Group-Pair Convolutional Neural Networks for Multi-View based 3D Object Retrieval

Zan Gao[1,2,*], **Deyu Wang**[1,2], **Xiangnan He**[3], **Hua Zhang**[1,2]
[1] Key Laboratory of Computer Vision and System, Tianjin University of Technology
Ministry of Education,Tianjin, 300384, China
[2]Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology,
Tianjin University of Technology, Tianjin, 300384, China
[3]School of Computing, National University of Singapore, 117417, Singapore
zangaonsh4522@gmail.com, xzero3547w@163.com, xiangnanhe@gmail.com, hzhang62@163.com

Abstract

In recent years, research interest in object retrieval has shifted from 2D towards 3D data. Despite many well-designed approaches, we point out that limitations still exist and there is tremendous room for improvement, including the heavy reliance on hand-crafted features, the separated optimization of feature extraction and object retrieval, and the lack of sufficient training samples. In this work, we address the above limitations for 3D object retrieval by developing a novel end-to-end solution named Group Pair Convolutional Neural Network (**GPCNN**). It can jointly learn the visual features from multiple views of a 3D model and optimize towards the object retrieval task. To tackle the insufficient training data issue, we innovatively employ a pair-wise learning scheme, which learns model parameters from the similarity of each sample pair, rather than the traditional way of learning from sparse label–sample matching. Extensive experiments on three public benchmarks show that our **GPCNN** solution significantly outperforms the state-of-the-art methods with 3% to 42% improvement in retrieval accuracy.

Introduction

With the rapid development of 3D model capturing tools and computing power, there are an increasing number of 3D objects in various domains (Mavar-Haramija, Prats-Galino, and Notaris 2015; Gao et al. 2017), such as computer vision, medical simulation, architectural design, computer graphics and computer-aided design. In contrast to object retrieval on 2D images, retrieving objects from 3D data is a more practical and realistic task. As such, addressing 3D object retrieval is a relevant and timely research topic and has attracted much attention in recent years (Liu et al. 2016; Leng et al. 2015; Liu et al. 2015).

Early works on 3D object retrieval are largely based on 3D models, where low-level feature-based methods (Ip et al. 2002; Osada et al. 2002; Mademlis et al. 2009) and high-level structure-based methods (LENG et al. 2009) have been employed. As these methods require the 3D models to be explicitly available, it limits the range of applications of these methods. Recently, extensive research efforts have been dedicated to view-based 3D object retrieval methods (Liu et al. 2015; Nie, Liu, and Su 2016) owing to the highly discriminative property of multiple views in representing 3D objects (Ohbuchi et al. 2008; Ohbuchi and Furuya 2009). Several visual descriptors have been proposed, including the light-field descriptors (LFDs) (Chen et al. 2003), elevation descriptors (EDs) (Shih, Lee, and Wang 2007), bag of visual features (BoVF) (Ohbuchi and Furuya 2009), and compact multi-view descriptors (CMVDs)(Ohbuchi and Furuya 2009). These view-based methods share a common advantage, that is, being invariant to articulation and global deformation of the 3D models. Along this line, many retrieval algorithms (Steinbach, Karypis, and Kumar 2000; Gao et al. 2011; Leordeanu and Hebert 2005; Cho, Lee, and Lee 2010; Gao et al. 2016) have also been developed such as, Hausdorff distance (HAUS) and Nearest Neighbor (NN) (Steinbach, Karypis, and Kumar 2000), weighted bipartite graph matching (WBGM) (Gao et al. 2011), spectral matching (SM) (Leordeanu and Hebert 2005), and reweighted random walks matching (RRWM) (Cho, Lee, and Lee 2010) select representative views from the query or candidate model, updating the matching degree of each view in an iterative way. More recently, Class-statistics matching method with pair-constraint (CSPC) (Gao et al. 2016) converts the view-based distance measure to object-based distance measure.

Despite many recent efforts, we observe that most of them rely on hand-crafted features, such as LFDs, BoVF, and CMVD. As such, these methods have a relatively low robustness; particularly, when it comes to different datasets with different 3D object properties — such as illumination conditions, scales and view variations — their performance vary greatly.

Along another line, deep learning techniques have been employed to address the 3D related tasks, such as object classification (Socher et al. 2012) and content-based PET image retrieval (Liu et al. 2014). These works show that the features extracted by Convolutional Neural Networks (CNNs) are more robust and effective, leading to better performance than hand-crafted features. However, we point out that a key limitation of these methods is that the feature extraction and model training are performed separately. As a result, there lacks necessary interactions between the feature extraction and model training — the feature extractor only extracts general features without knowing which characters are more important for the retrieval model, and the retrieval

model cannot supply any guidance for the feature extractor. To tackle this, several recent efforts have tried to design end-to-end deep architecture for the task, such as SPP-NET (He et al. 2014), 2ch-2stream (Zagoruyko and Komodakis 2015), MV-CNN (Su et al. 2016) and Siamese network (Wang, Kang, and Li 2015). While deep models are highly expressive, they meanwhile require a large number of training samples to ensure the model can learn useful patterns rather than overfitting the data. However, existing methods all employ a point-wise learning scheme, which requires a large number of 3D samples — which are difficult to obtain — for effective training.

To address the aforementioned issues, we develop a novel multi-view based 3D retrieval method named *Group Pair Convolutional Neural Network* (**GPCNN**), which unifies the strength of multi-view representation of 3D models with effective deep features learned by CNNs. To alleviate the sparsity issue of limited training samples, we employ a pair-wise learning scheme, which performs training on each pair of 3D samples by preserving their similarity. As each 3D sample is represented as a group of images captured from multiple views, we also term each sample pair as "group pair". By learning from each pair of groups, we can extend the number of training samples from $O(N)$ to $O(N^2)$ where N denotes the number of labelled 3D samples, reducing the demand for labelled 3D data significantly. Lastly, we adopt the contractive loss as the optimization objective, so as to implement the idea that 3D samples of the same category should be more similar than that of different categories. This optimization target lends support to the retrieval process directly and allows the learning of **GPCNN** in an end-to-end manner.

