MAT is used, not generated

MAT-Net: Medial Axis Transform Network for 3D Object Recognition

Jianwei Hu[1,2], **Bin Wang**[1,2*], **Lihui Qian**[1,2], **Yiling Pan**[1,2], **Xiaohu Guo**[3], **Lingjie Liu**[4] and **Wenping Wang**[4]

[1]School of Software, Tsinghua University, China

[2]Beijing National Research Center for Information Science and Technology (BNRist), China

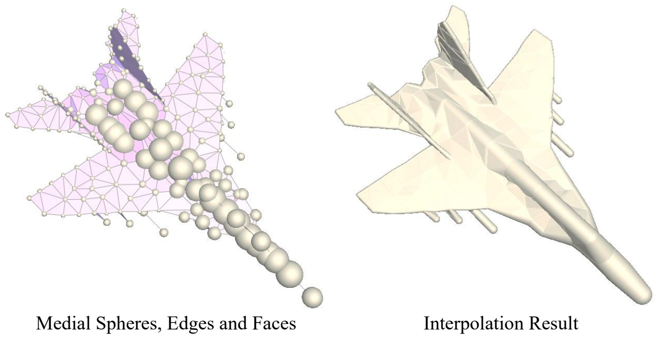
[3]Department of Computer Science, The University of Texas at Dallas, United States of America

[4]Department of Computer Science, The University of Hong Kong, Hong Kong

{hjw17, qlh17, pyl16}@mails.tsinghua.edu.cn, wangbins@tsinghua.edu.cn, xguo@utdallas.edu, liulingjie0206@gmail.com, wenping@cs.hku.hk

Abstract

3D deep learning performance depends on object representation and local feature extraction. In this work, we present MAT-Net, a neural network which captures local and global features from the Medial Axis Transform (MAT). Different from K-Nearest-Neighbor method which extracts local features by a fixed number of neighbors, our MAT-Net exploits effective modules Group-MAT and Edge-Net to process topological structure. Experimental results illustrate that MAT-Net demonstrates competitive or better performance on 3D shape recognition than state-of-the-art methods, and prove that MAT representation has excellent capacity in 3D deep learning, even in the case of low resolution.



Medial Spheres, Edges and Faces

Interpolation Result

Figure 1: The representation of Medial Axis Transform (MAT). The medial mesh is composed of 256 spheres and their connecting edges and faces. The right image shows the linear interpolation of the spheres along medial edges and faces. It can be seen that medial spheres in airfoil have small radii than those in fuselage, which illustrates that medial spheres encode local volume information for 3D shapes.

1 Introduction

Shape representation and neural network architecture are the research hotspots of 3D deep learning. Although deep learning has achieved remarkable results in the field of 2D images, it is difficult to be applied directly to 3D shapes, due to the complexity of the spatial and topological relationship between surface samples. Voxel representation [Wu *et al.*, 2015] is a regular 3D binary grid, which can be input to a standard convolution operation. But the memory and computation costs grow cubically as the resolution increases. Multi-view-based methods [Su *et al.*, 2015] express 3D objects as multi-view 2D images, by taking the advantage of pre-trained 2D CNN architectures. However, all views need to be pre-generated, and it is unclear how to determine the number and distribution of views to cover the 3D shape while avoiding self-occlusions. Point cloud is a typical 3D object representation that can be easily obtained from scanning devices, which has been exploited in point-based deep learning methods [Qi *et al.*, 2017a]. But they also ignore the information of topology structure in geometric presentation, leading to relatively low accuracy results in classification tasks.

This situation motivates us to find a 3D shape representation that has the following properties while avoiding the above mentioned problems: adapting to typical CNNs, expressing local and multi-scale information of shape, utilizing topological structure, and having low computational complexity. A representation is called a *complete shape descriptor* if it can be used to reconstruct the original shape. Medial Axis Transform (MAT) [Blum, 1967] is exactly a complete shape descriptor, and it contains information that jointly describes geometry, topology, symmetry, and thickness properties of a shape in a very compact fashion.

Medial axis of a shape is the set of all points having at least two closest points on the shape boundary. Medial axis transform (MAT) is a shape descriptor including the medial axis together with the associated radius function of the maximally-inscribed spheres, which can be used to reconstruct the original shape. There has been strong evidences on the importance of medial axis on human's perception of shapes. For example, physiologists have found that neurons in the primary visual cortex (V1) show strong response to the 2D medial axis of a textured figure [Kimia, 2003], and neurons in the inferior temporal cortex (IT) encode 3D medial axis information [Hung *et al.*, 2012]. Despite the important links between MAT and human cognitive system, there is still no study of 3D deep learning methods from the perspective of

[*]Corresponding Author

MAT, to the best of our knowledge. In this paper, we integrate the MAT of 3D shapes into the design of deep neural network, called *MAT-Net*, for 3D shape recognition.

In Q-MAT [Li *et al.*, 2015], the MAT is approximated by a *medial mesh*, which is a 2D simplicial complex consisting of spheres, edges, and triangle faces. A sphere is expressed as a four-dimensional vector: position coordinates (x, y, z) and radius (r). The medial mesh gives a linear approximation to the true MAT through linear interpolation of the spheres along its edges and faces. Q-MAT [Li *et al.*, 2015] proposes a method for simplifying an initial medial mesh to obtain a geometrically accurate and structurally simple MAT representation. Figure 1 shows an example of simplified medial mesh.

