This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2904460, IEEE Transactions on Image Processing

3D2SeqViews: Aggregating Sequential Views for 3D Global Feature Learning by CNN with Hierarchical Attention Aggregation

Zhizhong Han, Honglei Lu, Zhenbao Liu, *Member, IEEE,* Chi-Man Vong, *Senior Member, IEEE,* Yu-Shen Liu, *Member, IEEE,* Matthias Zwicker, *Member, IEEE,* Junwei Han, *Senior Member, IEEE,* C.L. Philip Chen, *Fellow, IEEE*

***Abstract*—Learning 3D global features by aggregating multiple views is important. Pooling is widely used to aggregate views in deep learning models. However, pooling disregards a lot of content information within views and the spatial relationship among the views, which limits the discriminability of learned features. To resolve this issue, *3D to Sequential Views* (3D2SeqViews) is proposed to more effectively aggregate sequential views using convolutional neural networks with a novel hierarchical attention aggregation. Specifically, the content information within each view is first encoded. Then, the encoded view content information and the sequential spatiality among the views are simultaneously aggregated by hierarchical attention aggregation, where view-level attention and class-level attention are proposed to hierarchically weight sequential views and shape classes. View-level attention is learned to indicate how much attention is paid on each view by each shape class, which subsequently weights sequential views through a novel recursive view integration. Recursive view integration learns the semantic meaning of view sequence which is robust to the first view position. Furthermore, class-level attention is introduced to describe how much attention is paid on each shape class, which innovatively employs the discriminative ability of the fine-tuned network. 3D2SeqViews learns more discriminative features than the state-of-the-art, which leads to the outperforming results in shape classification and retrieval under three large-scale benchmarks.**

***Index Terms*—3D global feature learning, View aggregation, Sequential views, Hierarchical attention aggregation, CNN.**

I. Introduction

TWO dimensional views can be used to represent both manifold and non-manifold 3D shapes. This advantage alleviates the difficulty of direct learning of 3D features from irregular 3D shapes (i.e. arbitrary vertex number, irregular vertex topology and orientation ambiguity on 3D surface) [1]–[3] by deep learning models, which makes learning 3D features from multiple views important for 3D shape analysis [4]–[8], such as 3D shape classification and retrieval [9]. By taking multiple views around a 3D shape, 3D shape features can be learned by aggregating the information of views [4], [5], [7], [8], [10]–[12], where the key lies in an efficient and effective view aggregation. To fully benefit from the powerful learning ability of deep learning models, it is critical to learn 3D features by view aggregation in the end-to-end parameter optimization procedure.

Max or mean pooling is widely used for view aggregation in deep learning models [4], [5], [7], [8], [10], [11]. As a procedure originally designed for information abstraction, pooling only employs the max or mean value of each dimension across all view features to learn 3D shape features. Although pooling is able to eliminate the rotation effect of 3D shapes to some extent, a lot of content information within views and the spatial relationship among the views are inevitably lost, leading to the limited discriminability of learned features. Therefore, it is still a research challenge to learn 3D features by more effectively aggregating the content information and spatial relationship of multiple views in deep learning models.

To tackle this challenge, a novel deep learning model, *3D to Sequential Views* (3D2SeqViews), is proposed to aggregate sequential views for 3D global feature learning by convolutional neural networks (CNN) with a novel hierarchical attention aggregation. To increase the discriminability of learned features, 3D2SeqViews aggregates not only the content information within all sequential views but also the sequential spatiality among the views. In our work, sequential views are sequentially taken around each 3D shape on a circle, which aims to alleviate the difficulty of capturing spatial relationship among multiple unordered views, such as the ones taken on a unit sphere. Specifically, in order to prevent the view content information loss caused by pooling [4], [5], [7], [8], [10], [11], 3D2SeqViews first encodes the content information within each sequential view by convolution in CNN. This is facilitated by sequential views due to the explicit view order in the view sequence, which makes 3D2SeqViews learn

Z. Han is with the Tsinghua University and the University of Maryland, College park (email: h312h@mail.nwpu.edu.cn).

Z. Liu, J. Han are with the Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China (email: liuzhenbao, jhan@nwpu.edu.cn).

Yu-Shen Liu and Honglei Lu are with the Tsinghua University, Beijing 100084, P. R. China (liuyushen@tsinghua.edu.cn, 454638641@qq.com). Yu-Shen Liu is the corresponding author.

Matthias Zwicker is with the University of Maryland, College park, 20737, USA (email: zwicker@cs.umd.edu).

C.M. Vong is with Dept. of Computer and Information Science, University of Macau, Macau 99999, China (email: cmvong@umac.mo).

C. L. P. Chen is with the Faculty of Science and Technology, University of Macau, Macau 99999, China (email: philip.chen@ieee.org).

This work was supported by National Key R&D Program of China (2018YFB0505400), the National Natural Science Foundation of China under Grant 61472202, 61672430, National Science Foundation under Grant 1813583, Shaanxi Key Research and Development Program under Grant S2019-YF-ZDCXL-ZDLGY-0227, Aeronautical Science Fund under Grant BK1829-02-3009, and NWPU Basic Research Fund under Grant 3102018jc-c001, the University of Macau Research Grant MYRG2018-00138-FST, MYRG2016-00134-FST, FDCT Grant 273/2017/A.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2904460, IEEE Transactions on Image Processing

more comprehensive characteristics in the view sequence than existing methods. Then, the encoded view content information and the sequential spatiality among the views are simultaneously aggregated to learn 3D global features by a novel hierarchical attention aggregation. Finally, a softmax layer is employed to guide the parameter optimization by minimizing the classification errors of 3D shapes.

In hierarchical attention aggregation, view-level attention and class-level attention are proposed to hierarchically weight sequential views and shape classes. View-level attention is learned to indicate how much attention is paid on each view by each shape class, while class-level attention is further introduced to describe the attention paid on each shape class which employs the discriminative ability of fine-tuned networks. In addition, a novel recursive view integration is proposed to weight the encoded view content information by view-level attention while preserving the sequential spatiality among the views, which enables 3D2SeqViews to learn the semantic meaning of view sequence that is robust to the first view position. Our significant contributions are listed as below:

i) A novel deep learning model called 3D2SeqViews is proposed for 3D global feature learning by aggregating sequential views. It not only encodes the content information within all sequential views but also preserves the sequential spatiality among the views.
ii) A novel view aggregation in CNN called hierarchical attention aggregation is proposed to simultaneously aggregate the content information and sequential spatiality in a view sequence, where view-level attention and class-level attention are proposed to get comprehensively combined to significantly increase the discriminability of learned features.
iii) The sequential spatiality captured by a novel recursive view integration improves the limited ability of CNN for learning from sequential data, which enables 3D2SeqViews to learn the semantic meaning of view sequence that is robust to the first view position.
iv) The discriminative ability of fine-tuned network for low-level view feature extraction is innovatively employed by 3D2SeqViews through class-level attention in hierarchical attention aggregation, which is an important source to enhance the discriminability of learned features but ignored by existing methods.

This paper is organized as follows. The related studies are reviewed in Section II. The details of 3D2SeqViews are presented in Section III. Experimental setup and results with analysis are shown in Section IV and Section V, respectively. Finally, a conclusion is drawn in Section VI.

II. Related work

The deep learning methods for 3D feature learning are reviewed in this section. These methods are categorized in terms of different raw 3D representations that are learned from, including meshes, voxels and views. In the reviewed methods, the procedures employed for view aggregation are emphasized to highlight the novelty and the significance of hierarchical attention aggregation proposed in 3D2SeqViews.

A. Mesh-based methods

3D mesh is an important type of 3D shape representations. A 3D mesh is composed of vertices which are connected by edges. Mesh-based methods mainly aim to learn the geometrical and spatial information from triangle faces of 3D mesh. To directly learn features from 3D meshes, different deep learning models have been proposed. Han et al. [1] proposed circle convolutional restricted boltzmann machine to learn 3D local features based on a novel circle convolution in an unsupervised way. To learn global features by hierarchically abstracting from local information, Han et al. [2] further proposed mesh convolutional restricted boltzmann machine, which simultaneously encodes the geometry of local regions and the spatiality among them. Jonathan et al. [13] learned 3D features from hand-crafted features on 3D meshes by a novel geodesic convolutional neural networks. To explore the feasibility of learning features in the spectral domain, Davide et al. [14] proposed localized spectral convolutional networks to perform supervised local feature learning. By encoding the spatial relationships among virtual words on 3D meshes, Han et al. proposed deep spatiality [15] to simultaneously learn 3D global and local features with novel coupled softmax. However, these methods can only be used to learn features from smooth manifold meshes.

B. Voxel-based methods

Voxel-based methods learn 3D features from voxels which represent 3D shapes by the distribution of corresponding binary variables. These methods usually employ deep learning models to capture the patterns of correlation among the binary variables involved in each 3D shape. Wu et al. [16] proposed 3D ShapeNets to learn global features from voxelized 3D shapes based on convolutional restricted boltzmann machine. Sharma et al. [17] employed fully convolutional denoising autoencoder to robustly perform unsupervised global feature learning by decomposing and reconstructing voxelized 3D shapes. Girdhar et al. [18] combined voxels and views of 3D shapes to learn global features by a novel T-L network based on CNN. With the generative adversarial training, Wu et al. [19] learned 3D global features by a novel 3DGAN which is composed of a generator and a discriminator. By analysing the reason why the performances of voxel-based methods are always not as good as view-based methods, Qi et al. [10] employed CNN to learn global features from novel voxel representations, where max pooling is used to aggregate information captured from different orientations. To speed up the training, Wang et al. [12] proposed O-CNN to learn global features based on a novel octree data structure. To learn local features from voxels, Han et al. [3] proposed a novel voxelization permutation strategy to eliminate the effect of rotation and orientation ambiguity on 3D surface. Although voxel-based methods have the advantage of generating 3D shapes, these methods require heavily computational cost and their performances in shape discrimination are always worse than the following view-based methods.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2904460, IEEE Transactions on Image Processing

C. View-based methods

View-based methods try to understand each 3D shape from different viewpoints. These methods learn the feature of a 3D shape from a set of view images captured from the 3D shape.

Light field descriptor (LFD) [20] is the pioneer view-based 3D descriptor to extract 3D global features, which employs the features of 2D silhouettes in multiple views taken around a 3D shape. Instead of learning global features by aggregating multi-view information, LFD evaluates the dissimilarity between two shapes by comparing the corresponding two view sets in a greedy way. By the same strategy, GIFT [6] measures the difference between two 3D shapes by the Hausdorff distance between their corresponding view sets. These methods employ a greedy strategy to compare views for the evaluation of the difference between two 3D shapes, which avoids to identically align 3D shapes before pairwise 3D shape comparison. In contrast, RotationNet [21] was proposed to learn global features by treating pose labels as latent variables which are optimized to self-align in an unsupervised manner.

