https://doi.org/10.2352/ISSN.2470-1173.2018.10.IMAWM-419

3D Shape Retrieval using Volumetric and Image CNNs: A Meta-Algorithmic Approach

Ruiting Shao*, Purdue University, West Lafayette, IN, USA
Yang Lei, Jian Fan, Jerry Liu, HP Labs, Palo Alto, CA, USA

Abstract

We propose a deep learning method to retrieve the most similar 3D well-designed model that our system has seen before, given a rough 3D model or scanned 3D data. We can either use this retrieved model directly or use it as a reference to redesign it for various purposes. Our neural network consists of 3 different neural networks (sub-nets). The first neural network deals with object images (2D projection) and the other two deals with voxel representations of the 3D object. At the last stage, we combine the results of all 3 sub-nets to get the object classification. Furthermore, we use the second to last layer as a feature map to do the feature matching, and return a list of top N most similar well-designed 3D models.

Introduction

Background

3D object classification and identification is a fundamental challenge of computer vision in 3D object creation, printing, and digital manufacturing. For example, for a target object, we can first scan it to get a 3D point cloud and then transfer to a CAD model. From the well-designed CAD models, we can find the most similar one for our particular application.

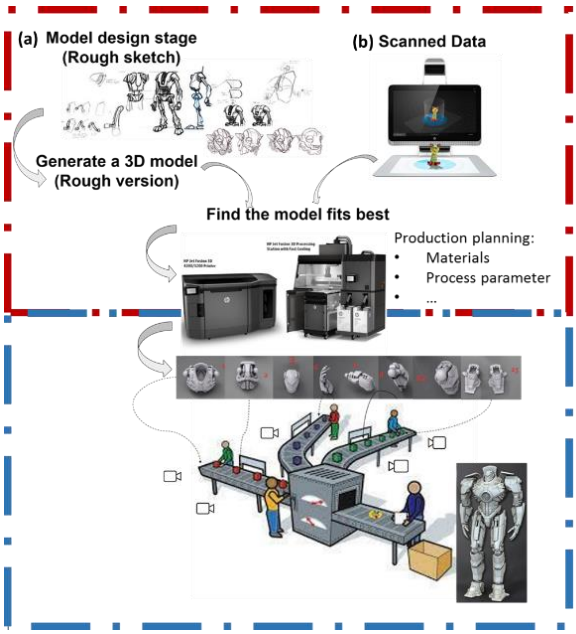

Figure 1: An illustration of how 3D shape classification and retrieval may be used in manufactory. If we want to manufacture a toy robot, we can (a) start with rough sketches for different parts and then use the sketches to generate 3D model, or (b) use scanned 3D data directly from some exiting object. Based on the achieved raw 3D model, it can apply the system to find the best fit model, then the individual parts of the toy robot are manufactured. Later on, cameras will be disposed as manufacturing facilities to track these individual parts through post-processing and enable automatic assembly by using this system.

* This work was done when Ruiting was an intern at HP Labs.

Meanwhile, we are living in a 3D world. Being able to correctly identify 3D content in the real world plays a key role in robotic navigation and 3D digital manufacturing.

Typically, there are two effective ways for 3D object classification: manually crafted features and learned features. Manually crafted features, which are all called shape descriptors, are extracted by applying human defined rules. Typically, different types of shape descriptors have difference feature preferences, like different types of geometry essence. But they are not robust enough for all data types. The complexity of computing 3D shape descriptors also varies, because it is mainly determined by the complexity of the 3D model.

In the meantime, many methods are using computer learned features. Currently, 2D image classification algorithms based on Convolutional Neural Network (CNN) are quite mature, thus extending them into 3D object classification problem is natural. But it needs to be carefully considered what types of 3D representation should be used, as representation schemes will affect the system performance. To achieve better performance on CNN, we need to preserve sufficient 3D information while reducing the computation complexity. It is the major challenge and the reason why some researcher [1] proposed using multiple 2D views to train a CNN. Recently, plenty of 3D meshed surface models are available in various fields, like IKEA dataset [2], ShapeNet [3], Princeton Shape Benchmark (PSB) [4] and Princeton ModelNet [5]. As a result, more attention has been paid to dealing with either 2.5D information (RGB-D image) or the 3D model directly. Some researchers [6, 7] developed the idea of using volumetric representation for 3D object and fed it into a CNN acting as an automated feature learning method.

