in the testing set, while the SPR method succeed in about 98% of the objects we used, of which 7625 objects in the training set and 1909 in the testing set.

	airplane	bathtub	bed	desk	guitar	chair	monitor	night.	person
GR	2.60	11.34	8.43	12.34	1.98	9.13	9.07	6.66	**2.19**
Poisson & Q-MAT+	10.63	15.57	16.27	24.41	5.15	21.84	14.40	14.52	14.25
SPR & Q-MAT+	3.50	13.19	12.32	20.85	2.92	12.01	15.22	12.94	4.46
P2MAT-NET	1.53	4.69	4.60	4.82	2.52	4.03	3.54	6.10	4.74
P2MAT-NET-S	1.28	3.97	3.80	4.67	1.55	2.50	3.26	5.16	4.52
P2MAT-NET-N-S	**1.23**	**3.43**	**3.65**	**4.40**	**1.32**	**2.53**	**2.67**	**4.63**	3.00

Table 1: Average Hausdorff errors of different classes of objects reconstructed from 2048 points, normalized with respect to the diagonal length of the corresponding surface, and represented in percentage. “P2MAT-NET” represents the results computed with the predicted spheres of P2MAT-NET, “-S” is by applying the sphere-bounding strategy, and “-N” is by applying the normal-refinement strategy. The best results of each type of objects are in bold font.

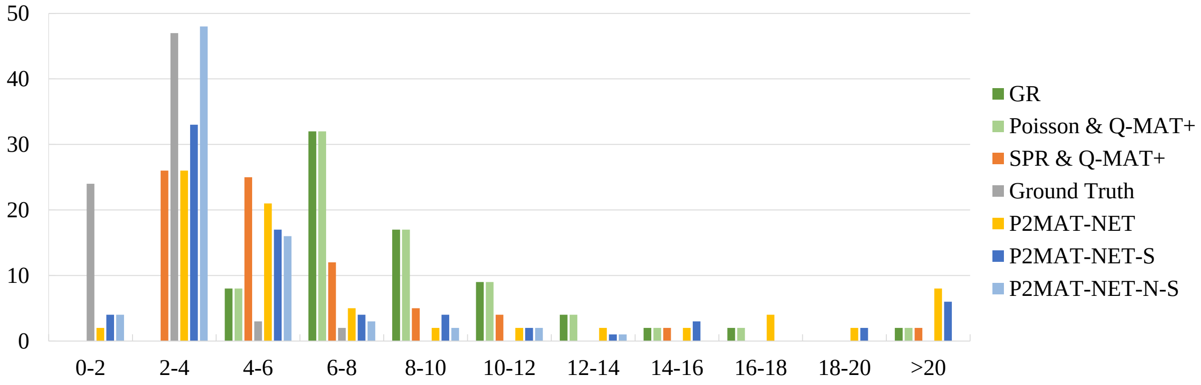

Figure 11: Error distribution of reconstructed mesh or computed MAT for the models of beds in the testing set.

For the GR method, we use the code provided by the authors to reconstruct mesh from the point set. However, most of the meshes reconstructed with the GR method Lu et al. (2018) could not be used for computing MAT with Q-MAT+ Pan et al. (2019) because there is no guarantee that they are closed and manifold.

Table 1 shows the average Hausdorff distance of different types of objects, and it reveals that our methods have much smaller Hausdorff distance than the Poisson & Q-MAT+ method, the SPR & Q-MAT+ method on all classes of objects, and the GR method on most classes except person. Note that the Hausdorff distances are normalized with respect to the diagonal length of the corresponding surface, and represented

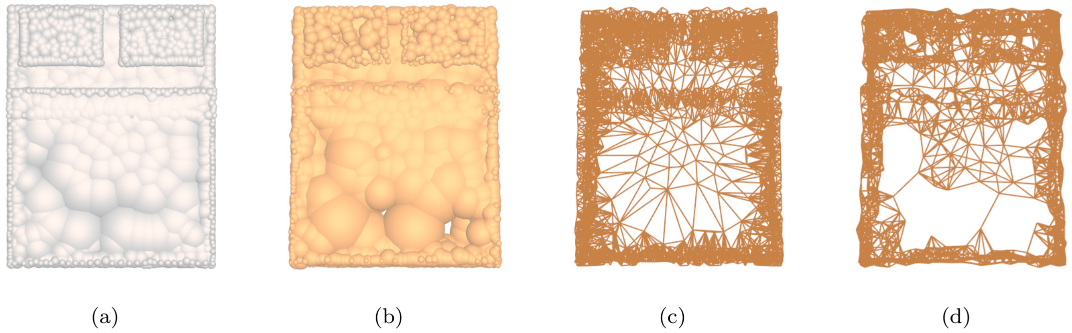

Figure 12: Holes in the our predicted MAT with 2048 points. (a) The ground truth spheres, (b) refined predicted spheres with both sphere-bounding and normal-refinement strategies, (c) the ground truth edges, (d) our constructed edges from spheres in (b).

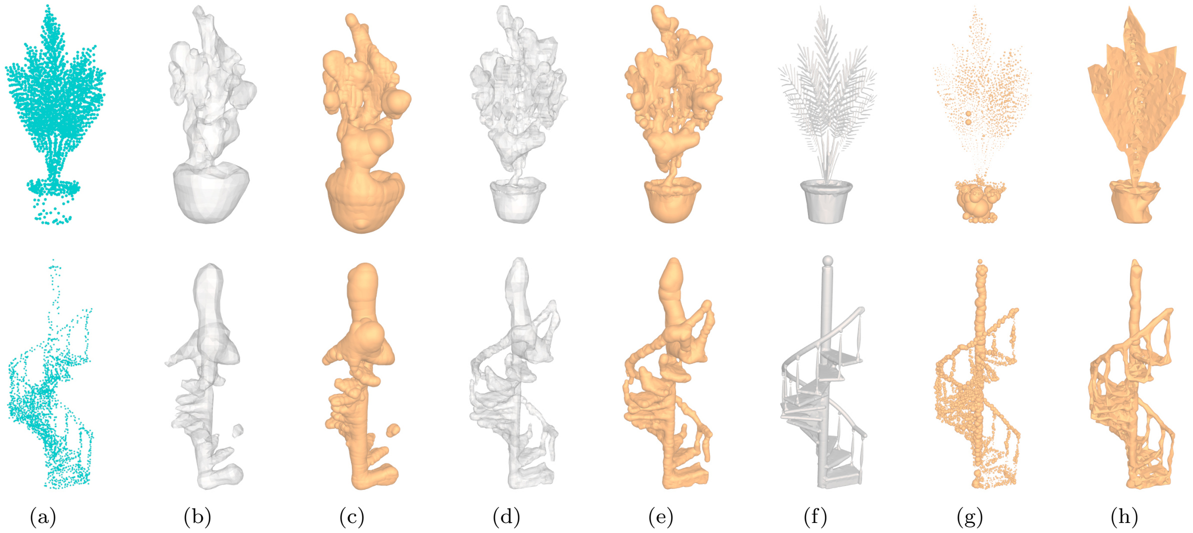

Figure 13: Comparison with the Poisson & Q-MAT+ method and the SPR & Q-MAT+ method with 2048 points as input. (a) The testing point cloud, (b) the Poisson reconstructed mesh, (c) the result of Q-MAT+ from the surface of (b), (d) the SPR reconstructed mesh, (e) the result of Q-MAT+ from the surface of (d), (f) the ground truth MAT, (g) the refined predicted spheres of P2MAT-NET with both the sphere-bounding and normal-refinement strategies, (h) the computed MAT with connectivity from spheres in (g).