There are several difficulties that we need to address when designing and exploiting MAT-Net for 3D deep learning.

1. Deep learning needs large data set for training, but there is no MAT data set with a large number of samples. ModelNet40 [Wu *et al.*, 2015] is one of the most popular data set on 3D object classification. Converting it to MAT representation is a non-trivial task. This is because MAT computation often needs closed manifold mesh surface. In ModelNet40, a lot of shapes are not watertight or are non-manifold. We repaired the meshes of majority objects in ModelNet40, and used Q-MAT to construct an open MAT data set called *ModelNet40-MAT* for research community.

2. Similar to point clouds, the medial spheres (x, y, z, r) are unordered and cannot be directly input to typical CNNs. PointNet [Qi *et al.*, 2017a] provides a solution for feature extraction of point clouds or other coordinate-based representation. Except for position coordinates (x, y, z), other local features (e.g. radius r of sphere as in our case) may be added as additional dimensions.

3. Another difficulty is local feature extraction from its topological structure. Although the medial sphere already expresses the local volumetric feature by its radius, it is better to extract different resolutions of local features from these spheres. A convenient method is to use the strategy in PointNet++ [Qi *et al.*, 2017b] to sample and group neighboring spheres around a given center sphere. However, this method still cannot utilize the topological structure of MAT.

In this paper, we propose two facilitating modules: *Group-MAT* and *Edge-Net*. The Group-MAT groups the unordered spheres into a regular local data structure by referring to their edge information. Then Edge-Net is designed to extract features of local shape. The idea is to use max-pooling function to select the max response sphere features from the neighbor spheres. The detailed description is given in Section 3.3.

The **contributions** of this paper are as follows:

- We present the first deep neural network architecture that can learn the features of MAT, for 3D object recognition.
- By utilizing MAT's edge information, we design Group-MAT and Edge-Net modules to capture local features from its topological structure, which achieves remarkable performance on 3D shape classification task, even for MATs with very few number of spheres only.
- We construct an open MAT dataset: ModelNet40-MAT, by repairing the majority of 3D models in ModelNet40.

2 Related Work

2.1 Shape Representation for 3D Analysis

Hand-Crafted Features. 3D shapes can be represented using either histograms or bag of features models, such as point feature histograms and normal histograms. Other representations include Light Fight Descriptor [Chen *et al.*, 2010], Heat Kernel Signatures [Bronstein *et al.*, 2011], and Spherical Harmonics [Kazhdan *et al.*, 2003], to name a few.

Voxel Grids. 3D ShapeNets [Wu *et al.*, 2015] uses convolutional deep belief network to learn probability distribution of binary information on 3D voxel grids. Similar to regular data in 2D images, voxel grids are proposed to represent 3D shapes because they are compatible with 3D convolutional neural networks. A similar approach was proposed in VoxNet [Maturana and Scherer, 2015]. Volumetric-MVCNN [Qi *et al.*, 2016] and FusionNet [Hegde and Zadeh, 2016] also combine the voxels and images. Using voxel grid representation achieves good results on a variety of recognition tasks. However, it is constrained by the grid resolution and computational cost.

2D Images. By taking advantages of pre-trained 2D convolutional neural networks [Krizhevsky *et al.*, 2012], multi-view methods [Su *et al.*, 2015; Qi *et al.*, 2016; Li *et al.*, 2018] have achieved excellent performance on shape classification tasks. PANORAMA-based methods [Shi *et al.*, 2015; Sfikas *et al.*, 2017] extract the panoramic representation that preserves feature continuity of the 3D models and achieves a performance above or comparable to the state-of-the-art.

Kd-tree and Octree. Kd-Net [Klokov and Lempitsky, 2017] uses a kd-tree structure to form a computational graph, and computes a sequence of hierarchical representation. O-CNN [Wang *et al.*, 2017] presents an octree-based convolutional neural network, which represents the 3D shapes with octrees and performs 3D CNN operations on the sparse octants occupied by the surfaces of 3D shapes. The octree structure stores the octants information computed from surfaces. Then it can be input into the common deep network to realize shape analysis tasks including object classification, shape retrieval, and shape segmentation.