Related work

Shape descriptors

There are several works that try to design shape descriptors for 3D objects based on different 3D data representations. For the same target object, using point-cloud, meshes, or voxel representation will result in different shape descriptor. Also, the resolution and sparsity of the 3D representation will lead to quite different results.

Generally, the shape descriptors will be classified into two types: global descriptors and local descriptors. In the past, the surface normal and curvature were used as a kind of shape descriptor. Recently, more types have been designed, like Light Field Descriptor [8], Fourier descriptor [9], Heat Kernel Signature (HKS) [10], Scale Invariant Heat Kernel Signature (SI-HKS) [11], etc. Then we can apply traditional machine learning method, like PCA, to perform 3D objects classification. But the handcrafted shape descriptors may not robust to all different type of 3D objects. Thus, we want to find a way to automatically learn 3D features, which leads to applying a neural network to the problem. For

example, the DeepSD method [10] first does some preprocess (HKS) on input shapes to get a rough shape feature, and then the shape features are fed into two deep neural networks to get the deep shape descriptor.

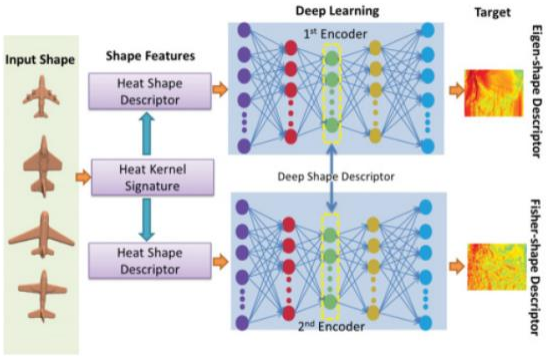

Figure 2: The pipeline of learning 3D Deep Shape Descriptor (DeepSD) [10].

Each image for the multi-view representation is first passed to an image-based CNN (CNN_1) separately, then aggregated at the view-pooling layer, and at last go through another image-based CNN (CNN_2). All the parameters for the first part (CNN_1) are shared among all views.

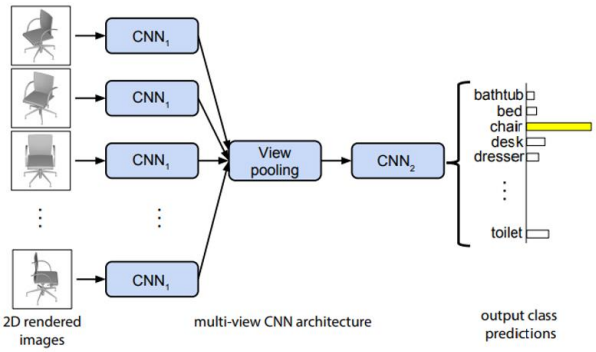

Figure 3: Illustration of multi-view CNN architecture [1].

MV-CNN

Multi-view Convolution Neural Network (MV-CNN) is the state-of-art 3D object classification method based on 2D projections. For each 3D object, multiple 2D projections are generated. Typically, it has 2 different camera set-ups. In the first camera setup, we assume all 3D objects are in the upright orientation along a consistent axis (e.g. z-axis). Under this assumption, it will generate 12 render views with 30 degrees from the horizontal line. For the 2nd camera setup, we do not assume the 3D objects are consistently upright. In this case, it will place 20 virtual cameras at the corners of an icosahedron and generate 4 render views for each camera. Thus, each 3D object will yield total 80 views.

In this paper, we proposed a method that combine several sub neural networks together to achieve better results in 3D object classification and identification. The flowchart of our method is shown in Figure 4.

Method

The main contributions of this paper are summarized as follows:

