Information Sciences 547 (2021) 984–995

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Hierarchical multi-view context modelling for 3D object classification and retrieval

An-An Liu [a], Heyu Zhou [a], Weizhi Nie [a,*], Zhenguang Liu [b], Wu Liu [c], Hongtao Xie [d], Zhendong Mao [d], Xuanya Li [e], Dan Song [a,*]

[a] School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China
[b] Zhejiang Gongshang University, Hangzhou 310018, China
[c] AI Research of JD, Beijing 100105, China
[d] School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China
[e] Baidu Inc., Beijing 100105, China

ARTICLE INFO

Article history:
Received 14 March 2020
Received in revised form 17 September 2020
Accepted 24 September 2020
Available online 2 October 2020

Keywords:
Multi-view learning
3D object retrieval
3D object classification

ABSTRACT

Recent advances in 3D sensors and 3D modelling software have led to big 3D data. 3D object classification and retrieval are becoming important but challenging tasks. One critical problem for them is how to learn the discriminative multi-view visual characteristics. To address it, we proposes a hierarchical multi-view context modelling method (HMVCM). It consists of four key modules. First, the module of view-level context learning is designed to learn visual context features with respect to individual views and their neighbours. This module can imitate the human need to look back and forth to identify and compare the discriminative parts of individual 3D objects based on a joint convolutional neural network (CNN) and bidirectional long short-term memory (Bi-LSTM) network. Then, a multi-view grouping module is introduced to split views into several groups based on their visual appearance. A raw group-level representation can be obtained by the weighted sum of the view-level descriptors. Furthermore, we employ the Bi-LSTM to exploit the context among adjacent groups to generate group-wise context features. Finally, all group-wise context features are fused into a compact 3D object descriptor according to their significance. Extensive experiments on ModelNet10, ModelNet40 and ShapeNetCore55 demonstrate the superiority of the proposed method.

1. Introduction

With the rapid development of 3D techniques and computer graphics hardware, advanced 3D equipment contributes to the availability of large-scale 3D object databases (e.g., ModelNet and Google 3D Warehouse) in a wide range of fields, such as geospatial science, civil engineering and medical engineering [1–4]. Since the number of 3D objects is growing rapidly, effective and efficient 3D object classification and retrieval algorithms are urgently desired for large-scale 3D object management.

The typical 3D object classification and retrieval methods can be classified into model-based and view-based methods [5,3,6]. For model-based methods, 3D representation is directly generated from 3D data [3,7], such as voxels, point clouds

[*] Corresponding authors.
E-mail addresses: weizhinie@tju.edu.cn (W. Nie), dan.song@tju.edu.cn (D. Song).

https://doi.org/10.1016/j.ins.2020.09.057

and meshes. Although model-based methods can perform well for 3D object analysis, the 3D CNN induced expensive computations, and the inconvenient obtainment of 3D data in real scenarios restricts its application. For view-based methods, the mainstream methods enable the 2D CNN to extract the visual features on multiple views rendered from the specific 3D object to generate multi-view representation [5]. Inspired by the mature and successful application of deep learning on image classification and retrieval fields, view-based methods are usually superior to model-based methods.

1.1. Motivation

Although view-based methods have achieved significant progress in 3D object classification and retrieval in recent years, there still exist two critical problems.

1.1.1. How to represent the discriminative multi-view characteristics of individual 3D objects

The popular view-based methods [8] usually extract 3D visual representation by leveraging view images with respect to individual viewpoints while ignoring the visual transition among multiple viewpoints. However, view images are not always reliable for shape description. As shown in Fig. 1, if the blue and orange view images are selected for 3D object representation, it is difficult to identify their correct categories (car and desk) since the visual appearance of individual views is not sufficiently discriminative. Comparatively, the visual transition among adjacent view images can contribute more to 3D visual representation. However, merely relying on the spatial position of the virtual camera to capture the visual transition among adjacent view images is not sufficiently discriminative since the view content and transition cannot be fully explored. As shown in Fig. 1, the visual appearance of the views in the same group (views with the identical number) is similar, and consequently, the visual transition among them is insignificant. Comparatively, the views in different groups have significant visual differences, and consequently, the group-wise context information is more discriminative. Therefore, hierarchical multi-view context discovery among adjacent view images and groups can benefit from discriminative visual representation for 3D object classification and retrieval.

1.1.2. How to integrate multi-view information for visual representation

The representative view-based methods [9,10] employ CNNs with the maximum operation across multi-view features for visual representation. The maximum operation can only preserve the maximum response of filters from multiple views while ignoring potential contextual information among adjacent views. Actually, it is quite intuitive that humans can easily recognize a 3D object by identifying its salient viewpoints. As shown in Fig. 1, if we select the green and gold view images as representative views or assign these views with high weights for both 3D objects, we can easily classify them. However, the representative views from different viewpoints may be visually similar to each other. Only focusing on these similar representative views is unreasonable, and it may cause two issues: 1) the visual transition between informative and uninformative views cannot be fully exploited, and 2) the discriminative and diverse information conveyed in the multi-view sequence is neglected. As shown in Fig. 1, the views in the same group, with similar visual content, should contribute almost equally to visual representation, while the diverse views in different groups, with the large discrepancy of visual discrimination, should make different contributions. Therefore, it is critical to explore the discrimination of individual views and the relationship among them for multi-view fusion.

To handle these problems, we propose a hierarchical multi-view context modelling method (HMVCM) for 3D object classification and retrieval. As shown in Fig. 2, HMVCM mainly contains four modules: view-level context learning, multi-view grouping module, group-level context learning and group context fusion module. First, for the rendered multi-view images of individual 3D objects, the 2D CNN pre-trained on ImageNet is implemented to extract the low-level visual features. We design the Bi-LSTM, which is a combination of two LSTM modules but with the opposite input order, to learn the visual transition among adjacent views to generate view-level context features. Then, we propose the multi-view grouping module to divide the views into several groups according to the visual appearance and employ the weighted sum strategy to obtain the group-wise features. Obviously, directly fusing the raw group-wise features for 3D object representation is not reasonable enough since the visual discrimination among adjacent groups has not been explored. Therefore, we design a group-level context learning module. Since the groups associated with multiple views may not equally contribute to the representation of 3D objects, we proposed the group context fusion module to fuse them adaptively for 3D object representation. The gen-

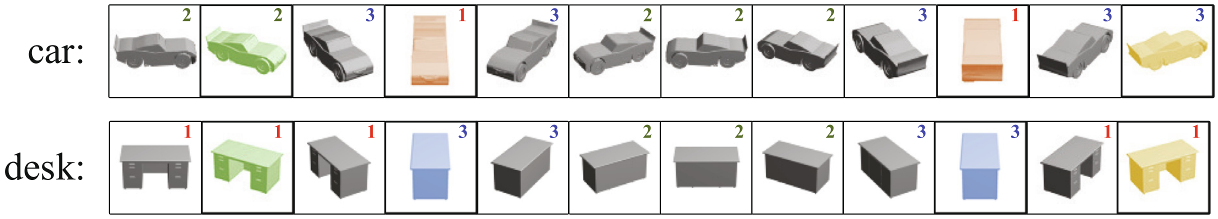

Fig. 1. Visualization of multi-view image sets of 3D objects from ModelNet40 by HMVCM. The views with identical numbers are visually similar to each other, and consequently, they can be grouped into the same category.