2.2 Feature Capturing on Point Clouds

PointNet [Qi *et al.*, 2017a] and PointNet++ [Qi *et al.*, 2017b] are the pioneering methods to directly process point cloud by utilizing symmetric function for disordered input. Although PointNet has excellent result on point cloud learning, it ignores the features extraction of local structures. PointNet++ proposes a set abstraction level to extract multiple scales of local patterns and combine them intelligently according to local point densities. It constructs local region sets by finding "neighboring" points around the center points. Inspired by this idea, PointCNN [Li *et al.*, 2018] proposed

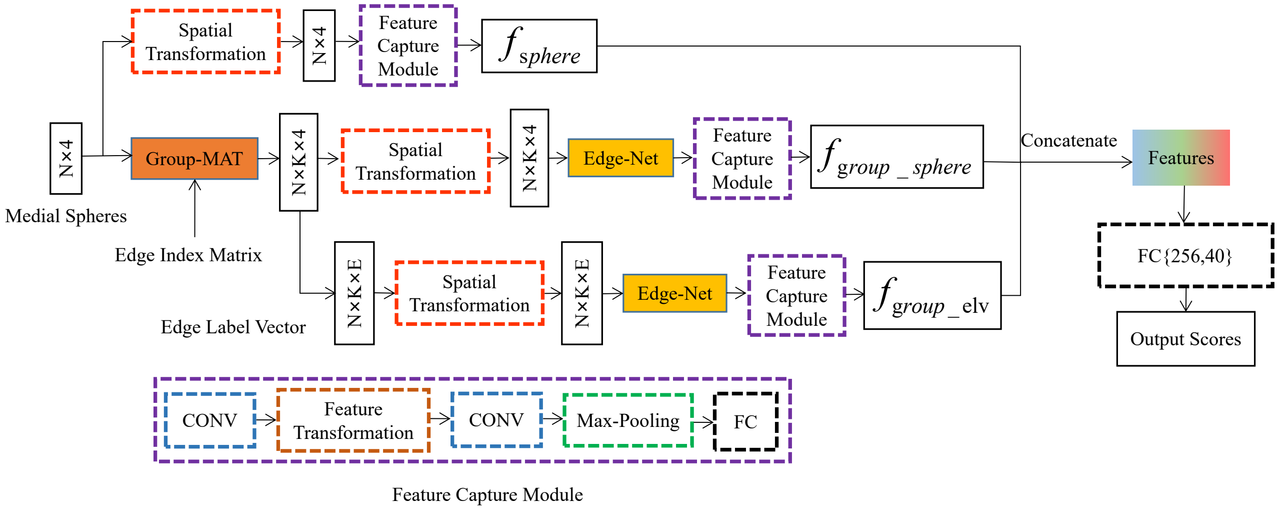


Figure 2: MAT-Net Architecture. The network takes N medial spheres, $N \times K$ edge index and $N \times K$ edge mask as inputs. Red dashed boxes denote transformation network from Spatial Transformer Networks. Purple dashed boxes denote feature capture module.

X-Conv operator that weights and permutes input points and features before they are processed by a typical convolution. GraphCNN [Wang *et al.*, 2018] also recomputes a graph by using nearest neighbors in the feature space produced by each neural network layer.

3 The Method

3.1 Simplified Medial Mesh

Q-MAT [Li *et al.*, 2015] uses quadratic error minimization to compute a structurally simple, geometrically accurate, and compact simplicial complex representation of the MAT. A triangle mesh M_s, called *medial mesh* is used to approximate the MAT of a 3D shape S. Each vertex $\mathbf{s}_i$ of M_s represents a medial sphere and is denoted as a 4D point $\mathbf{s} = (x, y, z, r)^\top$ which contains the center $(x, y, z)^\top$ and radius r of the sphere. e_{ij} is the edge of M_s between two medial spheres $\mathbf{s}_i$ and $\mathbf{s}_j$, in which case $\mathbf{s}_i$ and $\mathbf{s}_j$ are called neighboring spheres.

Each edge of the medial mesh defines an enveloping volume primitive. The primitive given by the edge e_{ij}= $\{\mathbf{s}_i, \mathbf{s}_j\}$ is swept by the family of spheres defined by the linear interpolation of $\mathbf{s}_i$ and $\mathbf{s}_j$, that is, $(1-t)\mathbf{s}_i + t\mathbf{s}_j, t \in [0, 1]$. It comprises two spherical caps joined by a truncated cone, called a *medial cone*. A medial cone can be seen as a part of a 3D shape, such as the human arm. Meanwhile a medial sphere may have several edges connecting with other medial spheres. All composed medial cones can express various shapes. We call $\mathbf{s}_i$ and all of its neighbor spheres as the *neighbor data*. Experiments show that using neighbor data of MAT can improve classification accuracy remarkably.

As shown in Figure 1, the MAT sampled and interpolated with a few medial spheres can still represent a 3D object very well. The question is how to use these data for designing an appropriate architecture of neural networks.

3.2 MAT-Net Architecture

Our full network architecture is visualized in Figure 2. The inputs of MAT-Net include medial axis sphere information and edge information. The sphere information is represented by an $N \times 4$ matrix:

$$\mathbf{S} = \{\mathbf{s}_i \in \mathbb{R}^4, i = 0, \cdots, N-1\},$$

where N is the total number of spheres, i is sphere's index, and $\mathbf{s}_i = (x_i, y_i, z_i, r_i)^\top$. The edge information is represented by two $N \times K$ matrices. One is called *Edge Index Matrix*:

$$\mathbf{D} = \{d_{ik}, i = 0, \cdots, N-1; k = 0, \cdots, K-1\},$$

where d_{ik} is the index of the k-th neighbor sphere of $\mathbf{s}_i$. We set the number of neighbors to be a constant K. If a medial sphere's number of neighbors is less than K, we set the redundant $d_{ik} = -1$. Another one is called *Edge Mask Matrix*:

$$\mathbf{M} = \{m_{ik}, i = 0, \cdots, N-1; k = 0, \cdots, K-1\}.$$

If $d_{ik} \geq 0$, then $m_{ik} = 1$; otherwise $m_{ik} = 0$. Figure 3 shows the illustration of these two matrices.