1. Data augmentation – how we rendered the 3D objects into images and voxels to ensure that the rich 3D information is

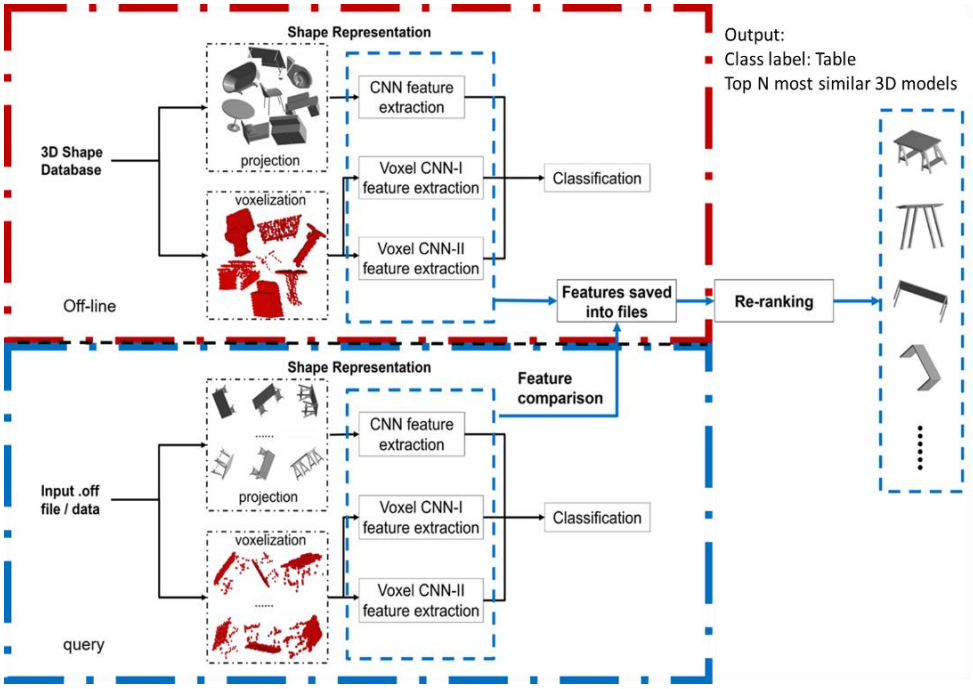

Figure 4: The entire flowchart for the offline training and real-time classification and retrieval process. Three neural networks are designed to process the 2D projection and 3D voxelized data separately. Data generated from the same viewing angle are put through the neural networks at the same time. The class/model that gets receives the highest number of votes among all viewing angles is selected as the final output.

preserved through 2D projection images and low-resolution data (voxels).
2. Provide flexibility on the number of views used during classification and retrieval. We pushed the technology one step further towards requirements for industry adoption.
3. Majority vote – the method that helps to make final decision based on outputs from multiple views.

Dataset

We applied our method on Princeton ModelNet dataset [5] which contains 127,915 CAD Models within 662 object categories. It also has two widely used subset, ModelNet10 and ModelNet40. ModelNet10 contains 10 popular object categories with 3991 training CAD models and 908 testing models. All these models have been manually aligned according to a certain predefined pose. ModelNet40 contains 40 distinct classes with a total of 9843 training models and 2468 testing models. We use the same train-test split provided by the authors and mainly test our method on the two subsets. The 3D CAD models in ModelNet dataset are in the form of polygon mesh, containing coordinates and all the vertices in the mesh and the IDs of the nodes forming a polygon.

For both ModelNet10 and ModelNet40, we randomly selected 100 training objects for each class to do the training section in our neural network. Also in our method, we do not require all the objects to be manually aligned.

Data Representation

Image representation: for a given view point, the 3D object is projected into a 2D image with a fixed light source.

Volumetric representation: the 3D object is represented by binary voxels with specific resolution – 1 if the voxel is valid and 0 otherwise. The volumetric representation is adapted to the size of the polygon mesh.

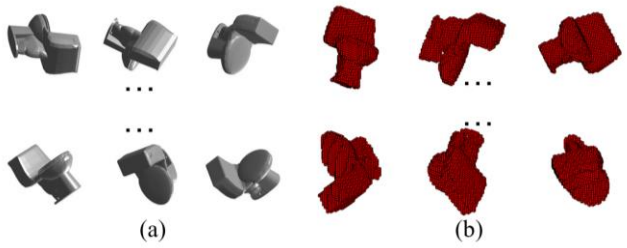

Figure 5: (a) Image representation of 3D object, here we choose the image size to be 224 pix × 224 pix. (b) Volumetric representation of 3D object, here we choose the voxel size to be 30 unit × 30 unit × 30 unit.

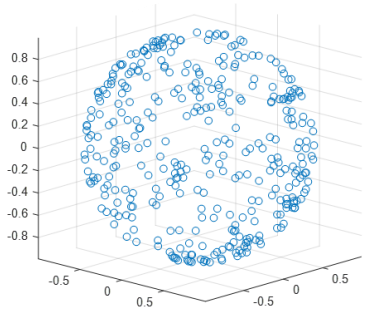

Figure 6: An example of how 360 random viewpoints distributed evenly on a unit sphere.

