IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 5, 2020, accepted July 5, 2020, date of publication July 15, 2020, date of current version July 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009333

Multi-View Tree Structure Learning for 3D Model Retrieval and Classification in Smart City

AN-AN LIU, (Member, IEEE), ZHENLAN ZHAO, WENHUI LI, AND DAN SONG
School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

Corresponding authors: Wenhui Li (liwenhui@tju.edu.cn) and Dan Song (dan.song@tju.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (61772359, 61872267, 61902277), the grant of Tianjin New Generation Artificial Intelligence Major Program (19ZXZNGX00110,1 8ZXZNGX00150), the Open Project Program of the State Key Lab of CAD & CG, Zhejiang University (Grant No. A2005, A2012).

ABSTRACT The application of digital products in smart city results in ever-increasing 3D model data and how to obtain the relevant 3D model becomes a crucial issue. In this paper, we propose the Multi-View Tree Structure (MVTS) learning for 3D model retrieval and recognition. MVTS contains three key consecutive modules. Firstly, the visual feature learning module extracts the visual features of multiple views. Then, we design a score matrix to estimate the value of contextual information between view pairs. Based on the score matrix, a maximum spanning tree is constructed to further explore the contextual information within multiple views. Then, we utilize the bidirectional Tree-LSTM to encode the contextual information among views and the spatial information of tree structure and optimize the tree parameters. After that, the tree attention strategy is adopted to explore the importance of each view. Comparing to existing methods, our proposed method explores the spatial information of 3D model without the requirement of specific camera settings, which is more suitable for real applications. Moreover, our method jointly realizes the feature learning, view-wise contextual information and tree spatial information encoding and view importance estimating, which enhances the discrimination of the 3D model representation. Extensive experimental results on Modelnet40 and ShapeNetCore55 demonstrate the superiority of our method.

INDEX TERMS 3D model retrieval, multi-view representation, long short-term memory, smart city.

I. INTRODUCTION

With the development of digital and smart city, massive and complex data are produced by various applications, e.g, unmanned vehicle, intelligent manufacturing and smart transportation. Due to the sophisticated structure, modality and appearance of data, it is difficult to process these data by using regular management approach or software. Therefore, it is essential to design efficient mechanisms to learn and represent these data for intelligently management and application within the smart city environment.

A. MOTIVATION AND OVERVIEW

Three-dimensional (3D) model data is an important information carrier in smart city and it gets a lot of attentions because of the convenience and effectiveness to describe the virtual industrial products. In recent years, the development of 3D modeling technology and low-cost acquisition equipment (Figure.1) have brought ever-increasing 3D data. Because of its advantages in shortening design cycle, reducing new product development risk and investment cost, 3D data has received much attention and wide application. Thus, how to effectively retrieve and classify 3D models in the dataset has become a crucial issue.

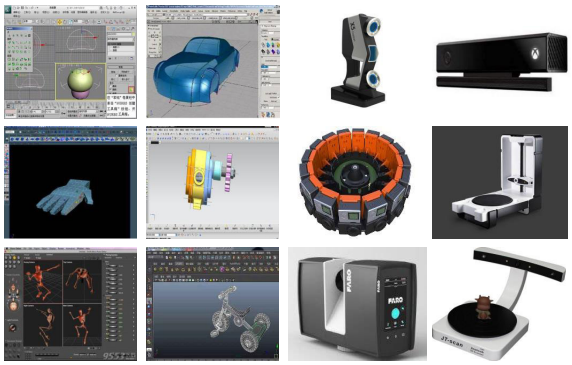


FIGURE 1. 3D modeling softwares and low-cost acquisition equipments.

The associate editor coordinating the review of this manuscript and approving it for publication was Long Xu.

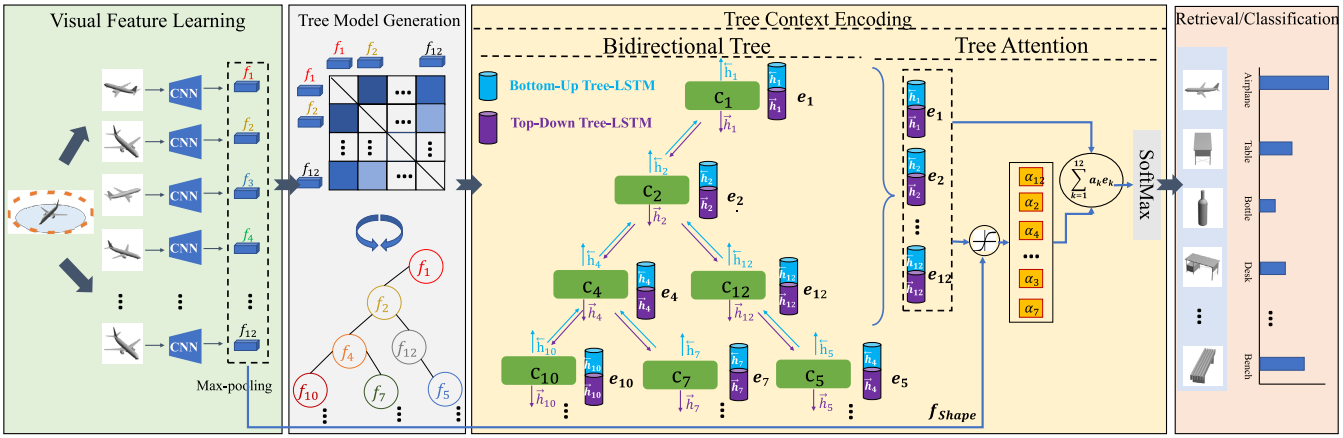


FIGURE 2. The flowchart of our proposed method is demonstrated as above, which mainly contains three consecutive modules: visual feature learning, tree model generation and tree contextual encoding. Firstly the feature of model views are extracted independently by CNN. In tree model generation module, a score matrix is calculated to measure the contextual information value of view pairs. Based on this matrix, the tree model is constructed recursively. Then, we adopt the bidirectional Tree-LSTM to encode the contextual and spatial information. The bidirectional (top-down and bottom-up) Tree-LSTM enables the hidden state to obtain richer and deeper contextual information inside the view sequence. All output hidden states of Tree-LSTM and the shape feature that is obtained by view-level max pooling will be inputted into the tree attention module together, where the weights of all hidden states will be calculated. The weighted sum of hidden states after applying the tree attention is employed as the final shape descriptor for the following retrieval or classification task.