The main contributions of this work are summarized as follows.

- We develop a novel deep learning solution **GPCNN** for 3D object retrieval, which is an end-to-end approach that seamlessly fuses the learning of visual features from multiple views with the retrieval model.

- We propose a simple yet effective scheme to lower the demand for large number of labelled 3D samples for training deep learning models. The idea is to artificially enrich the training data by performing pair-wise learning on the similarity of samples, and use a contractive loss to support the retrieval task effectively.

- We conduct extensive experiments on three benchmarks of 3D objects, demonstrating that our proposed **GPCNN** significantly outperforms state-of-the-art 3D object retrieval methods, including CSPC (Gao et al. 2016) and Siamese CNN (Chopra, Hadsell, and Lecun 2005).

The rest of the paper is organized as follows. The related work is first given in Section **Related Work**. After formulating the problem in Section **Problem Formulation**, we elaborate our proposed **GPCNN** solution in Section **Proposed Method**. We then conduct experiments to evaluate our method in Section **Experiments**. Lastly, we conclude the paper in Section **Conclusion**.

Related Work

In recent years, more and more 3D model retrieval methods are proposed and they can roughly be divided into two categories: (1) model-based methods; (2) view-based methods, Thus, we will introduce these methods respectively. Moreover, the related deep learning methods are also discussed.

- 3D object retrieval method based on models. In fact, model-based methods are proposed in early work, where an explicit 3D model data for retrieval is required, and then a lot of visual feature representations are proposed for describing 3D model, such as, leverage geometric moments (Yang and Albregtsen 1996), Fourier descriptors (Persoon and Fu 1977), surface distributions (Lu et al. 2014) and shape descriptors (Polewski et al. 2015). These feature representations are very popular, and they are often employed in different kinds of retrieval algorithms, for example, 3D shape histogram (Ankerst et al. 1999) is proposed as an intuitive and powerful similarity model for 3D objects. Meanwhile, quadratic form distance functions to account for errors of measurement are employed to allow a particular flexibility; A novel retrieval method based on the shape feature of 3D models is proposed in (Osada et al. 2001) where the shape distribution sampled from the 3D model is constructed as the digital signature of an object and further it is utilized to compute the similarity between different models; In fact, all of the 3D patterns from the model are employed for retrieval and classification. When no model information is available, a 3D model construction procedure is required to generate the virtual model using a collection of images. However, 3D model reconstruction is computationally expensive, and its performance is highly restricted by the sampled images. Therefore, the practical applications of model-based methods are seriously limited.

- 3D object retrieval method based on views. Recently, since view-based methods are independent of 3D models, and can be realized simply with the multi-view representation of models (Nie, Liu, and Su 2016), thus, it has attracted much more attention. Even more so, it will be very easy for this approach to directly extend to the retrieval in real objects, which has promising applications in e-business and location-based mobile applications. For example, Zernike moments and Fourier descriptors (Chen et al. 2003) are firstly extracted for each view image, and then the nearest neighbor method is utilized for the similarity measure between different models; A novel feature descriptor, elevation descriptor (Shih, Lee, and Wang 2007) is proposed for 3D object representation, which is invariant to translation, rotation and scaling of 3D models; In (Ansary, Daoudi, and Vandeborre 2007), X-means is used to select representative views and then Bayesian models is applied to compute the similarity between different models; A general framework for 3D object retrieval is proposed in (Gao et al. 2012), where camera array restriction is free, and each object can be represented by a free set of views. The proposed CCFV model can be generated on the basis of the query Gaussian models by combining the positive matching model and the negative

matching model. This method can remove the constraint of static camera array settings for view capturing and can be applied to any view-based 3D object database. A novel Compact Multi-View Descriptors (CMVD) is proposed in (Ohbuchi and Furuya 2009) for 3D model representation where camera arrays are set at the 18 vertices of a 32-hedron to capture the CMVD, and these cameras are uniformly distributed. Wang et al. (Wang et al. 2016) investigate the discriminative information of each view in dataset, and then the reverse distance metric is employed. Although these algorithms can have good performance, they are hand-crafted features whose robustness is limited.

- Visual representation based on deep learning for 3D object retrieval. Visual feature representations play an important role in the object retrieval, and they can be divided into local feature representation (SIFT, SURF) and global feature representation (Zernike, HOG, HSV). However, for 3D object retrieval, there is a high requirement for more discriminative feature representations. Thus, deep learning is often employed to learn more powerful visual representation for challenging tasks (Liu et al. 2017; **?**). For example, Socher et al. (Socher et al. 2012) proposed a model based on convolutional neural networks (CNN) to learn feature for 3D object classification, and then the classic SVM was utilized to handle classification problem. Liu et al. (Liu et al. 2014) proposed accurate content-based PET images retrieval where high-level ROI features with deep learning architecture was employed. He et al.(He et al. 2014) proposed SPP-net, whose pyramid pooling was robust to object deformations, and it could generate a fixed-length representation regardless of image size/scale. Zagoruyko et al. (Zagoruyko and Komodakis 2015) discussed how to directly learn the visual representation from image data. LonchaNet (Gomez-Donoso et al. 2017) was proposed which was a deep learning architecture for point clouds classification with providing a low computation cost. Multi-view CNN (MVCNN) (Su et al. 2016) architecture was proposed for 3D shape classification, where multiple views can be simultaneously employed. Wang et al (Wang, Kang, and Li 2015) proposed Siamese network which was consisted of two chains for cross domain (sketch-view) matching. Although these existing CNNs can obtain satisfying on some tasks, the large number of labeled training samples is required. However, the training samples in 3D datasets are very small, and it will be difficult for us to train the CNN. Moreover, most of the state-of-the-art CNN architectures only have one branch, and most of them are designed for classification task. Therefore, for view based 3D object retrieval problem, the existing CNN architectures are inadequate and low performance.