Data augmentation

In our method, we use two methods to represent the 3D object. For a 3D object, different viewpoints results in different 2D projection images, as shown in Figure 5 (a). And for the allotropic voxels, if the resolution of volumetric representation is high, they may look similar; but if the resolution is low, it may vary. In our method, the higher resolution of volumetric, the more time-consuming the system is. Thus, we choose to use 30×30×30 voxels to represent the target objects as shown in Figure 5 (b). For the entire system we randomly generate 360 different viewpoints which are evenly distributed on a sphere and pointing toward the centroid of the mesh as shown in Figure 6. Furthermore, the number of viewpoints for our method is not fixed and can be whatever the number we choose. Typically, we choose 36, 60, 90, 120, and 360.

Majority vote

Our work is inspired by FusionNet [12], which uses similar data representation and neural network design. It also tries to combine three different neural networks to improve the classification accuracy. One shortcoming of FusionNet is that once the neural network has been trained, the number of views used for 3D classification and retrieval is fixed. However, in practice the number of views that we can generate may be restricted; or under some circumstance, we need more views to help us better identify the objects of interest. Therefore, for the 3D digital manufacturing application, we cannot use FusionNet directly. In our approach, we use majority vote at the end of neural networks instead of view max-pooling in sub neural networks. Imagine, for each view, we get an image and corresponding voxel data, then we feed them into the three CNNs. At last, we fuse the outputs from all three sub-CNNs and make the final decision. The majority vote happens after the network fusion. For each view, we get a class ID. Then the majority vote counts how many votes each class gets from all views. The one with highest number of votes is the final agreed output class. By using this method, the number of views required for 3D objects will be more flexible, and the number of training parameters will be much smaller (in the max-pooling layer, it involves massive calculation and is very time-consuming).

In the training stage, we still generate images and voxels from multiple views for 3D objects to do the training. In the testing stage, we do a majority vote at the end of the softmax layer. In other words, in the classification step, the final class ID is the one that gets most of the votes. In the identification step, we treat the second to last layer as features and calculate its distance to the features from the training section within the decision class. Thus, for each view, we have a closet view of some object in the training set. Having done this for each view, we can find an object in the decision class, which gets most of the votes from all views.

This step dramatically increases the accuracy of object classification and retrieval. For example, in 120 views/120 views of training/testing task, the classification accuracy without majority vote is 81.15%, and the classification accuracy with majority vote is 93.03%.

Image-based CNN

In this image-based neural network, we apply a typical 2D image-based CNN to achieve our goal. In this work, we choose AlexNet as our basic neural network and perform some modification to make it more suitable for our dataset. We render multiple 2D projections of CAD models, which is represented in polygon mesh. Since the CAD model does not contain any color information, the projection is just a gray scale image. The direction of light source

for the 3D object when doing the projection is fixed compared to the viewpoint.

Volumetric CNN

We apply two well defined CNNs for volumetric data which form a large feature space in the off-line training stage. When we do the query part, we get a final result by using the majority vote method across multiple views. Similar to the image-based CNN, we still use 2D convolution to aggregate useful information across a direction of the object (this direction is fixed along the entire work).

Network 1: VCNNI

This volumetric CNN is trying to mimic the working principle of 'X-ray scanning', using a kernel with size, k x k x length, along a fixed direction. Here we call it 'anisotropy' convolution layer. The size, k, may be selected based on the input voxels size. In some examples, k is 1, 3, or 5. It consists of three 'anisotropy' convolution layers and two fully connected layers. The final layer works as a classifier, the size should be the same as the number of classes in training dataset.

Network 2: VCNNII

This neural network is the same as the FusionNet VCNNII [12], which is inspired by the inception module in GoogleLeNet. It concatenates output from different kernel size, so the key feature across multiple scales will be maintained. The filters we use are of size 1x1, 3x3 and 5x5. The 1x1 kernel tries to abstract information in the receptive field and encode a higher representational power without much additional computation cost. Since our volumetric data is not that big, and we don't need to reduce the computational complexity, this neural network contains 2 inception modules, followed by a convolution layer and 2 fully connected layers.