As shown in Figure 2, the medial spheres are processed into two branches. The top branch is similar with the original PointNet. The Spatial Transformation module can predict an affine transformation matrix by neural networks and directly apply this transformation to input data. It aligns all input set to a canonical space before feature extraction, which improves the classification performance. Feature Transformation extends this idea to the alignment in the feature space. To be exact, Transformation module learns a $K \times K$ feature map from $N \times K$ input features. Then multiplies the $N \times K$ input features and $K \times K$ feature map, and get a transformed $N \times K$ features. This performs better than using $N \times K$ input features directly. Feature Transformation just has lager K value than Spatial Transformation. After transformation, we

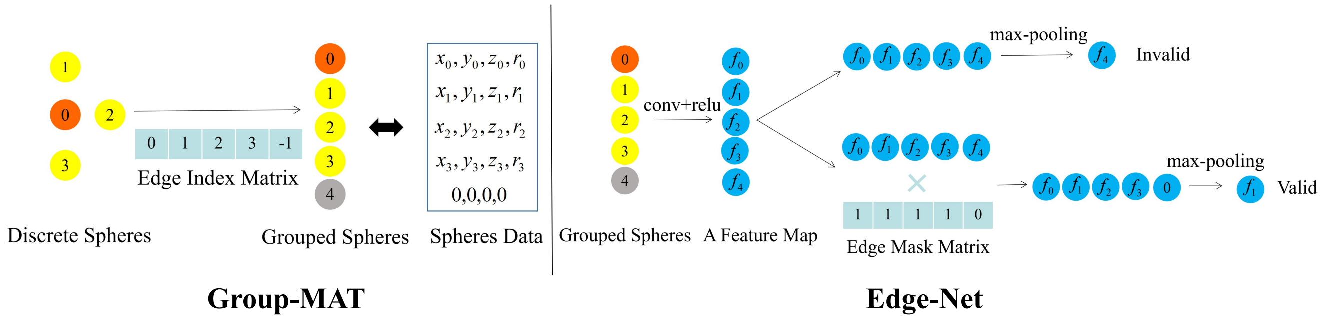


Figure 3: Group-MAT and Edge-Net. We take four medial spheres as an example to describe a local shape. Center medial sphere (red) has three neighbor spheres (yellow). In Group-MAT module, Edge Index Matrix (Here we assume $K = 5$) is used to group the center sphere's neighbors, and gray sphere is the filled sphere with $(0, 0, 0, 0)^\top$. In Edge-Net, utilizing the convolution operation and relu function, grouped spheres are computed to a feature map $\mathbf{F}_0 = \{f_0, \cdots, f_4\}$. Because using relu function, $f_i \geq 0$. In order to eliminate the interference caused by f_4, we take the element-wise product between $\mathbf{F}_0$ and Edge Mask Matrix $\mathbf{M}_0$. Finally, a max-pooling operation will find the maximum responsive sphere feature of the current feature map.

use multiple layer perceptrons and max-pooling to obtain a feature vector f_{sphere}.

In the bottom branch, we utilize topology structure that groups the neighbor data according to Edge Index Matrix and Edge Mask Matrix. Group-MAT constructs a $N \times K \times 4$ sphere tensor $\mathbf{T}$ by filling medial spheres to their corresponding positions. This operation is shown in Figure 3. The $K \times 4$ matrix $\mathbf{T}_i$, $i = 0, \cdots, N-1$ can be seen as a local shape patch centered around sphere $\mathbf{s}_i$. Let us use $\delta = \mathbf{s}_i - \mathbf{s}_j$ to denote the offset between the spheres i and j. We define Edge Label Vector (ELV) to be: $\mathbf{l}_{ij} = (\delta_x, \delta_y, \delta_z, \delta_r, r_i, r_j, \|\delta\|)^\top$. Here the dimension of ELV is $E = 7$. We construct an ELV tensor that is of dimension $N \times K \times E$ to represent local neighbor information of all spheres [Simonovsky and Komodakis, 2017]. Then we use two sub-branches with Edge-Net to learn local shape's features. After Feature Capture Module, we obtain two feature vectors: f_{group_sphere} and f_{group_elv}. Then the three features f_{sphere}, f_{group_sphere}, and f_{group_elv} are concatenated into global features. Finally MAT-Net uses fully connected networks and soft-max layers to classify these features.

3.3 Group-MAT and Edge-Net

Suppose the sphere $\mathbf{s}_i$ has K_i neighbor spheres. In order to extract features from a local shape centered around $\mathbf{s}_i$, a simple and convenient way is to group the neighbor spheres into a $K_i \times 4$ matrix. We can extract a feature map by using a $K_i \times 4$ kernel. But in the medial mesh, spheres have different number of neighbors K_i, thus a fixed-size kernel cannot be applied directly. We choose a large enough number K for all spheres (i.e. $K_i \leq K$), and then fill the redundant value of the kernel with 0 if $K_i < K$. In this method, a $K \times 4$ kernel can be used to represent the neighbor data of each sphere. Obviously, the neighbor data of different spheres have different numbers of filling elements. In addition, the spheres in the neighbor data do not have a reasonable order.