in percentage. Figure 11 illustrates the error distribution of computed MAT for the models of beds in the testing set. One of the reasons that some objects have large error for our methods is because the connectivity of the MAT causes some large holes in local domain, as shown in Figure 12. We have also compared the average Hausdorff distance with these two methods on different resolutions. Due to the limit of space, please see the results in the appendix for more details. For the Poisson & Q-MAT+ method and the SPR & Q-MAT+ method, the large error is mainly caused by the poor quality mesh reconstructed from the sparse point samples, as illustrated in Figure 6 and Figure 13. Figure 14 shows the comparison results with the GR method.

Table 2 shows the average Hausdorff errors of our predicted MAT in different resolutions. These results demonstrate that our network is able to learn proper transformations and obtain MAT with small errors even from sparse point clouds.

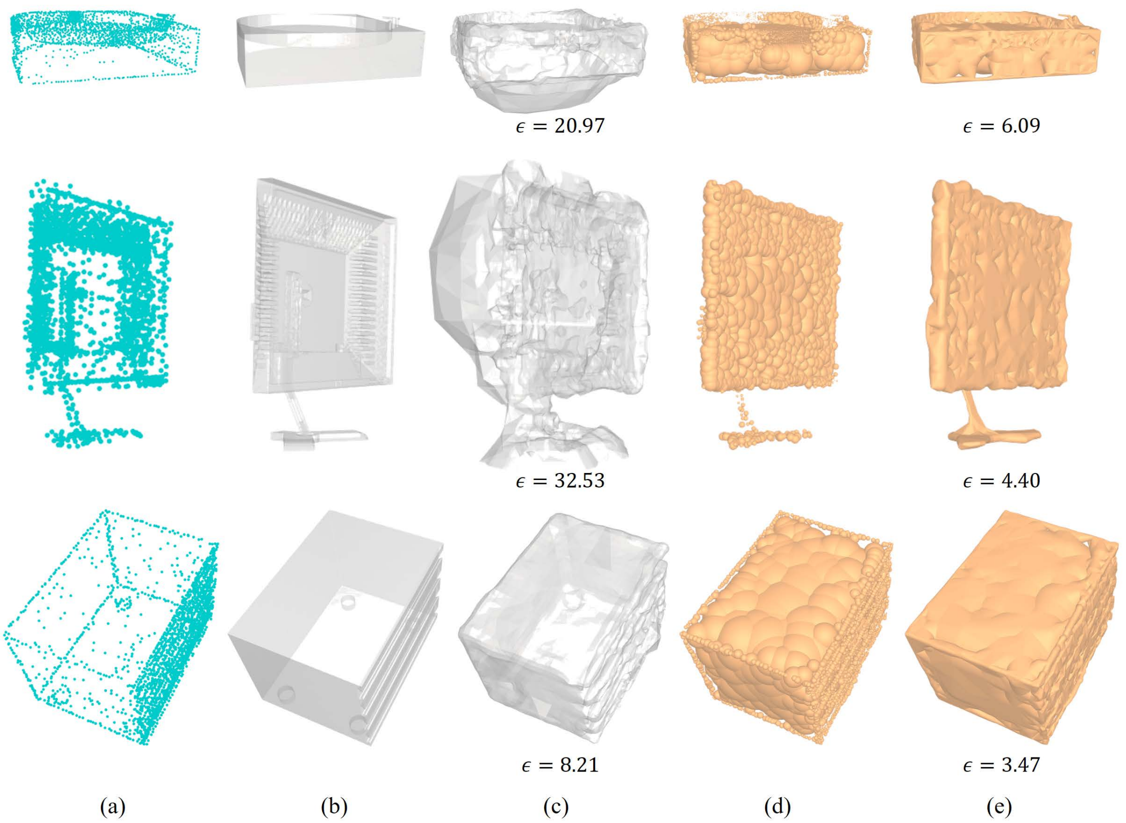

Figure 14: Comparison with the GR method on mesh reconstruction with 2048 points as input. (a) The testing point cloud, (b) the surface mesh, (c) reconstructed mesh with the GR method, (d) the refined predicted spheres with sphere-bounding strategy, (e) the computed MAT with connectivity from spheres in (d).

	airplane	bed	chair	sofa	guitar	monitor	person
256	3.42 / 1.22	6.25 / 2.47	4.40 / 1.84	5.12 / 2.50	2.16 / 0.92	4.15 / 1.73	4.67 / 2.31
	5.20 / 4.39	13.06 / 12.80	11.45 / 7.26	7.60 / 6.07	4.00 / 2.44	10.98 / 6.24	6.78 / 5.13
512	2.48 / 1.03	5.11 / 2.13	3.38 / 1.62	4.24 / 2.16	1.71 / 0.78	3.19 / 1.54	3.20 / 1.94
	3.10 / 2.85	10.69 / 5.86	8.92 / 5.50	6.03 / 5.75	3.12 / 2.03	9.28 / 4.73	4.65 / 4.90
1024	1.81 / 0.88	3.99 / 1.80	2.46 / 1.31	3.52 / 1.93	1.28 / 0.69	2.50 / 1.42	2.32 / 1.89
	2.10 / 1.64	8.04 / 5.27	5.52 / 3.74	4.75 / 4.79	1.86 / 1.75	5.51 / 4.17	3.28 / 4.05
2048	1.07 / 0.38	2.00 / 1.14	1.23 / 0.68	2.52 / 0.98	0.65 / 0.34	1.17 / 0.70	1.74 / 1.43
	0.98 / 1.17	2.50 / 3.61	1.78 / 2.43	2.49 / 3.95	1.04 / 1.29	1.98 / 2.65	2.05 / 2.97

Table 2: Average Hausdorff errors of different classes of objects, learning from point clouds with different sizes (256,512,1024 and 2048). For each size, the two lines represent the errors of MATs from (1st line) ground truth, (2nd line) our method of P2MAT-NET-N-S, respectively. "# / #" represent the errors of "Surface to MAT / MAT to Surface".

4.3. 3D Shape Classification from Point Clouds

The computed MAT can be used as an intermediate descriptor for downstream applications such as 3D shape classification from point clouds. In this section, we use the computed MAT of the 79.02% objects in ModelNet40 in shape classification task by using MAT-Net Hu et al. (2019). We generate MAT for those point clouds in the testing set, and use the sphere-bounding and normal-refinement strategies to optimize the predicted spheres, then construct the connectivity of MAT with the default parameters. Here we denote our method as “Ours w/ MAT-Net”.