The 3D model retrieval & recognition is broadly classified into two groups, the model-based method and the view-based method. The early research mainly focuses on the model-based method. These methods try to design the handcraft features to extract the surface geometry, volume and skeleton information to represent the 3D model data [1]–[3]. However, the performance of these method is limited by the complex topology structure and the diversity of categories. Moreover, inspired by the breakthrough progress made in the field of 2D image and videos, many researches focus on the view-based methods, which obtain multiple views of the 3D model with virtual camera set [4]–[6]. View-based methods usually extract representative views from the view set or pooling the multiple views into one view to represent the corresponding 3D model and the performance heavily depends on the ability of view selection strategy. On the other hand, to explore the sequential information of multiple views, a lot of work mine multi-view information by using the LSTM model or design the strategy to group the multi-view into subset [7]. However, these methods mainly have several critical problems: 1) It might lose the useful information by adopting max-pooling strategy to fuse multi-view representations. 2) It is sensitive to the order of views when using the sequence-based methods to learn the 3D model representation, such as Long short-term memory (LSTM), which is difficult to model the spatial characteristic of the 3D model. 3) Due to the difference in rendering angle, different views have different importance for shape representation and they should be given different weights.

To address the above problems, we propose a Multi-View Tree Structure Learning for 3D Model Retrieval and Classification (MVTS). The pipeline of our method is show in Figure. 2. Our method has three key modules. Firstly, we adopt deep CNN to extract the multi-view features. Then, we design a score matrix in which each cell indicates the contextual information value of the corresponding view pair. Based on the score matrix, the tree structure model is proposed to model the spatial characteristic information of 3D model and the bidirectional Tree-LSTM is adopted to encode the contextual and spatial information. Moreover, we adopt the tree attention strategy to model the importance of each view in model representation. In this way, our method can effectively avoid the negative influence caused by different view orders and model structure information based on multiple views. The weighted sum of all hidden states are utilized as the final model feature, which can be used for model retrieval based on distance metric or model classification after being passed through the softmax layer.

B. CONTRIBUTION

The main contributions of this paper are summarized as follows:

- We propose a novel tree structure learning method for view-based 3D model retrieval and recognition. Different from depending on the sequential information of multiple views, our method designs the tree structure to explore the spatial information without the requirement of specific camera settings, which is suitable for real applications.
- Our method introduce the bidirectional Tree-LSTM to encode the contextual information between view pairs and the spatial information of the tree structure. Besides, the tree attention strategy is adopted to estimate the view importance. Our model jointly realizes the feature learning, view-wise contextual information and tree spatial information encoding and view importance estimating,

which helps the network to obtain more discriminative representation of 3D models.
- We conduct extensive experiments on two large-scale 3D model datasets and the experimental results demonstrate the superiority of our method.

The rest of this paper is organized as follows. We overview the existing related work in Section II and detail the proposed method in Section III. The Section IV describes the experiment setting and Section V discusses the experimental results. Finally, we conclude this work in Section VI.

II. RELATED WORK

Generally, the existing methods for 3D model retrieval and classification can be grouped into model-based and view-based methods. In this section, we will introduce some representative methods of both categories as follows.

A. MODEL-BASED 3D MODEL RETRIEVAL

3D models have spatial structures and complex geometries with diverse variations. In model-based 3D model retrieval methods, the 3D model feature is extracted based on the model characteristics directly, such as grid, voxelized 3D network, mesh and point cloud. Overall, the geometric information and the topological or skeletal graph structures of a 3D model are popular model information adopted in model-based methods. [8]. The distance of random surface points, angle, area and volumes can be utilized for similarity measurement between 3D models. Wu *et al.* [9] designed the 3D convolutional restricted Boltzmann machine to learn the global features from voxelized 3D models. Furuya and Ohbuchi [10] designed a novel aggregation network to extract the rotation-invariant 3D local features and unified these features in a single architecture. Tabia and Laga [11] used the descriptor's covariance matrix to encode different feature forms and types into a single compact descriptor to represent 3D models. Then the Riemannian manifold is used to calculate the geodesic distance of the paired models. Charles *et al.* [12] proposed to the make use of point cloud descriptions for 3D models for classification. Dominguez *et al.* [13] proposed the graph-based method to apply transfer learning strategy on 3D point cloud data, which was demonstrated to have the ability to represent the unforeseen test models. You *et al.* [14] integrated point clouds and multi-view data into 3D model recognition. Shi and Rajkumar [15] designed a graph neural network, named Point-GNN, to predict the category and model of the object that each vertex in the graph belongs to. Although the model-based methods can make full use of the structure of 3D objects, they are computationally expensive due to the complexity of 3D models, especially when the model reconstruction are required in practical application. Moreover, the regular poor reconstruction performance and high computational complexity limit their flexible application in real applications.

B. VIEW-BASED 3D MODEL RETRIEVAL

Different from model-based methods in which the 3D model information is required directly, view-based methods employ the 3D model Information indirectly by view rendering procedure. Besides,multi-view representation can be achieved by view information fusion [16]. Generally, model views are captured by setting virtual cameras around the model. Then, we can select a representative view feature or fuse all views into a compact descriptor to represent the model. The GIFT [17] is a well-designed 3D shape search engine. In GIFT, the projection rendering and the view feature extraction was accelerated by using CNN with GPU and the embedding of two inverted files (F-IF and S-IF) are adopted to enable real-time retrieval. However, only a single view is utilized in GIFT. Many algorithms focus on utilizing the multi-view sequence to build a compact shape descriptor. One typical view fusing method is the MVCNN proposed by Su *et al.* [18]. In MVCNN, a novel CNN architecture is proposed, where the view-level pooling layer is added after all convolutional layers. The view-level pooling layer can combine multiple view information into a single and compact model descriptor.The view fusion operation can represent the model effectively and provide better retrieval or classification performance. Feng *et al.* [19] introduced a group view CNN framework, named GVCNN, for hierarchically correlated modeling to produce a discriminative and compact 3D model representation. The grouping strategy is used to solve the problem that the inherent hierarchical correlation and distinguish ability between views are not well utilized. In addition to fusing all view information, selecting more important views at first and just ensembling their information is also an effective way. Stereographic Projection Neural Network (SPNet) proposed by Yavartanoo *et al.* [20] is a valid model descriptor learning method. After employing the stereographic projection to transform the 3D volume into 2D planar image, a shallow 2D CNN is presented for object category estimation. At last, view selection and view ensembling are added to enhance the final performance.