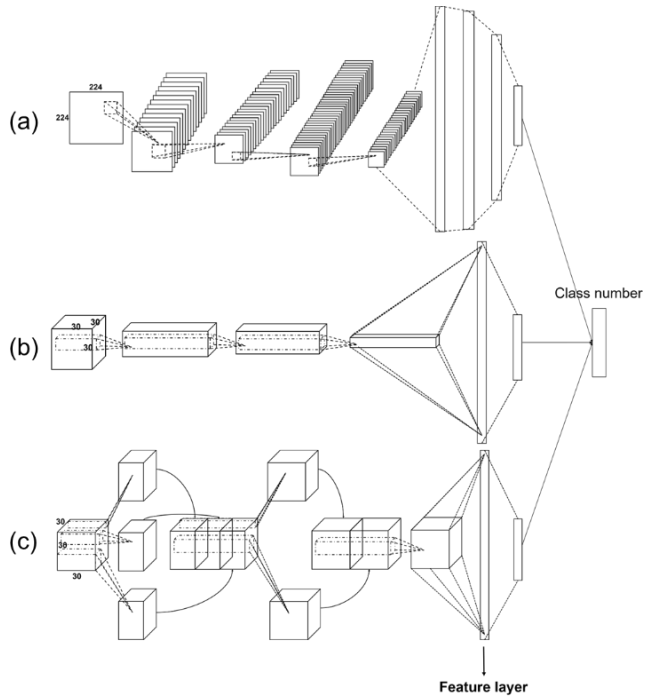

Figure 7: the entire structure of meta-NN. (a) Image input CNN, learning the shape information; (b) Voxel input CNNI, mimicking the principle of x-ray, learning the density of 3D objects; (c) Voxel input CNNII, applying inception module, using different kernel size to learn the invariant feature of 3D objects.

Result

In this paper, we run the classification task on ModelNet10. Each object in training and testing set is rendered from 120 viewing angles to generate the 2D images and corresponding voxel representations. As the classification accuracy shown in Table 1, our proposed Meta-NN approach achieves the accuracy of 93.03%, outperforming the Panorama-NN [13] and 3DShapeNets [5]. The performance is very close to FusionNet [12] and yet offers more flexibility for industry applications. FusionNet uses 20 images and 60 voxels as input with a priori that all the input 3D objects must be manually aligned according to certain predefined pose. Figure 8 shows the confusion matrix of the classification task on ModelNet10. It seems that the classification accuracy of "table/desk" and "nightstand/dresser" classes is a bit lower than the other classes. When we examined the dataset, we found that some 3D objects in "night stand" class is similar to the "dress" class, the same for "table" and "desk" classes.

Table 1: Classification accuracy comparison among different methods.

# of viewing points	Algorithm	ModelNet10 classification (accuracy)
120	Meta-NN (proposed method)	93.03%
N/A	Panorama-NN	91.1%
N/A	3DShapeNets	83.5%
20 images & 60 voxels	FusionNet	93.11%

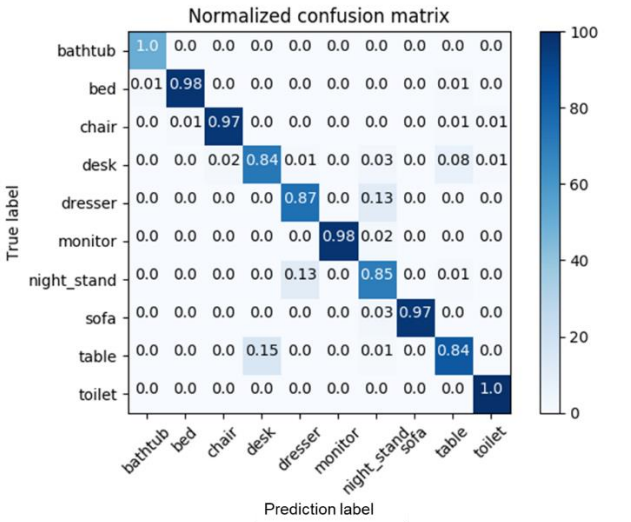

Figure 8: The confusion matrix for classification accuracy of Meta-NN (proposed method) running on ModelNet10 with 120 viewpoints.

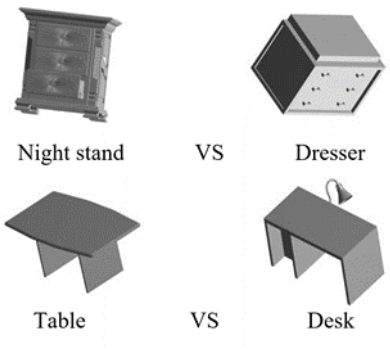

Figure 9: Similar 3D object examples from night stand / dresser classes (upper row) and table / desk classes (lower row) that confused our algorithm.