We compare our method with state-of-the-art methods: PointNet Qi et al. (2017a), PointNet++ Qi et al. (2017b), PointCNN Li et al. (2018b), O-CNN Wang et al. (2017) and DGCNN Wang et al. (2019). We use the codes of these methods provided by their authors on Github. For PointNet, PointNet++, PointCNN and DGCNN, we use the same point sets as ours. The grouping method of PointNet++ is multi-scale grouping(MSG). The PointCNN is trained on the same setting of network structure and hyper parameters configured in modelnet_x3_l4. For O-CNN, we generate the octree data with the original meshes with the default octree depth 6. The resolution of leaf octants is 64^3.

Method	Rep.	#In	Acc.	#In	Acc.	#In	Acc.	#In	Acc.
O-CNN(6)	octree	64^3	89.5	-	-	-	-	-	-
PointNet	xyz	256	86.1	512	86.9	1024	87.8	2048	87.3
PointNet++(MSG)	xyz	256	86.1	512	87.1	1024	88.3	2048	88.8
	xyz+normal	256	87.1	512	88.9	1024	90.4	2048	90.2
PointCNN	xyz+normal	256	87.8	512	88.3	1024	89.8	2048	**91.3**
DGCNN	xyz+normal	256	87.8	512	88.3	1024	89.9	2048	-
Ours w/ MAT-Net(sphere)	xyzr	256	90.8	512	91.2	1024	**91.0**	2048	91.2
Ours w/ MAT-Net(3 features)	xyzr	256	**91.8**	512	**91.7**	1024	90.9	2048	91.0

Table 3: Object classification accuracy on the same 79.02% objects of ModelNet40 dataset. “#In” represents the number of point cloud except that it is the resolution of leaf octants for O-CNN. “Acc.” represents the overall accuracy.

Table 3 compares the classification performance of different methods. The object classification results of all methods are computed without voting. “MAT-Net (3 features)” in the table represents using the spheres, the Edge Index Matrix, and the Edge Mask Matrix computed by the K nearest neighbors ($K = 16$) of spheres by using MAT-Net Hu et al. (2019). We can see that even using only the spheres, MAT-Net (spheres) can get an accuracy better than PointNet, PointNet++, O-CNN and DGCNN with different sizes of point cloud. The results of PointNet, PointNet++ and DGCNN are not as good as the results reported in the corresponding papers Qi et al. (2017a,b), which may be caused by the non-uniformly and sparsely sampled point sets that we use. Compared with using only the spheres, adding the local edge features to MAT-Net does not improve the overall accuracy when using 1024 or 2048 spheres for each shape, because of the redundant information residing in the topology of high resolution MAT, as compared to the best classification results of using 256 spheres in MAT-Net Hu et al. (2019). This further proves the competitive advantage of MAT by using less information in shape classification tasks.

5. Discussion and Future Work

In this paper, we propose a neural network, namely P2MAT-NET, to transform sparse point clouds to spheres approximating MAT of the 3D shape. To the best of our knowledge, this is the first method that can compute MAT from point clouds of such sparsity, i.e., only 256 / 512 / 1024 / 2048 points for each object. P2MAT-NET is effective in learning the pattern of point cloud, and stable even for transforming incomplete or noisy point clouds. Despite the fundamental breakthrough, there are still several limitations in our current method, which we plan to address in the future.

Firstly, although our P2MAT-NET demonstrates the best performance in computing MAT from point clouds, the way for constructing the connectivity of MAT is only a simple attempt. There is no guarantee that our connectivity construction can either re-produce or converge to the ground truth. We would like to explore this challenging geometric and topological problem in the future.

Secondly, MAT is naturally a very compact representation of the surface, and it does not need to have the same cardinality as the number of sampling points on surfaces. As shown in MAT-Net Hu et al. (2019), MAT with 256 spheres can already provide very good classification accuracy for 3D shapes. How to learn medial spheres of less cardinality than the size of point set, instead of through the one-to-one displacement, is another interesting topic for the future.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable suggestions. The paper is supported by the Xiamen Municipal Bureau of Science and Technology in 2018(No. 3502Z20184058), the Natural Science Foundation of Fujian Province in 2019 (No.2019J01601), the foreign cooperation project of Fujian Provincial Department of Science and Technology in 2018(No. 2018I0015) and the creation fund project of Fujian Provincial Department of Science and Technology in 2018(No. 2019C0021).

References

Amenta, N., Bern, M., 1999. Surface reconstruction by Voronoi filtering. Discrete & Computational Geometry 22, 481 – 504.

Amenta, N., Choi, S., Kolluri, R.K., 2001. The power crust, in: Proceedings of ACM Symposium on Solid Modeling, pp. 244 – 266.

Ben-Shabat, Y., Lindenbaum, M., Fischer, A., 2019. Nesti-net: Normal estimation for unstructured 3d point clouds using convolutional neural networks, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10112–10120.

Berger, M., Levine, J.A., Nonato, L.G., Taubin, G., Silva, C.T., 2013. A benchmark for surface reconstruction. ACM Transactions on Graphics 32, 20:1–20:17.

Berkiten, S., Halber, M., Solomon, J., Ma, C., Li, H., Rusinkiewicz, S., 2017. Learning detail transfer based on geometric features. Computer Graphics Forum 36, 361–373.

Blum, H., 1967. A transformation for extracting new descriptors of shape. Models for the Preception of Speech & Visual Form 19, 362–380.

Guerrero, P., Kleiman, Y., Ovsjanikov, M., Mitra, N.J., 2018. PCPNet: Learning local shape properties from raw point clouds. Computer Graphics Forum 37, 75–85. doi:10.1111/cgf.13343.

Hu, J., Wang, B., Qian, L., Pan, Y., Guo, X., Liu, L., Wang, W., 2019. MAT-Net: Medial axis transform network for 3d object recognition, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'19), pp. 774–781.

Huang, Z., Carr, N., Ju, T., 2019. Variational implicit point set surfaces. ACM Transactions on Graphics 38, 124:1–124:13.

Kazhdan, M., Bolitho, M., Hoppe, H., 2006. Poisson surface reconstruction, in: Proceedings of the Eurographics Symposium on Geometry Processing (SGP'06), pp. 61–70.

Kazhdan, M., Hoppe, H., 2013. Screened poisson surface reconstruction. ACM Transactions on Graphics 32, 29:1–29:13.

Li, J., Chen, B.M., Lee, G.H., 2018a. SO-Net: Self-organizing network for point cloud analysis, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18), pp. 9397–9406.

Li, P., Wang, B., Sun, F., Guo, X., Zhang, C., Wang, W., 2015. Q-MAT: Computing medial axis transform by quadratic error minimization. ACM Transactions on Graphics 35.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018b. PointCNN: Convolution on x-transformed points, in: Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS'18), pp. 828–838.

Lu, W., Shi, Z., Sun, J., Wang, B., 2018. Surface reconstruction based on the modified gauss formula. ACM Transactions on Graphics 38, 2:1–2:18.

Ma, J., Bae, S.W., Choi, S., 2012. 3D medial axis point approximation using nearest neighbors and the normal field. The Visual Computer 28, 7 – 19.

Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., Bronstein, M.M., 2017. Geometric deep learning on graphs and manifolds using mixture model CNNs, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17), pp. 5425–5434.