To solve the problem of only partial views are obtainable in real applications, Kanezaki *et al.* [21] proposed a novel CNN-based network called RotationNet. The RotationNet takes the multiple views of a model as input and estimates both the pose and category jointly. The pose alignment strategy ensures high accuracy in both model categorization and pose estimation by sharing view-specific descriptor across classes. Besides, RotationNet can learn the viewpoint labels without object labels. In the field of autonomous driving, due to the association of complementary devices such as LiDAR, Beltrán *et al.* [22] proposed an effective LiDAR-based 3D model detection method(BirdNEt) in driving environments. Firstly, in order to conduct bird's eye view projection, the laser information is projected into a new cell encoding. Then both the object position and its course are estimated by CNN. At last, 3D oriented detections are calculated in a post-processing phase. BirdNet also introduces

pedestrians and cyclist detection with only BEV images and can be applied in real-time scenes. Chen *et al.* [23]proposed Multi-View 3D networks (MV3D), where a sensory-fusion framework is proposed to predict oriented 3D bounding boxes with both the RGB images and LIDAR point cloud as input. The sparse 3D point cloud can be encoded with a compact multi-view representation. In addition, they introduced a deep fusion scheme to combine region-wise features and ensure the interactions between intermediate layers.

III. METHODOLOGY

In this section, we firstly overview the whole framework of our method. Then, we introduce each module of the framework in detail.

A. OVERVIEW

Our framework contains three consecutive modules as shown in Fig 2.

- **View Feature Learning.** After obtaining the view set for each 3D model, we utilize this module to encode the multi-view information and extract the visual feature for each view.
- **Tree Model Construction.** This module designs a score matrix where each cell indicates the relevance between the corresponding view pair. Based on the score matrix, a maximum spanning tree will be constructed to process the multi-view features and the tree structure will be used to initialize the next module.
- **Tree Model Context Encoding.** This module utilizes the tree structure to discover the contextual information of multiple views. Firstly, the bidirectional Tree-LSTM is used for contexual and spatial information encoding. Then the tree attention mechanism is designed to explore the importance of each view and get the unified representation by merging the multi-view contextual information with different importance.

Our framework can be trained in an end-to-end manner and the details of above three modules are descried as follows.

B. VIEW FEATURE LEARNING

In order to obtain the multi-view representation of 3D models, we set a group of virtual cameras around the model to capture its rendered views based on Phong reflection model [18]. By changing the interval angle between virtual cameras, the number of rendered views of each model can be changed as well. For instance, when the angle is set to 30°, the view number will be 12, which is employed as the default setting in all our experiments. Given N 3D models, the view set for all 3D models is denotes as $V = \left\{v_i^t|_t = 1, \ldots, S\right\}_{i=1}^{N}$, where S is the view number. v_i^t denotes the t-th view of model i. Due to the simpler architecture, less parameters and competing performance of Alexnet, it is employed as the backbone network to extract view feature in our method. AlexNet is composed of five convolutional layers, $conv_1 - conv_5$, and three fully connected layers, $fc_6 - fc_8$. For retrieval and classification task, the output of fc_7 is usually used as the visual representation of individual views. We utilize the $F = \left\{f_i^t|_{t=1,\ldots,S}\right\}_{i=1}^{N}$ to denote the view feature set, where $f_i^t \in R^{1\times D}$, D is the dimension of visual feature vector.

C. TREE MODEL CONSTRUCTION

Long Short-Term Memory (LSTM) network is a representative and special type of recurrent neural network (RNN). It has a more complex and precise computational unit so that the sequence information over time can be preserved. LSTM is mainly designed to solve the problem of gradient disappearance and gradient explosion during long sequence training and performs well on various sequence modeling tasks. However, the structure of normal LSTM a linear chain, which is too simplistic to capture complex spatial information. Therefore, we developed LSTM with tree structure rather than chain structure so that the information from each child can be incorporated selectively. It overcomes the limitation of LSTM architecture that only allows information to be spread in sequential order. The input gate, output gate, a memory cell and the hidden state form a basic unit of Tree-LSTM. The states of all possible child units decide whether the gating cell and the memory cell should update or not. The hidden state of Tree-LSTM is composed from the input vector and arbitrary number of child units instead of the input of the current time step and the hidden state of preceding time step. Therefore, richer network typologies are obtained to enable each Tree-LSTM unit to incorporate richer and more comprehensive information from multiple child units.

The model views are obtained by rendering the model from different perspectives. Thus, the views are related to each other because they can be regarded as progressive sequence. For example, the views rendered by adjacent virtual cameras are similar, which makes the corresponding view pair contains richer contextual information. Thus, before constructing the tree, a score matrix M need to be learnt to estimate the value of the contextual information between view pairs. In our method, the score matrix is computed based on view pair information, formulated as:

$$M_i(x, y) = D(f_i^x, f_i^y) \tag{1}$$

where f_i^x and f_i^y are the features of v_i^x and view v_i^y, where v_i^x and v_i^y are the x-th and y-th view of model i respectively and they are the output of fc_7 with the dimension of 4096. D is the distance for similarity measure, which can be Euclidean distance. Since M as a $S \times S$ symmetric adjacency matrix, a maximum spanning tree can be constructed by adopting the Prim's algorithm [24]. Firstly, the vertices set of the graph are separated into two disjoint subsets. One contains the vertices that have been included in the maximum spanning tree while the other contains the vertices not yet included. In our implementation, all view of a model are utilized as the vertices of graph and the edge weight between two vertices is the corresponding value of matrix M. The maximum spanning tree will be constructed recursively by comparing the weights of all edge between the nodes included in the tree and the nodes

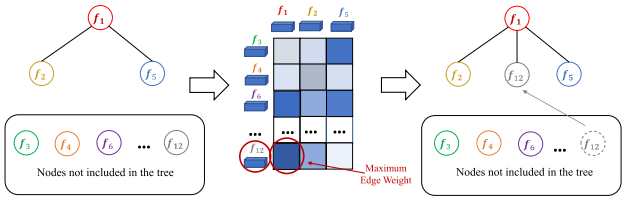


FIGURE 3. Illustration of tree generation procedure. Given the root node or part nodes of the tree, we connect the node of the tree by utilizing maximum weight between the nodes and current tree. Then the node that corresponds to the maximum weight will be connected to the tree. The edge weight is the corresponding cell value of the score matrix M that indicates the contextual information between view pairs.

not included in the tree. Then the node of the largest weight edge will be selected and connected to the corresponding tree node, which will update the spanning tree as well as the components of the two subsets. The largest edge weight indicates the maximum correlation and validity. Figure. 3 shows how the maximum spanning tree is constructed. This algorithm ensures the final spanning tree to have the maximum edge weight sum regardless the selection for the initial root node, which will be verified experimentally later.

