GVCNN: Group-View Convolutional Neural Networks for 3D Shape Recognition

Yifan Feng[†], Zizhao Zhang[‡], Xibin Zhao[‡], Rongrong Ji[†], Yue Gao[‡]*
[‡]KLISS, School of Software, Tsinghua University
[‡]Beijing National Research Center for Information Science and Technology
[†] School of Information Science and Engineering, Xiamen University
evanfeng97@gmail.com, rrji@xmu.edu.cn, {zz-z14,zxb,gaoyue}@tsinghua.edu.cn

Abstract

3D shape recognition has attracted much attention recently. Its recent advances advocate the usage of deep features and achieve the state-of-the-art performance. However, existing deep features for 3D shape recognition are restricted to a view-to-shape setting, which learns the shape descriptor from the view-level feature directly. Despite the exciting progress on view-based 3D shape description, the intrinsic hierarchical correlation and discriminability among views have not been well exploited, which is important for 3D shape representation. To tackle this issue, in this paper, we propose a group-view convolutional neural network (GVCNN) framework for hierarchical correlation modeling towards discriminative 3D shape description. The proposed GVCNN framework is composed of a hierarchical view-group-shape architecture, i.e., from the view level, the group level and the shape level, which are organized using a grouping strategy. Concretely, we first use an expanded CNN to extract a view level descriptor. Then, a grouping module is introduced to estimate the content discrimination of each view, based on which all views can be splitted into different groups according to their discriminative level. A group level description can be further generated by pooling from view descriptors. Finally, all group level descriptors are combined into the shape level descriptor according to their discriminative weights. Experimental results and comparison with state-of-the-art methods show that our proposed GVCNN method can achieve a significant performance gain on both the 3D shape classification and retrieval tasks.

1. Introduction

With the development of imaging and 3D reconstruction techniques, 3D shape recognition have become a fundamental task in computer vision with broad application prospects. Within the proliferation of deep learning, various deep networks have been investigated for 3D shape recognition, such as 3D ShapeNets [26], PointNet [7], VoxNet [14]. Among these methods, view-based method has performed best so far. In view-based method, the input data are the views taken from different angles, which can be easily captured comparing to other methods, like point cloud structure and polygon mesh. Using deep learning schemes for view representation typically refers to exploiting well-established models, such as VGG [21], GoogLeNet [23] and ResNet [9]. Besides, comparing with model-based methods, such as 3D ShapeNets [26], view-based methods can obtain much more views by rendering the 3D model.

Designing discriminative descriptor is the fundamental issue towards optimal 3D shape recognition. Although deep learning methods on 2D images have been well investigated in recent years, it is still at the beginning for describing multi-view based 3D shapes. In recent papers, the multi-view based methods, such as Multi-View Convolutional Neural Networks (MVCNN and MVCNN-MultiRes) [22, 18] usually employ a view pooling operation to generate the shape level description from the view descriptors. These methods have made the milestone for 3D shape recognition and achieve the current state-of-the-art performance. We note that all views are treated equally to generate the shape descriptor in exiting methods. However, the content relationship and the discriminative information of the views have left unexplored, which limits the performance of shape descriptors a lot. On one hand, some views are similar to each other, while the others are diverse. These similar views should contribute similarly to the shape descriptor. On the other hand, some views are more discriminative for shape recognition. Under such circumstances, it is important to further investigate the content relationship to mine the discriminative information from these views.

To tackle this issue, in this paper, we propose a group-view convolutional neural network (GVCNN) framework, which contains hierarchical view-group-shape architecture

*Corresponding author.
This work was finished when Yifan Feng visited Tsinghua University.

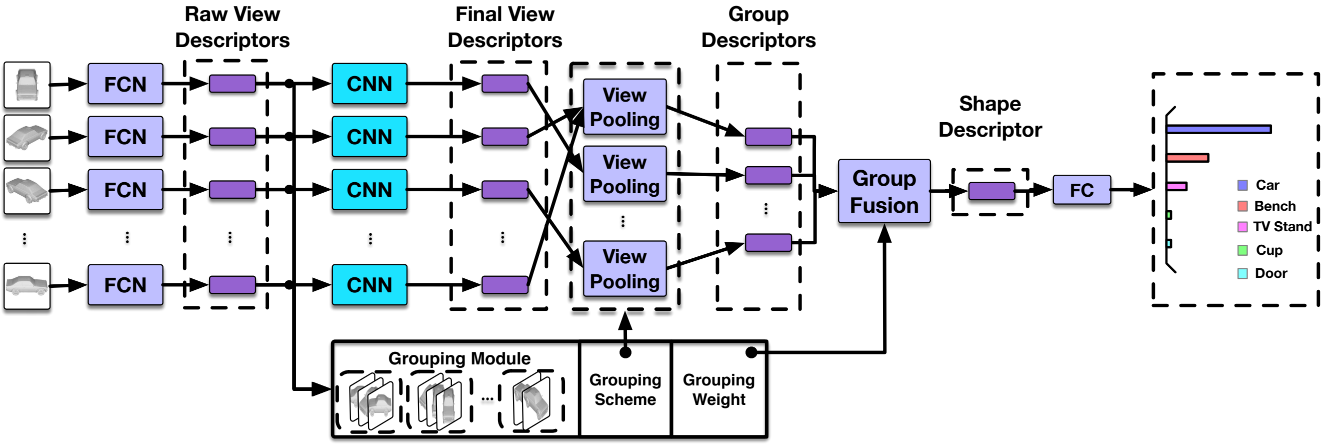


Figure 1. The Group-View CNN framework for 3D shape recognition.