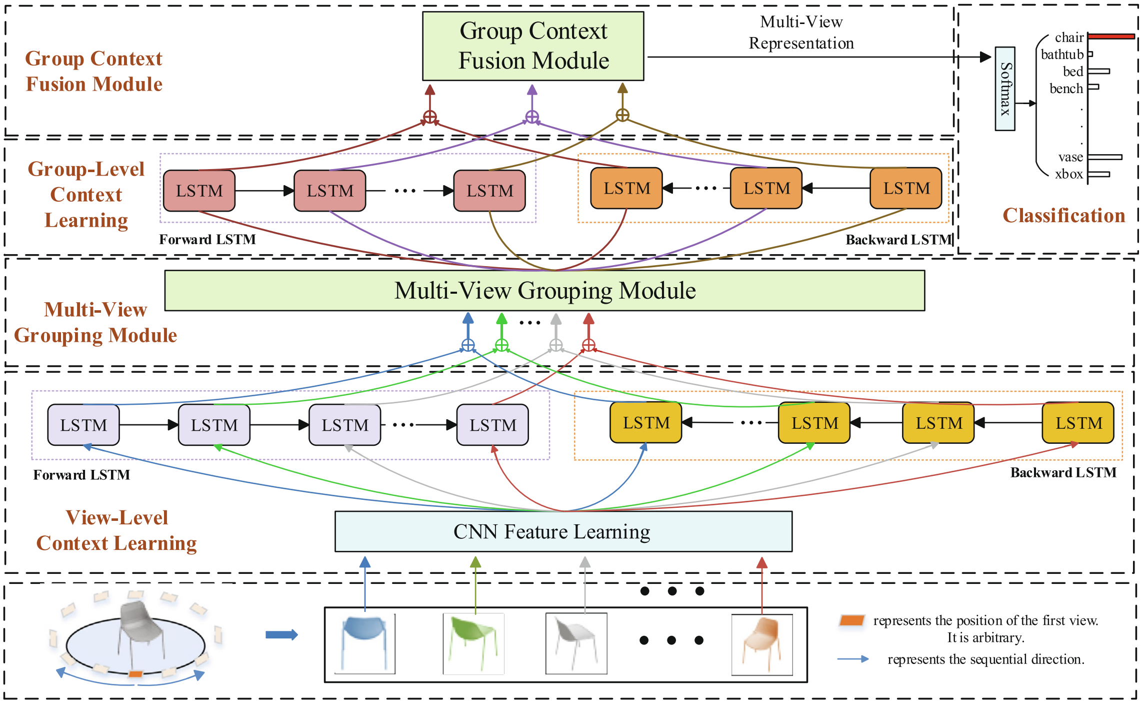
Group Context Fusion Module
Group Context Fusion Module
Multi-View Representation
Softmax
chair
bathtub
bed
bench
vase
xbox
Classification
Group-Level Context Learning
LSTM
Forward LSTM
Backward LSTM
Multi-View Grouping Module
Multi-View Grouping Module
View-Level Context Learning
CNN Feature Learning
represents the position of the first view. It is arbitrary.
represents the sequential direction.

Fig. 2. The hierarchical multi-view context modelling method for 3D object classification. For 3D object retrieval, the multi-view representation can be directly used to measure the similarity between pair-wise 3D objects.

erated 3D object descriptor can convey both the visual appearance of one 3D object and the visual appearance transition when changing viewpoints or groups.

1.2. Contributions

The main contributions of this paper are as follows.

- Different from traditional view-based methods, this method can hierarchically explore the multi-view context by modelling visual transition when changing viewpoints or view groups for the discriminative representation of 3D objects.
- Different from the current max-pooling fusion strategy, our method can take advantage of view-wise discrimination and multi-view correlations by designing the group-level context learning module. Furthermore, we propose the group context fusion module to adaptively compute the weights of group-wise context features and fuse them into a compact 3D descriptor.
- We conduct extensive experiments on three popular 3D object datasets, ModelNet10, ModelNet40 and ShapeNetCore55. The comparison experiments show the superiority of this method.

The rest of this paper is organized as follows. We introduce the related works in Section 2. Section 3 details the proposed method. Section 4 introduces the experimental settings. Section 5 presents the experimental results. Finally, we conclude this paper in Section 6.

2. Related works

The state-of-the-art methods can be divided into two main categories: view-based methods and model-based methods. We discuss the representative methods in this section.

2.1. Model-based methods

Model-based methods aim to learn 3D descriptors directly from 3D data, such as polygon meshes, voxels and point clouds. Brock et al. proposed 3D ShapeNets [11] to learn global features from voxelized 3D objects based on a convolutional restricted Boltzmann machine. Wu et al. [12] proposed a 3D generative adversarial network (3D-GAN). This method generated a 3D object descriptor from a probabilistic space by combining volumetric convolutional networks and generative adversarial nets. Qi et al. [13] proposed PointNet, which can directly take unordered point sets as input and output the object category. Qi et al. [14] further proposed PointNet++. This method employed the metric space to exploit local structures based on PointNet. Li et al. [15] presented SO-Net, which built a self-organizing map (SOM) to model the spatial distribution of point clouds. Then, SO-Net described the 3D object by performing feature extraction on SOM nodes and individual points. In addition, sparse representation learning [16–19] is often used for model-based methods. For example, Wan et al. [20] proposed integrating local shape descriptors into a global shape descriptor based on sparse representation. Although model-based methods can achieve good performance for 3D classification and retrieval tasks, the expensive computational costs restrict its application in real scenarios.

2.2. View-based methods

Compared with model-based methods that implicitly deal with 3D information, view-based methods describe 3D objects using a group of multiple views [5,21,22]. Chen et al. [23] described the 3D object based on the orthogonal projections. This method encoded one hundred orthogonal projections by both Zernike moments and Fourier descriptors as features. Su et al. [9] proposed a multi-view CNN (MVCNN). This method employed a full stride channel-wise max-pooling operation to generate the object descriptor from the visual view features. To increase the performance of channel-wise max-pooling operations, Wang et al. [24] clustered the views captured from each object and then pooled the view features in each cluster. Liu et al. [25] proposed a multi-view latent variable model (MVLVM). This method consisted of an undirected graph model with latent variables, which automatically exploited the correlative information among multiple views in both feature and spatial domains. Feng et al. [26] proposed a group-view convolutional neural (GVCNN) architecture. This method contained a hierarchical view-group-shape architecture of content descriptions to discover the view content relationship and view discrimination. Sarkar et al. [27] designed a novel multilayered height-map (MLH) representation for 3D objects. In addition, this method also provided an efficient method for fusing information from different views of 3D objects. Han et al. [28] proposed sequential views to sequential labels (SeqViews2SeqLabels). This method contained an encoder-decoder structure based on long short-term memory (LSTM) with an attention mechanism, which learned 3D global features by simultaneously fusing the content and spatial information among multiple views. Zhang et al. [29] proposed an inductive multi-hypergraph learning approach for 3D object classification. This method made full use of the higher-order correlation among the training set. It is noted that the individual views for each 3D shape contribute differently to the 3D object description. However, these existing view-based methods conducted pooling strategies on all views equally, ignoring the discriminative information contained in the view sequence. In addition, they do not consider the visual transition among adjacent viewpoints, which is crucial for view context information discovery. In this paper, the proposed HMVCM framework addresses these issues by its hierarchical multi-view context modelling structure.

3. Proposed method

The architecture of the HMVCM is illustrated in Fig. 2. It mainly contains four consecutive steps: View-level context learning, multi-view grouping module, group-level context learning and group fusion module. We will detail them as follows.