The resultant maximum spanning tree is a sparse graph with multi-branch and merely a single type of connection exists, which is inappropriate to distinguish the hierarchical relationship and the parallel relationship when it encodes the contextual information. Therefore, we convert the tree into a Left-Child Right-Sibling (LCRS) tree [25]. LCRS tree is a different kind of an n-ary tree in which a tree node only refers to two nodes at most, i.e. its first (leftmost) child and its immediate next sibling, rather than refer to its every child node. The leftmost child node will become the left branch while the immediate next sibling node will become the right branch. The left branch corresponds to the hierarchical relationship between nodes and the right branch corresponds to the parallel relationship between nodes. The transformation from original spanning tree to LCRS tree is revealed as Figure. 4 below. Compared with chain structure and the original spanning tree, the LCRS tree is more efficient and dynamic. In addition, with the hierarchical and parallel information, the tree structure can deeply mine the structural information of 3D model.

D. TREE MODEL CONTEXT ENCODING

1) BIDIRECTIONAL TREE CONTEXT ENCODING

After constructing the tree model based on the score matrix M, we adopt the bidirectional Tree-LSTM to encode the contextual information among model views and the spatial information of the maximum spanning tree structure. In the LSTM structure, a new hidden state h_n and cell state is generated by last hidden state h_n, last cell state c_n and current input f^x. While in the Tree-LSTM structure, we update these states by using the states of child nodes. For node j of the tree model, the unit equations of the tree structure are formulated as follows:

$$\tilde{h}_j = \sum_{k \in C(j)} h_k \tag{2}$$

$$i_j = \sigma\left(W^{(i)} x_j + U^{(i)} \tilde{h}_j + b^{(i)}\right) \tag{3}$$

$$f_{jk} = \sigma\left(W^{(f)} x_j + U^{(f)} h_k + b^{(f)}\right) \tag{4}$$

$$o_j = \sigma\left(W^{(o)} x_j + U^{(o)} \tilde{h}_j + b^{(o)}\right) \tag{5}$$

$$u_j = \tanh\left(W^{(u)} x_j + U^{(u)} \tilde{h}_j + b^{(u)}\right) \tag{6}$$

$$c_j = i_j \odot u_j + \sum f_{jk} \odot c_k \tag{7}$$

$$h_j = o_j \odot \tanh\left(c_j\right) \tag{8}$$

For a node j in a tree, $C(j)$ denotes the children set of node j and $k \in C(j)$. Each unit of Tree-LSTM consists of a collection of vectors including an input gate i_j, an output gate o_j, a memory cell c_j and a set of forget gate $\{f_{jk} | k \in C(j)\}$ of the children nodes. The entries of the gating vectors i_j, o_j, f_{jk} are in [0,1]. In the transition equations of Tree-LSTM, x_j is the input of node j, σ denotes the logistic sigmoid function and

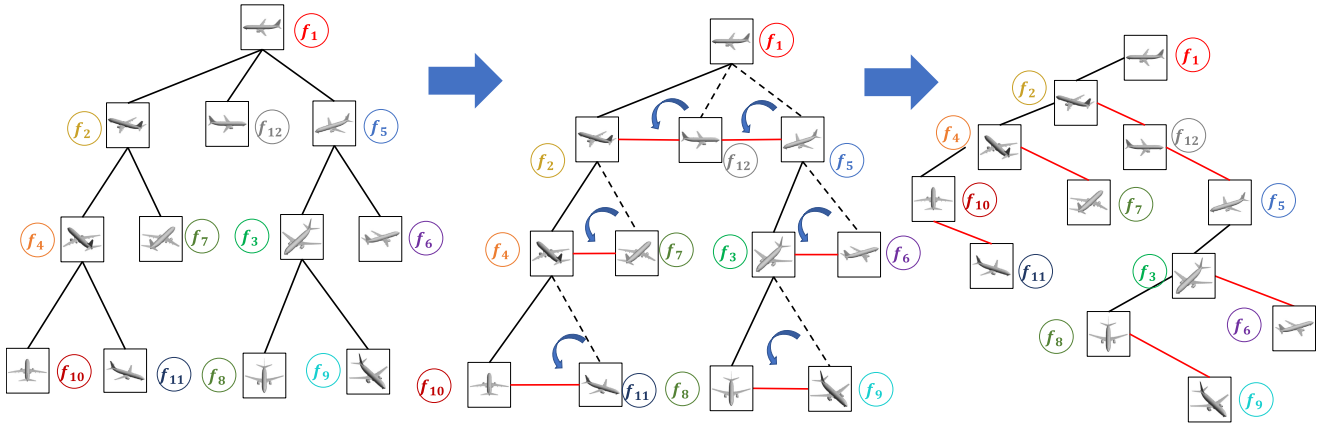


FIGURE 4. Illustration of transforming the maximum spanning tree into the LCRS tree. The left child of the node (denoted in black) in LCRS correspond to the leftmost child node and the right child of the node (denoted in red) in LCRS correspond to the immediate next sibling node in the maximum spanning tree.

$\odot$ denotes the elementwise multiplication. All parameters in these equations need to be learned and can be regarded as the encoding correlations between the input vector x_j, the component vectors of the tree structure unit and the hidden state h_k of all its children nodes. Given the view feature set $F_i = \{f_i^t|_{t=1,\ldots,S}\}$, the contextual encoding can be defined as:

$$E = \text{BiTreeLSTM}\left(\{f_i^t\}_{t=1,2,\ldots,S}\right) \tag{9}$$

where $E = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_S]$ is the encoded view-level context. Each $\boldsymbol{e}_i = \left[\overrightarrow{\boldsymbol{h}}_i; \overrightarrow{\boldsymbol{h}}_i\right]$ denotes the concatenation of the top-down ($\rightarrow$) and bottom-up ($\leftarrow$) hidden states, which ensure the hidden state to capture more comprehensive information.

$$\overrightarrow{\boldsymbol{h}}_i = \text{TreeLSTM}\left(z_i, \overrightarrow{\boldsymbol{h}}_p\right) \tag{10}$$

$$\overleftarrow{\boldsymbol{h}}_i = \text{TreeLSTM}\left(z_i, \left[\overleftarrow{\boldsymbol{h}}_l; \overleftarrow{\boldsymbol{h}}_r\right]\right) \tag{11}$$

where the subscripts p, l, r denotes the parent, left child and right child of node i, respectively.

2) TREE ATTENTION

The above tree architecture treats every node in the tree structure equally. However, multiple views captured from different perspectives of the model contain information of different degrees of importance. Thus, we apply the attention mechanism over the tree components. The attentive Tree-LSTM can assign different weights to all children in the tree structure, i.e. different views will be assigned with different weights based on their contribution to the final model representation.