We treat a local shape as a $K \times 4$ matrix with some filling value of $(0, 0, 0, 0)$. As shown in Figure 3, Group-MAT groups the spheres by referring to Edge Index Matrix and reshapes $\mathbf{s}_i$'s local shape data into a $K \times 4$ matrix $\mathbf{T}_i$. Then Edge-Net uses a kernel $\mathbf{w}$ of size 4×1 and bias vector $\mathbf{b}$ of size $K \times 1$ to compute a feature map:

$$\mathbf{F}_i = relu(\mathbf{T}_i * \mathbf{w} + \mathbf{b}), \quad i = 0, \cdots, N-1. \tag{1}$$

We use rectified linear unit as activation function to insure that $\mathbf{F}_i \geq 0$. In Figure 3, $\mathbf{F}_0 = (f_0, \cdots, f_4)$. Directly using max-pooling operation will fail to extract maximum responsive feature $F_{i,max}$ because of those $F_{i,r}$ where r corresponds to the filling entries, i.e., $d_{i,r} = -1$. For example, f_4 of Figure 3. In order to solve this problem, we make an element-wise product between $\mathbf{F}_i$ and Edge Mask Matrix $\mathbf{M}_i$, to let $F_{i,r} = 0$ for all filling entries r. Finally, by using max-pooling we get $F_{i,max}$ from the neighbor data of $\mathbf{s}_i$:

$$\begin{aligned} \mathbf{F}_i^* &= prod(\mathbf{F}_i, \mathbf{M}_i), \\ F_{i,max} &= max - pooling(\mathbf{F}_i^*). \end{aligned} \tag{2}$$

Since multiple convolution kernels are used, different kernels make different maximum responsive sphere features. This means that the local features of neighbor spheres can be represented as a set of maximum responsive sphere features from all feature maps. Our following experiments show that using Group-MAT and Edge-Net greatly improves the classification performance.

4 Experiments

4.1 Datasets

We evaluate MAT-Net on ModelNet40-MAT (rigid object) and SHREC15 [Lian *et al.*, 2015] (non rigid object).

- **ModelNet40-MAT**: ModelNet40 is a popular 3D data set to verify the classification capability of 3D deep learning methods. To prepare the input MAT data for our MAT-Net, we need to compute MAT for the 3D data in ModelNet40. The MAT computation typically needs the 3D surface to be uniformly sampled, and be a closed manifold. But the majority of 3D models in

Method	Representation	Input Size	Overall Accuracy(%)
PointNet	xyz points	1024×3	90.2
PointNet++(msg)	xyz points	1024×3	90.5
	xyz points + normals	1024×6	91.3
PointCNN	xyz points + normals	1024×6	91.6
O-CNN(6)	octree	64^3	87.4
MVCNN, 12×	images	$224 \times 224 \times 3$	90.2
MAT-Net(f_{sphere})	xyzr	256×4	90.8
MAT-Net(f_{sphere} + f_{group_sphere})	xyzr	256×4	92.0
MAT-Net(f_{sphere} + f_{group_elv})	xyzr	256×4	92.8
MAT-Net(f_{sphere} + f_{group_sphere} + f_{group_elv})	xyzr	128×4	91.1
	xyzr	256×4	**93.2**
	xyzr	512×4	92.9
	xyzr	1024×4	92.4

Table 1: Object classification in 83.2% objects of ModelNet40 data set. Contents in parentheses represent the features used.

Sphere Number	Accuracy(xyz)	Accuracy($xyzr$)
128	88.8%	90.2%
256	89.7%	90.8%
512	90.4%	91.3%
1024	90.3%	91.1%

Table 2: Comparing the effects without and with radius used.

ModelNet40 do not satisfy these requirements. Consequently, Q-MAT [Li *et al.*, 2015] can not be applied directly to compute MAT. We successfully repaired 83.2% of all 3D models in ModelNet40 and constructed a MAT data set, named *ModelNet40-MAT*, which has multi-resolution MAT data for each 3D model.

- **SHREC15**: The database has 1200 watertight meshes which are equally classified into 50 categories. Each category contains 24 shapes which generated from an original 3D shape by implementing various pose transformations. Because all shapes are watertight and manifold, we can directly generate simplified medial meshes by Q-MAT. We use five fold cross validation to acquire classification accuracy.

4.2 Mesh Repair

The repairing method is inspired by the virtual camera algorithm [Wang *et al.*, 2017]. We first place several cameras on the external ball, and sample the first intersected point between the ray emitted by the camera the faces of the 3D model. The parallel rays are uniformly cast toward the object. In this way, we can delete the point whose normal is different from the ray direction to fix the normals of the faces. This method can fix most of faces. However, there could be some occluded surfaces inside the 3D model, no matter how many cameras are placed around it. So we use a dynamic camera-placing strategy. If two intersected points of neighboring rays from the same virtual camera are far away from each other, there may be a gap on the faces of the model, so we can place the new camera in the further point position. This method generates surface samples that are more uniform and complete. Finally we use Poisson surface reconstruction to obtain a closed manifold surface from these samples.