3.1. View-level context learning

Given one 3D object, stored as polygon meshes, we can generate N view images by the Phong reflection mode [30]. We set the virtual camera array based on [9]. Moreover, we consider more choices of the view numbers to validate the proposed method. The rendered view images, $\boldsymbol{V} = \{\boldsymbol{v_1}, \boldsymbol{v_2} \ldots, \boldsymbol{v_N}\}$, are passed through 2D CNNs to generate the low-level visual features, $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2} \ldots, \boldsymbol{x_N}\}$.

To imitate the human need to look back and forth to identify and compare the discriminative parts of individual 3D objects, we design the Bi-LSTM network to integrate bidirectional visual transition for view-level context discovery. Bi-LSTM consists of two LSTM networks, the forward LSTM and the backward LSTM. Each LSTM is employed to model the visual transition among adjacent viewpoints in a specific direction. The output hidden states of the LSTM of the t^{th} viewpoint can be considered as the view-level context feature since it conveys the visual transition among the view images ahead of this time point. Considering that there exist two directions for capturing views, the hidden states of both LSTMs can be concatenated to represent the view-wise context feature with respect to each time point. Specifically, when the visual feature $\boldsymbol{x_t}$ in $\boldsymbol{X}$ is fed into the LSTM, the memory units and gates can be updated by:

$$\boldsymbol{i_t} = \sigma(\boldsymbol{W_i x_t} + \boldsymbol{U_i h_{t-1}} + \boldsymbol{b_i}) \tag{1}$$

$$\boldsymbol{o_t} = \sigma(\boldsymbol{W_o x_t} + \boldsymbol{U_o h_{t-1}} + \boldsymbol{b_o}) \tag{2}$$

$$\boldsymbol{f_t} = \sigma(\boldsymbol{W_f x_t} + \boldsymbol{U_f h_{t-1}} + \boldsymbol{b_f}) \tag{3}$$

$$\tilde{\boldsymbol{c_t}} = \tanh(\boldsymbol{W_c x_t} + \boldsymbol{U_c h_{t-1}} + \boldsymbol{b_c}) \tag{4}$$

$$\boldsymbol{c_t} = \boldsymbol{i_t} \odot \tilde{\boldsymbol{c}}_{\boldsymbol{t}} + \boldsymbol{f_t} \odot \boldsymbol{c_{t-1}} \tag{5}$$

where $\boldsymbol{i_t}, \boldsymbol{f_t}, \boldsymbol{o_t}, \boldsymbol{c_t},\ and\ \tilde{\boldsymbol{c}}_{\boldsymbol{t}}$ are the input gate, the forget gate, the output gate, the old cell state and the new cell state, respectively; $\odot$ represents the wise-element product; LSTM finally outputs the most relative parts in the memory cell $\boldsymbol{c_t}$ as the hidden states, $\boldsymbol{h_t} = \boldsymbol{o_t} \odot tanh(\boldsymbol{c_t})$. For the N-view images, we can obtain the hidden states $\{\boldsymbol{h_1}, \boldsymbol{h_2} \ldots, \boldsymbol{h_N}\}$ with respect to each view point.

3.2. Multi-view grouping module

Obviously, the learned view-level context features cannot equally contribute to the visual representation of one 3D object. However, it is difficult to assign weights to individuals based on human knowledge. Motivated by the popular attention mechanism [31], which simply utilizes the visual features of individual images to compute the corresponding attention for video clip representation, we designed the multi-view grouping module. Specifically, we first feed the view-level context feature $\boldsymbol{h_t}$ through a fully connected layer to obtain the discrimination weight α_t of each view by

$$\alpha_t = \mathrm{sigmoid}(\boldsymbol{W}_\alpha \boldsymbol{h_t} + \boldsymbol{b}_\alpha) \tag{6}$$

Then, we employ the K-means algorithm to classify α_t into K clusters. Based on previous work [3] and our experimental analysis, we notice that views with similar visual appearances are usually attached similar weights. Therefore, we group the views into K clusters and then generate the group-level feature, which can reduce the redundant information caused by similar views. Specifically, the multi-view sequence can be divided into K groups $M = \{M_j\}_{j=1}^{K}, 1 \leqslant K \leqslant N$. For the views in the same group, we employ the weighted sum strategy to fuse the views to obtain the raw group-level representations. The *jth* group-level descriptor $\boldsymbol{G_j}$ can be defined as

$$\boldsymbol{G_j} = \frac{\sum_{i=1}^{N} \lambda_i \alpha_i \boldsymbol{h_i}}{\sum_{i=1}^{N} \lambda_i \alpha_i} Y \lambda_i = \{ 1 \boldsymbol{h_i} \in M_j 0 \boldsymbol{h_i} \notin M_j \tag{7}$$

3.3. Group-level context learning

Directly fusing the raw group-level features for 3D object representation is not reasonable enough since view-level context learning focuses on discovering the correlation among multiple views, while the visual discrimination of diverse view groups has not been explored. To solve this problem, we employ the Bi-LSTM network to integrate bidirectional visual transition for group-level context discovery. Specifically, when we feed the *tth* raw group-level features $\boldsymbol{G_t}$ into the Bi-LSTM, the hidden state $\tilde{\boldsymbol{h}}_{\boldsymbol{t}}$ can be formulated as:

$$\tilde{\boldsymbol{h}}_{\boldsymbol{t}} = \mathrm{Bi} - \mathrm{LSTM}(\tilde{\boldsymbol{h}}_{\boldsymbol{t-1}}, \boldsymbol{G_t}) \tag{8}$$

For the K groups, we can obtain the hidden states $\left\{\tilde{\boldsymbol{h}}_{\boldsymbol{1}}, \tilde{\boldsymbol{h}}_{\boldsymbol{2}} \ldots, \tilde{\boldsymbol{h}}_{\boldsymbol{K}}\right\}$. The *tth* hidden state $\tilde{\boldsymbol{h}}_{\boldsymbol{t}}$ can be regarded as the group-wise context feature of the *tth* view group since it can not only encode the information of the current group but also exploit neighbour correlation.

3.4. Group context fusion module

The visual difference between the views from different groups is relatively significant. Obviously, a simple mean/max-pooling strategy across different group-wise context features cannot encode all relevant information. To solve this problem, we propose adaptively aggregate group-wise context features into a compact 3D object descriptor according to their significance. Specifically, we feed the group-wise context feature $\tilde{\boldsymbol{h}}_{\boldsymbol{t}}$ through a fully connected layer to obtain $\boldsymbol{u_t}$ as a latent representation of $\tilde{\boldsymbol{h}}_{\boldsymbol{t}}$ by

$$\boldsymbol{u_t} = \tanh(\boldsymbol{W_u} \tilde{\boldsymbol{h}}_{\boldsymbol{t}} + \boldsymbol{b_u}) \tag{9}$$

Then, we select the representative group-wise context features by evaluating the correlation of $\boldsymbol{u_t}$ with the global multi-view context vector $\boldsymbol{u}$. $\boldsymbol{u}$ can be randomly initialized and jointly learned during the training procedure. The normalized weight, $\boldsymbol{w_t}$, can be formulated as:

$$\boldsymbol{w_t} = \frac{\exp(\boldsymbol{u_t}^T \boldsymbol{u})}{\sum_t \exp(\boldsymbol{u_t}^T \boldsymbol{u})} \tag{10}$$

Finally, the 3D object descriptors can be formulated as the weighted sum of group features by

$$\boldsymbol{s} = \sum_t \boldsymbol{w_t} \tilde{\boldsymbol{h}}_{\boldsymbol{t}} \tag{11}$$

$\boldsymbol{s}$ can be used as the multi-view representation of one 3D object, which can be directly used for retrieval based on the Euclidean distance of pair-wise 3D objects or as the input of the softmax function for classification.

$$\boldsymbol{p} = \mathrm{softmax}(\boldsymbol{W_s s} + \boldsymbol{b_c}) \tag{12}$$

where $\boldsymbol{p} \in \mathbb{R}^d$ is a probability vector; the dimension of d is equal to the category number.