Problem Formulation

In a nutshell, the 3D object retrieval problem is formulated as: given a 3D object (query), retrieving the matching or relevant 3D objects (documents), and ranking the documents according to the similarity with the query.

Typically, there are two ways to get the 3D model for an object: either 1) directly obtain the 3D model by scanning the object with professional 3D capturing equipments, or 2) indirectly reconstruct the 3D model from images of multiple views of the object. However, both ways are quite costly to achieve in practice — professional 3D capturing equipments are usually expensive and inconvenient to carry, and reconstructing 3D models from images are computationally expensive when no model information is given. Moreover, it is also difficult to extract effective features for 3D models. As such, the applications of model-based retrieval approaches are quite limited in practice.

In this work, we focus on the view-based setting, which does not require the 3D model explicitly and is more practically plausible than model-based approaches. Particularly, we represent a 3D object (both query and gallery) as a group of images captured from different views, performing the retrieval task based on the multi-view representation of 3D objects. Note that in case of the 3D model is provided, we can easily get its multi-view representation by simulating virtual cameras to take pictures from different viewpoints of the object.

Proposed Method

The key to the retrieval task lies in measuring the similarity between two 3D objects. Considering that an object is represented as a group of images, an intuitive solution is to learn the features (aka. representation) for an image group, and then estimating the similarity with some statistical measures like cosine similarity. Nevertheless, such an intuitive solution has several flaws, making the retrieval performance unsatisfactory. First, it is unclear how to generate the representation for a group of images from the representation of each individual image; simply a pooling operation like average/max pooling will lose many useful information and cannot fully exploit the complementary information of different views. Second, the separated steps of feature learning and similarity measuring lack necessary interaction between the feature extractor and the retrieval model; as a result, the extractor has no information about which regions are more important for the retrieval task, making the results suboptimal.

Instead of measuring the image group similarity with statistical measures on extracted features, we unify the two steps and learn the similarity of two groups from data. The basic idea is that if two groups are in the same category (or assigned with the same label), they should express some visual similarity to a certain extent and have similar representation in the latent space. To implement this idea, we design a model to directly estimate the similarity of two image groups, using the labelling information to guide the learning of model parameters.

The GPCNN Solution

Figure 1 illustrates our **GPCNN** solution, which consists of three main components: the input layer of group-pair generation, the CNN-based deep architecture for feature learning, and contractive loss for model optimization and similarity

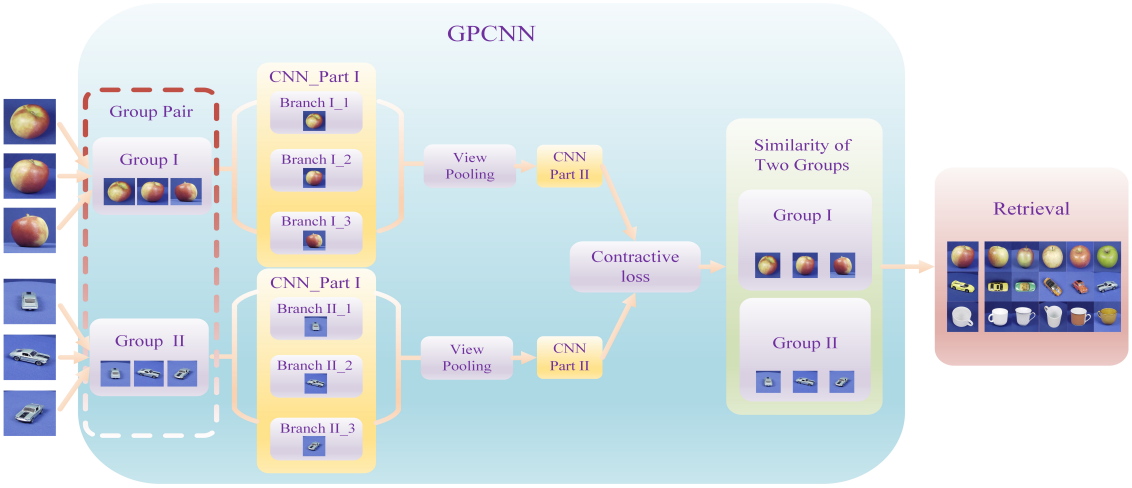


Figure 1: The architecture of our proposed **GPCNN** for 3D object retrieval. The apple object comes from query set, and the car is from the gallery set. Moreover, multiple samples are chosen from the query set and the gallery set respectively. These samples are then separately fed into the branch of CNN for obtaining feature maps. Furthermore, the view pooling scheme is employed to choose high-response feature maps from multi-view samples, and then CNN Part II is used to obtain effective descriptors. Finally, the value of the contractive loss between two groups is evaluated, which can also be employed as the similarity metric.

measurement. In what follows, we elaborate the components one by one.

Group-pair Generation Each training example is a pair of 3D objects, i.e., group of images in our multi-view setting. As the model learning is usually done in a stochastic manner, such as by using Stochastic Gradient Descent (SGD), we first randomly sample a 3D object from the labelled training data. We then pair the object with another randomly-sampled object of the same (different) category to form a positive (negative) training instance. For a positive sample, it is assigned with a target of value 1, and for a negative sample, its target value is 0.