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

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17), pp. 652–660.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space, in: Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS'17), pp. 5105–5114.

Rebain, D., Angles, B., Valentin, J.P.C., Vining, N., Peethambaran, J., Izadi, S., Tagliasacchi, A., 2019. LSMAT least squares medial axis transform. Computer Graphics Forum .

Sun, F., Choi, Y.K., Yu, Y., Wang, W., 2016. Medial meshes – a compact and accurate representation of medial axis transform. IEEE Transactions on Visualization and Computer Graphics 22, 1278–1290.

Tagliasacchi, A., Delame, T., Spagnuolo, M., Amenta, N., Telea, A., 2016. 3D skeletons: A state-of-the-art report. Computer Graphics Forum 35, 573 – 597.

Tang, J., Han, X., Pan, J., Jia, K., Tong, X., 2019. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single RGB images, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19), pp. 4541–4550.

The CGAL Project, 2019. CGAL User and Reference Manual. 4.14 ed., CGAL Editorial Board. URL: https://doc.cgal.org/4.14/Manual/packages.html.

Thiery, J.M., Guy, E., Boubekeur, T., 2013. Sphere-meshes: Shape approximation using spherical quadric error metrics. ACM Transactions on Graphics 32, 1.

Wang, C., Samari, B., Siddiqi, K., 2018. Local spectral graph convolution for point set feature learning, in: Proceedings of the European Conference on Computer Vision (ECCV'18), pp. 56–71.

Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y., Tong, X., 2017. O-CNN: Octree-based convolutional neural networks for 3d shape analysis. ACM Transactions on Graphics 36.

Wang, S., Suo, S., Ma, W., Pokrovsky, A., Urtasun, R., 2018. Deep parametric continuous convolutional neural networks, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18), pp. 2589–2597.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019. Dynamic graph CNN for learning on point clouds. ACM Transactions on Graphics (TOG) .

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3D ShapeNets: A deep representation for volumetric shapes, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15), pp. 1912 – 1920.

Yang, B., Yao, J., Guo, X., 2018. DMAT: deformable medial axis transform for animated mesh approximation. Computer Graphics Forum 37, 301–311.

Yifan, W., Wu, S., Huang, H., Cohen-Or, D., Sorkine-Hornung, O., 2019. Patch-based progressive 3d point set upsampling, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5951–5960.

Yin, K., Huang, H., Cohen-Or, D., Zhang, H., 2018. P2P-NET: Bidirectional point displacement net for shape transform. ACM Transactions on Graphics 37.

Yu, L., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A., 2018a. Ec-net: an edge-aware point set consolidation network, in: ECCV, pp. 386–402.

Yu, L., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A., 2018b. Pu-net: Point cloud upsampling network, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2790–2799.

Appendix A. Data Augmentation Factor

Some subsets in ModelNet40 are augmented by random rotation of each paired data to increase their sizes by 3 to 10 times. Table A.4 shows the augmentation factors for each subset of ModelNet40.

Fac.	airplane	bathtub	bed	bench	bottle	bookshelf	bowl	car	night.	cone
1	√		√		√	√				
3										
5				√				√		
10		√					√		√	√
Fac.	cup	curtain	desk	door	dresser	flower.	piano	guitar	keyboard	lamp
1										
3							√			
5			√		√			√	√	
10	√	√		√		√				√
Fac.	laptop	mantel	monitor	chair	person	glass.	plant	radio	range.	sink
1			√	√						
3		√								
5						√				
10	√				√		√	√	√	√
Fac.	sofa	stairs	stool	table	tent	tv stand	toilet	vase	wardrobe	xbox
1	√			√			√			
3						√				
5										
10		√	√		√			√	√	√

Table A.4: Augmentation factors for each subset in ModelNet40.

Appendix B. Selection of the Geometric Loss

We use the L2 norm of predicted sphere $\bar{\mathbf{m}}$ of input point $\mathbf{p}$ and its footprint sphere $\hat{\mathbf{m}}$ to measure the mismatch between them, other than the L1 norm between them. That's because that when using the L2 norm, the corresponding Hausdorff errors of computed MATs are smaller than using the L1 norm, as shown in Table B.5. "# / # " represent the errors of "P2MAT-NET" and "P2MAT-NET-N-S".

	airplane	chair
L1 norm	1.83 / 1.59	3.44 / 3.18
L2 norm	1.53 / 1.23	4.00 / 2.53

Table B.5: Average Hausdorff errors with different geometric loss on 2048 spheres.

Appendix C. Results on Computing MAT from Point Cloud

In this section, we show more results of P2MAT-NET for computing MAT from point cloud of different sizes (256, 512, 1024 and 2048 respectively), and more comparison results with the Power Crust method, the Poisson & Q-MAT+ method, the SPR & Q-MAT+ method and the QR method.

Appendix C.1. On the Size of Point Cloud

We test P2MAT-NET on point clouds with sizes of 256, 512, 1024 and 2048 respectively. Figure C.15–C.18 demonstrate that P2MAT-NET is able to learn proper transformations and obtain MAT even from sparse point clouds.

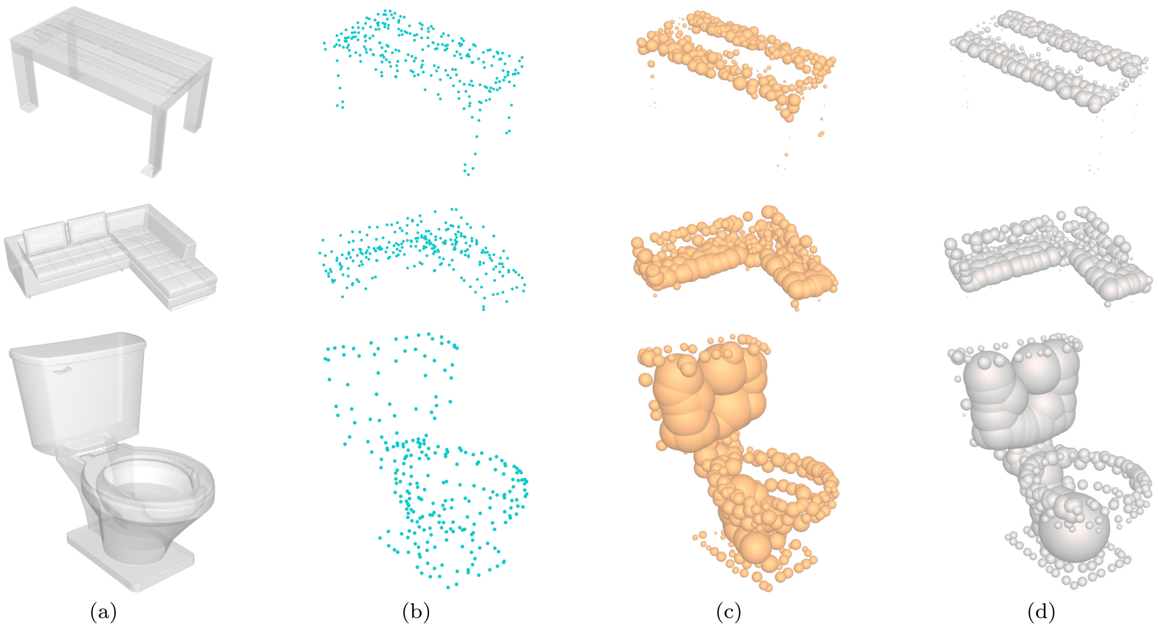

Figure C.15: More testing results on different types of shapes with 256 points as input. (a) The surface mesh, (b) testing point cloud, (c) the refined predicted spheres with both the normal-refinement strategy and the sphere-bounding strategy, (d) ground truth spheres.