4. Experimental settings

4.1. Dataset

We quantitatively evaluated HMVCM on three popular datasets, ModelNet10 [32], ModelNet40 [32] and ShapeNetCore55 [33]. We briefly introduce them as follows:

- **ModelNet10**: ModelNet10 consists of 4,899 3D objects in 10 categories. Specifically, 3,991 and 908 objects are used as the training set and the test set, respectively.
- **ModelNet40**: ModelNet40 contains 12,311 3D objects in 40 categories, which covers the 10 categories of ModelNet10. The entire dataset is divided into two parts: 9,843 3D objects serve as the training set, while the other 2,468 3D objects are used for testing.
- **ShapeNetCore55**: ShapeNetCore55 has indexed 51,300 3D objects from 55 common categories. The dataset is divided into the training/validation/test set by $70\%, 10\%, 20\%$ as [33]. In our experiment, the orientations of all 3D objects are aligned.

4.2. Evaluation criteria

In our experiments, we employed the classification accuracy to evaluate the 3D object classification. For 3D object retrieval, each 3D object from the test dataset is selected as a query, and the remaining 3D objects serve as the gallery dataset. We employed the most common retrieval criteria to evaluate the 3D object retrieval performance, including nearest neighbour (NN), first tier (FT), second tier (ST), F-measure (F), discounted cumulative gain (DCG), average normalized modified retrieval rank (ANMRR), mean average precision (mAP) and precise recall curve (PR-Curve) as [5]. To evaluate the performance of HMVCM on ShapeNetCore55, we directly adopt the evaluation code provided on the official website[1].

4.3. Implementation details

We applied AlexNet as the backbone CNN architecture. The last full-connection layer, fc7, was used to represent the low-level view feature. For model training, we fine-tuned the weights of AlexNet, pre-trained on ImageNet. This method was trained with the fixed learning rate (0.0001) in an end-to-end manner by SGD. For the three datasets, the mini-batch size was set to 16. In addition, we employed two strategies to avoid overfitting: 1) the weight decay strategy was adopted in the CNN weight, and 2) the dropout layer with the fixed-rate (0.5) was applied to Bi-LSTM. The dimension of the hidden state was empirically set to 1024. The view number, N, was set to 12. The group number, K, was set to 3. Section 5.2 further explains the influence of the view number and group number. For the efficiency of 3D object retrieval, each 3D object can be selected once for query. We evaluated the proposed method on the ShapeNetCore55 test, which has 10,366 3D objects. The average retrieval speed is approximately 23.6 ms per 3D object. The relevant configurations are detailed as follows: 1) Platform: MATLAB 2019a, 2) CPU: Intel(R) Core(TM) i5-8400 CPU @2.80 GHz 2.81 GHz; 3) GPU: GTX1080Ti, and 4). RAM: 32.0 GB.

5. Experimental results

5.1. Comparison with the state-of-the-art

We compared HMVCM with the representative methods for 3D object classification and retrieval. As shown in Tables 1 and 2, it is obvious that HMVCM can achieve the best classification and retrieval performances on ModelNet10 and ModelNet40 and compete with the representative methods on ShapeNetCore55. On ModelNet10, HMVCM can outperform the state-of-the-art methods with gains of 1.03–38.67% and 2.20–178.68% on the classification accuracy and mAP, respectively. On ModelNet40, HMVCM can outperform the state-of-the-art methods with gains of 0.95–14.61% and 1.95–37.48% on the classification accuracy and mAP, respectively. On ShapeNetCore55, HMVCM outperformed the others with gains of 6.35–327.60% on the microALL mAP and 17.32–194.83% on the macroALL mAP.

We make the following key observations.

[1] https://shapenet.cs.stanford.edu/shrec17/.

Table 1
Performance comparisons of 3D object classification and retrieval on ModelNet10 and ModelNet40.

Method	Train Configuration		3D Data Format	ModelNet40		ModelNet10	
	Pre-train	Fine tune		Accuracy	mAP	Accuracy	mAP
(1) SPH [34]	–	–	–	68.2%	33.3%	–	–
(2) LFD [23]	–	–	–	75.5%	40.9%	–	–
(3) 3D ShapeNets [32]	ModelNet40/10	ModelNet40/10	Volumetric	77.3%	49.2%	83.5%	68.3%
(4) VoxNet [35]	ModelNet40/10	ModelNet40/10	Volumetric	83.0%	–	92.0%	–
(5) MVCNN-MultiRes [10]	–	ModelNet40/10	Volumetric	91.4%	–	–	–
(6) LightNet [36]	–	ModelNet40/10	Volumetric	88.9%	–	93.9%	
(7) LP-3DCNN [37]	–	ModelNet40/10	Volumetric	92.1%	–	94.4%	
(8) GIFT [38]	–	ModelNet40/10	64 Views	83.1%	81.9%	92.4%	91.1%
(9) MVCNN [9], metric, 80×	ImageNet1K	ModelNet40/10	80 Views	90.1%	79.5%	–	–
(10) 3D2SeqViews [2], 12×	ImageNet1K	ModelNet40/10	12 Views	93.4%	90.8%	94.7%	92.1%
(11) SeqViews2SeqLabels [28], 12×	ImageNet1K	ModelNet40/10	12 Views	93.4%	89.1%	94.8 %	91.4 %
(12) MLVCNN [39]	ImageNet1K	ModelNet40/10	36 Views	94.1%	92.8%		
(13) PointNet [13]	–	ModelNet40/10	Point Cloud	89.2%	–	–	–
(14) PointNet++ [14]	–	ModelNet40/10	Point Cloud	90.7%	–	–	–
(15) KD-Network [40]	–	ModelNet40/10	Point Cloud	91.8%	–	–	–
(16) PointCNN [41]	–	ModelNet40/10	Point Cloud	91.8%	–	–	–
(17) DGCNN [42]	–	ModelNet40/10	Point Cloud	93.6%	–	–	–
(18) RS-CNN [43]	–	ModelNet40/10	Point Cloud	93.6%	–	–	–
(19) HMVCM	ImageNet1K	ModelNet40/10	12 Views	**94.57%**	**92.8%**	**95.70%**	**93.9%**

Table 2
Performance comparison on ShapeNet55Core. (The performances of the other methods were cited from [44]).