of content descriptions, i.e., from the view level, the group level and the shape level. In the beginning, GVCNN groups the views to generate the view level descriptors, and assigns individual groups with associated weights, leading to the group level description. Then, the group level description can be further weighted combined to generate the shape level description. In this way, the view content and the discriminativity can be jointly considered for shape recognition. More specifically, we first use an expanded CNN to extract a view level descriptor. Then, a grouping module is proposed to estimate the content-based discrimination for each view, based on which all views can be splitted into different groups according to their discrimination level. An intra-group view pooling scheme is further proposed to generate the group level description from view level descriptions. Finally, all group level descriptors are weighted ensembled to generate the shape level descriptor. In this way, we establish a three-layer description framework, i.e., view-group-shape, which differs from the existing view-to-shape pooling scheme. To evaluate the performance of the proposed GVCNN framework, we have conducted experiments on ModelNet40 dataset, with comparisons to the state-of-the-art methods [22][18][26][11][4]. Experimental results show that our proposed GVCNN method can achieve better performance on both 3D shape classification and retrieval tasks, which demonstrates the effectiveness of the proposed framework.

The main contributions of this paper are two-fold;

- We design a three-level 3D shape description framework, consisting of a view-based end-to-end network for shape recognition. Different from the traditional view-to-shape description, our framework is composed of the view, the group and the shape levels. In particular, we take the view content relationship and the view discrimination into consideration by introducing the group level representation. Compared to the view-to-shape strategy, our framework is much more effective on representing the discriminative information of 3D shapes.
- We propose a grouping module to group the views according to their content and the discriminative information. In this way, all views for each shape can be grouped into different clusters with associated weights. Quantitative results and comparisons have shown the merits of the proposed grouping scheme.

The rest of this paper is organized as follows. We first introduce the related work in Sec.2. We then present our proposed group-view CNN architecture in Sec.3. Experiments and discussions are provided in Sec.4. Finally, we conclude this paper in Sec.5.

2. Related Work

3D shape retrieval and recognition have been investigated in recent years. In this section, we briefly review some typical handcraft and deep learning descriptors.

2.1. Handcraft Descriptors

There have been plenty of handcraft 3D descriptors, which can be mainly divided into two categories, i.e., model-based methods [15, 5] and view-based methods [4]. One typical model-based method is the statistical models, which can be used to describe the distributions of the attributes. Osada *et al.* [15] employed the shape distribution to calculate the similarity based on distance, angle, area, and volume between random surface points. Akgul *et al.* [1] proposed a probabilistic generative descriptor of local shape properties for 3D shape retrieval. Different from the distribution based methods, transform-based methods employed signal processing techniques to describe 3D shapes by Fourier transform, spherical projection, etc. Tatsuma *et al.* [24] proposed the Multi-Fourier Spectra Descriptor (MFSD) by augmenting the feature vector with spectral clustering. MFSD was composed of four independent Fourier spectras with periphery enhancement, which was able to capture the inherent characteristics of an arbitrary 3D shape regardless of the dimension, orientation, and original location of the object. The shape-based descriptor was designed based on the native 3D representations of objects, such as voxel grid [26], polygon mesh [2, 10], local shape diameters measured at densely sampled surface points [3], or extensions of the SIFT and SURF descriptors to 3D voxel grids [13].

In recent years, view-based descriptor has attracted much attention, which describes 3D shape using a group of views. Compared with model-based methods that implicitly require the model information, view-base methods only need a group of images. For instance, Lighting Field descriptor [4] is the first typical view-based 3D descriptor, which is composed of a group of ten views, captured from the vertices of a dodecahedron over a hemisphere. In [6], the similarity between two 3D objects is measured as the probabilistic matching. In panoramic object representation for accurate model attributing (PANORAMA) [16], a set of panoramic views were generated from the 3D model to represent the model surface and the orientation. In [19], Shu *et al.* proposed to employ principal thickness images for 3D shape description and classification.