Besides the 2D rendered views, other different 2D representations are also employed to represent 3D shapes for deep learning models to learn. DeepPano [7] was proposed to learn features from panorama views using CNN, where each panorama view can be regarded as the seamless aggregation of multiple views captured on a circle. To eliminate the effect of rotation about the up-orientation, row-wise max pooling was introduced in DeepPano. With pose normalization, Sfikas et al. [22] used CNN to learn 3D global features from multiple panorama views which were stacked together in a consistent order. Similarly, Sinha et al. [23] proposed to learn features from hand-crafted features named as geometry images.

To encode information from multiple views through view aggregation, pooling becomes a widely used procedure in deep learning models. This manner was introduced in multi-view CNN [4] which learns global features by aggregating multiple views. To describe a 3D shape, the content information within all views is first max-pooled together before the global feature of the 3D shape is learned. Similarly, max pooling is also first employed to aggregate multiple views which are taken around local regions to learn local features for 3D shape segmentation or correspondence [5]. Instead of performing pooling first, 3D2SeqViews convolves the content information within all sequential views in a view sequence, which prevents the loss of content information caused by pooling.

To employ the content information within all views, Li et al. [24] concatenated all content information for hierarchical abstraction in the CNN-based model. By decomposing a view sequence into a set of view pairs, Johns et al. [25] classified each pair independently, and then, learned an object classifier by weighting the contribution of each pair, which allows 3D shape recognition over arbitrary camera trajectories. To perform pooling more efficiently, Wang et al. [8] proposed dominant set clustering to cluster views taken from each shape, where pooling is performed in each cluster respectively.

The issues of the view aggregation procedures in the aforementioned methods are analyzed in the following. View aggregation by pooling eliminates the effect of rotation on 3D shapes to some extent, however, it inevitably loses a lot of content information within views and the spatial relationship among the views which has been regarded an important information in computer vision area [26]. In addition, the spatial relationship between pairwise views is also disregarded by the view pair decomposition [25]. Although it is able to overcome the disadvantages of pooling by concatenation of all view content information [24], it is sensitive to the first view position in a view sequence. Xu et al. [27] employed the concept of attention to find next best view for depth acquisition, and then, found the most discriminative part in each view for part-based recognition.

To resolve the aforementioned issues, 3D2SeqViews employs a novel hierarchical attention aggregation to aggregate sequential views for 3D global feature learning. In hierarchical attention aggregation, the content information within all sequential views and the sequential spatiality among the views are effectively aggregated under hierarchically weighting view-level attention and class-level attention. With a novel recursive view integration, the sequential spatiality among sequential views is encoded to help 3D2SeqViews learn the semantic meaning of sequential views in the view sequence, which is robust to the first view position.

III. 3D2SeqViews

In this section, 3D2SeqViews is introduced in detail. First, the overview of 3D2SeqViews is presented. Then, the key elements, including sequential views capturing, low-level view feature encoding, and hierarchical attention aggregation are described in detail in the subsequent three subsections, respectively.

A. Overview

The framework of 3D2SeqViews is illustrated in Fig. 1. First, for each i-th 3D shape m^i in a training set of M 3D shapes, where $i \in [1, M]$, a view sequence $\boldsymbol{v}^i$ is obtained by capturing V sequential views v_j^i around m^i, such that $\boldsymbol{v}^i = [v_1^i, ..., v_j^i, ..., v_V^i]$ and $j \in [1, V]$, as shown in Fig. 1 (a). Then, the low-level feature $\boldsymbol{f}_j^i$ of each view v_j^i is encoded by row-wise convolution after extracted by a fine-tuned VGG19 network [28]. The VGG19 also provides the classification probability $\boldsymbol{p}_j^i$ of each view v_j^i to calculate the subsequent class-level attention, as shown in Fig. 1 (b). Finally, the global feature $\boldsymbol{F}^i$ of shape m^i is learned by aggregating the content information within all sequential views v_j^i in $\boldsymbol{v}^i$ and the sequential spatiality among v_j^i. This view aggregation is conducted under hierarchically weighting view-level attention and class-level attention by hierarchical attention aggregation, as shown in Fig. 1 (c).

To learn $\boldsymbol{F}^i$, the low-level features $\boldsymbol{f}_j^i$ of all sequential views in $\boldsymbol{v}^i$ are first stacked into a low-level view feature matrix $\boldsymbol{A}^i$ according to the sequential direction derived in $\boldsymbol{v}^i$. Then, several hidden convolutional layers are employed to perform row-wise convolution on $\boldsymbol{A}^i$ by row-wise convolution kernels, which abstracts the content information within each view v_j^i. The hidden convolutional layers shorten the low-level feature $\boldsymbol{f}_j^i$ of each view v_j^i in $\boldsymbol{v}^i$ and form an

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2904460, IEEE Transactions on Image Processing

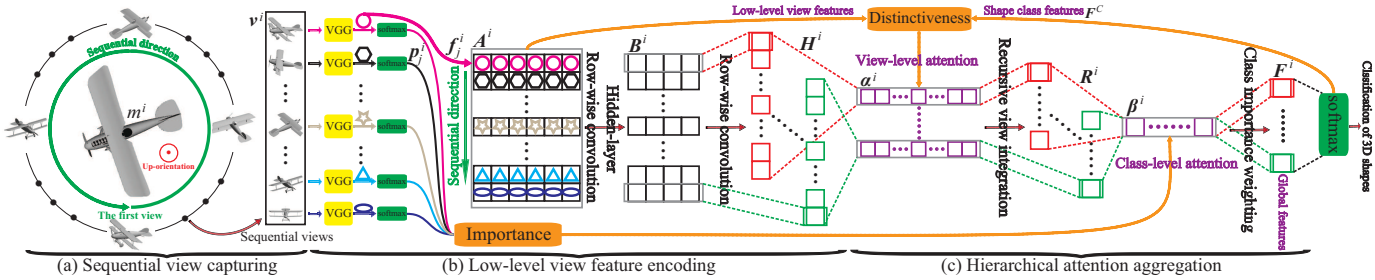


Fig. 1. The overview of 3D2SeqViews. The sequential views are first captured around each up-oriented 3D shapes on a circle in (a). Then, the low-level feature of each view is encoded by row-wise convolution after extracted by a fine-tuned VGG network, as shown in (b). Finally, the global 3D feature is learned by aggregating sequential views in hierarchical attention aggregation.

abstracted view feature matrix $\boldsymbol{B}^i$. Subsequently, another row-wise convolution layer is employed to further encode $\boldsymbol{B}^i$ into a set $\boldsymbol{H}^i$ of column-wise feature maps by row-wise convolution. Finally, $\boldsymbol{F}^i$ is learned from $\boldsymbol{H}^i$ by hierarchical attention aggregation, where view-level attention $\boldsymbol{\alpha}^i$ and class-level attention $\boldsymbol{\beta}^i$ are employed to weight $\boldsymbol{H}^i$ in a hierarchical manner. View-level attention $\boldsymbol{\alpha}^i$ weights the encoded content information of views with preserving the sequential spatiality among the views through the novel recursive view integration, which helps 3D2SeqViews learn the semantic meaning of the view sequence that is robust to the first view position. With introducing $\boldsymbol{\alpha}^i$ and $\boldsymbol{\beta}^i$, the discrminability of the learned global feature $\boldsymbol{F}^i$ are significantly increased.

B. Sequential view capturing

The V sequential views v_j^i are taken around the i-th 3D shape m^i on a circle, which forms a view sequence $\boldsymbol{v}^i$ as shown in Fig. 1 (a), where $j \in [1, V]$ and $i \in [1, M]$. The sequential views in $\boldsymbol{v}^i$ are uniformly distributed on the circle in order, where the cameras are elevated $30°$ from the ground plane, pointing to the centroid of the 3D shape. The first view in the view sequence is taken from a fixed position which can be randomly selected on the circle. Then, the subsequent views are taken with an angle interval of $360°/V$ in a consistent sequential direction. The sequential direction is determined by the right hand rule, i.e., the direction of wrapping one's right hand when the thumb is in the same direction of the up-orientation, as demonstrated by the green arrow surrounding the 3D shape in Fig. 1 (a).

Different from the traditional multi-view capturing [6], [20], the sequential views are captured on a circle rather than a unit sphere. Although the sequential views cannot fully cover the top or the bottom of 3D shapes, the content information within sequential views can be more efficiently aggregated with preserving the sequential spatiality among the views for 3D global feature learning.

C. Low-level view feature encoding

In this subsection, the low-level feature $\boldsymbol{f}_j^i$ of each view v_j^i in $\boldsymbol{v}^i$ is first extracted by a fine-tuned VGG19. Then, the content information within each view is abstracted by reducing the dimension of the low-level features $\boldsymbol{f}_j^i$ using row-wise convolution. Finally, the abstracted content information within all v_j^i is further encoded into a set $\boldsymbol{H}^i$ of column-wise feature maps by row-wise convolution. The content information and the sequential spatiality of $\boldsymbol{v}^i$ in $\boldsymbol{H}^i$ is subsequently aggregated by hierarchical attention aggregation which will be detailed in the next subsection.

Low-level view feature extraction. VGG19 is employed to extract the low-level feature $\boldsymbol{f}_j^i$ of each sequential view v_j^i in $\boldsymbol{v}^i$ from the i-th shape m^i. VGG19 is originally trained under ImageNet benchmark for large scale image classification [28].

VGG19 is formed by 19 weight layers which include 16 convolutional layers and 3 fully connected layers. With a softmax layer, VGG19 is capable of classifying images belonging to 1000 categories. Here, the VGG19 pre-trained under ImageNet is fine-tuned by all sequential views of 3D shapes in the training set, where each view v_j^i is classified into one of C shape classes by another softmax layer, as shown in Fig. 1 (b).

In Fig. 1 (b), when the view v_j^i is forwarded through the fine-tuned VGG19, its low-level feature $\boldsymbol{f}_j^i$ is extracted as a 4096 dimensional vector from the last fully connected layer. In addition, the classification probability $\boldsymbol{p}_j^i$ of sequential view v_j^i is also obtained from the softmax layer, which will be subsequently used for calculating class-level attention $\boldsymbol{\beta}^i$.