Method	microALL					macroALL				
	P@N	R@N	F1@N	mAP	NDCG	P@N	R@N	F1@N	mAP	NDCG
Kanezaki_RotationNet	81.0%	80.1%	**79.8%**	77.2%	86.5%	60.2%	63.9%	**59.0%**	58.3%	65.6%
Zhou_Improved_GIFT	78.6%	77.3%	76.7%	72.2%	82.7%	59.2%	65.4%	58.1%	57.5%	65.7%
Tatsuma_ReVGG	76.5%	80.3%	77.2%	74.9%	82.8%	51.8%	60.1%	51.9%	49.6%	55.9%
Furuya_DLAN	**81.8%**	68.9%	71.2%	66.3%	76.2%	**61.8%**	53.3%	50.5%	47.7%	56.3%
Thermos_MVFusionNet	74.3%	67.7%	69.2%	62.2%	73.2%	52.3%	49.4%	48.4%	41.8%	50.2%
Deng_CM-VGG5-6DB	41.8%	71.7%	47.9%	54.0%	65.4%	12.2%	**66.7%**	16.6%	33.9%	40.4%
Li_ZFDR	53.5%	25.6%	28.2%	19.9%	33.0%	21.9%	40.9%	19.7%	25.5%	37.7%
DMk_DeepVoxNet	79.3%	21.1%	25.3%	19.2%	27.7%	59.8%	28.3%	25.8%	23.2%	33.7%
SHREC16-Bai_GIFT	70.6%	69.5%	68.9%	64.0%	76.5%	44.4%	53.1%	45.4%	44.7%	54.8%
SHREC16-Su_MVCNN	77.0%	77.0%	76.4%	73.5%	81.5%	57.1%	62.5%	57.5%	56.6%	64.0%
HMVCM	75.3%	**85.1%**	74.6%	**82.1%**	**89.1%**	60.1%	**66.7%**	51.6%	**68.4%**	**78.1%**

- HMVCM can outperform the representative view-based methods. Fig. 3 presents some 3D shape retrieval samples based on HMVCM and MVCNN. We can notice that HMVCM can achieve better performance than MVCNN. Most view-based methods (e.g., MVCNN [9], MVCNN-MultiRes [10]) fuse multi-view visual features by designing deep neural networks with the maximum operation across multiple views. Theoretically, the maximum operation can maintain multi-view local saliency for visual representation. However, it loses the multi-view context, which is more discriminative for 3D object representation, as shown in Fig. 1. Comparatively, HMVCM can learn a bidirectional multi-view context among adjacent views according to their visual content information and adjacent groups according to their significance for 3D object representation. Therefore, HMVCM can outperform the others as expected.
- HMVCM can achieve better performances than the representative point-based methods. The point-based methods usually focus on local salient geometric features while ignoring the global correlation of different points. Moreover, it is difficult to deal with the disorder of points for feature computing. Comparatively, HMVCM can leverage both local visual appearance and global visual transition when changing viewpoints or groups. The group context fusion module can automatically compute the importance of multiple groups for fusion to infer the category of each 3D object.
- HMVCM is superior to the representative model-based methods. Model-based methods usually apply 3D convolutional neural networks on voxelized 3D objects. However, data sparsity caused by voxelized 3D objects cannot support the discovery of local and global structural features. In contrast, HMVCM can fully exploit the multi-view context for discriminative visual representation.

In addition, we can notice that HMVCM can achieve relatively better performance than the most recent methods, including LP-3DCNN [37], RS-CNN [43] and MLVCNN [39]. LP-3DCNN [37] introduces a rectified local phase volume block in 3D CNN to improve the feature learning capabilities ability on voxel data. However, the expensive computation cost of 3D

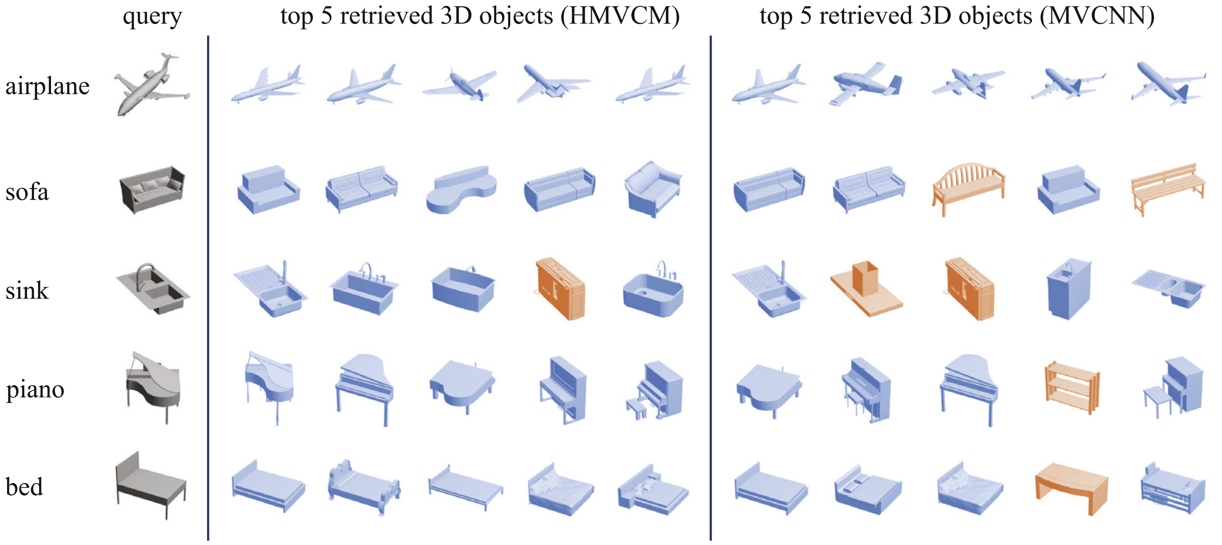
query
top 5 retrieved 3D objects (HMVCM)
top 5 retrieved 3D objects (MVCNN)
airplane
sofa
sink
piano
bed

Fig. 3. 3D shape retrieval examples based on HMVCM and MVCNN. The top 5 matches are shown for each query, with mistakes highlighted in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

CNN restricts its application in real scenarios. RS-CNN [43] can extend regular grid CNN to irregular configurations for point cloud analysis. This work is more suitable for environmental analysis for autonomous vehicles but not for 3D object classification and retrieval because of the unavailability of the point cloud for users. Comparatively, the backbone architecture in HMVCM is 2D CNN with less computation cost, and the 2D image can be easily captured by cellphones or cameras, which further boosts the demand of view-based methods. MLVCNN [39] proposes a multiloop-view 3D shape representation learning method, which also takes the view context information into consideration. However, it requires more than 24 images (3 loops) to obtain 94.1% classification accuracy. Comparatively, HMVCM requires 12 views to obtain competing performance and generalize well in camera-free settings. Specifically, the proposed method is free of specific camera constraints with arbitrary view numbers and view orders, which are more flexible for real applications. The experimental results in Section 5.2 and Section 5.4 can further validate it.

Furthermore, we also implemented the proposed method with VGG16 [45] and ResNet50 [16] on both ModelNet10 and ModelNet40. Table 3 presents the classification performance. We can notice that HMVCM with VGG16 and ResNet50 can achieve better performance due to the deeper network, but the improvement is insignificant. This comparison can further validate that the proposed hierarchical multi-view context modelling framework contributes more to these two tasks.

5.2. Analysis of the view/group number

Since the view number and the group number of individual 3D objects can have a direct influence on the classification and retrieval performances, we conducted comparison experiments for optimal view/group number selection. Specifically, we set the virtual camera array with the interval of the angle θ around the z-axis. θ was set to $\{90^\circ, 45^\circ, 30^\circ, 22.5^\circ\}$, which can independently yield $\{4, 8, 12, 16\}$ views for each 3D object. The group number K was set to $\{1, 3, 6, 9, 12\}$.