Appendix C.2. Comparison with Power Crust

We compare our method with Power Crust, which is a sampling-based method, on point cloud of size 2048. Taking the point cloud as the input of P2MAT-NET, we predict spheres with the same cardinality and refine them with the sphere-bounding strategy. Figure C.19 illustrates the comparison result on a guitar object. As mentioned, Power Crust could generate many isolated large spheres outside the object. Due to the limit of paper length, we show the result in this supplementary document. For a complete show of the computed MAT with Power Crust, Figure 19(c) and 19(d) are rendered in a different view from the other sub-figures. By comparing the result in Figure 19(d) and Figure 19(g), it is noticed that there are some isolated spheres with their centers outside the object. Besides, the computed edges in Figure 19(i) have

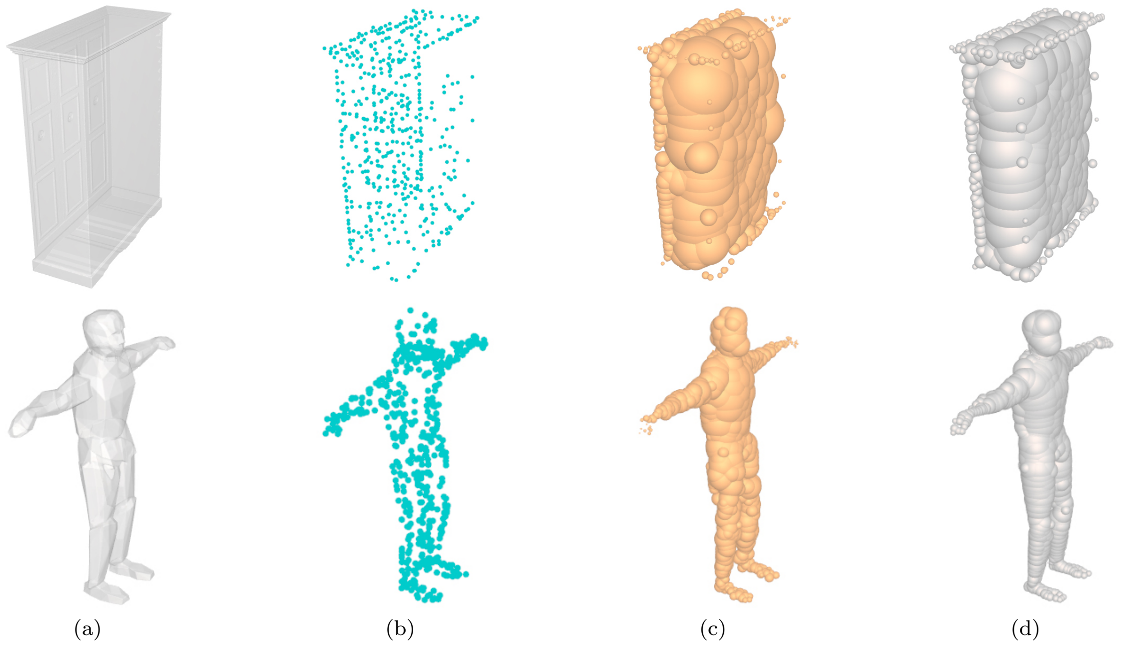

Figure C.16: More testing results on different types of shapes with 512 points as input. (a) The surface mesh, (b) testing point cloud, (c) the refined predicted spheres with both the normal-refinement strategy and the sphere-bounding strategy, (d) ground truth spheres.

large error with the ground truth. Figure C.20 shows more comparison results of the Power Crust and our method. Typically, for the result on the plant object in Figure C.20, quite a few edges of medial axis are computed. The centers of the computed spheres by Power Crust on the plant object and the lamp object are also illustrated in Figure 21(d) for a clear observation on the constructed edges of MAT by Power Crust.

Appendix C.3. Comparison with Poisson & Q-MAT+ Method and SPR & Q-MAT+

The Poisson & Q-MAT+ method or the SPR & Q-MAT+ is taken as a way for computing MAT from point cloud by computing MAT from the reconstructed surface mesh of the object. By assuming that the normals of point cloud is a priori, the Poisson Surface Reconstruction method is firstly used to reconstruct the surface mesh from the point cloud, and Q-MAT+ is then applied on the reconstructed surface mesh to compute the corresponding approximated MAT in the Poisson & Q-MAT+ method or in the SPR & Q-MAT+. For fair comparison, we take the same point set of size 2048 as input, the predicted spheres from P2MAT-NET are refined by both the normal-refinement and sphere-bounding strategies. Figure C.21 compares the results on the Poisson & Q-MAT+ method, the SPR & Q-MAT+ method and our P2MAT-NET-N-S method. It is noticed that the poor reconstructed mesh of the input point cloud leads to the poor MAT approximation of the input point cloud, even when the ground truth normal of the point cloud is given and used in the reconstruction of mesh. Table C.6 shows the comparison on the average Hausdorff errors of the Poisson & Q-MAT+, the SPR & Q-MAT+ method and our our P2MAT-NET-N-S method on point set with 256 and 2048 points.

Appendix C.4. Comparison with the GR method

The GR method could reconstruct meshes from the sparse point clouds. We use the two-sided Hausdorff distance to evaluate the reconstruction accuracy. Figure C.22 compares the results on the GR method and our method.