In existing public 3D benchmarks (Liu et al. 2016; Chen et al. 2003; Ess et al. 2008), the number of labelled 3D samples is usually small for each category. As such, the conventional point-wise learning strategy, as used in AlexNet, will suffer from the small data issue and easily be overfitting. Although the situation can be alleviated by performing fine-tuning on pre-trained models, we find the training can still be inadequate, evidenced by the relatively low performance. By expanding the sparse object–label relations to dense object–object data, we can increase the number of training samples by a magnitude. Moreover, it enables the direct estimation of the similarity of two 3D objects.

CNN-based Deep Architecture The core component of **GPCNN** has three main parts:

- 1) CNN-Part I Conventional CNNs like VGG-16 or AlexNet receives a single view (image) for training, which can not explicitly leverage the commonalities of multiple views in a class. In the field of 3D retrieval, the Siamese network receives a view pair as the input for training, which still fails to capture the potential relationship among multiple views. Here, our **GPCNN** receives two groups as the input, where each group contains multiple views. For each view, we utilize a CNN for feature extraction; To allow the learning of common patterns among multiple views, we enforce the CNNs share the same model parameters. More details about the CNN structure and settings can be found in Section Implementation Details.
- 2) View Pooling After extracting the feature maps for each view, we now consider how to model the interaction among multiple views of an object. The view pooling layer aims to implement the locally optimal screening of multiple views. Specifically, it performs element-wise operation on each feature map of a view, and then obtains locally high-response feature maps of each view. Finally, we convert the feature maps of multiple views into high-response feature maps of one view, so that it can implement views screening and initial association among views.
- 3) CNN-Part II The high-response feature maps obtained in the View Pooling layer are then fed into the CNN Part II. In this part, only one branch CNN is used for dealing with the association information cross multiple views by using high-response feature maps. After that, the high-response feature vectors can be extracted in the (fully connected layer, and since its location lies in seventh layer, thus, we called it as fc7 for abbreviation.) fc7 layer of CNN Part II, and these feature vectors serve as the features for a group.

Contractive Loss Function Since **GPCNN** is a double-chain CNN, there are two groups of feature vectors obtained in fc7s individually. The contractive loss function is utilized to receive these two feature vectors and compute loss of two groups according to their feature distances. In

detail, we randomly extract two groups of views $Z_{i,m}^G = \{v_1^G, v_2^G, v_3^G\} \in o_i^G$ and $Z_{j,n}^Q = \{v_1^Q, v_2^Q, v_3^Q\} \in o_j^Q$ from the gallery object i and query object j. The contractive loss function is defined in Eq. (1).

$$E = \frac{1}{2N} \sum_{n=1}^{N} \{y^n d^n + (1 - y^n) max(margin - d^n, 0)\}, \tag{1}$$

where N denotes the batch size, y^n denotes target label of nth pairwise sample (0 or 1) between group $Z_{i,m}^G$ and group $Z_{j,n}^Q$, $margin$ denotes the maximum distance boundary in the current batch, and d^n denotes the euclidean distance between group $Z_{i,n}^G$ and group $Z_{j,n}^Q$.

The contractive loss function describes the matching degree of two objects. In each iteration of SGD training, if two objects are the same category, the contractive loss function will reduce the feature distance between these two objects; otherwise, it will increase the feature distance between these two objects. After training, we use $d(Z_{i,n}^G, Z_{j,n}^Q)$ as the similarity metric of two groups, so as to simplify the whole retrieval process. In other words, our network architecture can output the similarity metric of two groups directly.

Implementation Details

In detail, we randomly extract three views from each object, and then the two groups of views diverge at a data layer and are sent into the two branches separately. Each branch has two parts CNN architectures. CNN Part I is the parallel processing of multiple views, and in this part, the group of views are sent through three parallel CNNs that share the same parameters. CNN Part I consists of five convolution layers $conv$1-5, and each convolution layer is followed by the Pooling layer and Rectified Linear Unit (ReLU). Particularly, there are two Batch Norm layers $norm$1-2 followed by con-1 and con-2, respectively. After Part I, We use the element-wise maximum operation for the views of each group in the view-pooling layer, so as to mine high-response feature maps to facilitate the comparison of two objects. The View-pooling layer is closely related to the max-pooling and maxout layer, with the only difference on the dimension where the pooling operation is performed. An alternative is the element-wise mean operation, which however shows weaker performance in our experiments. Moreover, we also observe that it should be placed close to the last convolutional layer (conv5). Next, the high-response views are sent through the CNN Part II, which consists of two inner product layers fc6-7 to deal with high-response views.

Experiments

In order to evaluate the retrieval performance of our **GPCNN**, we perform 3D object retrieval on three public 3D object datasets. We initially introduce the datasets and evaluation criteria, and then, the experimental setting is given. Meanwhile, we will evaluate **GPCNN** from three aspects respectively:1) We will discuss how many training samples we can obtain, and then compare with the training samples in the original dataset; 2) We will evaluate the performance of **GPCNN**, and then compare with hand-crafted feature representation by different retrieval algorithms; 3)In addition, we also compare **GPCNN** network architecture to VGG-16 and siamese convolutional neural network.

Dataset

In our experiments, three widely-used datasets are employed where each object in gallery set is firstly represented by a free set of views which means that these views can be captured from any direction without camera constraint. The details of these datasets are shown as follows:

- ETH 3D object dataset (Ess et al. 2008), where it contains 80 objects belonging to 8 categories, and each object from ETH includes 41 different view images;
- NTU-60 3D model dataset (Chen et al. 2003), where it contains 549 objects belonging to 47 categories, and each object from NTU-60 includes 60 different view samples;
- MVRED 3D category dataset (Liu et al. 2016), where it contains 505 objects belonging to 61 categories, and each object from MVRED includes 36 different view images;

Object samples from different datasets are shown in Fig.2 respectively.