Table 4 shows the classification and retrieval performances on two datasets. Before arriving at the optimal view number (12), the performance can be monotonously augmented by increasing the view number. It is quite understandable that it can convey a more discriminative multi-view context by increasing the view number. After reaching the peak performance, the performance will drop slightly by increasing the view number since redundant or even noise information might exist, which will have a negative influence on model learning. For the group number selection, we fix the view number (12) and then explore the effect caused by the different group numbers. As shown in Table 4, HMVCM with 12 views divided into 3 groups can achieve the best performances since the visual appearance of one 3D object from the same rendering direction has similar characteristics, and the directions can mainly be divided into three categories, including the side direction, the front

Table 3
Comparison on ModelNet10 and ModelNet40 with different backbone CNNs.

	HMVCM (AlexNet)	HMVCM (VGG16)	HMVCM (ResNet50)
ModelNet10	95.70%	96.12%	**96.21%**
ModelNet40	94.57%	95.30%	**95.55%**

Table 4
Performances with respect to the number of views/groups on ModelNet10 and ModelNet40.

View/Group Number	ModelNet10							ModelNet40						
	NN	FT	ST	DCG	F	ANMRR	ACC	NN	FT	ST	DCG	F	ANMRR	ACC
4/3	92.2	86.4	95.9	88.3	32.6	9.8	90.3	90.6	83.9	91.9	86.5	33.0	13.6	90.0
8/3	93.5	87.2	96.1	89.7	32.9	8.6	92.0	92.1	85.1	92.3	87.9	33.1	10.4	91.9
12/3	**97.9**	**91.0**	**98.6**	**93.3**	**34.1**	**7.1**	**95.7**	**97.1**	**91.1**	**96.9**	**93.8**	**36.0**	**6.6**	**94.6**
16/3	96.8	90.2	98.2	92.7	33.9	7.4	94.8	96.1	90.4	96.2	93.0	35.4	7.3	94.2
12/1	94.1	87.7	96.3	90.1	33.0	8.2	93.6	94.2	86.9	93.9	89.7	33.8	9.5	92.6
12/6	96.3	89.3	97.1	91.3	33.6	7.5	94.3	95.3	88.6	95.1	91.6	34.2	8.0	93.7
12/9	95.3	89.5	97.7	92.4	33.8	7.5	94.4	95.3	89.5	95.2	91.8	34.8	7.9	94.0
12/12	94.8	89.3	96.8	91.1	33.2	7.7	93.8	94.7	87.1	94.2	90.3	34.2	8.6	93.7

direction and the back direction. Therefore, in our experiments, the optimal view/group number was set to 12 and 3, respectively.

5.3. Analysis of the architecture of HMVCM

We conducted comparison experiments to validate the architecture of HMVCM for both view-level and group-level context learning. We compared four architectures: 1). MVCNN: MVCNN only aggregates multiple views by view pooling without considering the multi-view context. 2). Forward LSTM(F-LSTM)+G: The architecture of both view-level/group-level context learning only consists of the forward LSTM, as shown in Fig. 2. F-LSTM can only explore the visual transition between the current viewpoint/group and the previous viewpoints/groups in sequential views. '+G' means that we aggregate the group-wise context features according to their significance (Section 3.4). 3). Backward LSTM(B-LSTM)+G: The architecture of view-level/group-level context learning only consists of the backward LSTM, as shown in Fig. 2. B-LSTM can only explore the visual transition between the current viewpoint/group and the latter viewpoints/groups in the sequential views/groups. 4). HMVCM: This architecture integrates both F-LSTM and B-LSTM to take advantage of the bidirectional context for modelling.

Tables 5 and 6 show the classification and retrieval performances with respect to different settings. HMVCM can consistently outperform the others on both datasets since it can fully exploit multi-view context and further adaptively fuse them for 3D object representation. Comparatively, both F-LSTM+G and B-LSTM+G are competing and work worse than HMVCM since each of them loses one kind of complementary visual context. As expected, MVCNN works worse since it loses the bidirectional multi-view context, which is discriminative for 3D representation.

5.4. Analysis on view order

It is intuitive that the order of view capturing has a direct influence on sequential model learning. To validate whether HMVCM is constrained by the specific view order, we randomized the view order of the multi-view image sets of individual 3D objects 100 times for testing. We compared HMVCM with the common multi-view representation method with LSTM under different architectures. Since the variations of F-LSTM and B-LSTM are competing, we only selected F-LSTM for evaluation. We compared the following settings: 1. F-LSTM+L/Bi-LSTM+L: Since the last hidden state can encode the previous information, we directly utilized the last hidden state $\tilde{h}_K$ in the F-LSTM and Bi-LSTM architecture for 3D object representation; 2. F-LSTM+M/Bi-LSTM+M: Different from '+L', we directly computed the mean of all hidden states $\{\tilde{h}_i\}_{i=1}^{K}$ in the F-LSTM and Bi-LSTM architecture for 3D object representation; 3. F-LSTM+G/HMVCM: Different from '+L' and '+M', we leverage the

Table 5
Comparison on ModelNet10 with different view orders.

Method	Order							Random						
	NN	FT	ST	DCG	F	ANMRR	ACC	NN	FT	ST	DCG	F	ANMRR	ACC
MVCNN	90.6	78.9	88.6	82.2	31.3	18.5	91.7	90.6	78.9	88.6	82.2	31.3	18.5	91.7
B-LSTM+G	91.2	84.6	95.6	86.9	32.1	13.5	92.3	91.1	84.4	95.1	86.8	32.0	13.5	92.3
F-LSTM+G	92.2	84.5	95.5	86.8	32.1	13.6	92.2	91.3	83.9	94.8	86.7	31.9	13.9	92.2
F-LSTM+L	94.1	88.1	96.3	90.1	32.9	8.2	92.4	92.2	86.4	95.8	88.3	32.6	9.8	90.8
F-LSTM+M	91.7	84.1	94.7	86.7	31.5	14.1	92.0	91.6	74.1	94.8	81.9	30.8	14.3	91.9
Bi-LSTM+L	95.0	88.7	96.7	90.5	33.0	7.8	93.1	72.5	69.4	81.6	70.4	18.3	22.4	75. 3
Bi-LSTM+M	94.7	89.1	97.2	90.3	32.9	7.7	93.7	94.4	88.8	96.8	89.7	32.1	8.0	93.7
HMVCM	**97.9**	**91.0**	**98.6**	**93.3**	**34.1**	**7.1**	**95.7**	**97.4**	**90.9**	**97.2**	**92.7**	**33.8**	**7.8**	**95.7**

Table 6
Comparison on ModelNet40 with different view orders.

Method	Order							Random						
	NN	FT	ST	DCG	F	ANMRR	ACC	NN	FT	ST	DCG	F	ANMRR	ACC
MVCNN	87.8	79.1	87.8	82.0	31.3	18.1	90.1	87.8	79.1	87.8	82.0	31.3	18.1	90.1
B-LSTM+G	91.7	82.0	91.9	85.3	33.5	15.2	92.3	91.1	81.3	90.8	84.1	32.8	14.7	92.2
F-LSTM+G	91.9	81.2	92.1	85.6	33.6	15.4	92.2	90.9	81.4	90.6	83.7	32.9	14.3	92.1
F-LSTM+L	94.0	86.7	93.7	89.5	33.6	9.3	92.1	90.6	83.9	91.9	86.5	33.0	13.6	90.4
F-LSTM+M	91.9	84.6	92.1	87.6	33.4	10.6	92.0	91.5	84.1	92.0	87.2	33.2	12.2	91. 5
Bi-LSTM+L	95.4	89.5	95.2	91.8	35.1	7.6	93.0	77.4	69.6	77.5	72.2	21.5	31.2	73. 6
Bi-LSTM+M	95.3	88.3	95.1	90.9	34.6	8.0	92.8	94.8	87.4	94.3	89.7	33.8	8.5	92.7
HMVCM	**97.1**	**91.1**	**96.9**	**93.8**	**36.0**	**6.5**	**94.6**	**96.9**	**90.7**	**96.2**	**92.8**	**35.7**	**7.0**	**94.6**

module of the group context fusion module to adaptively fuse $\{\tilde{h}_i\}_{i=1}^{K}$ in the F-LSTM and HMVCM architectures for 3D object representation.