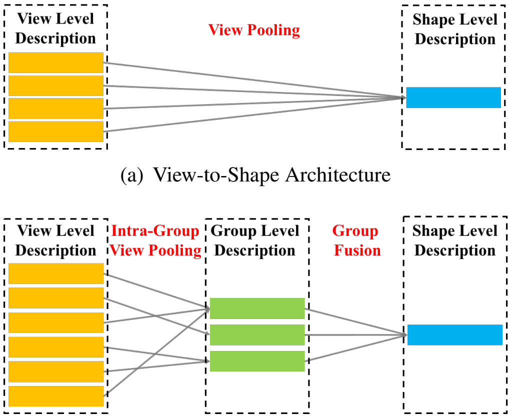
2.2. Deep Learning Based Descriptors

In recent years, deep learning methods have been widely investigated in 3D shape description. Su *et al.* [22] proposed a multi-view convolutional neural network (MVCNN), which first generated the feature for each view individually base on convolutional neural networks and then fused multiple views by a pooling procedure. MVCNN further employs a low-rank Mahalanobis metric [20] to improve the retrieval performance. To jointly utilize the model information and the view data, Qi *et al.* [18] combined view-based descriptor and volumetric-based descriptor by taking these two types of information into consideration in the network. More specifically, multi-resolution views were employed in [18]. In [8], Guo *et al.* proposed to a unified multi-view 3D shape retrieval method with a deep embedding network to handle the complex intra-class and inter-class variations, in which the deep convolutional network can be jointly supervised by classification loss and triplet loss. Xie *et al.* [28, 27] proposed a deep auto-encoder for 3D shape feature extraction. In [29], a progressive shape-distribution encoder was introduced to generate 3D shape representation.

Deep learning based methods have shown superior performance compared with the traditional handcraft descriptors. It is noted that multiple views for each 3D shape could have different importance on shape description. However, existing deep learning methods mainly conducted information pooling on all views equally, ignoring the discriminative information of different views, which limits the performance of existing methods.

3. Group-View Convolutional Neural Network

In this section, we introduce the proposed GVCNN framework in details. Compared with previous view-to-shape architecture, as shown in Fig. 2 (a), considering the relationship among the content of the views and the discriminativity of different views, we introduce a hierarchical view-group-shape framework. In our proposed GVCNN framework, a group level description is first generated from all the view level descriptors. In this step, the correlation among these views are taken into consideration by the grouping procedure, and the weights for different groups are also calculated to quantify the discriminativity of these groups of views. Then, we finally generate the shape level description by weighted combines these group level descriptions.



(a) View-to-Shape Architecture

(b) View-Group-Shape Architecture

Figure 2. The comparison between the traditional view-to-shape architecture and the proposed view-group-shape architecture for shape description.

Fig.1 illustrates the detailed flowchart of our proposed method. GVCNN employs the GoogLeNet as the base architecture. The “FCN” part is the top five convolutional layers of GoogLeNet. The “FC” part has appeared twice: One is the last layer of GVCNN to perform classifier, another is in Group Module to extract discrimination scores from mid-level representation (the output of “FC”). “CNN” is the same as GoogLeNet. The output of Group Module will fuse view descriptors to product the shape descriptor. Then the shape descriptor will be sent into one “FC” layer to get the final classification result. Given a 3D shape, we first take

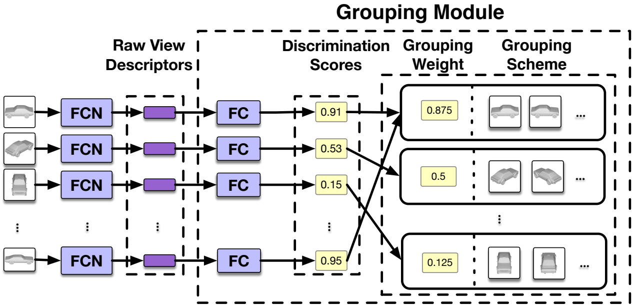


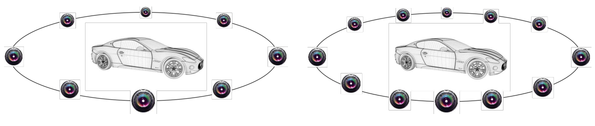
Figure 3. The group module use the same input views as the GVCNN. We use a FC layer to obtain the discrimination scores from raw view descriptors. Then, we group these views base on the discrimination scores to get the grouping scheme and grouping weights. The group scheme is used to supervise the intra-group view pooling. After the intra-group view pooling, the grouping weights are used for group combination to the shape descriptor.

a set of views captured from different angles. Each view is passed through the first part of the network (FCN) to get the raw descriptor in the view level. Then, the second part of the network (CNN) and the group module, are used to extract the final view descriptors together with the discrimination scores, separately. The discrimination scores are used to group these views and a intra-group view-pooling step is conducted to extract a group level descriptor. Finally, all group descriptors are combined into a shape level description according to their grouping weights produced by the grouping module.

3.1. Raw View Descriptor Generation

Given a 3D shape, which is usually stored as polygon meshes or point clouds, the first step is to generate a set of virtual images from the virtual 3D model. To capture the visual data of the 3D shape as completely as possible, we designed two types of predefined camera arrays, and generate rendering views from the 3D shape. The first camera array contains 8 cameras, which are set as a horizontal circle with 45 degrees interval. Therefore, there are 8 views for this camera array. The second camera array contains 12 cameras, which are set as a horizontal circle with 30 degrees interval. Therefore, there are 12 views for this camera array. In our experiments, these two types of multi-view data are employed. The two employed camera array settings are shown in Fig. 4 We note that the proposed framework has no constraint on the rendering method, and other multi-view capturing approaches can be also used in our method.