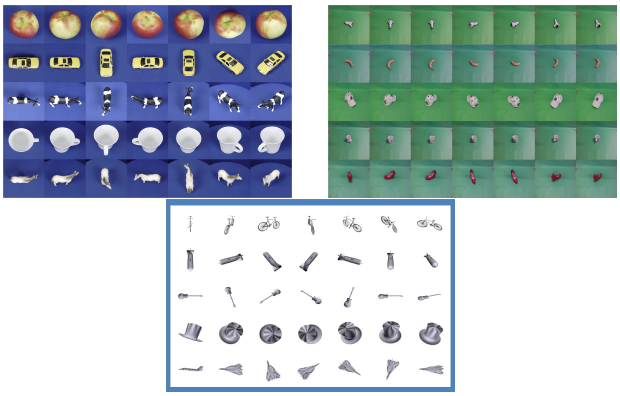


Figure 2: Object examples from different 3D datasets, from top to down, and left to right, samples come from ETH, NTU, MVRED datasets respectively

Evaluation Criteria

In order to fully assess the performance of different algorithms, seven evaluation criterions are employed to evaluate the retrieval performance, and they are Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), F-measure (F), Discounted Cumulative Gain (DCG) (Osada et al. 2001), Average Normalized Modified Retrieval Rank (ANMRR) and Precision-Recall Curve (PR-Curve). For the details, you can find them in (Liu et al. 2016; Chen et al. 2003; Ess et al. 2008).

Here, it is noted that for NN, FT, ST, F and DCG the bigger the better, but for ANMRR the smaller the better, and the greater of the area under PR-Curve, its performance is better. In addition, in order to fair competition, we also follow

the parameter setting in other papers, where k is set to ten, thus, we also compute the performance when k is used (each retrieval, top ten retrieval results are utilized to compute the evaluation criteria value).

Experimental Setting

For each dataset, the first 80% views in each object is utilzied as gallery set and the remaining views in each object is used as query set. When building the training dataset and validation dataset, we choose group-pair samples from the gallery set. The proportion of positive and negative group-pair sample is 1:3 for all datasets. In ETH, 10,000 positive group-pair samples and 30,000 negative group-pair samples from ETH gallery set are produced as training samples. In NTU and MVRED, 30,000 positive group-pair samples and 90,000 negative group-pair samples from NTU gallery set are collected as training samples. For the validation dataset, the same scheme is employed. In order to prove the stability of testing, we cluster the remaining views of each object into three subclusters, and then, these three views are considered as a group to represent the object.

In our experiments,for **GPCNN**, when it is utilzied into NTU and MVRED datasets, the sizes of convolution kernels of each layer in CNN Part I are 32, 64, 128,256 and 512 respectively, but when it is evaluated on ETH dataset, the sizes of convolution kernels of each layer in CNN Part I are 16, 32, 64, 128 and 256 respectively. As for VGG-16 and siamese convolutional neural network, the parameters are pre-trained on the ImageNet dataset, and then each 3D dataset is utilized to fine-tune the parameters with default settings.

Competing Methods

Several popular methods are implemented for comparison:

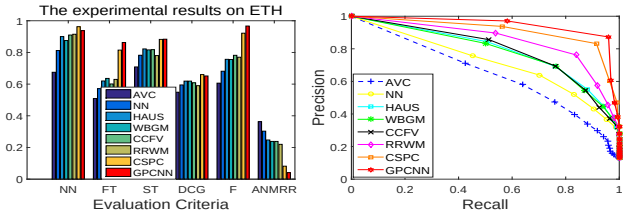
- Adaptive View Clustering (AVC) (Osada et al. 2001): AVC selects the optimal 2D characteristic views of a 3D model based on the adaptive clustering algorithm and then utilizes a probabilistic Bayesian method for 3D model retrieval.
- Camera Constraint Free Ciew (CCFV) (Gao et al. 2012): A CCFV model is generated on the basis of the query Gaussian models by combining the positive matching model and the negative matching model.
- Weighted Bipartite Graph Matching (WBGM) (Gao et al. 2011) WBGM builds the weighted bipartite graph only with the attributes of individual 2D views.
- Hausdorff distance (HAUS) & Nearest Neighbor (NN) (Steinbach, Karypis, and Kumar 2000): The Hausdorff distance is used to measure the maximum distance between a set and its nearest point in the other set. The nearest neighbor-based method is similar to HAUS.
- Class-Statistic and Pair-Constraint (CSPC) (Gao et al. 2016): The retrieval results from different retrieval algorithms are combined.
- Siamese Convolutional Neural Network (Chopra, Hadsell, and Lecun 2005; Bromley et al. 1993): Siamese Convolutional Neural Network takes a pair of samples instead of taking single sample as input, and the loss functions are usually defined over pairs.
- VGG-16 network (Chatfield et al. 2014): VGG-16 consists of five groups of convolution, and each group includes 3 convolution layers. Meanwhile, there are three fully connected layers and a Softmax classification layer. The inner product layer fc7 (after Rectified Linear Units, 4096-dimensional) is used as image descriptor.