Low-level view feature abstraction. To preserve the sequential spatiality among sequential views in $\boldsymbol{v}^i$, the low-level features $\boldsymbol{f}_j^i$ of all sequential views are stacked into a low-level view feature matrix $\boldsymbol{A}^i = [\boldsymbol{f}_1^i; ...; \boldsymbol{f}_j^i; ...; \boldsymbol{f}_V^i]$ and $\boldsymbol{A}^i \in \mathbb{R}^{V \times 4096}$. First, the low-level feature $\boldsymbol{f}_j^i$ of each view v_j^i is abstracted by row-wise convolution on $\boldsymbol{A}^i$ in N hidden layers $\boldsymbol{D}_n^i$, where $n \in [1, N]$. The hidden layer $\boldsymbol{D}_n^i \in \mathbb{R}^{E_n \times V \times q_n}$ is produced by E_n row-wise convolution filters and encodes the content information within each view v_j^i with reducing the dimension of low-level view features into q_n. Then, an abstracted view feature matrix $\boldsymbol{D}_N^i$ is obtained from $\boldsymbol{A}^i$, where $E_N = 1$. The matrix $\boldsymbol{D}_N^i$ is denoted as $\boldsymbol{B}^i \in \mathbb{R}^{V \times D}$ with $D = q_N$ for more clear representation.

Low-level view feature encoding. Another row-wise convolution is conducted on $\boldsymbol{B}^i$ to further encode the content information within each view by K row-wise filters $\{\boldsymbol{k}_t\}$, where $\boldsymbol{k}_t \in \mathbb{R}^{1 \times D}$ and $t \in [1, K]$. For each row-wise convolution filter $\boldsymbol{k}_t$, a column-wise feature map $\boldsymbol{h}_t^i \in \mathbb{R}^{V \times 1}$ is obtained by convolving across $\boldsymbol{B}^i$ row-by-row, where $\boldsymbol{h}_t^i = sig(\boldsymbol{B}^i * \boldsymbol{k}_t)$, $*$ is the row-wise convolution and sig is the sigmoid function. Then, all column-wise feature maps $\boldsymbol{h}_t^i$ obtained by $\{\boldsymbol{k}_t\}$ form

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2904460, IEEE Transactions on Image Processing

a set of feature maps $\boldsymbol{H}^i = \{\boldsymbol{h}_t^i | t \in [1, K]\}$.

D. Hierarchical attention aggregation

The global feature $\boldsymbol{F}^i$ of the i-th shape m^i is learned from $\boldsymbol{H}^i$ by hierarchical attention aggregation. Hierarchical attention aggregation aims to aggregate the encoded content information within sequential views and the sequential spatiality among the views for learning 3D global features. In addition, the two kinds of attention, i.e., view-level attention $\boldsymbol{\alpha}^i$ and class-level attention $\boldsymbol{\beta}^i$, hierarchically weight sequential views and shape classes in this view aggregation process.

View-level attention. In order to facilitate 3D2SeqViews to conduct the classification of 3D shapes, view-level attention $\boldsymbol{\alpha}^i$ is learned to indicate how much attention is paid on each view v_j^i in $\boldsymbol{v}^i$ by each shape class c, where $c \in [1, C]$ and C is the number of shape classes.

Intuitively, each shape class focuses more on the views that are more distinctive to the shape class when contributing to the classification of 3D shapes. For example, because of self-occlusion, shape class "bowl" focuses more on some views of a cup with a handle if the handle does not appear in these views, while shape class "cup" focuses more on other views where the handle appears. Thus, view-level attention is proposed to indicate how much attention is paid on each view by each shape class, which shows the distinctiveness of each view to each shape class.

In addition, the distinctiveness indicated by view-level attention is measured by the similarity between each view and each shape class in our work. This is because a specific shape class focuses more on some views which are more similar to the common characteristics of the shape class. For example, the views of the cup that shape class "bowl" focuses more on are more similar to the characteristics of shape class "bowl" than the ones of shape class "cup".

For sequential views of a cup, the view-level attention learned by 3D2SeqViews is briefly visualized in Fig. 2. The view-level attention paid by shape class "bowl" and shape class "cup" is shown by bars in different colors. To better visualize which shape class pays more attention on a specific view, the line of ratio between view-level attention paid by shape class "bowl" and shape class "cup" is also shown, and the isoline of unit ratio is used for reference. Since the handle of the cup is occluded by the body in view 1 and view 2, shape class "bowl" pays more attention on these views than shape class "cup", as shown by the ratio upon the isoline. In contrast, when the handle appears in view 3, view 4 and view 5, shape class "cup" pays more attention than shape class "bowl". Although the handle also appears in views 6, view 7, and views 8, it is very hard to distinguish the handle from the body because of the resolution of the views. Thus, shape class "bowl" pays more attention than shape class "cup" again. This example illustrates the rationale of the proposed view-level attention $\boldsymbol{\alpha}^i$ which is detailed in the following.

The view-level attention $\boldsymbol{\alpha}^i$ measures the distinctiveness of each view v_j^i to each shape class c by the similarity between the low-level view feature matrix $\boldsymbol{A}^i$ and the shape class features $\boldsymbol{F}^C$. The view-level attention $\boldsymbol{\alpha}^i$ is a $C \times V$ matrix,

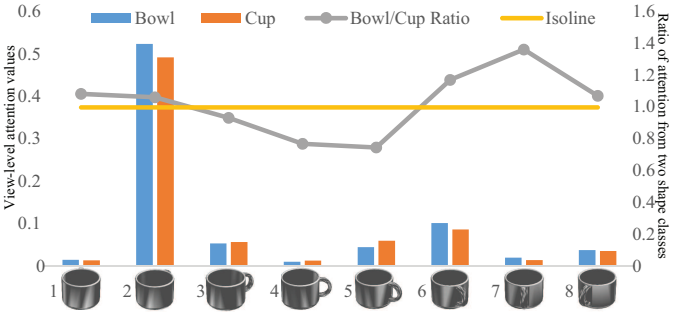


Fig. 2. The illustration of view-level attention learned for a cup. The view-level attention on different views is shown by bars, where the colors indicate different shape classes, such as shape class "bowl" and shape class "cup". The line of ratio between view-level attention from the two shape classes is also shown, where the isoline indicating the unit ratio is used for reference. As shown by the attention values, a shape class tries to focus more on the views that are more distinctive to the shape class. This is achieved by measuring the distance between the low-level view feature and shape class feature in a common space learned in our model.

where its entry $\boldsymbol{\alpha}^i(c, j)$ denotes the attention paid on the j-th sequential view v_j^i of shape m^i by the c-th shape class. The entry $\boldsymbol{\alpha}^i(c, j)$ indicates the distinctiveness of the j-th sequential view v_j^i to the c-th shape class, which is defined as below,

$$\boldsymbol{\alpha}^i = \boldsymbol{W}_A(\boldsymbol{A}^i)^{\mathrm{T}} + \boldsymbol{F}^C\boldsymbol{W}_C + \boldsymbol{b}_V, \tag{1}$$

where $\boldsymbol{W}_A$, $\boldsymbol{W}_C$, and $\boldsymbol{b}_V$ are learnable parameters. $\boldsymbol{F}^C$ is a matrix formed by all shape class features $\boldsymbol{f}^c$ which are innovatively employed as the parameters learned in the softmax layer for the classification of 3D shapes, where $\boldsymbol{F}^C = [\boldsymbol{f}^1, ..., \boldsymbol{f}^c, ..., \boldsymbol{f}^C]$ and $\boldsymbol{f}^c$ is the feature of the c-th shape class. To make our description more clearly, $\boldsymbol{F}^C$ will be detailed at the end of this subsection. $\boldsymbol{W}_A$ and $\boldsymbol{W}_C$ respectively project $\boldsymbol{A}^i$ and $\boldsymbol{F}^C$ into a common subspace for the calculation of similarity between $\boldsymbol{A}^i$ and $\boldsymbol{F}^C$, where $\boldsymbol{b}_V$ is used as a bias term. In addition, for the c-th shape class, its attention $\boldsymbol{\alpha}^i(c, 1:V)$ to all sequential views v_j^i in $\boldsymbol{v}^i$ are normalized by softmax as follows,

$$\boldsymbol{\alpha}_{norm}^i(c, 1:V) = softmax(\boldsymbol{\alpha}^i(c, 1:V)), \tag{2}$$

where $\boldsymbol{\alpha}_{norm}^i$ denotes the normalized $\boldsymbol{\alpha}^i$ for clearer description.

Recursive view integration. Each feature map $\boldsymbol{h}_t^i \in \mathbb{R}^{V\times 1}$ in $\boldsymbol{H}^i$ is weighted by view-level attention $\boldsymbol{\alpha}_{norm}^i$ through the novel recursive view integration, which highlights the encoded content information within sequential views that should be paid more attention by each shape class. Recursive view integration not only aggregates the encoded content information $\boldsymbol{h}_t^i(j)$ of the j-th view v_j^i with view-level attention $\boldsymbol{\alpha}_{norm}^i$ but also preserves the sequential spatiality among the views. The recursive view integration on a feature map $\boldsymbol{h}_t^i$ with $\boldsymbol{\alpha}_{norm}^i$ is defined as,

$$\boldsymbol{r}_t^i = \boldsymbol{\alpha}_{norm}^i \otimes \boldsymbol{h}_t^i, \tag{3}$$

where $\otimes$ denotes the recursive view integration. $\boldsymbol{r}_t^i \in \mathbb{R}^{C\times 1}$ is the result of recursive view integration on $\boldsymbol{h}_t^i$, whose element

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2904460, IEEE Transactions on Image Processing

is $\boldsymbol{r}_t^i(c) = \boldsymbol{\alpha}_{norm}^i(c, 1:V) \otimes \boldsymbol{h}_t^i(1:V)$. $\boldsymbol{r}_t^i$ comprehensively encodes the sequential spatiality among views and the content information encoded in $\boldsymbol{h}_t^i$ along with weighting the attention paid by each shape class c. Specifically, with the attention paid by the c-th shape class $\boldsymbol{\alpha}_{norm}^i(c, 1:V)$, the c-th element $\boldsymbol{r}_t^i(c)$ of $\boldsymbol{r}_t^i$ is obtained by the recursive view integration on a feature map $\boldsymbol{h}_t^i$, as defined by,

$$\begin{aligned} &\boldsymbol{r}'(1) = \boldsymbol{\alpha}_{norm}^i(c,1)\boldsymbol{h}_t^i(1), \\ &\boldsymbol{r}'(j) = (1 - \boldsymbol{\alpha}_{norm}^i(c,j))\boldsymbol{r}'(j-1) + \boldsymbol{\alpha}_{norm}^i(c,j)\boldsymbol{h}_t^i(j), \\ &\boldsymbol{r}_t^i(c) = \boldsymbol{r}'(V), \text{and } 2 \leq j \leq V, \end{aligned} \tag{4}$$

where $\boldsymbol{r}'$ is an intermediate variable vector for better understanding of recursive view integration, the iteration is continued until j reaches V from 2, and $\boldsymbol{r}'(1)$ is initialized by $\boldsymbol{\alpha}_{norm}^i(c,1)\boldsymbol{h}_t^i(1)$ when $j = 1$. Finally, $\boldsymbol{r}_t^i(c)$ is assigned by the last element of $\boldsymbol{r}'$. This procedure is further illustrated in Fig. 3 with a sequence of $V = 4$ views.