Given such a set of views for each 3D shape, we design a full convolutional network (FCN) to extract the raw view descriptors, as shown in Fig. 1. Compared with deeper CNN, shallow FCN could have more position information, which is needed for the followed grouping module. And the deeper CNN will have the content information which could represent the view feature better.



(a) 8 Views

(b) 12 Views

Figure 4. The camera array settings for 8 views and 12 views.

3.2. Grouping Module

The grouping module aims to learn the group information to assist in mining the relationship among views. In order to make the grouping module better integrated into the convolutional neural network, we designed a unique grouping mechanism.

Formally speaking, there is an output unit connected to the last layer of FCN by a FC layer. Given a set of views $S = \{I_1, I_2, \cdots, I_N\}$, the output of this unit is denoted as $\{O_{I_1}, O_{I_2}, \cdots, O_{I_N}\}$. We use a function $\xi(\cdot)$ to quantify the discrimination of a view, which is defined as

$$\xi(I_i) = sigmoid\Bigg(log\Big(abs\big(O_{I_i}\big)\Big)\Bigg). \tag{1}$$

We notice that the output of sigmoid function will approach to 0 or 1 when the input of sigmoid function is larger than 5 or less than -5. Thus we add the abs and log function before the sigmoid function. After getting the discrimination score of each view, we divide the range of discrimination score $(0, 1)$ into N sub-range with the same length. Views with discrimination scores in the same sub-range belong to the same group. Thus we divide the N views into M groups $\{G_1, G_2, \cdots, G_M\}$. Note that $1 \leq M \leq N$ because there may exist sub-ranges that have no views falling into it. The merit of this grouping scheme is that we don't have to fix the number of input views N and the number of groups M, which is more flexible and practical.

As mentioned above, the group module not only decides which group each view belongs to, but also determines the weight of each group when conducting group fusion. The more discriminative group should have higher weights and vice versa. Thus we define the weight of group G_j as:

$$\xi(G_j) = \frac{Ceil(\xi(I_k) \times |G_j|)}{|G_j|} \qquad I_k \in G_j \tag{2}$$

In this way, we can have both the grouping scheme (with group information) and the grouping weights, which can be used for the following intra-group view pooling and group fusion procedures.

3.3. Intra-Group View Pooling

Given the view descriptors and the generated grouping information, the objective here is to conduct intra-group view pooling towards a group level description.

After the grouping procedure, the views in one group share similar content and also are with close discriminations. Here, all the views in the same group pass through a view pooling layer to get a group level description. Let D_{I_i} be the view descriptor of I_i, and D_{G_j} be the group descriptor of G_j. The relationship between G_j and I_i can be written as

$$D(G_j) = \frac{\Sigma_{i=1}^{N} \lambda_i D_{I_i}}{\Sigma_{i=1}^{N} \lambda_i}, \tag{3}$$

$$\lambda_i = \begin{cases} 1 & I_i \in G_j, \\ 0 & I_i \notin G_j. \end{cases}$$

The intuition behind Eq.3 is that the views in the same group have the similar discrimination, which are assigned the same weight.

After this step, we can have several group level descriptors and the corresponding weights.

3.4. Group Fusion

To generate the shape level description, all these group level descriptors should be further combined. Therefore, we conduct a weighted fusion process using all group descriptors according to Eq.2 to get the final 3D shape descriptor $D(S)$

$$D(S) = \frac{\Sigma_{j=1}^{M} \xi(G_j) D(G_j)}{\Sigma_{j=1}^{M} \xi(G_j)}. \tag{4}$$

In this way, the groups containing more discriminative views contribute more to the final 3D shape descriptor $D(S)$ than those containing less discriminative views. By using these hierarchical view-group-shape description framework, the important and discriminative visual content can be discovered in the group level, and thus emphasized in the shape descriptor accordingly.

3.5. Classification and Retrieval

Classification. Given C classes in the classification task, the output of the last layer in our network architecture is a vector with C elements, *i.e.*, $V = \{v_1, v_2, \cdots, v_C\}$. Each element represents the probability that the subject belongs to that category. And the category with the largest value is the category it belongs to.

Retrieval. In GVCNN, the shape descriptor comes from the output of group fusion module, which is more representative than the view descriptor extracted from single view. And we directly use it for 3D shape retrieval. For two 3D shape X and Y, x and y is the shape descriptor extracted from GVCNN. Concretely, we use Euclidean distance between two 3D shapes in retrieval. The distance metric formula is defined as:

$$d(X, Y) = \|x - y\|_2. \tag{5}$$

We further adopt a low-rank Mahalanobis metric. We learn a Mahalanobis metric W that directly projects GVCNN descriptors to a new space, in which the intra-class distance is smaller and inter-class distance is larger. We use the large-margin metric learning algorithm and implementation from [20].