The Number of Training Samples

The details of three datasets are shown in Table.1. It shows that if we employ the original data as the training dataset, the number of samples has only 3280 in ETH dataset. However, since multiple-view samples from different objects are chosen as a group-pair samples, thus, we can easily obtain 10660×10660 samples whose number of group-pair samples is far more than the number of views in the original dataset. In detail, there are only 3280, 32940 and 18180 samples for training and validating network in ETH, NTU-60 and MVRED respectively. In this way, it is easy to lead to over-fitting when training CNN network, moreover, it also overlooks the generality of multi-view objects. However, when group-pair samples are utilized, multiple views are combined into a pair, thus, we have far more samples to train and validate the networks. Moreover, the generalization ability of CNN can be improved and the common information from multiple views can be fully explored.

Performance Evaluation and Comparison

We firstly assess the performance of **GPCNN** on three 3D datasets, and then compare it with the state-of-the-art algorithms, whose performances are obtained by running the codes offered by the authors with default settings. In addition, in the state-of-the-art algorithms, the efficient Zernike feature is extracted for all datasets. Their results are shown in Fig.3, Fig.4 and Fig.5. From Fig.3 (a) and Fig.3 (b), we can observe that since ETH is a small dataset which only includes 80 objects, the state-of-the-art algorithms can have good retrieval performances. Since the group-pair scheme is utilized, we can obtain a lot of training samples for **GPCNN**, thus, **GPCNN** also can achieve a gain of 4%-36%, 1%-34%, 3%-26% on FT, ST, F and obtain a decline of 4%-28% on ANMRR.



(a) Retrieval Comparison (b) PR Comparison

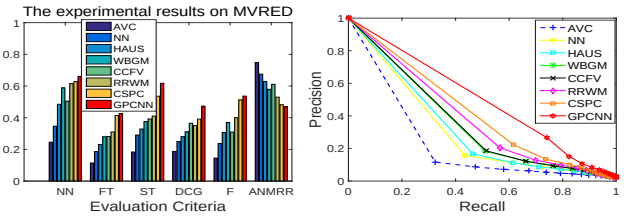
Figure 3: Retrieval result comparison and PR Curve comparison of different 3D model retrieval algorithms on ETH datasets

In MVRED dataset, the retrieval comparison obviously indicates that **GPCNN** can consistently outperform others.

Table 1: Comparison of the number of samples with single-view input and multiple-view inputs

Data Sets	Original Samples			Group-Pair Samples			
	Objects	Views (one object)	Samples	Views in Group	Groups (one object)	Group pairs (two objects)	All Group Pairs (all objects)
ETH	80	41	3280	3	10660	10660×10660	$3160 \times 10660 \times 10660$
NTU-60	549	60	32940	3	34220	34220×34220	$150426 \times 34220 \times 34220$
MVRED	505	36	18180	3	7140	7140×7140	$127260 \times 7140 \times 7140$

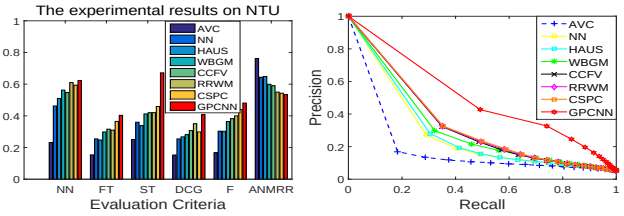
From Fig.4 (a) and Fig.4 (b), **GPCNN** achieves a gain of 3.5%-42%, 1.5%-27.5%, 8%-39%, 8.3%-28%, 2.5%-34.5% on NN, FT, ST, DCG, F and obtain a decline of 2%-29% on ANMRR.



(a) Retrieval Comparison (b) PR Comparison

Figure 4: Retrieval result comparison and PR Curve comparison of different 3D model retrieval algorithms on MVRED 3D dataset

In NTU dataset, from Fig.5 (a) and Fig.5 (b), **GPCNN** outperforms 3%-42%, 4%-27%, 22%-41%, 11%-37% and 4%-32% on NN, FT, ST, DCG, F and obtain a decline of 2%-29% on ANMRR. In addition, for the PR comparison, Fig.3 (b), Fig.4 (b), Fig.5 (b) shows the PR curve comparison of different 3D object retrieval algorithms on ETH, NTU-60 and MVRED datasets respectively. From them, we can see that our **GPCNN** obviously outperforms all other algorithms on MVRED and NTU datasets, but it is still a litter better than CSPC method on ETH dataset. it is noted that the work of CSPC was published on ACM MM 2016 where different retrieval results from different retrieval algorithms are combined. In conclusion, **GPCNN** is efficient and effective, which significantly outperforms the state-of-the-art methods.



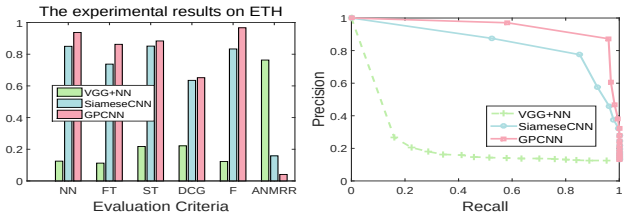
(a) Retrieval Comparison (b) PR Comparison

Figure 5: Retrieval result comparison and PR Curve comparison of different 3D model retrieval algorithms on NTU dataset

Compared with other CNNs

In order to further prove the superiority of **GPCNN**, in this section, we will compare **GPCNN** with the Siamese Convolutional Neural Network and VGG-16 network. For siamese convolutional neural network, a pair of views will be received as training unit, but in **GPCNN**, group pair samples will be input, accordingly, there is no View-Pooling in Siamese Convolutional Neural Network. For VGG-16, this network is one branch network which can only receive one view and use Softmax layer to implement image classification. To implement 3D object retrieval, the inner product layer fc7 of VGG-16 (after Rectified Linear Units, 4096-dimensional) is used as image descriptor, and then we use NN method to finish 3D object retrieval. Their results are shown in Fig.6, Fig.7 and Fig.8 respectively.