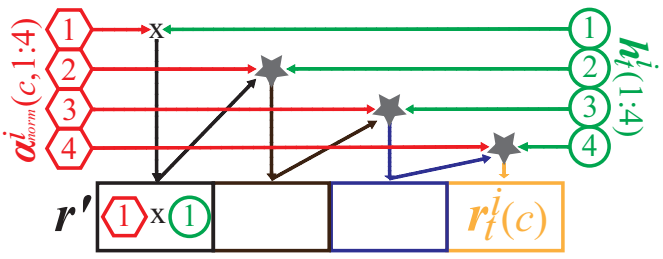


Fig. 3. The recursive view integration is illustrated by weighting normalized attention weights $\boldsymbol{\alpha}_{norm}^i(c, 1:V)$ on a feature map $\boldsymbol{h}_t^i(1:V)$ of a sequence of $V = 4$ views. The calculation involved in the second row of Eq. (4) is represented by a symbol of star. The recursive view integration not only encodes the content information of each view but also preserves the sequential spatiality among the views.

Recursive view integration is defined as a form of recursive filtering as shown in Eq. (4). It is able to encode the sequential spatiality among the sequential views, and moreover, makes the 3D2SeqViews learn the semantic meaning of view sequence which is robust to the first view position. This is because recursively weighting $(1 - \boldsymbol{\alpha}_{norm}^i(c,j))$ significantly reduces the influence of the first view position but keeps concentrating on the sequential spatiality among the successive sequential views.

Similar to $\boldsymbol{H}^i$, all column-wise feature maps $\boldsymbol{r}_t^i$ form another set $\boldsymbol{R}^i$ of feature maps, such that $\boldsymbol{R}^i = \{\boldsymbol{r}_t^i | t \in [1, K]\}$. Subsequently, $\boldsymbol{R}^i$ is weighted by class-level attention $\boldsymbol{\beta}^i$ to highlight the shape classes that are focused more by 3D2SeqViews, which represents the importance of each shape class to the classification of 3D shape m^i.

Class-level attention. Because of our limited computational capacity, VGG19 is not jointly trained with entire 3D2SeqViews in an end-to-end manner, which is a compromise that is widely adopted by existing methods. However, the discriminative ability of fine-tuned network, such as VGG19, was always ignored by existing methods, which should be an important source to increase the discriminability of learned global features although it is hard to use. To resolve this issue, class-level attention is introduced in hierarchical attention aggregation to employ the discriminative ability of the fine-tuned VGG19 by 3D2SeqViews.

Class-level attention $\boldsymbol{\beta}^i$ presents a shape class prior to show the importance of shape classes to the classification of the i-th 3D shape m^i, which comes from the softmax layer in the fine-tuned VGG19. The fine-tuned VGG19 not only extracts low-level features of sequential views but also learns the discriminative information for the classification of sequential views. Class-level attention $\boldsymbol{\beta}^i$ helps 3D2SeqViews adopt this important information source.

$\boldsymbol{\beta}^i$ is innovatively calculated using the classification probability $\boldsymbol{p}_j^i$ of sequential views provided by the fine-tuned VGG19, where $\boldsymbol{p}_j^i \in \mathbb{R}^{1\times C}$. VGG19 is fine-tuned to minimize the classification error of each sequential view v_j^i into one of C shape classes, where the classification probability $\boldsymbol{p}_j^i$ of v_j^i is provided to indicate which shape class v_j^i belongs to. With $\boldsymbol{p}_j^i$, class-level attention $\boldsymbol{\beta}^i \in \mathbb{R}^{1\times C}$ is calculated by averaging $\boldsymbol{p}_j^i$ of all sequential views in $\boldsymbol{v}^i$, as defined below,

$$\boldsymbol{\beta}^i = \frac{1}{V}\sum_{j=1}^{V} \boldsymbol{p}_j^i. \tag{5}$$

Class importance weighting. Global feature $\boldsymbol{F}^i$ of shape m^i is finally learned by weighting $\boldsymbol{R}^i$ using $\boldsymbol{\beta}^i$, which highlights the dimensions of each feature map $\boldsymbol{r}_t^i$ in $\boldsymbol{R}^i$ according to the importance of shape classes to the classification of 3D shape m^i. Each element $\boldsymbol{F}^i(t)$ of $\boldsymbol{F}^i$ is obtained by multiplying $\boldsymbol{\beta}^i$ with $\boldsymbol{r}_t^i$ in $\boldsymbol{R}^i$, as defined as follows,

$$\boldsymbol{F}^i(t) = \boldsymbol{\beta}^i \boldsymbol{r}_t^i \tag{6}$$

where $\boldsymbol{F}^i = [\boldsymbol{F}^i(1), ..., \boldsymbol{F}^i(t), ..., \boldsymbol{F}^i(K)] \in \mathbb{R}^{1\times K}$ is a K dimensional vector which is employed for the classification of m^i by a softmax layer. The classification probability $\boldsymbol{P}^i$ provided by the softmax layer is used to classify shape m^i into one of C shape classes, as defined by,

$$\boldsymbol{P}^i = softmax(\boldsymbol{F}^i \boldsymbol{W}_F + \boldsymbol{b}_F). \tag{7}$$

In Eq. (7), $\boldsymbol{W}_F$ and $\boldsymbol{b}_F$ are learnable parameters. The c-th element of $\boldsymbol{P}^i$, $P^i(l^i = c|\boldsymbol{F}^i)$, is the probability that m^i belongs to the c-th shape class, i.e., $l^i = c$. In addition, $\boldsymbol{W}_F \in \mathbb{R}^{K\times C}$ is innovatively used as the features of all shape classes. Here, $\boldsymbol{F}^C$ in Eq. (1) is the transpose of $\boldsymbol{W}_F$ for the calculation of view-level attention $\boldsymbol{\alpha}^i$, such that $\boldsymbol{F}^C = \boldsymbol{W}_F^{\mathrm{T}}$.

E. Learning inference

Finally, the parameters involved in 3D2SeqViews are optimized by minimizing the negative log-likelihood O over M 3D shapes in the training set, as defined below, where $Q^i(l^i = c)$ is the ground truth label,

$$O = -\frac{1}{M}\sum_{i=1}^{M}\sum_{c=1}^{C} Q^i(l^i = c) \log P^i(l^i = c). \tag{8}$$

The involved parameters can be optimized by back propagation of classification errors of 3D shapes. It is worth noting that the gradient for updating $\boldsymbol{W}_F$ comes from two parts. One part

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2904460, IEEE Transactions on Image Processing

is from the softmax layer for the classification of 3D shapes, i.e., $\partial O/\partial \boldsymbol{W}_F$, the other is from the calculation of view-level attention, i.e., $\partial \boldsymbol{\alpha}^i/\partial \boldsymbol{W}_F$. Finally, $\boldsymbol{W}_F$ is iteratively updated with the learning rate ε as follows,

$$\boldsymbol{W}_F \leftarrow \boldsymbol{W}_F - \varepsilon(\frac{\partial O}{\partial \boldsymbol{W}_F} + \frac{\partial \boldsymbol{\alpha}^i}{\partial \boldsymbol{W}_F}). \quad (9)$$

The advantage of Eq. (9) lies in that $\boldsymbol{W}_F$ can be learned more flexibly for optimization convergence, which can be regarded as a skip connection across 3D2SeqViews. This is because $\boldsymbol{W}_F$ is innovatively employed as the shape class features for the learning of view-level attention $\boldsymbol{\alpha}^i$. In addition, the shape class features $\boldsymbol{W}_F$ also enable $\boldsymbol{\alpha}^i$ to simultaneously observe the encoded content information of views in $\boldsymbol{H}^i$ and the original content information of views in $\boldsymbol{A}^i$, which makes 3D2SeqViews comprehensively understand the 3D shape m^i.

IV. Experimental setup

In this section, different shape benchmarks and performance measures for shape classification and retrieval are respectively described for evaluating the 3D global features learned by 3D2SeqViews. In addition, the setup of parameters involved in 3D2SeqViews is also discussed.

A. Benchmarks and evaluations

The shape classification and retrieval experiments are conducted under three well-known large-scale 3D shape benchmarks, including ModelNet40 [16], ModelNet10 [16] and ShapeNetCore55 [29].

ModelNet40 and ModelNet10 are two subsets of ModelNet which contains 151,128 3D shapes categorized into 660 shape classes. As smaller subsets, ModelNet40 is formed by 40 shape classes with a total of 12,311 3D shapes, while ModelNet10 comprises 4,899 3D shapes split into 10 shape classes. The training and testing sets of ModelNet40 consist of 9,843 and 2,468 shapes, respectively. In addition, the training and testing sets of ModelNet10 consist of 3,991 and 908 shapes, respectively. ShapeNetCore55 is a subset of the ShapeNet dataset, and it contains 51,190 3D shapes of 55 shape classes.

In classification experiments, the metrics employed for evaluating the performances of different methods include *average instance accuracy* and *average class accuracy*. In retrieval experiments, *mean Average Precision* (mAP), *Precision and Recall* (PR) curves, *precision* (P), *recall* (R), *F1 score* (F1) and *Normalized Discounted Cumulative Gain* (NDCG) are presented to compare the performances of different methods under different benchmarks.

B. The setup of parameters

In this subsection, the key parameters involved in 3D2SeqViews are set by exploring their impacts on the performance of 3D2SeqViews in shape classification under ModelNet40. The average instance accuracy is used as the metric for the performance comparison.

In 3D2SeqViews, the key parameters include the number K of row-wise kernels for low-level view feature encoding, dropout ratio, the learning rate ε, the number V of sequential views in the view sequence captured around each 3D shape and the number N of hidden row-wise convolution layers for low-level view feature abstraction.