4. Experiments

In this section, we first provide the experiments on 3D shape classification and retrieval, and also discuss the results and comparison with the state-of-the-art methods. Following we provide the experiments on investigating the grouping module of our proposed framework. In the last part, we investigate the influence of the number of views on the performance of 3D shape recognition.

4.1. 3D Shape Classification and Retrieval

To evaluate the performance of the proposed GVCNN method, we have conducted 3D shape classification and retrieval experiments on the Princeton ModelNet dataset [25]. ModelNet is composed of 127,915 3D CAD models from 622 object categories. We further subsample ModelNet40 as a the subset of ModelNet, which contains 40 popular object categories. We follow [26] to conduct the training/testing split.

In experiments, our GVCNN is compared with the Multi-view CNN by Su *et al.* [22], MVCNN-MultiRes by Qi *et al.* [18], which employs multi-resolution views, 3D ShapeNets by Wu *et al.* [26], Spherical Harmonics descriptor (SPH) by Kazhdan *et al.* [11], which is a typical model-based method, Lighting Field descriptor (LFD) by Chen *et al.* [4], which is a typical view-based method, PointNet by Qi *et al.* [17], which is a typical point clouds method, and KD-Network by Klokov *et al.* [12].

The experimental results and comparison among different methods are demonstrated in Tab. 1. The proposed GVCNN with 8 views achieves the best classification accuracy of 93.1%. It has gains of 3.44% and 1.86% compared with MVCNN with 80 views and the MVCNN-MultiRes, respectively. In the retrieval experiments, GVCNN with 8 views and 12 views achieves the best retrieval mAP of 79.7% and 81.3%, respectively, which largely boosts from MVCNN with 80 views of 70.4%.

When the low-rank Mahalanobis metric learning is further included, all compared methods can achieve better performance on the retrieval task. In our method, the descriptors extracted from the GVCNN is a 2,048-dimensional vector. We use the large-margin metric learning to learn a projection matrix M, which projects the sparse matrix of 2,048 dimensions to another subspace of 128 dimensions. Then, we use the projected shape descriptors to represent 3D shapes for retrieval. By using a learned metric, GVCNN with 8 views achieves an mAP of 84.5% and GVCNN with

Method	Training Config.		Test Config.	Classification	Retrieval
	Pre train	Fine tune	#Views	(Accuracy)	(mAP)
(1)SPH[11]	-	-	-	68.2%	33.3%
(2)LFD[4]	-	-	-	75.5%	40.9%
(3)3D ShapeNets[26]	ModelNet40	ModelNet40	-	77.3%	49.2%
(4)MVCNN[22], 12×	ImageNet1K	ModelNet40	12	89.9%	70.1%
(5)MVCNN[22], metric,12×	ImageNet1K	ModelNet40	12	89.5%	80.2%
(6)MVCNN[22], 80×	ImageNet1K	ModelNet40	80	90.1%	70.4%
(7)MVCNN[22], metric, 80×	ImageNet1K	ModelNet40	80	90.1%	79.5%
(8)MVCNN-MultiRes[18]	-	ModelNet40	-	91.4%	-
(9)PointNet[17]	-	ModelNet40	-	89.2%	-
(10)KD-Network[12]	-	ModelNet40	-	91.8%	-
(11)MVCNN(GoogLeNet), 8×	ImageNet1K	ModelNet40	8	92.0%	74.62%
(12)MVCNN(GoogLeNet), metric, 8×	ImageNet1K	ModelNet40	8	92.0%	83.3%
(13)MVCNN(GoogLeNet), 12×	ImageNet1K	ModelNet40	12	92.2%	74.1%
(14)MVCNN(GoogLeNet), metric, 12×	ImageNet1K	ModelNet40	12	92.2%	83.0%
(15)GVCNN, 8×	ImageNet1K	ModelNet40	8	**93.1**%	79.7%
(16)GVCNN, metric, 8×	ImageNet1K	ModelNet40	8	**93.1**%	84.5%
(17)GVCNN, 12×	ImageNet1K	ModelNet40	12	92.6%	81.3%
(18)GVCNN, metric, 12×	ImageNet1K	ModelNet40	12	92.6%	**85.7**%

* metric=low-rank Mahalanobis metric learning

Table 1. Classification and retrieval results on the ModelNet40 dataset. On the top are results using state-of-the-art 3D shape descriptors. MVCNN(GoogLeNet) means we use the GoogLeNet as the base architecture and add view pooling layer like MVCNN. And the position of its view pooling layer is the same as the fusion module of GVCNN. The GVCNN architecture outperforms the view-based methods, especially for retrieval.