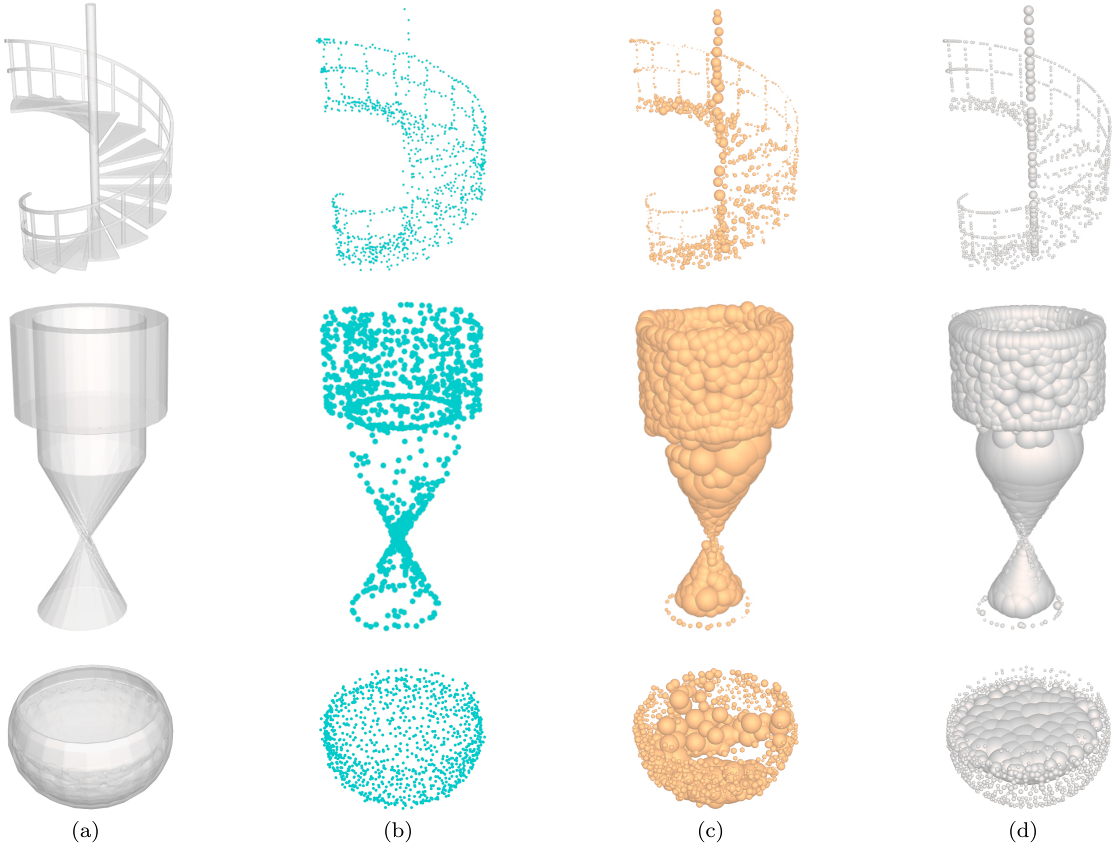

Figure C.17: More testing results on different types of shapes with 1024 points as input. (a) The surface mesh, (b) testing point cloud, (c) the refined predicted spheres with both the normal-refinement strategy and the sphere-bounding strategy, (d) ground truth spheres.

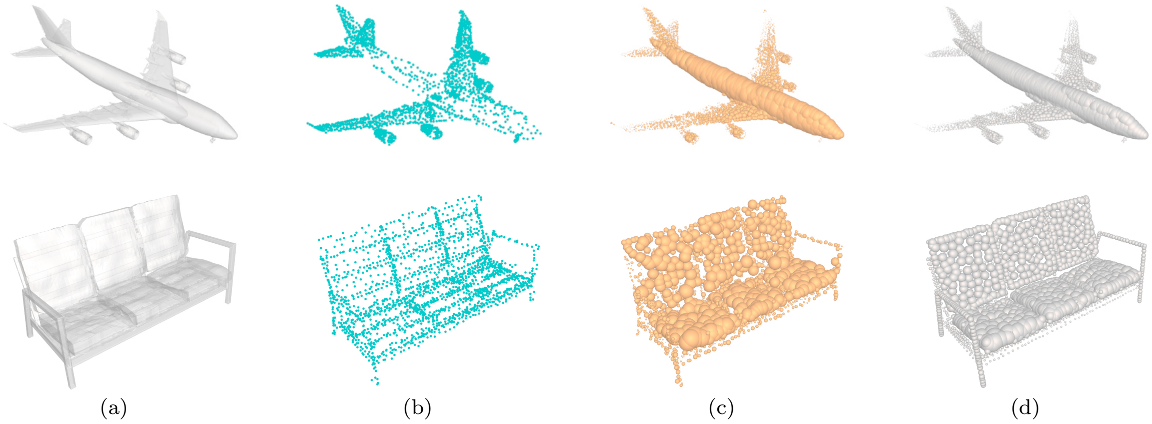

Figure C.18: More testing results on different types of shapes with 2048 points as input. (a) The surface mesh, (b) the testing point cloud, (c) the refined predicted spheres with both the normal-refinement strategy and the sphere-bounding strategy, (d) ground truth spheres.

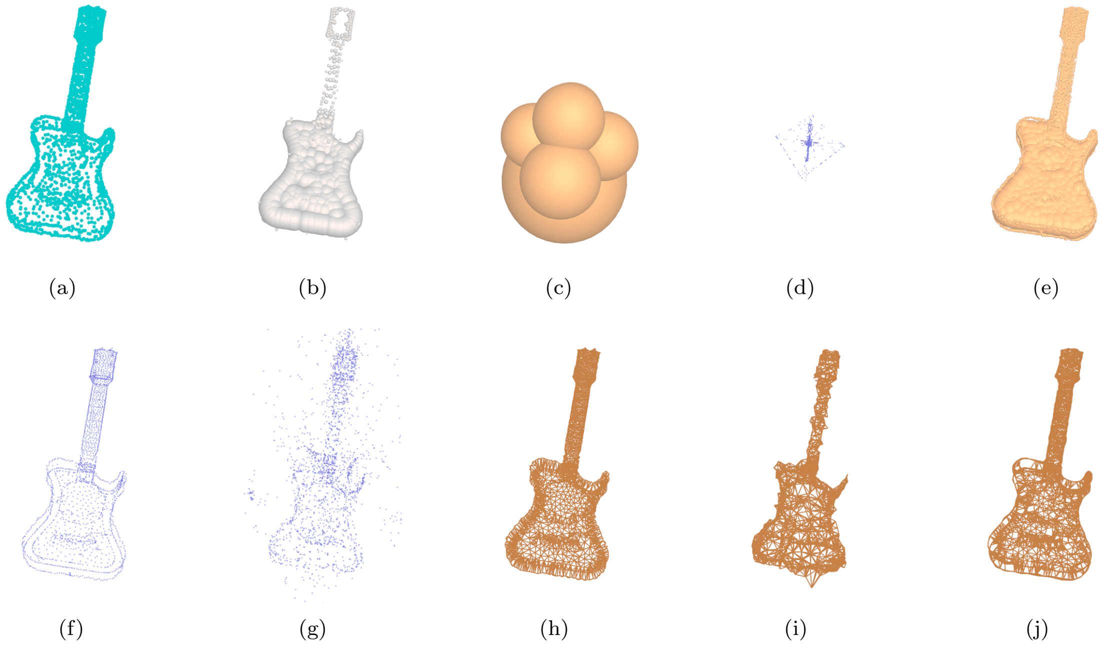

Figure C.19: Comparison with Power Crust on MAT computation on guitar with 2048 points as input. (a) The testing point cloud, (b) the ground truth medial spheres, (c) the spheres computed by the Power Crust method with (d) the centers of the spheres, (e) our refined predicted spheres with the sphere-bounding strategy, (f) the centers of ground truth spheres, (g) partial of the centers of spheres computed by Power Crust, (h) the ground truth medial edges, (i) the computed medial edges of Power Crust, and (j) our computed medial edges from spheres in (c).

	airplane	bed	chair	sofa	guitar	monitor
256	23.97 / 7.80	16.95 / 16.24	28.98 / 19.11	15.49 / 10.43	33.15 / 8.19	19.43 / 16.52
	12.83 / 23.05	13.81 / 19.26	18.39 / 27.46	10.37 / 8.89	5.59 / 10.86	12.49 / 23.35
	5.20 / 4.39	13.06 / 12.80	11.45 / 7.26	7.60 / 6.07	4.00 / 2.44	10.98 / 6.24
2048	10.49 / 4.45	11.66 / 13.27	19.07 / 11.27	8.53 / 6.71	4.45 / 3.98	8.67 / 12.55
	2.95 / 2.75	6.00 / 11.78	5.52 / 10.44	5.13 / 5.01	2.45 / 2.21	4.11 / 15.14
	0.98 / 1.17	2.50 / 3.61	1.78 / 2.43	2.49 / 3.95	1.04 / 1.29	1.98 / 2.65

Table C.6: Average Hausdorff errors of different classes of objects, learning from point clouds with different sizes (256/2048). For each size, the three lines represent the errors of MATs from (1st line) the Poisson & Q-MAT+ method, (2nd line) the SPR & Q-MAT+ method, (3rd line) our method of P2MAT-NET-N-S, respectively. "# / #" represent the errors of "Surface to MAT / MAT to Surface".