The number K of row-wise kernels and dropout ratio. In this experiment, the impacts of the number K of row-wise kernels and dropout ratio on the performance of 3D2SeqViews are comprehensively investigated. To explore the raw effect of K, the row-wise convolution in hidden layers is not used, i.e. $N = 0$ and $\boldsymbol{A}^i = \boldsymbol{B}^i$. In addition, the dropout is employed on the low-level view feature matrix $\boldsymbol{A}^i$.

The learning rate ε is set to 0.000002. K is iteratively selected from $\{32, 64, 128, 256, 512, 1024\}$. Similarly, the dropout ratio is iteratively selected from $\{0.7, 0.5, 0\}$. The result comparison is presented in Table I.

TABLE I
K AND DROUPOUT RATIO COMPARISON UNDER MODELNET40, $\varepsilon = 0.000002$.

Accuracy(%)	Dropout=0.7	Dropout=0.5	Dropout=0
K=32	90.88	92.10	92.10
K=64	93.19	92.50	93.07
K=128	93.03	93.07	92.99
K=256	93.15	93.03	93.15
K=512	93.15	93.27	**93.31**
K=1024	93.15	93.23	93.23

Comparison results show that the performance of 3D2SeqViews can be improved by increasing K. In addition, the dropout ratio only affects the performance of 3D2SeqViews when K is small, such as $K = 32$. The average instance accuracy achieves 93.31% with $K = 512$ and dropout ratio of 0. In addition, these results indicate that the performance is slightly affected by the dropout. This is because although there is information loss using dropout, it can be compensated by using more row-wise kernels. This observation also shows that there is no overfitting issue in our current network. Therefore, dropout is not employed in the following experiments.

The learning rate ε. In this experiment, we explore how the learning rate affects the optimization of parameters in 3D2SeqViews. ε is set to each candidate from $\{0.000001, 0.000002, 0.000004, 0.000016, 0.0001, 0.001, 0.01\}$ which are 0.5, 1, 2, 8, 50, 500 and 5000 times of 0.000002 employed in the former experiment. As the comparison shown in Table II, the results obtained with appropriate learning rates are quite well, such as $\{0.000001, 0.000002, 0.000004\}$. In addition, the result obtained with learning rate of 0.000004 is better than the ones obtained in the former experiment, which achieves up to 93.40%. While the performance is degenerated gradually with bigger learning rates, such as $\{0.000016, 0.0001, 0.001, 0.01\}$. In the following experiments, ε is set to 0.000004.

TABLE II
THE LEARNING RATE ε COMPARISON UNDER MODELNET40, K=512.

$\varepsilon \times 0.000001$	1	2	4	16	100	1000	10000
Accuracy%	93.19	93.31	**93.40**	93.23	92.46	92.17	32.30

The number V of sequential views in $\boldsymbol{v}^i$. In this experiment, the effect of the number V of sequential views is explored. Note that $V = 12$ sequential views in the view sequence captured around each 3D shape are employed to learn global features in the former experiments. As the comparison shown in Table III, both the performances of 3D2SeqViews under ModelNet40 and ModelNet10 keep improved along with increasing V until $V = 12$. In the following experiments, $V = 12$ sequential views in the view sequence captured around each shape are used to learn 3D global features.

TABLE III
V COMPARISON UNDER MODELNET, K=512, $\varepsilon = 0.000004$.

View nunmber	3	6	12	24
ModelNet40 Accuracy(%)	92.10	93.07	**93.40**	92.75
ModelNet10 Accuracy(%)	94.49	94.60	**94.71**	94.60

The number N of hidden row-wise convolution layers. In this experiment, we explore whether the performance of 3D2SeqViews could be further improved by additional row-wise convolution in hidden layers for low-level view feature abstracting. Specifically, we incrementally add $N = 2$ hidden row-wise convolution layers to abstract $\boldsymbol{A}^i$, where $\boldsymbol{D}_1^i$ employs $E_1 \in \{16, 32, 64\}$ row-wise convolution kernels to obtain multiple feature maps from $\boldsymbol{A}^i$, and $\boldsymbol{D}_2^i$ employs $E_2 = 1$ row-wise convolution kernel to combine these feature maps to form $\boldsymbol{B}^i$. In addition, the width of $\boldsymbol{A}^i$ is not reduced by the two hidden row-wise convolution layers, i.e., $q_1 = q_2 = 4096$, which aims to explore the effect of hidden row-wise convolution layers with fairly comparing the results of former experiments. As shown in Table IV, different number K of row-wise convolution kernels, such as $\{64, 128, 256, 512\}$, are employed for the performance comparison. Comparing with the results in Table I, the degenerated performances imply that the added hidden row-wise convolution layers cause overfitting. This experiment also demonstrates that the hidden row-wise convolution layers are capable of increasing the learning ability of 3D2SeqViews. According to the scale of dataset, the hidden row-wise convolution layers are not employed in the following experiments, that is, $N = 0$, $\boldsymbol{A}^i = \boldsymbol{B}^i$ and $D = 4096$.

TABLE IV
N COMPARISON UNDER MODELNET40, $\varepsilon = 0.000004$.

Accuracy(%)	E_1=16	E_1=32	E_1=64
K=64	91.49	91.73	89.91
K=128	91.10	91.57	91.33
K=256	91.20	91.57	92.05
K=512	91.05	91.86	91.92

V. RESULTS AND ANALYSIS

In this section, the performance of 3D2SeqViews is evaluated by comparing with the state-of-the-art methods in shape classification and retrieval under ModelNet40, ModelNet10 and ShapeNetCore55, respectively. For fair comparison, the results obtained by the state-of-the-art methods are computed from the single modality, such as image, voxel or point cloud.

A. Shape classification

ModelNet40. Under ModelNet40, the performance comparison in shape classification is shown in Table V, where the modality and numbers of views are also presented. The evaluation metrics, both average class precision and average instance precision, are presented in Table V if they are available in the literature.

TABLE V
CLASSIFICATION COMPARISON UNDER MODELNET40, K=512, $\varepsilon = 0.000004$.

Methods	Modality	Views	Class(%)	Instance(%)
SHD	Mesh	-	68.23	-
LFD	Image	10	75.47	-
PyramidHoG-LFD	Image	20	87.2	90.5
Fisher vector [4]	-	12	84.8	-
3DShapeNets [16]	Voxel	12	77.32	-
DeepPano [7]	Image	1	77.6	-
Geometry image [23]	Image	1	83.9	-
VoxNet [30]	Voxel	-	83.0	-
VRN [31]	Voxel	24	-	91.33
FPNN [32]	Voxel	-	88.4	-
T-L Network [18]	Voxel	-	74.4	-
3DGAN [19]	Voxel	-	83.3	-
PointNet [33]	Point	1	86.2	89.2
PointNet++ [34]	Point	1	-	91.9
FoldingNet [35]	Point	1	-	88.4
Octree [12]	Voxel	12	90.6	-
PANORAMA [22]	Image	6	90.70	-
Pairwise [25]	Image	12	90.7	-
GIFT [6]	Image	64	89.5	-
Dominant Set [8]	Image	12	-	92.2
Su-MVCNN [4]	Image	80	90.1	-
MVCNN [10]	Image	20	89.7	92.0
MVCNN-Sphere [10]	Voxel	20	86.6	89.5
Spherical projection [36]	Image	36	-	93.31
RotationNet [21]	Image	12	-	90.65
SO-Net [37]	Point	1	87.3	90.9
VGG(ModelNet40)	Image	1	-	89.47
VGG(Voting)	Image	12	90.27	92.50
Ours	Image	12	**91.51**	**93.40**
Ours1	Image	12	**91.64**	**93.27**
Ours(Start)	Image	12	90.83	93.27
Ours(No finetune)	Image	12	80.74	83.43

Using the sequential views captured around 3D shapes in the training set of ModelNet40, VGG is fine-tuned by classifying each sequential view into one of 40 shape classes ($C = 40$). The accuracy of single view classification is 89.47%, as the result named as "VGG(ModelNet40)". By voting the classification results of all sequential views in a view sequence, namely "VGG(Voting)", the instance accuracy of classifying 3D shapes is 92.50%. Fine-tuning is important for VGG to extract low-level view features. This is because VGG is pre-trained with color images from ImageNet while the sequential views are captured without colors. To verify this point, the results listed as "Ours(No finetune)" are obtained by training 3D2SeqViews under low-level view features extracted from pre-trined VGG. As analysis before, they are unsatisfactory, comparing to our best results described in the following paragraph.

Using the low-level view features from the fine-tuned VGG19, the results of 3D2SeqViews listed as "Ours" achieve 91.51% and 93.40%, as shown in the bold numbers. Our results are the best among all reported results in terms of both average class accuracy and average instance accuracy. All the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2904460, IEEE Transactions on Image Processing

compared methods learn 3D shape features from three different modalities. We find that the methods learning from voxel or point clouds are usually with the worst performance in shape classification, although they have the ability of generating 3D shapes using voxel or points. View-based methods usually perform better but these methods are still suffering from the loss of content information and spatiality among views caused by pooling and the understanding of the ambiguous views. To resolve these issues, 3D2SeqViews employs recursive view integration with view-level attention mechanism, which makes 3D2SeqViews achieve the best results. With hierarchical attention aggregation, 3D2SeqViews can aggregate views more effectively. This enables 3D2SeqViews to learn more discriminative features from only $V = 12$ views than the methods learning from $V = 20$ views or more, such as MVCNN-Sphere [10], Spherical projection [36], and Su-MVCNN [4]. For fair comparison, the result of VRN [31] is presented with a single CNN, where twice more views than ours are employed, the result of RotationNet [21] is presented with views taken by the default camera system orientation which keeps identical with other methods, and the result of Spherical projection [36] is presented with the same type of views as ours. In addition, our another set of results listed as "Ours1", obtained with another set of initialized parameters, achieves 91.64% and 93.27%, which are also state-of-the-art results in terms of average class accuracy. The comparison between "Ours" and "Ours1" implies that the unbalanced number of shapes in each shape class makes average class accuracy and average instance accuracy not positively correlated.

3D2SeqViews is able to learn the semantic meaning of a view sequence by aggregating sequential views using hierarchical attention aggregation, which makes 3D2SeqViews insensitive to the first view position. To explore this point, the result listed as "Ours(Start)" is obtained by training 3D2SeqViews with random first view position. Although the first view position is not fixed for training, the results obtained by "Ours(Start)" are comparable to our best results listed as "Ours".