Based on the hidden states set $e_1, e_2, \ldots, e_S$ produced by the Tree-LSTM encoding of all view features and an auxiliary vector f_{shape}, the wight α_k of each hidden state can be computed by the tree attention mechanism. Firstly, an affine transformation on each hidden state h_k is conducted to calculate a vector m_k, formulated as:

$$m_k = \tanh\left(W^{(m)}e_k + U^{(m)}f_{shape}\right) \tag{12}$$

where $W^{(m)}$ and $U^{(m)}$ are the parameter matrices and f_{shape} is the representation feature vector of all view features which is obtained by the view-level max-pooling. Then, the attention weight α_k of h_k is computed as follows:

$$\alpha_k = \frac{w^T m_k}{\sum_{j=1}^{S} w^T m_j} \tag{13}$$

where w is the parameter vector. After computing the weights of all hidden states, the vector g is calculated by the weighted sum of all hidden states, formulated as:

$$g = \sum_{k=1}^{s} \alpha_k e_k \tag{14}$$

At last, the new hidden state $\tilde{h}$ is obtained by employing another affine transformation on g as follows:

$$\tilde{h} = \tanh\left(\mathbf{W}^{(a)}g + \mathbf{b}^{(a)}\right) \tag{15}$$

As the new hidden state $\tilde{h}$ contains the contextual information between model views and the contribution of each view feature to the model representation, it is utilized as the final model descriptor. On retrieval task, the model pairwise distance will be calculated based on $\tilde{h}$. While on classification task, the softmax function will be used as follows:

$$\hat{y} = \text{softmax}\left(W_y\tilde{h} + b_y\right) \tag{16}$$

IV. DATASET AND EXPERIMENT SETTINGS

In this section, we introduce the dataset, evaluation criteria and other implementation details of our method in detail.

A. DATASET

In order to evaluate the effectiveness of our proposed method, several comparison experiments are conducted on two mainstream and challenging 3D model datasets, ModeleNet40 and ShapeNetCore55.The model samples are shown in Figure. 5 and the introduction of the two datasets are described as follows:

- ModelNet40 [9]: ModelNet40 is the subset of Princeton ModelNet dataset, which is composed of 12311 common CAD models from 40 categories. In our implementation, we follow the official setting and take 9843 models for training and 2468 models for testing.
- ShapeNetCore55 [26]: ShapeNetCore55 is the subset of the complete ShapeNet dataset, which contains 51300 common CAD models from 55 categories. The official splitting way of ShapeNetCore55 is to portion the train subset, validation subset and test subset with 70%, 10% and 20%, respectively. In our implementation, we adopt the official splitting way.

B. EVALUATION CRITERIA

For 3D model retrieval task, each model of test dataset will be employed once as the query. In order to quantify the performance of our proposed method on retrieval task, we employ the Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), F measure (F), Discounted Cumulative Gain (DCG), Average Normalized Modified Retrieval Rank (ANMRR) and Mean Average Precision (mAP) as the evaluation criteria to quantify the retrieval performance [16].

To evaluate our method on the ShapeNetCore55, we following the work [26] to adopt the macro-averaged and micro-averaged measure, where macro-averaged measure employs the evaluation criteria to compute the unweighted average over the entire dataset while the micro-averaged measure utilizes the aforesaid criteria to compute a weighted average which takes the number of 3D models of each category into consideration.

C. IMPLEMENTATION DETAILS

We utilize the Alexnet as the backbone CNN architecture for feature extraction. AlexNet contains five convolutional layers and three fully connected layers. During training, we fine-tune the weights of AlenxNet that has been

FIGURE 5. 3D model samples on Modelnet40.

TABLE 1. Comparison experiments on ModelNet40.

	Method	Classification (Accuracy)	Retrieval (mAP)
Shape-based	3DShapeNets [9]	77.0%	-
	3D-GAN [27]	83.3%	-
	VSL [28]	84.5%	-
	binVoxNetPlus [29]	85.47%	-
	PointNet [30]	89.2%	-
	kd-Networks [31]	91.8%	-
	3D-A-Nets [32]	90.5%	80.1%
	G3DNet [13]	91.13%	-
	PointNet++ [33]	91.9%	-
View-based	DeepPano [34]	77.6%	76.8%
	GIFT [17]	83.1%	81.9%
	Geometry Image [35]	83.9%	53.1%
	Multiple Depth Maps [36]	87.8%	-
	MVCNN [18]	90.1%	79.5%
	PANORAMA-NN [37]	90.7%	83.5%
	Pariwise [38]	90.7%	-
	MVCNN-MultiRes [39]	91.4%	-
	MVTS (Our)	**93.4%**	**87.3%**

pre-trained on ImageNet to update the parameters of our network. We adopt SGD for optimizing and the learning rate is fixed to 5e-5. Besides, we adopt the weight decay strategy during training and employ an element-wise dropout mask in Tree-LSTM. The dimension of hidden state in our method is set to 1024 and the view number is set to 12.

V. EXPERIMENT AND DISCUSSION

In this section, in order validate the effectiveness and efficiency of our proposed method, we conducted several comparative experiments on ModelNet40 and ShapeNetCore55. Firstly, we compare the performance of our proposed methods with several state-of-the-art methods on 3D model retrieval task. Then we explore the influence of the variation in view number to validate the robustness of our method and verify the effectiveness of different modules via the sensitivity analysis on network modules. Besides, we confirm the theory that the selection for tree root node don't affect the final result experimentally. Finally, a visualization experiment is conducted to show the retrieval performance intuitively.

A. COMPARISON WITH THE STATE-OF-THE-ART METHODS

In this section, the experimental results of our proposed method and several representative state-of-the-art methods of both view-based category and model-based category are revealed for comparison. The comparison results of ModelNet40 and ShapeNetCore55 are showed in Table 1 and Table 2, respectively.

On ModelNet40 (Table 1), our method outperforms other benchmark methods in both classification accuracy and retrieval mAP criteria. Specifically, comparing to the existing methods, the classification accuracy and retrieval mAP are improved by 1.6% to 16.4% and 3.8% to 34.2&, respectively. On ShapeNetCore55 (Table 2), we employ the same criteria to evaluate the corresponding performance. We follow the

TABLE 2. Comparison of performance (%) on the ShapeNet55 normal version with official testing method.