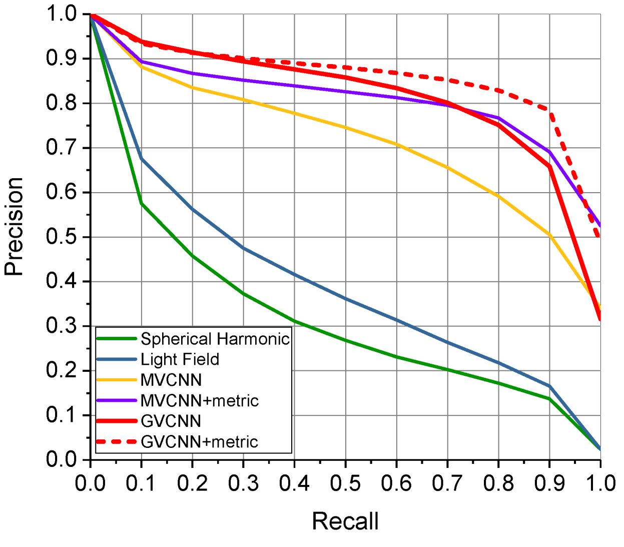


Figure 5. Precision-recall curves for compared methods on the task of 3D shape retrieval on the ModelNet40 dataset. In these experiments, 12 views are used in both MVCNN and GVCNN methods. Our method (GVCNN+metric) significantly outperforms the state-of-the-art on this task and achieves 85.7% mAP.

12 views achieves an mAP of 85.7%, which are the best compared to all methods. Both results demonstrate the effectiveness of the proposed GVCNN.

Note that GVCNN employs the GoogLeNet as the base architecture, which differs from MVCNN that uses ImageNet pre-trained VGG-m as the base architecture. To evaluate the contribution of the base architectures, we further conduct experiments of MVCNN with GoogLeNet, whose results are shown in Tab.1. It is clear that the use of GoogLeNet can improve the performance of MVCNN. For example, MVCNN(GoogLeNet) with 12 views achieves gains of 2.7% and 2.8% with metric learning compared with MVCNN [22]. Using the same base architecture, GVCNN with 12 views using metric learning achieves 0.4% and 2.7% gains compared with MVCNN(GoogLeNet) in the recognition and retrieval tasks, respectively.

Fig.5 quantizes the precision-recall curves of all compared methods. For MVCNN and GVCNN, 12 views are used. As shown, GVCNN and GVCNN+metric significantly outperform MVCNN and MVCNN+metric, respectively.

Our performance is dedicated to the following reasons. GVCNN contains a grouping module, which can identify view groups and also assign weights for each group. In this way, similar views can be grouped together and the features can be pooling in each group, rather than pooling on all views. Compared to MVCNN, our grouping can be re-

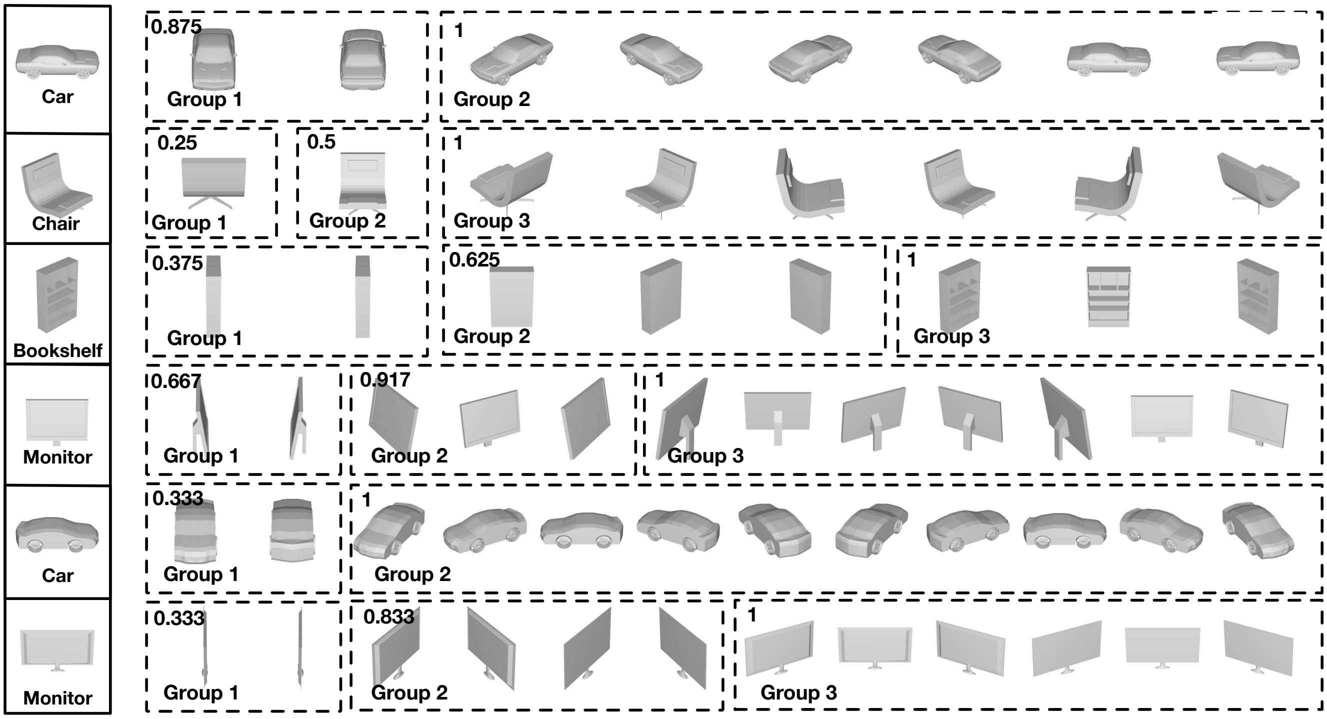


Figure 6. In the figure, each line is the output of group module in a query. The first three lines are the query with 8 views. And the last three lines are the query with 12 views. The weight of each group is shown in the upper left corner of each group box.