ModelNet10. We further evaluate the performance of 3D2SeqViews under ModelNet10 in shape classification. All the results are compared in Table VI.

Under ModelNet10, the low-level view features are also extracted by a fine-tuned VGG19. In this experiment, VGG19 is fine-tuned by classifying each sequential view into one of 10 shape classes ($C = 10$). The accuracy of single view classification is 91.87%, as the result named as "VGG(ModelNet10)". By voting the classification results of all sequential views in a view sequence, namely "VGG(Voting)", the average instance accuracy of classifying 3D shapes is 93.83%. With low-level view features extracted by the fine-tuned VGG19, we obtain the results listed as "Ours", "Ours(256)", "Ours(Maxpool)", "Ours(Meanpool)" and "Ours(No recursive)".

As the results shown as "Ours", 3D2SeqViews achieves the best results under ModelNet10, where average class accuracy and average instance accuracy achieve up to 94.68% and 94.71%, respectively. Considering that the shapes for training in ModelNet10 are less than the ones in ModelNet40, we try to explore whether the performance of 3D2SeqViews could be further improved by a smaller number K of row-wise convolution kernels, such as $K = 256$. However, as the results listed as "Ours(256)", the results are slightly degenerated to 94.43% and 94.49%, but they are still the state-of-the-art results among all reported results.

TABLE VI
CLASSIFICATION COMPARISON UNDER MODELNET10, K=512, $\varepsilon = 0.000004$.

Methods	Modality	Views	Class(%)	Instance(%)
SHD	Mesh	-	79.79	-
LFD	Mesh	10	79.87	-
3DShapeNets [16]	Voxel	12	83.54	-
DeepPano [7]	Image	1	85.5	-
Geometry image [23]	Image	1	88.4	-
VoxNet [30]	Image	-	92.0	-
VRN [31]	Voxel	24	-	93.8
3DGAN [19]	Voxel	-	91.0	-
ORION [38]	Voxel	-	93.8	-
FoldingNet [35]	Point	1	-	94.4
PANORAMA [22]	Image	6	91.12	-
Pairwise [25]	Image	12	92.8	-
GIFT [6]	Image	64	91.5	-
RotationNet [21]	Image	12	-	93.84
3DDescriptorNet [39]	Voxel	-	-	92.4
SO-Net [37]	Point	1	93.9	94.1
VGG(ModelNet10)	Image	1	-	91.87
VGG(Voting)	Image	12	93.83	93.83
Ours	Image	12	**94.68**	**94.71**
Ours(256)	Image	12	**94.43**	**94.49**

ShapeNetCore55. In this experiment, the performance of 3D2SeqViews in shape classification is evaluated under ShapeNetCore55. 3D2SeqViews is trained by 12 sequential views ($V = 12$) rendered without colors. In addition, we also explore whether sequential views rendered with colors can be used to train 3D2SeqViews better. The sequential views with colors are downloaded from the main page of ShapeNet, however, there are only 8 sequential views ($V = 8$) to represent each 3D shape. The results are shown in Table VII.

TABLE VII
CLASSIFICATION COMPARISON UNDER SHAPENET, K=512, $\varepsilon = 0.000004$.

Methods	Modality	Views	Class(%)	Instance(%)
VGG(ShapeNetCore55)	Image	1	-	83.85
VGG(Voting)	Image	12	71.84	86.78
Ours	Image	12	**74.07**	84.58
Ours(1024)	Image	12	**72.65**	82.95
VGG1(ShapeNetCore55)	Image	1	-	83.68
VGG1(Voting)	Image	8	76.03	87.04
Ours1	Image	8	**76.12**	86.29
Ours1(1024)	Image	8	**76.38**	86.46

In Table VII, the results named as "VGG(ShapeNetCore55)" and "VGG1(ShapeNetCore55)" are obtained by fine-tuning VGG under the views without colors and the views with colors, respectively, where the classification of 3D shapes obtained by voting across sequential views are correspondingly listed as "VGG(Voting)" and "VGG1(Voting)". Because of the highly unbalanced number of shapes in each shape class, we only present our best results in terms of average class accuracy, as listed as "Ours" and "Ours1" which are obtained by learning from low-level view features employed in "VGG(ShapeNetCore55)" and "VGG1(ShapeNetCore55)",

respectively. The comparison between these results implies that the color is slightly helpful to increase the performance of 3D2SeqViews in terms of average class accuracy, i.e., from 74.07% to 76.12%. Based on the consideration that there are more 3D shapes in ShapeNetCore55 than the ones in ModelNet40, we also try to explore whether the performance of 3D2SeqViews could be improved by increasing the number K of row-wise convolution kernels, i.e., from 512 to 1024, as the results of "Ours(1024)" and "Ours1(1024)". However, the results of "Ours(1024)" or "Ours1(1024)" are comparable to "Ours" or "Ours1", respectively, which implies that the 512 row-wise convolution kernels ($K = 512$) are sufficiently good to learn from shapes under the scale of ShapeNetCore55.

B. Ablation studies

In this subsection, we conduct ablation studies to demonstrate the contribution of elements involved in 3D2SeqViews. First, we explore how much the performance of 3D2SeqViews relies on the VGG19 for low-level view feature extraction under ModelNet40. Then, we highlight the advantage of our hierarchical attention aggregation over widely used pooling under ModelNet40 and ModelNet10. Finally, we highlight the effect of recursive view integration under ModelNet40 and ModelNet10.

To explore the effect of VGG19, we replace VGG19 by several other state-of-the-art neural networks for image classification, including VGG16 [28], Resnet50 [40], and Resnet101 [40] respectively. Similar to the VGG19, we fine-tune these networks using the same single view in our training set, and then, use these fine-tuned networks to extract low-level view features. As shown in Table VIII, we see that these low-level view feature extraction networks perform slightly different. However, 3D2SeqViews can always achieve the state-of-the-art results using the low-level view features extracted by all these different networks. This observation shows that 3D2SeqViews does not rely on a particularly fine-tuned network for low-level view feature extraction. In addition, we also show the results obtained by voting the single view classification from each of these networks, listed as "*(Voting)". Our outperforming results over voting indicates that 3D2SeqViews can improve the discriminability of learned features by aggregating more information from multiple views. Moreover, by hierarchical attention aggregation, the issues in current view aggregation are resolved.

Then, we highlight the effect of hierarchical attention aggregation by replacing it with the widely used pooling in view aggregation. In this experiment, we compare our best results under ModelNet40 in Table V and ModelNet10 in Table VI with the ones obtained using max pooling and mean pooling, respectively. Specifically, $\boldsymbol{H}^i$ is pooled by max or mean calculation across the content information within all sequential views that is encoded in each feature map. As shown in Table IX, our proposed hierarchical attention aggregation outperforms "Ours(Maxpool)" and "Ours(Meanpool)" in terms of both average instance accuracy and average class accuracy. This is because max pooling or mean pooling loses a lot of content information within sequential views, and disregards the spatial relationship among the views. On the other hand, hierarchical attention aggregation can simultaneously aggregate the content information within all sequential views and the sequential spatiality among the views, where recursive view integration effectively weights view-level attention with preserving the sequential spatiality.

TABLE VIII
LOW-LEVEL VIEW FEATURE EXTRACTION NETWORK COMPARISON UNDER MODELNET40, K=512, $\varepsilon = 0.000004$.

Methods	Modality	Views	Class(%)	Instance(%)
VGG19	Image	1	-	89.47
VGG19(Voting)	Image	12	90.27	92.50
Ours(VGG19)	Image	12	**91.51**	**93.40**
VGG16	Image	1	-	89.34
VGG16(Voting)	Image	12	90.01	92.30
Ours(VGG16)	Image	12	**91.10**	**93.20**
Resnet50	Image	1	-	89.35
Resnet50(Voting)	Image	12	89.44	92.10
Ours(Resnet50)	Image	12	**90.38**	**93.35**
Resnet101	Image	1	-	89.46
Resnet101(Voting)	Image	12	90.60	92.54
Ours(Resnet101)	Image	12	**91.40**	**93.40**

TABLE IX
COMPARISON BETWEEN HIERARCHICAL ATTENTION AGGREGATION AND POOLING FOR VIEW AGGREGATION UNDER MODELNET40 AND MODELNET10, K=512, $\varepsilon = 0.000004$.

	ModelNet40		ModelNet10	
Methods	Class(%)	Instance(%)	Class(%)	Instance(%)
Ours	**91.51**	**93.40**	**94.68**	**94.71**
Ours(Maxpool)	90.20	92.59	94.41	94.49
Ours(Meanpool)	90.77	92.99	94.53	94.60

Furthermore, we highlight the effect of recursive view integration by replacing it with linear weighting. In other words, the sequential spatiality is disregarded by directly multiplying view-level attention $\boldsymbol{\alpha}^i_{norm}$ with each feature map in $\boldsymbol{H}^i$. As shown in Table X, compared to our best results under ModelNet40 in Table V and ModelNet10 in Table VI, the results listed as "Ours(No recursive)" degenerate slightly, which is caused by the lack of encoding the sequential spatiality among the views. However, "Ours(No recursive)" are still at the state-of-the-art level with the help of view-level attention and class-level attention.

TABLE X
THE EFFECT OF RECURSIVE VIEW INTEGRATION UNDER MODELNET40 AND MODELNET10, K=512, $\varepsilon = 0.000004$.

	ModelNet40		ModelNet10	
Methods	Class(%)	Instance(%)	Class(%)	Instance(%)
Ours	**91.51**	**93.40**	**94.68**	**94.71**
Ours(No recursive)	90.61	93.11	94.31	94.49

C. Attention visualization

In this subsection, the view-level attention and class-level attention learned by 3D2SeqViews under ModelNet40 are visualized, which demonstrates how 3D2SeqViews understands 3D shapes by analysing sequential views. In Fig. 4, view-level attention $\boldsymbol{\alpha}^i_{norm}$ on sequential views in $\boldsymbol{v}^i$ from all shape

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2904460, IEEE Transactions on Image Processing

classes is visualized as a matrix, such as the matrices of an airplane in Fig. 4 (a) and two different bookshelves in Fig. 4 (b) and (c), where $\boldsymbol{\alpha}^i_{norm}$ is transposed for better demonstration. The (j, c)-th entry of the matrix represents the attention paid on the j-th view by the c-th shape class. 3D2SeqViews learns the view-level attention matrices of two bookshelves with similar patterns which are much different from the one of airplane. In addition, class-level attention $\boldsymbol{\beta}^i$ employed by 3D2SeqViews is also visualized on top of each view-level attention, where each circle indicates the attention paid on each shape class by 3D2SeqViews. As shown in Fig. 4 (a), (b) and (c), class-level attention could provide valuable information to 3D2SeqViews for the learning of highly discriminative global features via employing the discriminative ability learned by the fine-tuned network. Moreover, view-level attention and class-level attention enable 3D2SeqViews to effectively combine the content information and the sequential spatiality in a view sequence with the discriminability of fine-tuned network by hierarchical attention aggregation.