Method	microALL			macroALL		
	F1@N	mAP	NDCG	F1@N	mAP	NDCG
Kanezaki_RotationNet [40]	**79.8%**	77.2%	86.5%	**59.0%**	58.3%	65.6%
Zhou_Improved_GIFT [40]	76.7%	72.2%	82.7%	58.1%	57.5%	65.7%
Tatsuma_ReVGG [40]	77.2%	74.9%	82.8%	51.9%	49.6%	55.9%
Furuya_DLAn [40]	71.2%	66.3%	76.2%	50.5%	47.7%	56.3%
Thermos_MVFusionNet [40]	69.2%	62.2%	73.2%	48.4%	41.8%	50.2%
Deng_CM-VGG5-6DB [40]	47.9%	54.0%	65.4%	16.6%	33.9%	40.4%
Li_ZFDR [40]	28.2%	19.9%	33.0%	19.7%	25.5%	37.7%
DMk_DeepVoxNet [40]	25.3%	19.2%	27.7%	25.8%	23.2%	33.7%
SHREC16-Bai_GIFT [41]	68.9%	64.0%	76.5%	45.4%	44.7%	54.8%
SHREC16_Su_MVCNN [41]	76.4%	73.5%	81.5%	57.5%	56.6%	64.0%
Wang [41]	24.6%	60.0%	77.6%	16.3%	47.8%	69.5%
Kd-network [42]	45.1%	61.7%	81.4%	24.1%	48.4%	72.6%
MVTS (Our)	76.9%	**80.2%**	**87.8%**	58.6%	**64.7%**	**74.2%**

official splitting way and employ the 70% training subset along with the 10% validation subset for network training, the test subset is used for retrieval and classification tasks. The performance in Table 2 indicates that our proposed method can outperforms the existing methods by around 3.0%-61% in microALL mAP and by around 6.4%-41.5% in macro mAP. In addition, we have several important observations as follows:

- Our method is consistently competitive compared with other representative view-based and model-based methods for both 3D model retrieval and classification tasks, which demonstrates the superiority and efficiency of our proposed method.
- Previous view-based methods usually just select one representative view from the view sequence of the model, or employ simple view-level aggregation strategy, like the max-pooling (eg. MVCNN) method to fuse multiple views. Although some generally typical information of all views is preserved, the correlation among views are often neglected, causing the lack of valuable information. However, by employing the Tree-LSTM architecture, the intrinsic correlation among views are encoded so that the valuable information among views can be preserved to boost the final retrieval performance.
- Generally, the model-based methods usually adopt the topological or locally geometric structure information of 3D models directly, which results in the lack in global information of the 3D model. However, our proposed method can not only extract the view-level feature of individual view to capture local information, but also obtain the compact feature of whole view sequence of the 3D model to capture the global information. Combining the local information with the global information of the 3D model helps get better performance.

B. SENSITIVITY ANALYSIS ON VIEW NUMBER

In order to investigate the influence of view number of models on the retrieval and classification tasks, we conducted the comparative experiment by varying view number. In detail, we change the intervals between the virtual cameras when rendering the views from the 3D model. The corresponding angle θ of interval is set to {180°, 90°, 60°, 45°, 36°, 30°} so that the view number of each model can be set to {2, 4, 6, 8, 10, 12}, respectively. The experiment results are showed in Table 3. From the results, we can find that with the view number increasing, the retrieval performance is becoming better. This is because more views can provide richer and more diverse information to represent the 3D model more comprehensively. The performance reaches the peak when the view number is set to 12. Specifically, when the view number is set to 12, it outperforms other view number settings by 0.9%-8.0%, 1.0%-6.4%, 0.8%-14.6%, 0.8%-7.5%, 1.2%-9.7% on NN, FT, ST, F and DCG, respectively. For ANMRR, the performance decreases by 1.2%-10.2% comparing to using other view numbers.

TABLE 3. Retrieval performance on ModelNet40 with different view numbers.

View Number	NN	FT	ST	F	DCG	ANMRR
2	0.851	0.797	0.802	0.262	0.794	0.217
4	0.878	0.813	0.831	0.289	0.816	0.183
6	0.893	0.824	0.857	0.301	0.841	0.158
8	0.910	0.839	0.929	0.319	0.858	0.139
10	0.922	0.851	0.940	0.329	0.879	0.127
12	**0.931**	**0.861**	**0.948**	**0.337**	**0.891**	**0.115**

C. SENSITIVITY ANALYSIS ON DIFFERENT MODULES

To validate the effectiveness of different modules in our framework, we evaluate the our method with different modules on Modelnet40 and the results are shown in Table. 4. The ''Base model'' indicates that we only employ the tree structure with one direction. The ''MVTS w/o A'' means our framework without attention module. The ''MVTS w/o B''

TABLE 4. Retrieval performance on ModelNet40 with different modules.

Structure	NN	FT	ST	F	DCG	ANMRR
Base model	0.890	0.786	0.886	0.308	0.829	0.168
MVTS w/o A	0.915	0.821	0.910	0.320	0.864	0.135
MVTS w/o B	0.920	0.849	0.929	0.331	0.879	0.127
MVTS	**0.931**	**0.861**	**0.948**	**0.337**	**0.891**	**0.115**

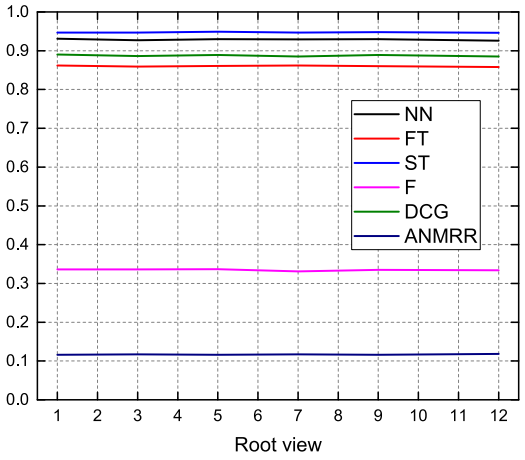


FIGURE 6. The experiment result for different root node selection on ModelNet40. The horizontal axis is the ranking index of the root node after sorting the views by their sum of similarity scores in an ascendant order.

means our framework does not contain the bidirectional module. Comparing to the "Base model", the "MVTS w/o A" utilizes the bidirectional tree structure to capture the spatial contextual information and outperforms the "Base model" by 2.5%, 3.5%, 2.4%, 1.2%, 3.5%, 3.3% in terms of NN, FT, ST, F, DCG and ANMRR criteria, respectively, which demonstrates the bidirectional module of our method. The similar observation can be found by comparing the "Base model" with the "MVTS w/o A". The MVTS with all modules gets the best performance, which further proves the effectiveness of our method.

D. THE INFLUENCE OF ROOT NODE SELECTION ON RETRIEVAL RESULT

From the theoretical analysis, the selection for root node don't affect the structure of the final spanning tree. Because at each step, the edge with the highest weight will be selected, which ensures the final spanning tree to be the maximum spanning tree with the maximum edge weight sum. To confirm this experimentally, we conduct a comparison experiment on ModelNet40, in which the root node is varied. Firstly, the sum of similarity scores of each view is calculated as:

$$\mathrm{Sum}_{v_i^d} = \sum M_i(f_i^d, .) \tag{17}$$

where v_i^d denotes the d-th view of model i and $Sum_{v_i^d}$ is the corresponding sum of similarity scores. Then $Sum_{v_i^0} \ldots Sum_{v_i^d} \ldots Sum_{v_i^S}$ are sorted in an ascendant order. According to this order, we repeat the experiment S times by selecting the different view as the root node to construct the tree. The experiment results are revealed in Figure. 6. From the results we can find that the selection for root node don't affect the retrieval result, which demonstrates the robustness of our method.