Appendix D. Comparison on Average Accuracy of Object Classes

The computed MAT from the point cloud can be used as an intermediate descriptor for downstream applications, 3D shape classification in our experiments, by using MAT-Net. In this section, we compare our methods "Ours w/ MAT-Net (sphere)" which uses only the refined predicted spheres and "Ours w/ MAT-Net (3 features)" which uses both the spheres and the topolofy information of MATs, with the state-of-the-art symmetric pooling-based approaches processing features of point clouds. Table D.7–D.10 report the average accuracy of those object classes with significant differences in classification accuracy, with point clouds of different sizes (256, 512, 1024 and 2048 respectively).

Method	avg.class	bathtub	ward.	radio	cone	person	dresser	desk
PointNet++	79.2	0.79	0.73	0.55	**0.82**	0.81	**0.80**	0.71
PointCNN	78.0	0.45	0.67	0.55	0.64	0.88	0.76	0.63
DGCNN	79.9	0.76	0.73	0.60	**0.82**	0.88	0.73	**0.81**
Ours w/ MAT-Net(sphere)	85.0	0.86	**0.93**	0.65	0.73	**1.00**	0.74	0.79
Ours w/ MAT-Net(3 features)	**86.5**	**0.90**	0.73	**0.75**	0.73	**1.00**	0.74	**0.81**

Table D.7: The classification accuracy of those object classes with significant differences in accuracy with 256 points (spheres).

Method	avg.class	bathtub	bench	cone	person	stairs	stool	tv.
PointNet++	79.5	0.79	0.75	**0.82**	0.69	0.56	0.75	0.85
PointCNN	81.2	0.55	0.75	0.73	**0.94**	0.88	0.69	0.80
DGCNN	79.3	0.76	0.75	**0.82**	0.88	0.63	**0.81**	0.80
Ours w/ MAT-Net(sphere)	84.9	**0.83**	**0.81**	0.73	0.88	**0.94**	**0.81**	**0.93**
Ours w/ MAT-Net(3 features)	**85.2**	0.76	**0.81**	**0.82**	**0.94**	0.88	**0.81**	0.85

Table D.8: The classification accuracy of those object classes with significant differences in accuracy with 512 points (spheres).

Method	avg.class	bathtub	bottle	cone	night.	range.	table	stool
PointNet++	**83.9**	**0.90**	0.95	0.64	0.65	0.93	0.86	0.81
PointCNN	83.6	0.62	0.91	0.73	0.78	0.88	0.79	0.75
DGCNN	79.9	0.76	0.95	0.73	0.69	0.91	0.85	0.81
Ours w/ MAT-Net(sphere)	83.1	0.83	0.95	**0.82**	**0.82**	**0.96**	**0.92**	0.75
Ours w/ MAT-Net(3 features)	83.8	0.79	**0.98**	0.73	0.71	0.91	**0.92**	**0.88**

Table D.9: The classification accuracy of those object classes with significant differences in accuracy with 1024 points (spheres).

Method	avg.class	bathtub	bench	cup	night.	person	range.	table
PointNet++	83.8	0.76	**0.81**	0.50	0.59	0.81	0.93	0.86
PointCNN	85.5	0.79	**0.81**	0.50	0.71	**0.94**	0.86	0.81
Ours w/ MAT-Net(sphere)	**85.6**	0.76	0.69	**0.58**	**0.89**	0.81	**0.94**	0.77
Ours w/ MAT-Net(3 features)	83.5	**0.83**	0.56	0.42	**0.85**	0.88	0.93	**0.87**

Table D.10: The classification accuracy of those object classes with significant differences in accuracy with 2048 points (spheres).