garded as a mid-level pooling, which is better than treating all views equally. Besides, the group weights can be used to better ensemble such groups. It is noted that some views could be very discriminative for shape recognition, while some others may be not. As an accommodation, the proposed method can generate a weight for each view group to identify whether it is good for recognition. Therefore, the weighted fusion leads to better performance compared to direct pooling on all views.

4.2. On the Grouping Module

In our pipeline, the grouping module plays an important role. We further investigate this grouping module, whose objective is to identify the content that whether they are discriminative to the corresponding labels. It is expected that the views in the same group could share similar content with closer discriminativity. We have demonstrated some grouping examples in Fig. 6.

As shown in this figure, similar content of the same shape can be grouped together. For example, in the first example, all 8 car views are divided into two groups. The first group is mainly the front and back views of the car, while the second is the side views. In the fourth example, all 8 chair views are divided into three groups. The first group is from the back direction, the second group is from the front direction, and the third is from side directions. Similar observations can be obtained from other examples. These results can demonstrate that the proposed grouping module is effective on clustering visual content.

Another important property of the grouping module is the weight estimation of different groups. Some views could be highly discriminative for the 3D shape, while the others may be not. Therefore, we further investigate the learned weights for different groups of views. Here we take the first shape in Fig. 6 as an example. The two groups are with weights of 0.875 and 1, respectively. Comparing to the back and front views of the car shape, the side views are much more discriminative, and thus *Group 2* are assigned with a higher weights compared with *Group 1*. In the example of chair, the first group is in the back view, which is quite similar to a monitor and thus not very useful for identifying its true category. The second group is the front view, which is slightly better than the first group, as it has a clearer chair shape. Compared with these two groups, the third group is the side view, which has clear chair shape, and is quite useful for recognition. In this example, the weights for these three groups are 0.25, 0.50, and 1, corresponding to the discriminative power of these views.

4.3. On the Number of Views

Another important issue is the number of views for each shape. We have also quantitatively evaluate its influence

Training Config. #Views	Test Config. #Views	Classification (accuracy)
8	1	70.0%
8	2	71.2%
8	4	91.1%
8	8	93.1%
8	12	91.5%
8	8*	84.3%
12	1	75.0%
12	2	76.8%
12	4	90.3%
12	8	92.1%
12	12	92.6%
12	12*	85.3%

Table 2. The comparison of different number of input views. The first five lines are the network trained on 8 views. The last five lines are the network trained on 12 views. Both have bad performance with the number of input views less than four.

on the classification performance. More specifically, we fix the number of input views for training, and generate two networks, one from the training data with 8 views and the other from the training data with 12 views. Note that it may be not feasible to have exactly the same number of views or have exactly the same view direction as the training data. In practice, it is possible to have just several randomly captured views or just a few number of views. In the testing stage, we have varied the number of views from 1 to 12 for both networks. The experimental results on the classification task are provided in Tab. 2.

Clearly, when the number of views is quite small, such as 1 or 2, the classification performance is very poor. This is reasonable that too few views lost much information of the 3D shape. With more views, such as 4, 8 or more, the performance increases very fast and becomes much stable. For instance, given 4 views, the network trained with 12 views can achieve a classification accuracy of 90.3%, while given 8 views, the accuracy can be further improved to 92.1%.

We also have investigated the influence of view generation. We first generate a pool of views for each shape. More specifically, we extract 80 viewpoints from 80-face semiregular polyhedron which is generated from the icosahedron using butterfly subdivision with 42 vertices. Then, we randomly select 8 and 12 views from these 80 views and conduct shape recognition. We repeat 10 times and the average performance and the standard deviation are reported in Tab. 2, denoted as 8^* and 12^*, respectively. When the views are randomly selected, the performance becomes worse. Actually, if all the views are captured from the identical or close direction, it turns to the case of using just 1 or a few views, which will significantly degrade the performance. However, if we just randomly capture views from different directions, the performance could be very steady.

One circle			Half circle		
Test #views	Train #views	Accu- racy	Test #views	Train #views	Accu- racy
8	12	91.6%	8	12	88.8%
12	12	91.8%	12	12	90.3%
8	8	90.2%	8	8	87.9%
12	8	91.3%	12	8	87.6%

Table 3. The comparison of different generating-view conditions. In the left subtable, the views are randomly selected from 80 horizon views. In the right subtable, the views are randomly selected from half of the circle directions.