E. RESULTS VISUALIZATION

The visual retrieval results on ModelNet40 are presented in Figure. 7. The left column shows the query models and the

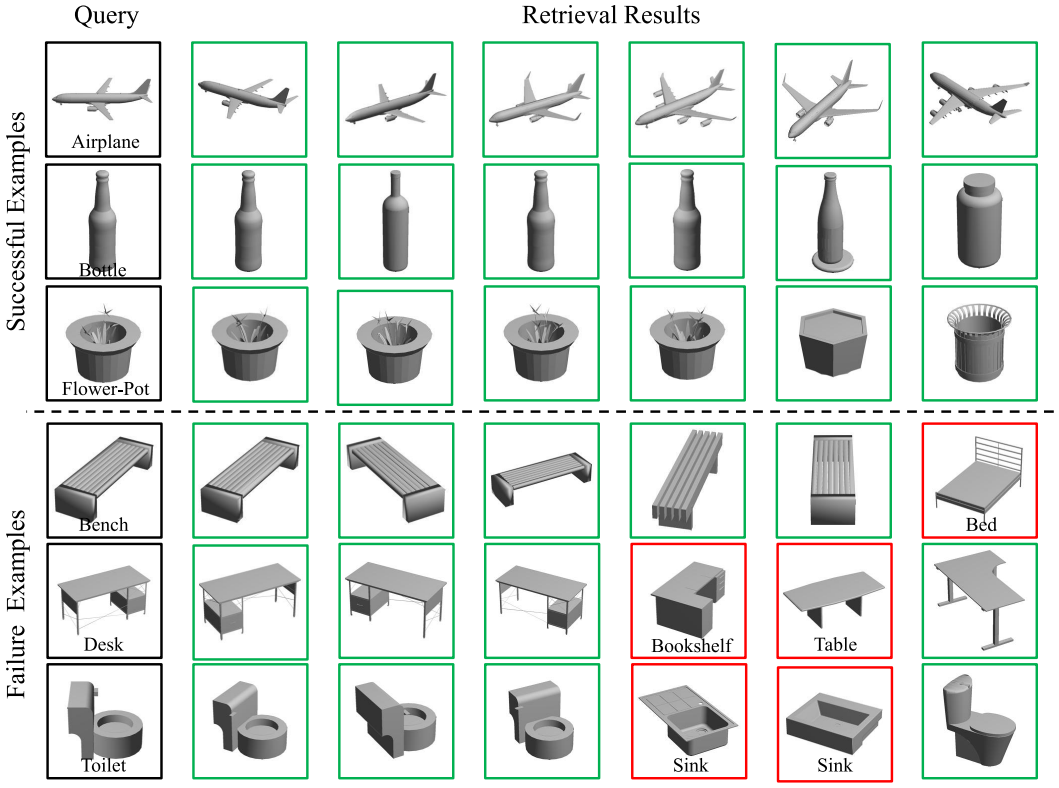


FIGURE 7. Examples of retrieval results on ModelNet40.

rest columns are retrieval results. In this figure, the 3D models surrounded by the green box denote the correct retrieval results and the models surrounded by red box denote the false retrieval results. When we use the desk as query model, the retrieval results contain the Bookshelf and Table, which are the different classes while have similar visual pattern. Consequently, although there exist failure cases, the retrieved results are still visually similar to the query model, which demonstrates the robustness and effectiveness of our proposed method.

VI. CONCLUSION

In this paper, we propose a Multi-View Tree Structure (MVTS) learning method for 3D model retrieval and recognition. Different from the traditional LSTM methods, in which the specific view order is needed to explore the structure information of multiple views, our method adopts the tree structured LSTM to model this information and abandon the limitation of view order for multiple views. Moreover, we introduce the Tree-LSTM with binary directions to help encode the contextual information between view pairs and the spatial information of the tree structure better. In order to distinguish the contribution of each view to the final model representation, the tree attention strategy is adopted to assign higher weights to more important views. The weighted average sum of hidden states after applying tree attention is utilized as the final model descriptor. The experimental results on ShapeNet55 and ModelNet40 are revealed and discussed to demonstrate the superiority of our proposed method. In the future, we intend to construct the multiple trees by using the different visual features to enrich the contextual information and further boost its performance in both 3D shape retrieval and classification tasks.