We also do the comparison between the different combination of the number of training viewpoints and testing viewpoints. From table 2, we find that the combination of the number of training and testing viewpoints can be very flexible. The best performance is achieved using 120 views for both training and testing. This gives us a hint that 120 views are adequate to acquire almost all useful information to discrete 3D object for ModelNet10. The number of views may increase when the 3D objects get subtler.

Table 2: Classification accuracy of different combination of the viewpoints between training and testing stage.

# of viewpoints		Test		
		36	120	360
Train	36	91.59%	87.61%	87.61%
	120	91.92%	93.03%	92.37%
	360	90.93%	91.92%	91.84%

Furthermore, we evaluated the system's performance on 3D object retrieval. In this case, the 360 viewpoints of the same 3D model are divided into training set and testing set. In this manner, we will have ground truth for each testing sample. The retrieval accuracy of the method for the bathtub class is 87.13%, for example.

Conclusions

Since working with the direct 3D data has become more popular, we propose a way to do this task; and based on the results, it seems work well. Our method does not have restrictions on the number of viewpoints and does not require the 3D objects to be manually aligned.

Also, we have several directions to explore in the future. One is to experiment with other types of sub neural networks dealing with either 2D image or 3D voxels. What if the input 3D object is just a partial object? We want to improve the accuracy of each subnet, whatever the integrity of the 3D object. Another direction is to explore different resolutions for the voxels representation. The resolution will have a significant impact on the volumetric represent of the 3D object. If we choose a lower resolution, the outline of the object will be conserved but we will lose detail information. On the other hand, higher resolution will require higher computation complexity and memory consumption.

References

[1] H. Su, S. Maji, E. Kalogerakis and E. Learned-Miller, "Multi-view Convolutional Neural Networks for 3D Shape Recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015.

[2] J. J. Lim, H. Pirsiavash and A. Torralba, "Parsing IKEA Objects: Fine Pose Estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[3] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," arXiv:1512.03012 [cs.GR], 2015.

[4] P. Shilane, P. Min, M. Kazhdan and T. Funkhouser, "The Princeton Shape Benchmark," in *Shape modeling applications*, 2004.

[5] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang and J. Xiao, "3D ShapeNets: A Deep Representation for Volumetric Shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[6] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan and L. J. Guibas, "Volumetric and Multi-View CNNs for Object Classification on 3D Data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[7] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[8] D. Chen, X. Tian, Y. Shen and M. Ouhyoung, "On Visual Similarity Based 3D Model Retrieval," *Computer graphics forum,* vol. 22, no. 3, pp. 223-232, 2003.

[9] D. Saupe and D. V. Vranić, "3D Model Retrieval with Spherical Harmonics and Moments," in *Joint Pattern Recognition Symposium*, Berlin, Heidelberg, 2001.

[10] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu and . E. Wong, "3D Deep Shape Descriptor," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[11] M. M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[12] V. Hegde and R. Zadeh, "FusionNet: 3D Object Classification Using Multiple Data Representations," arXiv:1607.05695 [cs.CV], 2016.

[13] K. Sfikas, T. Theoharis and I. Pratikakis, "Exploiting the PANORAMA representation for convolutional neural network classification and retrieval," in *Eurographics Workshop on 3D Object Retrieval*, 2017.

Author Biography

Ruiting Shao *received her BS in Biomedical Engineering from Beijing Institute of Technology (2014) and is pursuing her PhD degree in Electrical and Computer Engineering from Purdue University. Her work is focused on image forgery and tracking the trajectory of 3D object.*

Yang Lei *received her BS in Electrical Engineering from Sichuan University (2009) and her PhD in Electrical Engineering from Purdue University (2014). Since then she has worked as a researcher at HP Labs in Palo Alto, CA. Her work has focused on 3D imaging systems and applications in object recognition and tracking.*

Jian Fan *is currently a principal research engineer at HP Labs. Jian holds a Ph.D. degree in computer engineering from the University of Florida. His research interests include image processing, document image processing and computer vision.*

Jerry Liu *is currently a Senior Research Manager at HP Labs, Palo Alto, California. His research interests are in data analysis and sensor systems, with over 20 issued patents in these fields. Jerry earned his Master of Engineering degree and Bachelor of Science degree in Electrical Engineering from Cornell University, Ithaca, NY.*