Our ModelNet40-MAT has $10,243$ MAT objects in 40 categories, about 83.2% of the original ModelNet40. We use $8,208$ objects for training and $2,035$ objects for testing. For each 3D object, we use Q-MAT to compute the MAT data of different numbers of spheres: 128, 256, 512, and 1024. All medial sphere centers and radii are normalized into a unit ball. To ensure a fair comparison, all compared methods are run on the same 83.2% objects of ModelNet40.

4.3 Implementation Details

Our network is implemented with TensorFlow on an NVIDIA TITAN Xp. All experiments are trained with a large-enough number of neighbors $K = 16$. We jitter the initial medial spheres (with random translation of Gaussian distribution $\mathcal{N}(0, 0.008)$ and clipped to 0.01 to generate the augmented spheres. Edge-Net has only 1 convolution layer with output channel of 32. In $group\_sphere$ and $group\_elv$ sub-branches, Edge-Net is also used after the Spatial Transformation module. Batch-normalization and relu activation are applied to other layers.

The first CONV module of Feature Capture Module has two convolution layers and the second has three convolution layers. Their filters size are {64,64,128,256,1024}. The convolution kernel size of the first layer is 1×4, and the rest are 1×1. FC module has two fully connected layers and filters size are {512,256}. The three branch features are 1×256 vectors. The concatenated feature is a 1×768 vector. The loss function includes cross entropy loss for classification and $L2$ loss of feature transformation matrix. The batch size of ModelNet40-MAT classification is 32, and 16 for SHREC15.

For fair comparison, we reproduced the representative 3D CNN methods on the same 83.2% objects of ModelNet40. The grouping method of PointNet++ is multi-scale grouping (MSG). The PointCNN is trained on modelnet_x3_l4, which includes the configured network structure and hyper parameters. We generated the octree data, and rotated them to 12 orientations. The resolution of leaf octants is 64^3. MVCNN

	overall	avg.class	radio	table	vase	wardrobe	bench	plant	lamp	door	person
PointNet++	0.912	0.86	0.75	0.86	0.78	0.86	0.67	0.69	0.45	**1.00**	**1.00**
MAT-Net(f_{sphere})	0.908	0.86	0.65	0.84	0.77	0.84	0.53	**0.78**	**0.82**	**1.00**	0.94
MAT-Net(3 features)	**0.932**	**0.89**	**0.90**	**0.97**	**0.90**	**0.97**	**0.87**	0.72	0.64	0.94	0.94

Table 3: Accuracy of different classes of 3D objects.

Method	Input Size	Input feature	Accuracy(%)
PointNet++	1024 × 3	XYZ	60.18
PointNet++		Intrinsic features (Euclidean)	94.49
PointNet++		Intrinsic features (Non-Euclidean)	96.09
MAT-Net(f_{sphere})	256 × 4	XYZR	95.58
MAT-Net(f_{group_sphere})	256 × 4	XYZR	96.25
MAT-Net($f_{sphere} + f_{group\_sphere}$)	256 × 4	XYZR	**96.42**

Table 4: Non-Rigid object classification on SHREC15.

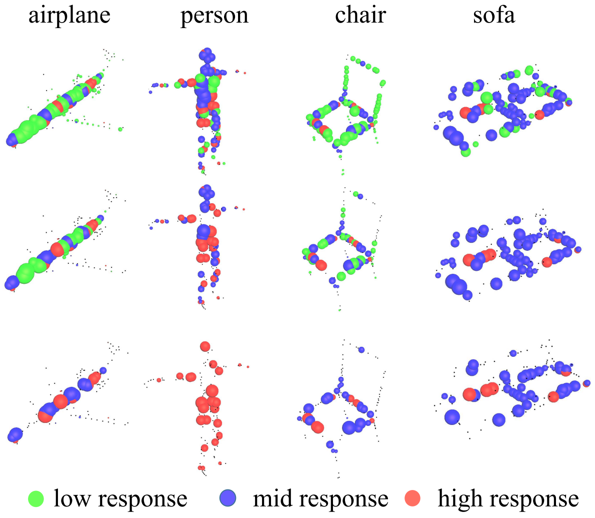


Figure 4: The responsive spheres of MATs with 256 medial spheres. From top to bottom, every row shows the responsive spheres of different objects under incremental frequency threshold. The third row only shows the mid and high responsive spheres. In the first column, the high responsive spheres are gathered in fuselage of airplanes, which have the ability to distinguish airplane from other categories.

is pre-trained on the ImageNet1K data set. We fine-tuned this model on multi-view data (views=12). The object classification results of all methods are computed without voting.