From these figures, we can find that the performance of Siamese Convolutional Neural Network is much better than that of VGG-16. In other words, the performance of Siamese network is more efficient than that of single network, such as VGG-16. In addition,in ETH dataset,Fig.6(a) and Fig.6 (b) demonstrate that **GPCNN** achieves a gain of 8%-90%, 13%-62%, 3%-67%, 2%-43%, 13%-94% on FT, ST, F and obtain a decline of 11%-72% on ANMRR.

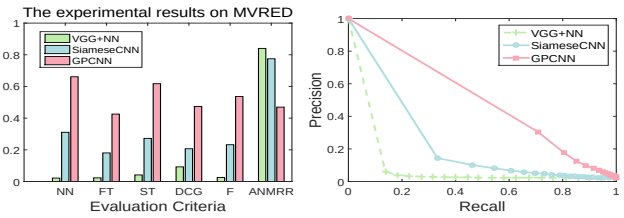


(a) Retrieval Comparison (b) PR Comparison

Figure 6: Retrieval result comparison and PR Curve comparison of different 3D model retrieval algorithms on ETH datasets

In MVRED dataset, the retrieval comparison also obviously indicates that **GPCNN** can consistently outperform others. From Fig.7 (a) and Fig.7 (b), **GPCNN** achieves a gain of 35%-64%, 24%-40%, 34%-57%, 27%-38%, 30%-51% on NN, FT, ST, DCG, F and obtain a decline of 31%-7% on ANMRR.

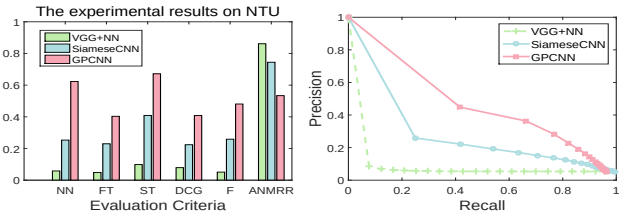
In NTU dataset, from Fig.8 (a) and Fig.8 (b), **GPCNN** outperforms 37%-57%, 18%-36%, 27%-31%, 18%-33% and 23%-43% on NN, FT, ST, DCG, F and obtain a decline of 21%-33% on ANMRR. In addition, for the PR



(a) Retrieval Comparison (b) PR Comparison

Figure 7: Retrieval result comparison and PR Curve comparison of different 3D model retrieval algorithms on MVRED 3D datasets

curve comparison, Fig.6 (b), Fig.7 (b), Fig.8 (b) show the PR curve comparison of different CNNs on ETH, NTU-60 and MVRED datasets respectively. From them, we can see that our **GPCNN**'s performance significantly higher than others. In other words, group-pair and view pooling scheme can further improve the performance, and the latent complementary information is fully explored.



(a) Retrieval Comparison (b) PR Comparison

Figure 8: Retrieval result comparison and PR Curve comparison of different 3D model retrieval algorithms on NTU dataset

Conclusion

In this work, a novel end-to-end solution named Group Pair Convolutional Neural Network (**GPCNN**) is proposed which can jointly learn the visual features from multiple views of a 3D model and optimize towards the object retrieval task. Extend experiment results demonstrate that the group-pair network architecture is very useful, and it can reduce the requirement of the training samples in the original dataset. Moverover, view pooling scheme is effecient to explore the latent complementary information from different views, and multi-view samples can supply much more information, and generate compact descriptor with powerful discrimination for individual 3D objects. Finally, the end-to-end solution scheme can further improve the discrimination which will be much suitable for 3D object retrieval.

In the future work, we will explore some practical questions for **GPCNN**, including how to construct the group-pair samples, which views are most informative, how many group-pair samples are necessary for a given level of accuracy, how to choose the optimal number of views for each group.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No.61572357, No.61202168).

References

Ankerst, M.; Kastenmuller, G.; Kriegel, H.; and Seidl, T. 1999. 3d shape histograms for similarity search and classification in spatial databases. *Lecture Notes in Computer Science* 207–226.

Ansary, T. F.; Daoudi, M.; and Vandeborre, J. 2007. A bayesian 3-d search engine using adaptive views clustering. *IEEE Transactions on Multimedia* 9(1):78–88.

Bromley, J.; Bentz, J. W.; Bottou, L.; Guyon, I.; Lecun, Y.; Moore, C.; Sackinger, E.; and Shah, R. 1993. Signature verification using a siamese' time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence* 7(4):669–688.

Chatfield, K.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Return of the devil in the details: Delving deep into convolutional nets. *Computer Science*.

Chen, D.; Tian, X.; Shen, Y.; and Ouhyoung, M. 2003. On visual similarity based 3d model retrieval. In *Computer Graphics Forum*, 223–232.

Cho, M.; Lee, J.; and Lee, K. M. 2010. Reweighted random walks for graph matching. In *European Conference on Computer Vision*, 492–505.

Chopra, S.; Hadsell, R.; and Lecun, Y. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 539–546 vol. 1.

Ess, A.; Leibe, B.; Schindler, K.; and Gool, L. V. 2008. A mobile vision system for robust multi-person tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 1–8.

Gao, Y.; Dai, Q.; Wang, M.; and Zhang, N. 2011. 3d model retrieval using weighted bipartite graph matching. *Signal Processing Image Communication* 26(1):39–47.

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

Gao, Z.; Wang, D.; Zhang, H.; Xue, Y.; and Xu, G. 2016. A fast 3d retrieval algorithm via class-statistic and pair-constraint model. In *ACM on Multimedia Conference*, 117–121.