From Tables 5 and 6, we have several observations:

- HMVCM is independent of specific view orders and can effectively aggregate group-wise context features by adaptively computing the weights of each group. Although F-LSTM+M/Bi-LSTM+M can be independent of view order by mean pooling all hidden states, they work worse than HMVCM.
- The performances vary greatly in terms of the ordered view and the random view by taking the last hidden state as the 3D object descriptor. It shows that the order of the input sequence has a great influence on the performance of F-LSTM+L/Bi-LSTM+L.
- MVCNN is independent of the view order since it only implements maximum operation among multiple views for multi-view fusion.

5.5. Analysis of hierarchical multi-view context modelling

In this section, we qualitatively analyse the hierarchical multi-view context fusion network to understand how the multi-view grouping module discovers the visual similarity and discrimination of multiple views and how the group context fusion module adaptively computes the weights of group-wise context features. As shown in Fig. 4, we visualize the view-wise context feature weights α_t and the group-wise context weights w_t of some samples from ModelNet40 (e.g., bookshelf, monitor, sofa, bag). In Fig. 4, it is intuitive that the view images in green and blue usually contain more significant appearance and structural characteristics, and consequently, they are assigned with much higher weights compared with the view images in red. In addition, views with similar visual content can be grouped together, while views belonging to different groups make a great difference in visual appearance. These results are consistent with our expectations. For the monitor sample,

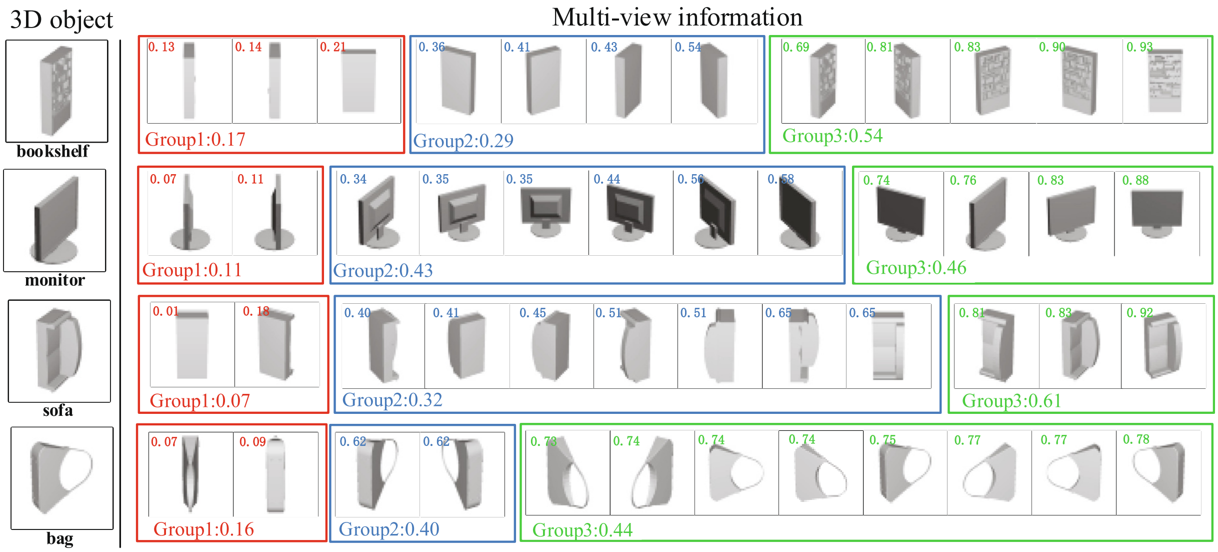

Fig. 4. Visualization of the weights of the view-wise context (top left) and group-wise context (bottom left) features on ModelNet40.

compared to the view images in red, the views in green can represent the spatial structure of the monitor and consequently have higher weights. In addition, we can also observe that the views in the same group are rendered from the adjacent directions of the 3D object. Specifically, the views in the three groups of the monitor are rendered from the side/back/front direction of the 3D object.

6. Conclusion

In this paper, we proposed a hierarchical multi-view context modelling method for 3D object classification and retrieval. In this method, we explored the multi-view context among adjacent view images and groups to discriminatively represent the visual characteristics of 3D objects. We compared HMVCM with the state-of-the-art methods and explored the influence of the view number, the group number, the view order, and multiple architecture variations. The qualitative analysis of hierarchical multi-view context modelling can intuitively illustrate how HMVCM adaptively fuses multi-view information. Extensive comparisons demonstrated the superiority of the proposed method. However, HMVCM is not necessarily suitable for real-world 3D object retrieval. In this paper, we aim to explore the multi-view context contained in the view sequence but neglect the visual characteristics in each view, which has a clear background, and the object can be easily recognized. Comparatively, real-world 3D objects have more complex backgrounds and abundant visual information. Therefore, it is necessary to explore the visual context at the view level or enhance the capability of feature representation to discover the visual characteristics in individual views. Due to the successful application of the spatial attention mechanism on Baidu's PaddlePaddle deep learning platform [46,47], we will design a spatial attention module to exploit the view-level context.

CRediT authorship contribution statement

An-An Liu: Conceptualization, Methodology. **Heyu Zhou:** Software, Writing - original draft. **Weizhi Nie:** Supervision. **Zhenguang Liu:** Writing - review & editing. **Wu Liu:** Software. **Hongtao Xie:** Validation. **Zhendong Mao:** Visualization. **Xuanya Li:** Investigation. **Dan Song:** Conceptualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (61772359, 61872267, 61902277), the grant of Tianjin New Generation Artificial Intelligence Major Program (19ZXZNGX00110, 18ZXZNGX00150), the Open Project Program of the State Key Lab of CAD & CG, Zhejiang University (Grant No. A2005, A2012), the grant of Elite Scholar Program of Tianjin University (2019XRX-0035) and the Baidu Pinecone Program.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, athttps://doi.org/10.1016/j.ins.2020.09.057.

References

[1] H. Abdul-Rashid, J. Yuan, B. Li, Y.L. et al., Extended 2d scene image-based 3d scene retrieval, in: 12th Eurographics Workshop on 3D Object Retrieval, 2019, pp. 41–48..

[2] Z. Han, H. Lu, Z. Liu, C. Vong, Y. Liu, M. Zwicker, J. Han, C.L.P. Chen, 3d2seqviews: Aggregating sequential views for 3d global feature learning by CNN with hierarchical attention aggregation, IEEE Trans. Image Process. 28 (8) (2019) 3986–3999.

[3] H. Zhou, A. an Liu, W. Nie, J. Nie, Multi-view saliency guided deep neural network for 3d object retrieval and classification, IEEE Trans. Multimedia 22 (6) (2020) 1496–1506..