4.4 3D Object Classification

Table 1 compares the classification performance of different methods. We can see that even using only 256 medial spheres MAT-Net (f_{sphere}) can get an accuracy better than PointNet's 90.2% and PointNet++'s 90.5% (without normals). By using three features simultaneously, MAT-Net (f_{sphere} + f_{group_sphere} + f_{group_elv}) gets an overall classification accuracy of 93.2%. Compared with using only the sphere feature f_{sphere}, adding the local shape feature f_{group_sphere} and f_{group_elv} produces improvements for accuracy, respectively. Note that the classification accuracy does not necessarily improve with the increase of MAT resolution. The low resolution MAT contains simple but concise topology, while in high resolution MAT the topology has redundant information. Thus, it will slightly affects the classification accuracy as shown in Table 1. It can be seen that the MAT resolution of 256 spheres produces the best results.

Sphere radius is an important characteristic of MAT, which encodes the volume of local shape. In order to explore the effect of radius for classification, we compare the performances of different resolutions of MAT without and with radius used. As shown in Table 2, by using only f_{sphere}, the accuracy is improved by around $0.8\% - 1.4\%$ after using radius. As shown in Table 3, we compare the average class accuracies of three methods, and list the object classes with significant differences in classification accuracy. MAT-Net (3 features) greatly improves overall accuracy and average class accuracy. Only in very few object classes, as shown in the last two columns of Table 3, MAT-Net has slightly-worse accuracy than PointNet++. Here we use 1024 points and normals for PointNet++, and 256 spheres for MAT-Net. For the ambiguous category in ModelNet40, such as plant (confused with flower pot and vase), the result of MAT-Net is not very satisfactory.

4.5 Classification on 3D Non-Rigid Object

We test our method on randomly divided SHREC15 benchmark. As shown in Table 4, PointNet++ achieves excellent classification performance when using non-Euclidean metric space and intrinsic features (including wave kernel signature, heat kernel signature and multi-scale Gaussian curvature). However, with the spatial coordinates as input features, PointNet++ achieves a classification accuracy of 60.18%. MAT-Net gets a much higher classification accuracy even using the f_{sphere} merely. The best classification accuracy of MAT-Net is 96.42% when concatenating the f_{sphere} and f_{group_sphere}.

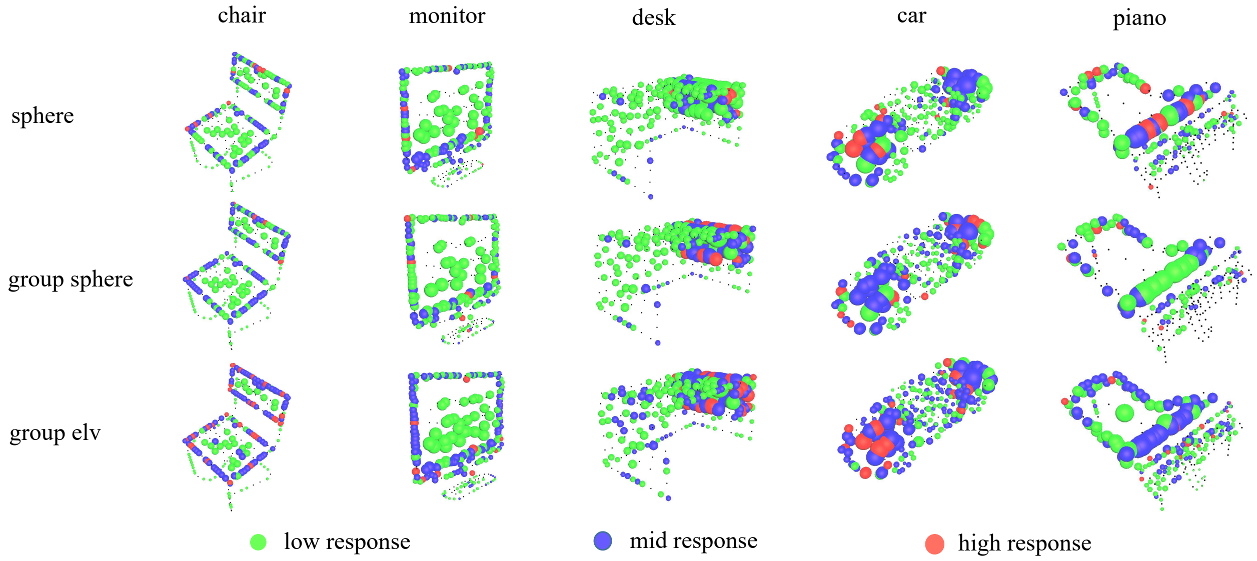


Figure 5: Responsive spheres of different branches of the MAT-Net architecture.

Method	Input	Overall accuracy (%)	Forward time (ms)
PointNet++	1024 points + normals	91.3	6.5
MAT-Net(f_{sphere})	256 spheres	90.8	4.7
MAT-Net(f_{group_sphere})	256 spheres	91.7	5.2
MAT-Net(3 features)	256 spheres	93.2	12.6

Table 5: Running time comparison.