Gao, Z.; Li, S. H.; Zhang, G. T.; Zhu, Y. J.; Wang, C.; and Zhang, H. 2017. Evaluation of regularized multi-task leaning algorithms for single/multi-view human action recognition. *Multimedia Tools and Applications* 1–24.

Gomez-Donoso, F.; Garcia-Garcia, A.; Garcia-Rodriguez, J.; Orts-Escolano, S.; and Cazorla, M. 2017. Lonchanet: A sliced-based cnn architecture for real-time 3d object recognition. In *International Joint Conference on Neural Networks*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(9):1904.

Ip, C. Y.; Lapadat, D.; Sieger, L.; and Regli, W. C. 2002. Using shape distributions to compare solid models. In *ACM Symposium on Solid Modeling and Applications*, 273–280.

LENG; Biao; Zheng; Xiaoman; and ZHANG. 2009. Mate: A visual based 3d shape descriptor. *Chinese Journal of Electronics* 18(2):291–296.

Leng, B.; Du, C.; Guo, S.; Zhang, X.; and Xiong, Z. 2015. A powerful 3d model classification mechanism based on fusing multi-graph. *Neurocomputing* 168:761–769.

Leordeanu, M., and Hebert, M. 2005. A spectral technique for correspondence problems using pairwise constraints. In *Tenth IEEE International Conference on Computer Vision*, 1482–1489 Vol. 2.

Liu, S.; Liu, S.; Cai, W.; Che, H.; Pujol, S.; Kikinis, R.; Fulham, M.; and Feng, D. 2014. High-level feature based pet image retrieval with deep learning architecture. *Journal of Nuclear Medicine* 55(S1):2028.

Liu, A.; Wang, Z.; Nie, W.; and Su, Y. 2015. Graph-based characteristic view set extraction and matching for 3d model retrieval. *Information Sciences* 320:429–442.

Liu, A. A.; Nie, W. Z.; Gao, Y.; and Su, Y. T. 2016. Multi-modal clique-graph matching for view-based 3d model retrieval. *IEEE Trans Image Process* 25(5):2103–2116.

Liu, A. A.; Nie, W. Z.; Yue, G.; and Su, Y. T. 2017. View-based 3-d model retrieval: A benchmark. *IEEE Transactions on Cybernetics* PP(99):1–13.

Lu, K.; Wang, Q.; Xue, J.; and Pan, W. 2014. 3d model retrieval and classification by semi-supervised learning with content-based similarity. *Information Sciences* 281:703–713.

Mademlis, A.; Daras, P.; Tzovaras, D.; and Strintzis, M. G. 2009. 3d object retrieval using the 3d shape impact descriptor. *Pattern Recognition* 42(11):2447–2459.

Mavar-Haramija, M.; Prats-Galino, A.; and Notaris, M. 2015. Interactive 3d-pdf presentations for the simulation and quantification of extended endoscopic endonasal surgical approaches. *Journal of Medical Systems* 39(10):127.

Nie, W. Z.; Liu, A. A.; and Su, Y. T. 2016. 3d object retrieval based on sparse coding in weak supervision. *Journal of Visual Communication and Image Representation* 37(C):40–45.

Ohbuchi, R., and Furuya, T. 2009. Scale-weighted dense bag of visual features for 3d model retrieval from a partial view 3d model. In *IEEE International Conference on Computer Vision Workshops*, 63–70.

Ohbuchi, R.; Osada, K.; Furuya, T.; and Banno, T. 2008. Salient local visual features for shape-based 3d model retrieval. In *IEEE International Conference on Shape Modeling and Applications*, 93–102.

Osada, R.; Funkhouser, T. A.; Chazelle, B.; and Dobkin, D. P. 2001. Matching 3d models with shape distributions. *Statistical Methods and Applications* 154–166.

Osada, R.; Funkhouser, T.; Chazelle, B.; and Dobkin, D. 2002. Shape distributions. *Acm Transactions on Graphics* 21(4):807–832.

Persoon, E., and Fu, K. 1977. Shape discrimination using fourier descriptors. *systems man and cybernetics* 7(3):170–179.

Polewski, P.; Yao, W.; Heurich, M.; Krzystek, P.; and Stilla, U. 2015. Detection of fallen trees in als point clouds using a normalized cut approach trained by simulation. *Isprs Journal of Photogrammetry and Remote Sensing* 105:252–271.

Shih, J. L.; Lee, C. H.; and Wang, J. T. 2007. A new 3d model retrieval approach based on the elevation descriptor. *Pattern Recognition* 40(1):283–295.

Socher, R.; Huval, B.; Bhat, B.; Manning, C. D.; and Ng, A. Y. 2012. Convolutional-recursive deep learning for 3d object classification. *NIPS* 665–673.

Steinbach, M.; Karypis, G.; and Kumar, V. 2000. A comparison of document clustering techniques. In *In KDD Workshop on Text Mining*.

Su, H.; Maji, S.; Kalogerakis, E.; and Learnedmiller, E. 2016. Multi-view convolutional neural networks for 3d shape recognition. In *IEEE International Conference on Computer Vision*, 945–953.

Wang, D.; Wang, B.; Zhao, S.; Yao, H.; and Liu, H. 2016. *Exploring Discriminative Views for 3D Object Retrieval*. Springer International Publishing.

Wang, F.; Kang, L.; and Li, Y. 2015. Sketch-based 3d shape retrieval using convolutional neural networks. *Computer Science* 1875–1883.

Yang, L., and Albregtsen, F. 1996. Fast and exact computation of cartesian geometric moments using discrete green's theorem. *Pattern Recognition* 29(7):1061–1073.

Zagoruyko, S., and Komodakis, N. 2015. Learning to compare image patches via convolutional neural networks. *CVPR* 4353–4361.