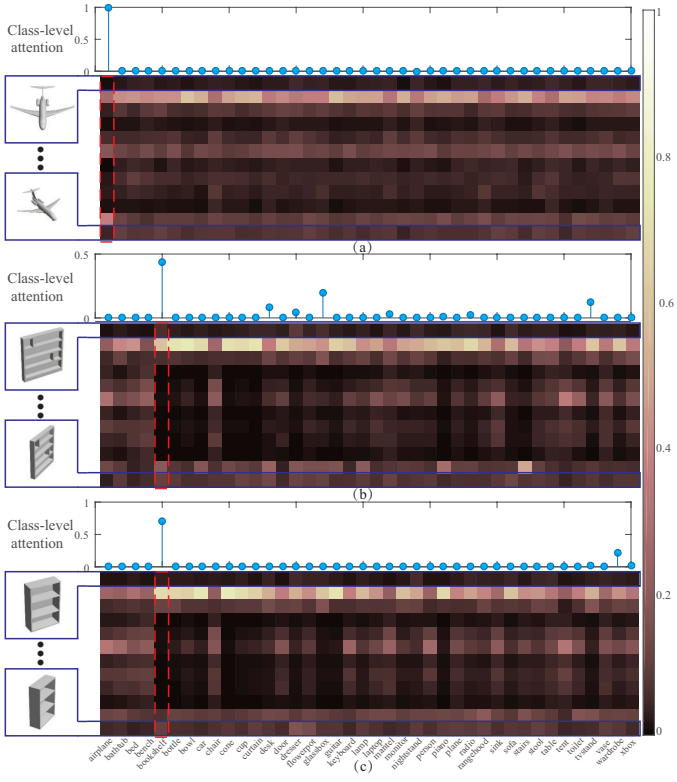


Fig. 4. The attention weights learned by 3D2SeqViews for one airplane and two bookshelves from ModelNet40, including view-level attention (demonstrated in a matrix) and class-level attention (demonstrated in a vector), as shown in (a), (b) and (c), respectively.

D. Shape retrieval

The performance of 3D2SeqViews is also evaluated using the learned global features for shape retrieval under ModelNet40, ModelNet10 and ShapeNetCore55, respectively. Under ModelNet40 and ModelNet10, our shape retrieval results are respectively produced with the global features learned by the trained 3D2SeqViews named as "Ours" in the corresponding Table V, Table VI.

Both training and testing sets are provided in ModelNet40 and ModelNet10. Thus, to comprehensively evaluate the performance of 3D2SeqViews in shape retrieval, four experiments are conducted under each benchmark. The four experiments are named as "Test-Test", "Test-Train", "Train-Train", and "All-All", indicating retrieval range formed by the sets that the query and the retrieved shapes come from, respectively. For example, "Test-Train" indicates that the shapes in the testing set are used as query to retrieve shapes from the training set.

TABLE XI
RETRIEVAL COMPARISON (MAP) UNDER MODELNET, K=512, $\varepsilon = 0.000004$.

Methods	Range	ModelNet40	ModelNet10
SHD	Test-Test	33.26	44.05
LFD	Test-Test	40.91	49.82
3DShapeNets [16]	Test-Test	49.23	68.26
Geometry image [23]	Test-Test	51.30	74.90
DeepPano [7]	Test-Test	76.81	84.18
su-MVCNN [4]	Test-Test	79.50	-
PANORAMA [22]	Test-Test	83.45	87.39
GIFT [6]	Random	81.94	91.12
Triplet-Center [41]	Test-Test	88.0	-
Ours	Test-Test	**90.76**	**92.12**
Ours	Test-Train	**93.51**	**95.26**
Ours	Train-Train	**98.76**	**99.82**
Ours	All-All	**96.98**	**98.48**

In Table XI, the comparison between 3D2SeqViews and the state-of-the-art methods is shown in terms of mAP, where the retrieval ranges are also presented. As shown by bold numbers, the proposed 3D2SeqViews completely outperforms the other compared methods in any range. Specially, in the "Test-Test" and "All-All", it achieves 90.76% and 96.98% under ModelNet40, while achieving 92.12% and 98.48% under ModelNet10. Comparing with GIFT [6] under ModelNet10 (best performing among the state-of-the-art methods), 3D2SeqViews only achieves a higher mAP about 1%, i.e., from 91.12% to 92.12%. However, the dataset used by GIFT is formed by randomly selecting 100 shapes from each shape class, which is much simpler than the whole benchmark that we used. Moreover, the corresponding PR curves of our results obtained under ModelNet40 and ModelNet10 are shown in Fig. 5 (a) and (b), respectively, where the PR curves of the results illustrate an excellent performance of 3D2SeqViews.

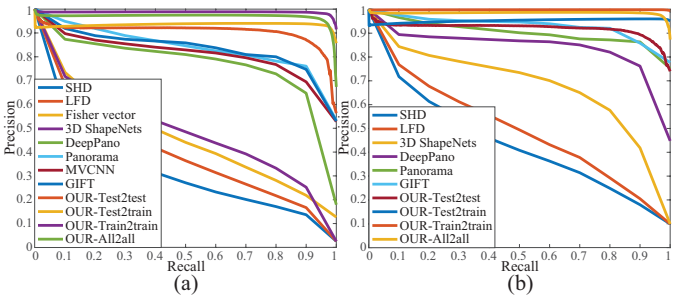


Fig. 5. The comparison between precision and recall cures obtained by different methods under (a) ModelNet40 and (b) ModelNet10.

Under the three subsets of ShapeNetCore55, i.e., training set, validation set and testing set, the performance of

3D2SeqViews in shape retrieval is compared with other state-of-the-art methods in terms of different metrics. Since there is no comparison results under training set and validation set in [42], the results of state-of-the-art methods under testing set are from the SHREC2017 retrieval contest [42], while the ones under training set and validation set are from the SHREC2016 retrieval contest [24], where these compared methods are shown as the same names in the contests. All involved 3D shapes under ShapeNetCore55 are normal, and not perturbed by rotation. In Table XII, the performances of 3D2SeqViews trained under the views without/with colors are presented, named as "Ours"/"Ours(C)". These two results are respectively produced with the learned features employed in the results of "Ours" and "Ours1(1024)" in Table VII. The comparison result shown in Table XII implies that the performance of 3D2SeqViews in shape retrieval is the best among all state-of-the-art methods under all subsets. In addition, the comparison between the results of "Ours" and "Ours(C)" also demonstrates that the colors in views for training do not significantly improve the performance of 3D2SeqViews in shape retrieval.

VI. Conclusion, limitation and future work

A. Conclusion

In this paper, 3D2SeqViews is proposed to resolve the loss of the content information and the spatial relationship caused by pooling for view aggregation in deep learning models. 3D2SeqViews is a novel deep learning model to learn 3D global features by aggregating sequential views. The proposed 3D2SeqViews is formed by CNN with a novel hierarchical attention aggregation, which effectively aggregates not only the content information within all sequential views but also the sequential spatiality among the views. In the novel hierarchical attention aggregation, view-level attention is successfully learned to indicate how much attention is paid on sequential views by each shape class for the classification of 3D shapes, which measures the distinctiveness between every pair of sequential view and shape class. The view content information is then weighted by view-level attention with preserving the sequential spatiality among the views using the novel recursive view integration. Moreover, class-level attention effectively employs the discriminative ability learned by the fine-tuned network in 3D2SeqViews, which further increases the discriminability of learned global features. The outperforming results verify that the hierarchical attention aggregation enables 3D2SeqViews to learn more discriminative features by more effectively aggregating sequential views than other state-of-the-art methods.

B. Limitation and future work

Although 3D2SeqViews achieves excellent performance on 3D global feature learning, it still suffers from a disadvantage. That is, 3D2SeqViews can only learn features by aggregating sequential views rather than any other kinds of unordered views, such as the views captured on a unit sphere centered at 3D shapes. This disadvantage prevents 3D2SeqViews from further increasing the discriminability of learned global features by more detailed characteristics of 3D shapes. From this inspiration, how to learn global features by aggregating unordered views is still eager to be resolved, which would be our next research topic in the future.

References

[1] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and X. Li, "Unsupervised 3D local feature learning by circle convolutional restricted boltzmann machine," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5331–5344, 2016.

[2] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and C. Chen, "Mesh convolutional restricted boltzmann machines for unsupervised learning of features with structure preservation on 3D meshes," *IEEE Transactions on Neural Network and Learning Systems*, vol. 28, no. 10, pp. 2268 – 2281, 2017.

[3] Z. Han, Z. Liu, J. Han, C. Vong, S. Bu, and C. Chen, "Unsupervised learning of 3D local features from raw voxels based on a novel permutation voxelization strategy," *IEEE Transactions on Cybernetics*, vol. 49, no. 2, pp. 481–494, 2019.

[4] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *International Conference on Computer Vision*, 2015, pp. 945–953.

[5] H. Huang, E. Kalegorakis, S. Chaudhuri, D. Ceylan, V. Kim, and E. Yumer, "Learning local shape descriptors with view-based convolutional neural networks," *ACM Transactions on Graphics*, 2017.

[6] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. J. Latecki, "GIFT: Towards scalable 3D shape retrieval," *IEEE Transaction on Multimedia*, vol. 19, no. 6, pp. 1257–1271, 2017.

[7] B. Shi, S. Bai, Z. Zhou, and X. Bai, "Deeppano: Deep panoramic representation for 3D shape recognition," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015.

[8] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant set clustering and pooling for multi-view 3D object recognition," in *Proceedings of British Machine Vision Conference*, 2017.

[9] Z. Han, Z. Liu, C.-M. Vong, Y.-S. Liu, S. Bu *et al.*, "BoSCC: Bag of spatial context correlations for spatially enhanced 3D shape representation," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3707–3720, 2017.

[10] C. R. Qi, H. Su, and M. Niebner, "Volumetric and multi-view cnns for object classification on 3D data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5648–5656.

[11] T. Furuya and R. Ohbuchi, "Deep aggregation of local 3D geometric features for 3D model retrieval," in *Proceedings of the British Machine Vision Conference*, 2016.