We also provide the experiments on 8/12 random views from the same horizon circle setting. We have generated 80 views from the horizon circle direction, and 8/12 random views were selected from the whole circle or just half of it for testing. The classification results are in Tab. 3. In the same training and testing configure, the classification of input with views from a circle outperforms input with views from the half circle. And in the same input condition (randomly select views from a circle or half of a circle), the accuracy of testing with 12 views is higher than that of 8 views. Training with 12 views in general outperforms training with 8 views.

5. Conclusions

In this paper, we proposed a GVCNN framework for 3D shape recognition. In this method, a hierarchical shape description framework is introduced, including the view, the group, and the shape level descriptor. The correlation among the views for each shape is taken into consideration, and the grouping information is utilized for shape representation. Compared with traditional methods, the proposed method not only considers the view level pooling, but also takes the group information in the pooling procedure. Experimental results and comparisons with the state-of-the-art methods have demonstrated the effectiveness of the proposed method. We have also investigated the influence of different numbers of views for 3D shape representation. The results indicate that more and relatively complete views can be better for 3D shape recognition.

Acknowledgements

This work was supported by National Key R&D Program of China (Grant No. 2017YFC0113000), and National Natural Science Funds of China (U1701262, U1705262, 61671267).

References

[1] C. B. Akgül, B. Sankur, Y. Yemez, and F. Schmitt. 3D model retrieval using probability density-based shape descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1117–1133, 2009. 2

[2] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics*, 30(1):1, 2011. 3

[3] S. Chaudhuri and V. Koltun. Data-driven suggestions for creativity support in 3D modeling. *ACM Transactions on Graphics*, 29(6):183, 2010. 3

[4] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On visual similarity based 3D model retrieval. In *Computer Graphics Forum*, volume 22, pages 223–232. Wiley Online Library, 2003. 2, 3, 5, 6

[5] I. Chiotellis, R. Triebel, T. Windheuser, and D. Cremers. Non-rigid 3D shape retrieval via large margin nearest neighbor embedding. In *European Conference on Computer Vision*, pages 327–342. Springer, 2016. 2

[6] Y. Gao, J. Tang, R. Hong, S. Yan, Q. Dai, N. Zhang, and T.-S. Chua. Camera constraint-free view-based 3D object retrieval. *IEEE Transactions on Image Processing*, 21(4):2269–2281, 2012. 3

[7] 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 *International Joint Conference on Neural Networks*, pages 1578–1584. IEEE, 2016. 1

[8] H. Guo, J. Wang, Y. Gao, J. Li, and H. Lu. Multi-view 3D object retrieval with deep embedding network. *IEEE Transactions on Image Processing*, 25(12):5526–5537, 2016. 3

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[10] I. In Kokkinos, M. M. Bronstein, R. Litman, and A. M. Bronstein. Intrinsic shape context descriptors for deformable shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 159–166. IEEE, 2012. 3

[11] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing*, volume 6, pages 156–164, 2003. 2, 5, 6

[12] R. Klokov and V. Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3D point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 863–872. IEEE, 2017. 5, 6

[13] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool. Hough transform and 3D surf for robust three dimensional classification. *European Conference on Computer Vision*, pages 589–602, 2010. 3

[14] D. Maturana and S. Scherer. Voxnet: A 3D convolutional neural network for real-time object recognition. In *IEEE International Conference on Intelligent Robots and Systems*, pages 922–928. IEEE, 2015. 1

[15] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, 2002. 2

[16] P. Papadakis, I. Pratikakis, T. Theoharis, and S. Perantonis. Panorama: A 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval. *International Journal of Computer Vision*, 89(2):177–192, 2010. 3

[17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1(2):4, 2017. 5, 6

[18] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3D data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5648–5656, 2016. 1, 2, 3, 5, 6

[19] Z. Shu, S. Xin, H. Xu, L. Kavan, P. Wang, and L. Liu. 3D model classification via principal thickness images. *Computer Aided Design*, 78:199–208, 2016. 3

[20] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *British Machine Vision Conference*, volume 2, page 4, 2013. 3, 5

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 1

[22] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–953, 2015. 1, 2, 3, 5, 6

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 1

[24] A. Tatsuma and M. Aono. Multi-fourier spectra descriptor and augmentation with spectral clustering for 3D shape retrieval. *The Visual Computer*, 25(8):785–804, 2009. 3

[25] The Princeton ModelNet. `http://modelnet.cs.princeton.edu/`. 5

[26] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 1, 2, 3, 5, 6

[27] J. Xie, G. Dai, F. Zhu, E. K. Wong, and Y. Fang. Deepshape: Deep-learned shape descriptor for 3D shape retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1335–1345, 2017. 3

[28] J. Xie, Y. Fang, F. Zhu, and E. Wong. Deepshape: Deep learned shape descriptor for 3D shape matching and retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1275–1283, 2015. 3

[29] J. Xie, F. Zhu, G. Dai, L. Shao, and Y. Fang. Progressive shape-distribution-encoder for learning 3D shape representation. *IEEE Transactions on Image Processing*, 26(3):1231–1242, 2017. 3