[4] S. Fu, W. Liu, D. Tao, Y. Zhou, L. Nie, Hesgcn: Hessian graph convolutional networks for semi-supervised classification, Inf. Sci. 514 (2020) 484–498.

[5] A. Liu, W. Nie, Y. Gao, Y. Su, View-based 3-d model retrieval: A benchmark, IEEE Trans. Cybern. 48 (3) (2018) 916–928.

[6] R. Hong, J. Pan, S. Hao, M. Wang, F. Xue, X. Wu, Image quality assessment based on matching pursuit, Inf. Sci. 273 (2014) 196–211.

[7] Y. Feng, J. Xiao, Y. Zhuang, X. Yang, J.J. Zhang, R. Song, Exploiting temporal stability and low-rank structure for motion capture data refinement, Inf. Sci. 277 (2014) 777–793.

[8] Y. Gao, M. Wang, D. Tao, R. Ji, Q. Dai, 3-d object retrieval and recognition with hypergraph analysis, IEEE Trans. Image Process. 21 (9) (2012) 4290–4303.

[9] H. Su, S. Maji, E. Kalogerakis, E.G. Learned-Miller, Multi-view convolutional neural networks for 3d shape recognition, in: ICCV, 2015, pp. 945–953..

[10] C.R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, L.J. Guibas, Volumetric and multi-view cnns for object classification on 3d data, CVPR (2016) 5648–5656.

[11] A. Brock, T. Lim, J.M. Ritchie, N. Weston, Generative and discriminative voxel modeling with convolutional neural networks, arXiv preprint arXiv:1608.04236 (2016)..

[12] J. Wu, C. Zhang, T. Xue, B. Freeman, J. Tenenbaum, Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling, NIPS (2016) 82–90.

[13] C.R. Qi, H. Su, K. Mo, L.J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, CVPR (2017) 77–85.

[14] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: NIPS, 2017, pp. 5105–5114..

[15] J. Li, B.M. Chen, G.H. Lee, So-net: Self-organizing network for point cloud analysis, CVPR (2018) 9397–9406.
[16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CVPR (2016) 770–778.
[17] L. Rebollo-Neira, D. Whitehouse, Sparse representation of 3d images for piecewise dimensionality reduction with high quality reconstruction, Array 1–2 (2019) 100001.
[18] M. Elad, M. Aharon, Image denoising via learned dictionaries and sparse representation, CVPR (2006) 895–900.
[19] X. Li, H. Shen, L. Zhang, H. Zhang, Q. Yuan, G. Yang, Recovering quantitative remote sensing products contaminated by thick clouds and shadows using multitemporal dictionary learning, IEEE Trans. Geosci. Remote. Sens. 52 (11) (2014) 7086–7098.
[20] L. Wan, S. Li, Z. Miao, Y. Cen, Non-rigid 3d shape retrieval via sparse representation, PG (2013).
[21] F. Xue, S. Xu, C. He, M. Wang, R. Hong, Towards efficient support relation extraction from RGBD images, Inf. Sci. 320 (2015) 320–332.
[22] M. Jian, C. Cui, X. Nie, H. Zhang, L. Nie, Y. Yin, Multi-view face hallucination using SVD and a mapping model, Inf. Sci. 488 (2019) 181–189.
[23] D. Chen, X. Tian, Y. Shen, M. Ouhyoung, On visual similarity based 3d model retrieval, Comput. Graph. Forum 22 (3) (2003) 223–232.
[24] C. Wang, M. Pelillo, K. Siddiqi, Dominant set clustering and pooling for multi-view 3d object recognition, BMVC (2017).
[25] A. Liu, W. Nie, Y. Su, 3d object retrieval based on multi-view latent variable model, IEEE Trans. Circuits Syst. Video Techn. 29 (3) (2019) 868–880.
[26] Y. Feng, Z. Zhang, X. Zhao, R. Ji, Y. Gao, GVCNN: group-view convolutional neural networks for 3d shape recognition, in: CVPR, 2018, pp. 264–272..
[27] K. Sarkar, B. Hampiholi, K. Varanasi, D. Stricker, Learning 3d shapes as multi-layered height-maps using 2d convolutional networks, ECCV (2018) 74–89.
[28] Z. Han, M. Shang, Z. Liu, C. Vong, Y. Liu, M. Zwicker, J. Han, C.L.P. Chen, Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by RNN with attention, IEEE Trans. Image Process. 28 (2) (2019) 658–672.
[29] Z. Zhang, H. Lin, X. Zhao, R. Ji, Y. Gao, Inductive multi-hypergraph learning and its application on view-based 3d object classification, IEEE Trans. Image Process. 27 (12) (2018) 5957–5968.
[30] B.T. Phong, Illumination for computer generated pictures, Commun. ACM 18 (6) (1975) 311–317.
[31] L. Yao, A. Torabi, K. Cho, N. Ballas, C.J. Pal, H. Larochelle, A.C. Courville, Describing videos by exploiting temporal structure, in: ICCV, 2015, pp. 4507–4515..
[32] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: A deep representation for volumetric shapes, CVPR (2015) 1912–1920.
[33] A.X. Chang, T.A. Funkhouser, L.J. Guibas, et al., Shapenet: An information-rich 3d model repository, arXiv preprint arXiv:1512.03012 (2015)..
[34] M.M. Kazhdan, T.A. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3d shape descriptors, SGP (2003) 156–164.
[35] D. Maturana, S. Scherer, Voxnet: A 3d convolutional neural network for real-time object recognition, in: IROS, 2015, pp. 922–928..
[36] S. Zhi, Y. Liu, X. Li, Y. Guo, Toward real-time 3d object recognition: A lightweight volumetric cnn framework using multitask learning, Comput. Graphics 10 (2017)..
[37] S. Kumawat, S. Raman, LP-3DCNN: unveiling local phase in 3d convolutional neural networks, CVPR (2019) 4903–4912.
[38] S. Bai, X. Bai, Z. Zhou, Z. Zhang, L.J. Latecki, GIFT: A real-time and scalable 3d shape search engine, in: CVPR, 2016, pp. 5023–5032..
[39] J. Jiang, D. Bao, Z. Chen, X. Zhao, Y. Gao, MLVCNN: multi-loop-view convolutional neural network for 3d shape retrieval, AAAI (2019) 8513–8520.
[40] R. Klokov, V.S. Lempitsky, Escape from cells: Deep kd-networks for the recognition of 3d point cloud models, in: ICCV, 2017, pp. 863–872..
[41] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, B. Chen, Pointcnn: Convolution on x-transformed points, in: NIPS, 2018, pp. 828–838..
[42] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph CNN for learning on point clouds, ACM Trans. Graph. 38 (5) (2019) 146:1–146:12..
[43] Y. Liu, B. Fan, S. Xiang, C. Pan, Relation-shape convolutional neural network for point cloud analysis, CVPR (2019) 8895–8904.
[44] M. Savva, F. Yu, H.S. et al., Large-scale 3d shape retrieval from shapenet core55, in: 10th Eurographics Workshop on 3D Object Retrieval, 2017, pp. 76–93..
[45] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, ICLR (2015).
[46] T.W.H.W. Yanjun Ma, Yu. Dianhai, Paddlepaddle: An open-source deep learning platform from industrial practice, Front. Data Comput. 1 (1) (2019).
[47] Paddlepaddle, Paddlepaddle: An easy-to-use, easy-to-learn deep learning platform, http://www.paddlepaddle.org/..