[12] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 72:1–72:11, 2017.

[13] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in *Proc. of the International IEEE Workshop on 3D Representation and Recognition*, 2015.

[14] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, "Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks," *Computer Graphics Forum*, vol. 34, no. 5, pp. 13–23, 2015.

[15] Z. Han, Z. Liu, C. Vong, Y.-S. Liu, S. Bu *et al.*, "Deep spatiality: Unsupervised learning of spatially-enhanced global and local 3D features by deep neural network with coupled softmax," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 3049–3063, 2018.

[16] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang *et al.*, "3D ShapeNets: A deep representation for volumetric shapes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[17] A. Sharma, O. Grau, and M. Fritz, "VConv-DAE: Deep volumetric shape learning without object labels," in *Proceedings of European Conference on Computer Vision*, 2016, pp. 236–250.

[18] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *Proceedings of European Conference on Computer Vision*, 2016, pp. 484–499.

[19] 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 *Advances in Neural Information Processing Systems*, 2016, pp. 82–90.

[20] 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.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2904460, IEEE Transactions on Image Processing

TABLE XII
RETRIEVAL COMPARISON UNDER SHAPENETCORE55, K=512, $\varepsilon = 0.000004$. ALL COMPARED METHODS WITHOUT CITATIONS TAKE THE SAME NAMES IN [42] (UNDER TESTING SET) OR [24] (UNDER VALIDATION AND TRAINING SETS).

		micro					macro				
Datasets	Methods	P@N	R@N	F1@N	mAP@N	NDCG@N	P@N	R@N	F1@N	mAP@N	NDCG@N
Tesing	Kanezaki	0.810	0.801	**0.798**	0.772	0.865	0.602	0.639	**0.590**	0.583	0.656
	Zhou	0.786	0.773	0.767	0.722	0.827	0.592	0.654	0.581	0.575	0.657
	Tatsuma	0.765	0.803	0.772	0.749	0.828	0.518	0.601	0.519	0.496	0.559
	Furuya	**0.818**	0.689	0.712	0.663	0.762	**0.618**	0.533	0.505	0.477	0.563
	Thermos	0.743	0.677	0.692	0.622	0.732	0.523	0.494	0.484	0.418	0.502
	Deng	0.418	0.717	0.479	0.540	0.654	0.122	0.667	0.166	0.339	0.404
	Li	0.535	0.256	0.282	0.199	0.330	0.219	0.409	0.197	0.255	0.377
	Mk	0.793	0.211	0.253	0.192	0.277	0.598	0.283	0.258	0.232	0.337
	Su	0.770	0.770	0.764	0.735	0.815	0.571	0.625	0.575	0.566	0.640
	Bai	0.706	0.695	0.689	0.640	0.765	0.444	0.531	0.454	0.447	0.548
	Taco [43]	0.701	0.711	0.699	0.676	0.756	-	-	-	-	-
	Ours	0.6002	**0.8030**	0.6107	**0.8428**	**0.9050**	0.1891	**0.8352**	0.2411	**0.7061**	**0.8537**
	Ours(C)	0.6128	**0.8035**	0.6158	**0.8521**	**0.9092**	0.1986	**0.8565**	0.2518	**0.7251**	**0.8616**
Validation	Su	0.805	0.800	**0.798**	0.910	0.938	0.641	0.671	**0.642**	0.879	0.920
	Bai	0.747	0.743	0.736	0.872	0.929	0.504	0.571	0.516	0.817	0.889
	Li	0.343	**0.924**	0.443	0.861	0.930	0.087	**0.873**	0.132	0.742	0.854
	Wang	0.682	0.527	0.488	0.812	0.881	0.247	0.643	0.266	0.575	0.712
	Tatsuma	0.306	0.763	0.378	0.722	0.886	0.096	0.828	0.140	0.601	0.801
	Ours	**0.8710**	0.1165	0.1653	**0.9540**	**0.9553**	**0.6426**	0.3742	0.3733	**0.9168**	**0.9405**
	Ours(C)	**0.8792**	0.1383	0.1904	**0.9496**	**0.9530**	**0.6486**	0.4258	0.4041	**0.9135**	**0.9388**
Training	Su	0.939	0.944	**0.941**	0.964	0.923	0.909	0.935	**0.921**	0.964	0.947
	Bai	0.841	0.571	0.620	0.907	0.912	0.634	0.452	0.472	0.815	0.891
	Li	0.827	**0.996**	0.864	0.990	0.978	0.374	**0.997**	0.460	0.982	0.986
	Wang	0.884	0.260	0.363	0.917	0.891	0.586	0.497	0.428	0.775	0.863
	Ours	**0.9902**	0.0058	0.0114	**0.9988**	**0.9843**	**0.9869**	0.0220	0.0422	**0.9987**	**0.9905**
	Ours(C)	**0.9970**	0.0059	0.0115	**0.9996**	**0.9845**	**0.9971**	0.0222	0.0426	**0.9997**	**0.9909**

[21] A. Kanezaki, Y. Matsushita, and Y. Nishida, "Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[22] 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, pp. 1–7.

[23] A. Sinha, J. Bai, and K. Ramani, "Deep learning 3D shape surfaces using geometry images," in *European Conference on Computer Vision*, 2016, pp. 223–240.

[24] M. Savva, F. Yu, H. Su, M. Aono, B. Chen *et al.*, "Shrec'16 track large-scale 3D shape retrieval from shapeNet core55," in *EG 2016 workshop on 3D Object Recognition*, 2016.

[25] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise decomposition of image sequences for active multi-view recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3813–3822.

[26] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, 2017, pp. 3856–3866.

[27] K. Xu, Y. Shi, L. Zheng, J. Zhang, M. Liu *et al.*, "3D attention-driven depth acquisition for object identification," *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 238:1–238:14, 2016.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[29] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang *et al.*, "ShapeNet: An information-rich 3D model repository," *CoRR*, vol. abs/1512.03012, 2015.

[30] D. Maturana and S. S., "Voxnet: A 3D convolutional neural network for real-time object recognition," in *International Conference on Intelligent Robots and Systems*, 2015, pp. 922–928.

[31] A. Brock, T. Lim, J. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," in *3D deep learning workshop (NIPS)*, 2016.

[32] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas, "FPNN: Field probing neural networks for 3D data." in *NIPS*, 2016, pp. 307–315.

[33] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[34] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5105–5114.

[35] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[36] Z. Cao, Q. Huang, and K. Ramani, "3D object classification via spherical projections," in *International Conference on 3D Vision*, 2017.

[37] J. Li, B. M. Chen, and G. H. Lee, "SO-Net: Self-organizing network for point cloud analysis," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[38] N. Sedaghat, M. Zolfaghari, E. Amiri, and T. Brox, "Orientation-boosted voxel nets for 3D object recognition," in *British Machine Vision Conference*, 2017.

[39] J. Xie, Z. Zheng, R. Gao, W. Wang, S.-C. Zhu, and Y. N. Wu, "Learning descriptor networks for 3D shape synthesis and analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[41] X. He, Y. Zhou, Z. Zhou, S. Bai, and X. Bai, "Triplet-center loss for multi-view 3D object retrieval," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

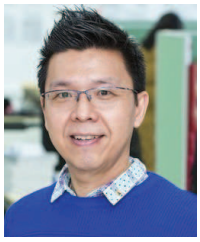
[42] M. Savva, F. Yu, H. Su, A. Kanezaki, T. Furuya *et al.*, "SHREC'17 Large-Scale 3D Shape Retrieval from ShapeNet Core55," in *Eurographics Workshop on 3D Object Retrieval*, 2017.

[43] T. S. Cohen, M. Geiger, J. Khler, and M. Welling, "Spherical CNNs," in *International Conference on Learning Representations*, 2018.

Zhizhong Han received his PhD degree from Northwestern Polytechnical University, China, 2017. He is currently a postdoctoral researcher in Department of Computer Science, University of Maryland, College Park, USA. He is also a research member of BIM group, Tsinghua University, China. His research interests include machine learning, pattern recognition, deep learning and digital geometry processing.

Honglei Lu is currently a master candidate in the School of Software at Tsinghua University. He received his BS in the School of Software, Tsinghua University. His research interests include machine learning and computer vision.

Zhenbao Liu (M'11) is currently a Professor with Northwestern Polytechnical University, China. He received the Ph.D. degree from the College of Systems and Information Engineering, University of Tsukuba, Tsukuba, Japan, in 2009. He was a visiting scholar with Simon Fraser University, Canada, in 2012. He has published approximately 50 papers in major international journals and conferences. His research interests include pattern recognition, computer vision, and shape analysis.



Chi-Man Vong (M'09-SM'14) received the M.S. and Ph.D. degrees in Software Engineering from the University of Macau in 2000 and 2005, respectively. He is currently an Associate Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau. His research interests include machine learning methods and intelligent systems.

Yu-Shen Liu (M'18) is an Associate Professor in School of Software at Tsinghua University, Beijing, China. He received his BS in mathematics from Jilin University, China, in 2000. He earned his PhD in the Department of Computer Science and Technology at Tsinghua University, China, in 2006. He spent three years as a post doctoral researcher in Purdue University from 2006 to 2009. His research interests include shape analysis, pattern recognition, machine learning and semantic search.

Matthias Zwicker is a professor at the Department of Computer Science, University of Maryland, College Park, where he holds the Reginald Allan Hahne Endowed E-nnovate chair. He obtained his PhD from ETH in Zurich, Switzerland, in 2003. Before joining University of Maryland, he was an Assistant Professor at the University of California, San Diego, and a professor at the University of Bern, Switzerland. His research in computer graphics covers signal processing for high-quality rendering, point-based methods for rendering and modeling, 3D geometry processing, and data-driven modeling and animation.

Junwei Han (M'12-SM'15) is a currently a Professor with Northwestern Polytechnical University, Xi'an, China. He received his Ph.D. degree in pattern recognition and intelligent systems from the School of Automation, Northwestern Polytechnical University in 2003. His research interests include multimedia processing and brain imaging analysis. He is an Associate Editor of IEEE Trans. on Human-Machine Systems, Neurocomputing, and Multidimensional Systems and Signal Processing.

C.L. Philip Chen (S'88CM'88CSM'94CF'07) received his M.S.degree in electrical engineering from University of Michigan, Ann Arbor, in 1985 and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, in 1988. After having worked at U.S. for 23 years as a tenured professor, as a department head and associate dean in two different universities, he is currently the Dean of the Faculty of Science and Technology, University of Macau, Macau, China and a Chair Professor of the Department of Computer and Information Science.