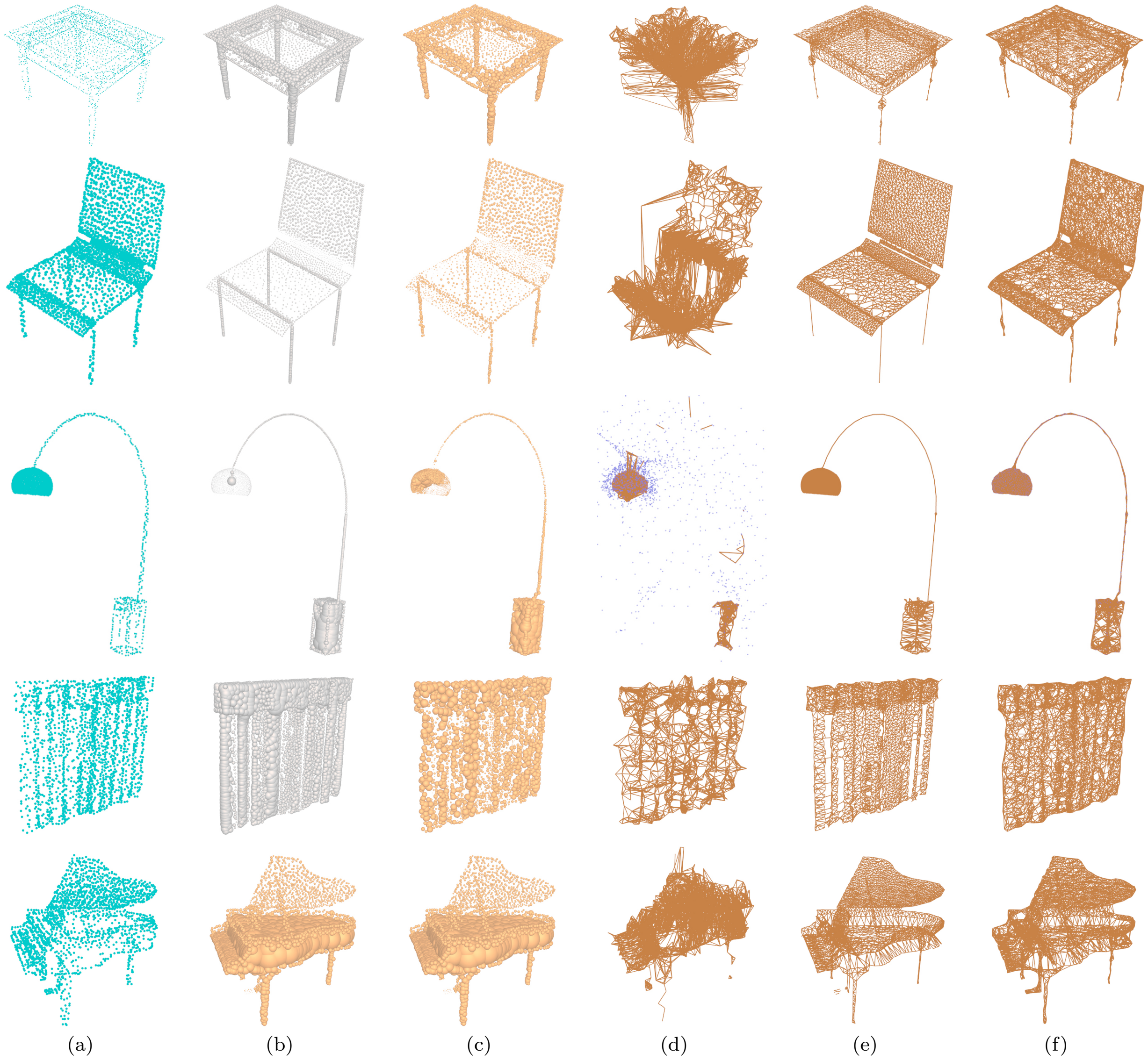

Figure C.20: More comparison results with the Power Crust method on MAT computation with 2048 points as input. (a) The testing point cloud, (b) the ground truth medial spheres, (c) our refined predicted spheres with the sphere-bounding strategy, (d) the computed medial edges of Power Crust, (e) the ground truth medial edges, (f) our computed medial edges from spheres in (c).

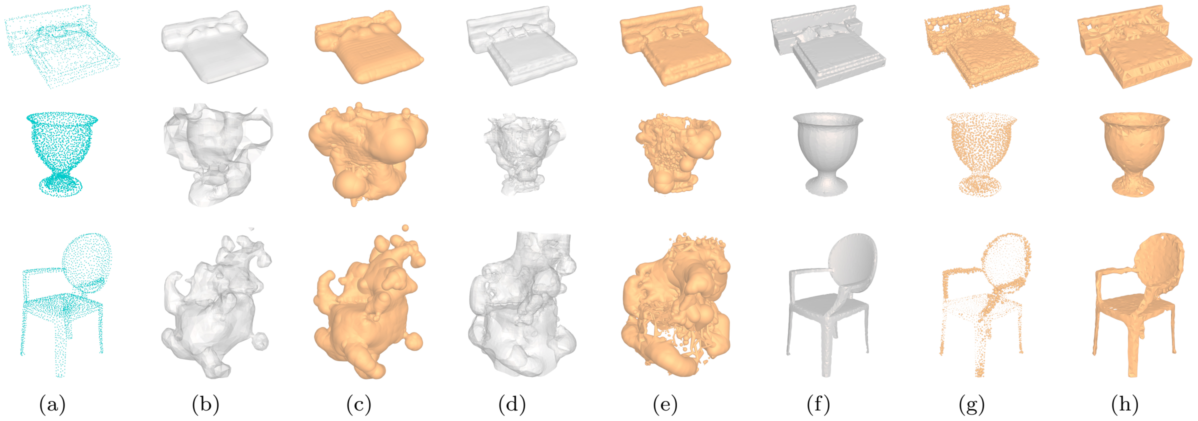

Figure C.21: More comparison results with the Poisson & Q-MAT+ method and the SPR & Q-MAT+ method with 2048 points as input. (a) The testing point cloud, (b) the Poisson reconstructed mesh, (c) the result of Q-MAT+ from the surface of (b), (d) the SPR reconstructed mesh, (e) the result of Q-MAT+ from the surface of (d), (f) the ground truth MAT, (g) the refined predicted spheres of P2MAT-NET with both the sphere-bounding and normal-refinement strategies, (h) the computed MAT with connectivity from spheres in (g).

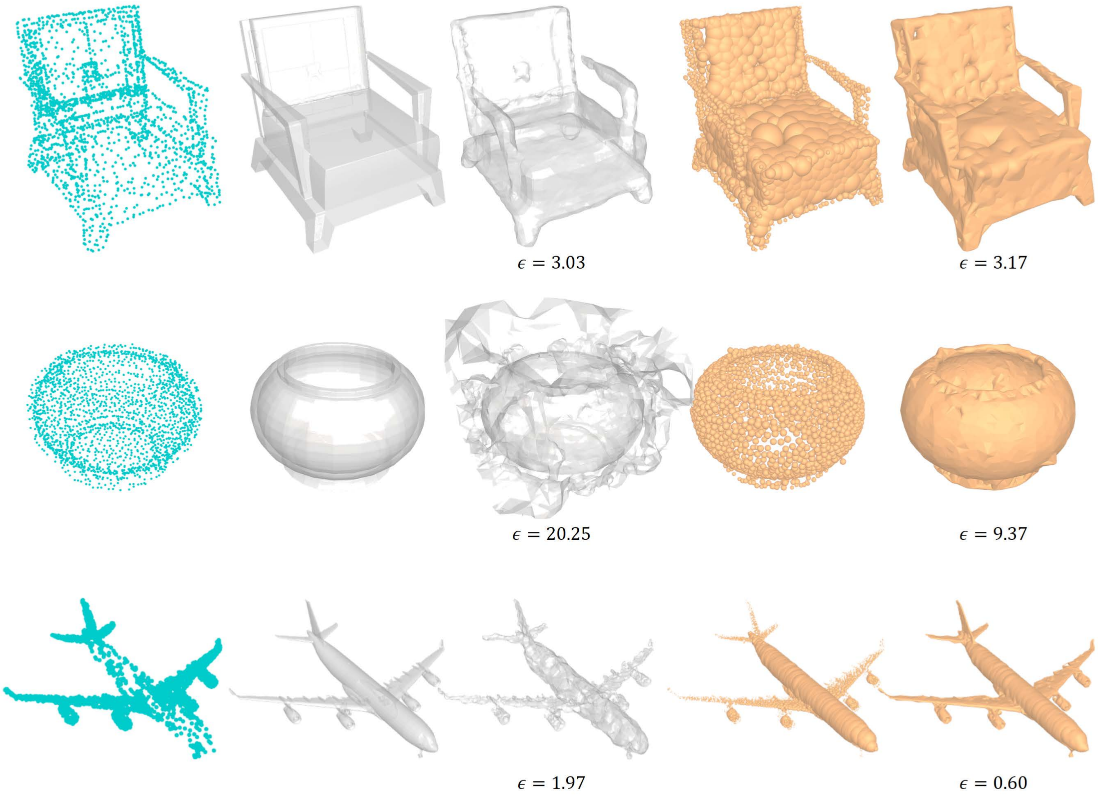

Figure C.22: More comparison results with the GR method on mesh reconstruction with 2048 points as input. (a) The testing point cloud, (b) the surface mesh, (c) reconstructed mesh with the GR method, (d) the refined predicted spheres with sphere-bounding strategy, (e) the computed MAT with connectivity from spheres in (d).