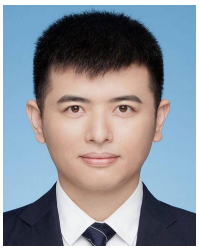
REFERENCES

[1] A. Frome, D. Huber, R. K. Kolluri, T. Bulow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 224–237.
[2] V. Barra and S. Biasotti, "3D shape retrieval using kernels on extended reeb graphs," *Pattern Recognit.*, vol. 46, no. 11, pp. 2985–2999, Nov. 2013.
[3] M. M. Kazhdan, T. A. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proc. Symp. Geometry Process.* (ACM International Conference Proceeding Series), vol. 43. Aire-la-Ville, Switzerland: Eurographics Association, 2003, pp. 156–164.
[4] H. Zeng, Q. Wang, and J. Liu, "Multi-feature fusion based on multi-view feature and 3D shape feature for non-rigid 3D model retrieval," *IEEE Access*, vol. 7, pp. 41584–41595, 2019.
[5] A.-A. Liu, S. Xiang, W.-Z. Nie, and D. Song, "End-to-End visual domain adaptation network for cross-domain 3D CPS data retrieval," *IEEE Access*, vol. 7, pp. 118630–118638, 2019.
[6] Y. Su, W. Li, W. Nie, D. Song, and A.-A. Liu, "Unsupervised feature learning with graph embedding for view-based 3D model retrieval," *IEEE Access*, vol. 7, pp. 95285–95296, 2019.
[7] A. Liu, N. Hu, D. Song, F. Guo, H. Zhou, and T. Hao, "Multi-view hierarchical fusion network for 3D object retrieval and classification," *IEEE Access*, vol. 7, pp. 153021–153030, 2019.
[8] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, and M. Aono, "A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries," *Comput. Vis. Image Understand.*, vol. 131, pp. 1–27, Feb. 2015.
[9] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1912–1920.
[10] T. Furuya and R. Ohbuchi, "Deep aggregation of local 3D geometric features for 3D model retrieval," in *Proc. Brit. Mach. Vis. Conf.*, York, U.K., R. C. Wilson, E. R. Hancock, and W. A. P. Smith, Eds., 2016, pp. 1–12.
[11] H. Tabia and H. Laga, "Covariance-based descriptors for efficient 3D shape matching, retrieval, and classification," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1591–1603, Sep. 2015.
[12] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.
[13] M. Dominguez, R. Dhamdhere, A. Petkar, S. Jain, S. Sah, and R. Ptucha, "General-purpose deep point cloud feature extractor," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1972–1981.
[14] H. You, Y. Feng, R. Ji, and Y. Gao, "Pvnet: A joint convolutional network of point cloud and multi-view for 3D shape recognition," in *Proc. 26th ACM Int. Conf. Multimedia*, Seoul, South Korea, S. Boll, K. M. Lee, J. Luo, W. Zhu, H. Byun, C. W. Chen, R. Lienhart, and T. Mei, Eds., 2018, pp. 1310–1318.
[15] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," 2020, *arXiv:2003.01251*. [Online]. Available: https://arxiv.org/abs/2003.01251
[16] A.-A. Liu, W.-Z. Nie, Y. Gao, and Y.-T. Su, "View-based 3-D model retrieval: A benchmark," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 916–928, Mar. 2018.
[17] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. J. Latecki, "GIFT: A real-time and scalable 3D shape search engine," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5023–5032.
[18] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.
[19] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "GVCNN: group-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 264–272.
[20] M. Yavartanoo, E. Kim, and K. M. Lee, "SPNet: Deep 3D object classification and retrieval using stereographic projection," in *Proc. 14th Asian Conf. Comput. Vis.*, vol. 11365, C. V. Jawahar, H. Li, G. Mori, and K. Schindler, Eds., 2018.
[21] A. Kanezaki, Y. Matsushita, and Y. Nishida, "RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5010–5019.
[22] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. de la Escalera, "Birdnet: A 3D object detection framework from lidar information," in *Proc. 21st IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Maui, HI, USA, W. Zhang, A. M. Bayen, J. J. S. Medina, and M. J. Barth, Eds., 2018, pp. 3517–3523.
[23] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6526–6534.
[24] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Syst. Tech. J.*, vol. 36, no. 6, pp. 1389–1401, Nov. 1957.
[25] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.
[26] A. X. Chang, T. A. Funkhouser, L. J. 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," 2015, *arXiv:1512.03012*. [Online]. Available: https://arxiv.org/abs/1512.03012
[27] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proc. Conf. Workshop Neural Inf. Process. Syst.*, Barcelona Spain, D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 82–90.
[28] S. Liu, L. Giles, and A. Ororbia, "Learning a hierarchical latent-variable model of 3D shapes," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 542–551.
[29] C. Ma, W. An, Y. Lei, and Y. Guo, "BV-CNNs: Binary volumetric convolutional networks for 3D object recognition," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*. London, U.K.: BMVA Press, Sep. 2017, pp. 1–12.
[30] A. Garcia-Garcia, F. Gomez-Donoso, J. Garcia-Rodriguez, S. Orts-Escolano, M. Cazorla, and J. Azorin-Lopez, "PointNet: A 3D convolutional neural network for real-time object class recognition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 1578–1584.

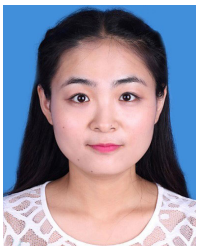
[31] R. Klokov and V. Lempitsky, "Escape from cells: Deep kd-networks for the recognition of 3D point cloud models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 863–872.
[32] M. Ren, L. Niu, and Y. Fang, "3D-a-nets: 3D deep dense descriptor for volumetric shapes with adversarial networks," 2017, *arXiv:1711.10108*. [Online]. Available: https://arxiv.org/abs/1711.10108
[33] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Conf. Workshop Neural Inf. Process. Syst.*, Long Beach, CA, USA, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5099–5108.
[34] A. Kanezaki, Y. Matsushita, and Y. Nishida, "RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," 2016, *arXiv:1603.06208*.
[35] A. Sinha, J. Bai, and K. Ramani, "Deep learning 3D shape surfaces using geometry images," in *Proc. 14th Eur. Conf. Comput. Vis.* Lecture Notes in Computer Science, vol. 9910, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Amsterdam, The Netherlands: Springer, 2016, pp. 223–240.
[36] P. Zanuttigh and L. Minto, "Deep learning for 3D shape classification from multiple depth maps," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3615–3619.
[37] K. Sfikas, T. Theoharis, and I. Pratikakis, "Exploiting the PANORAMA representation for convolutional neural network classification and retrieval," in *Proc. Eurographics Workshop 3D Object Retr.*, I. Pratikakis, F. Dupont, and M. Ovsjanikov, Eds., 2017, pp. 1–7.
[38] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise decomposition of image sequences for active multi-view recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3813–3822.
[39] C. R. Qi, H. Su, M. NieBner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5648–5656.
[40] M. Savva, F. Yu, H. Su, A. Kanezaki, and T. Furuya, "Large-scale 3D shape retrieval from ShapeNet Core55: SHREC'17 track," in *Proc. Workshop 3D Object Retr.* Aire-la-Ville, Switzerland: Eurographics Association, 2017, pp. 39–50.
[41] M. Savva, F. Yu, H. Su, M. Aono, and B. Chen, "Shrec16 track: Largescale 3D shape retrieval from shapenet core55," in *Proc. Eurographics Workshop 3D object Retr.*, vol. 10, 2016, pp. 1–11.

AN-AN LIU (Member, IEEE) received the B.Eng. and Ph.D. degrees from Tianjin University, Tianjin, China. He was a Visiting Scholar with the Robotics Institute, Carnegie Mellon University, where he worked with Prof. Take Kanade. He is currently a Professor with the School of Electronic Information Engineering, Tianjin University. His research interests include computer vision and machine learning.

ZHENLAN ZHAO is currently pursuing the master's degree with the School of Electrical and Information Engineering, Tianjin University. His research interests include computer vision, unsupervised learning, and 3D model retrieval.



WENHUI LI received the M.S. and Ph.D. degrees from the School of Electrical and Information Engineering, Tianjin University. He was an Intern Student with the SeSaMe Center, National University of Singapore. His research interests include computer vision, machine learning, and 3D model retrieval.



DAN SONG received the Ph.D. degree in computer science and technology from the Zhejiang University of China. Her research interests include computer graphics, computer vision, 3D human body reconstruction, and virtual fitting.