4.6 Feature Visualization

In order to explore the features that MAT-Net learned, we visualize the information that has been learned from max-pooling layer of Feature Capture Module. Max-pooling layer outputs a 1024-dimensional feature vector from 1024 feature maps of size $N \times 1$. A feature map includes the features of N spheres. The max-pooling layer will find out the maximum features of all feature maps. These sphere features are input to fully-connected layers, finally affecting the soft-max layer. We call these spheres *responsive spheres*, which have contribution to classification. As shown in Figure 4, we count the indexes of responsive spheres from max-pooling layer, and color these spheres by their frequency. Large frequency spheres have more contribution to classification. In order to eliminate the occlusion between the spheres and make it easy to be visualized, in each row we only show the spheres whose frequencies are larger than a certain threshold. We can observe that each class object learns its unique features that are important for the classification.

In Figure 5, we show the responsive spheres from three types of features proposed in our MAT-Net. The second and third rows show that adding edge information highlights local structures, especially for frame structures of 3D objects. It partially explains why concatenating all three types of features gets the best classification results.

4.7 Running Time

We record the forward time with a batch size 5 and using TensorFlow 1.4 with a single NVIDIA TITAN Xp. The first batch is neglected since there is some preparation for GPU. As shown in Table 5, MAT-Net gets faster and without large accuracy reduction when using $f_{spheres}$ merely. Concatenating three branch features leads to best classification result but increases the running time a bit, since the forward propagation will be executed three times.

5 Conclusion and Future Work

In this paper, we propose the MAT-Net architecture that extracts features of spheres and their topological structures from MAT of 3D shapes, which shows better performance on 3D classification task than state-of-the-art methods.

We would like to extend MAT-Net to explore other 3D representations with topology information. It is interesting to note that in our MAT-Net architecture, Group-MAT is used only once, and the neighbor index matrix is static. We would like to explore the dynamic updates of neighbor index matrix and the use of Edge-Net to learn hierarchical features.

Acknowledgements

This work was supported by the NSFC under Grant 61772301.

References

[Blum, 1967] Harry Blum. A transformation for extracting new descriptors of shape. *Models for Perception of Speech and Visual Forms, 1967*, pages 362–380, 1967.

[Bronstein *et al.*, 2011] Alexander Bronstein, Michael Bronstein, Leonidas Guibas, and Maks Ovsjanikov. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics*, 30(1):1–20, 2011.

[Chen *et al.*, 2010] Ding Yun Chen, Xiao Pei Tian, Yu Te Shen, and Ouhyoung Ming. On visual similarity based 3D model retrieval. *Computer Graphics Forum*, 22(3):223–232, 2010.

[Hegde and Zadeh, 2016] Vishakh Hegde and Reza Zadeh. FusionNet: 3D object classification using multiple data representations. In *Neural Information Processing Systems Workshop on 3D Deep Learning*, 2016.

[Hung *et al.*, 2012] Chia-Chun Hung, Eric T. Carlson, and Charles E. Connor. Medial axis shape coding in macaque inferotemporal cortex. *Neuron*, 74(6):1099–1113, 2012.

[Kazhdan *et al.*, 2003] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing*, pages 156–164, 2003.

[Kimia, 2003] Benjamin B. Kimia. On the role of medial geometry in human vision. *Journal of Physiology-Paris*, 97(2-3):155–190, 2003.

[Klokov and Lempitsky, 2017] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3D point cloud models. In *IEEE International Conference on Computer Vision*, pages 863–872, 2017.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems*, pages 1097–1105, 2012.

[Li *et al.*, 2015] Pan Li, Bin Wang, Feng Sun, Xiaohu Guo, Caiming Zhang, and Wenping Wang. Q-MAT:computing medial axis transform by quadratic error minimization. *ACM Transactions on Graphics*, 35(1):1–16, 2015.

[Li *et al.*, 2018] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and et al. PointCNN: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems 31*, pages 820–830. 2018.

[Lian *et al.*, 2015] Z. Lian, J. Zhang, S. Choi, H. ElNaghy, El-Sana, and et al. Non-rigid 3d shape retrieval. In *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2015.

[Maturana and Scherer, 2015] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D convolutional neural network for real-time object recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 922–928, 2015.

[Qi *et al.*, 2016] Charles R. Qi, Hao Su, Matthias Niessner, Dai, and et al. Volumetric and multi-view CNNs for object classification on 3D data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[Qi *et al.*, 2017a] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[Qi *et al.*, 2017b] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems 30*, pages 5099–5108. 2017.

[Sfikas *et al.*, 2017] Konstantinos Sfikas, Ioannis Pratikakis, and Theoharis Theoharis. Ensemble of panorama-based convolutional neural networks for 3D model classification and retrieval. *Computers & Graphics*, 2017.

[Shi *et al.*, 2015] Baoguang Shi, Song Bai, Zhichao Zhou, and Xiang Bai. DeepPano: Deep panoramic representation for 3-D shape recognition. *IEEE Signal Processing Letters*, 22(12):2339–2343, 2015.

[Simonovsky and Komodakis, 2017] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3693–3702, 2017.

[Su *et al.*, 2015] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *IEEE International Conference on Computer Vision*, pages 945–953, 2015.

[Wang *et al.*, 2017] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *ACM Transactions on Graphics*, 36(4):72, 2017.

[Wang *et al.*, 2018] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Bronstein, and et al. Dynamic graph CNN for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018.

[Wu *et al.*, 2015] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, and et al. 3D ShapeNets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.