Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-08967-7

Exploring contextual information for view-wised 3D model retrieval

Wenhui Li[1] · Yuting Su[1] · Zhenlan Zhao[1] · Tong Hao[2] · Yangyang Li[3]

Received: 3 September 2019 / Revised: 15 February 2020 / Accepted: 22 April 2020 /
Published online: 29 May 2020

Abstract
Recently, with the rapid development of digital technologies and its wide application, 3D model retrieval is becoming more and more important in graphic communities. In this task, how to effectively represent the 3D model and how to robustly measure similarity between pair-wise models are two crucial problems. In previous work, most papers dedicated to researching how to effectively using the visualize features to represent 3D model and using the visual information to measure the similarity. However, visual feature can not represent 3D model well because of the model variations in poses and illumination. To address this task, we propose an novel framework, which utilizes the visual and contextual information to construct the rank graphs and fuses these two graphs to enhance the similarity measure. When fusing visual and contextual information, we define four strategies to measure the similarity among models according to the relation between the query model and the gallery models. The extensive experimental results demonstrate the superiority of our proposed method compare against the state of the arts.

Keywords View-based model retrieval · Contextual information · Similarity measure

✉ Tong Hao
joyht2001@163.com

Wenhui Li
liwenhui@tju.edu.cn

Yuting Su
ytsu@tju.edu.cn

Zhenlan Zhao
zhenlantju@gmail.com

Yangyang Li
liyangyang@cetc.com.cn

[1] School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

[2] Tianjin Key Laboratory of Animal and Plant Resistance/College of Life Science, Tianjin Normal University, Tianjin 300387, China

[3] National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data (PSRPC), CAEIT, Beijing, China

1 Introduction

In recent years, with the development of computing systems and computing power, 3D model retrieval has received a lot of attentions [10, 19, 20] on various domains such as architectural design, computer-aided design, printing and digital entertainment. These applications have resulted in huge 3D model data. Therefore, how to design an data-driven method to represent 3D models [11, 27, 28] and how to effectively measure the similarity between two models [5, 6, 19] has become a crucial issue. The task of 3D model retrieval is that finding the similar 3D models from the existing dataset with the query 3D model. The existing methodologies can be grouped into two classes, model-based methods, which utilize and explore spatial information to represent 3D model and view-based methods, which translate 3D model into multiple views. Various methods developed novel shape descriptors [1, 12, 15, 26, 32, 35] to represent 3D models by using different types of spatial information, such as volumetric, polygonal mesh, point cloud and so on. However, it cannot always get 3D models in many applications. So it is necessary to reconstruct 3D models based on a carefully collected group of 2D views. Whereas, the reconstruction is computationally expensive and the sampling of the images needs to be fine-grained enough to reconstruct a reasonably good 3-D model, which is difficult to realize [7]. All these difficulties severely constrain the practical applications of model-based 3-D model retrieval methods.

The existing works show that view-based methods usually outperform model-based methods in term of retrieval accuracy. The view-based methods obtain multiple views by rending the 3D models and utilize the mature algorithms of 2D image to represent 3D model. The view-based methods have become a prevalent research direction in recent years because of the superior performance. The view-based methods can be divided into two classes with whether using the label information of 3D models. For supervised methods, a lot of works [4, 37] utilize the deep neural network to model the relation between multiple views of 3D models, which need huge human annotations. Because of the limitation of the label requirement, many researches [5, 6] focus on how to retrieval 3D models in unsupervised manner. which shows the promising performance.

However, as far as we know, the existing problems of most view-based retrieval methods are that they learn the discriminative representation of 3D model or construct the similarity graph by only using the multiple visual features, which cannot always get good performance because of the diversity of 3D models. In this paper, we propose an integrating contextual information with similarity measure method to solve the model retrieval task, which joint visual information and contextual information between query model and gallery models to measure the similarity. When processing the similarity, four- stages measure is designed to enhance the similarity measure among models. Our pipeline is summarized in Fig. 1. As shown in Fig. 1, our framework joints visual and contextual information to conduct two rank graphs, where the models are the nodes and the similarities are the edge weights. Subsequently, the similarity information and the disparity information are combined together to divide the gallery into four parts, which can separate the similar samples and dissimilar samples. Finally, we re-rank the similarities of four stages and get the final retrieval result.

The main contributions of this paper are followed as:

- We develop a novel 3D retrieval method, which can combine the visual information of the model and the contextual information of the model corresponding to other models to explore more plentiful information to improve the robustness of similarity measure.
- We define four strategies to measure the similarity among models according to the relation between the query model and the gallery models. The stages can make the similar models close to each other and make the dissimilar models far from each other.

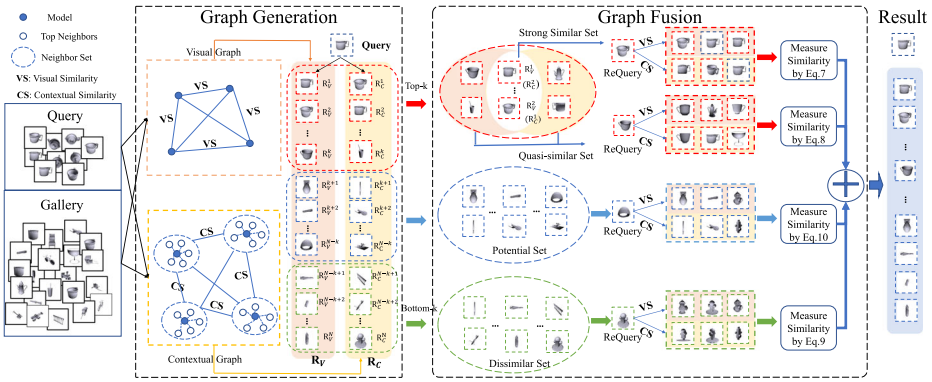

Fig. 1 The flowchart of our proposed method. Our method consists two key parts, graph generation and graph fusion. In the graph generation, we utilize visual similarity and contextual similarity to construct two graphs, respectively. In the graph fusion, we firstly split the rank graph into four parts by considering the overlap information between two graphs. Then, we define four similarity measure strategies with respect to the four parts to re-calculate the similarity. Finally, we obtain the final retrieval result by ranking the similarity

– Extensive experiments are conducted on ETH, MVRED and NTU datasets. The experimental results and the parameter analysis confirm the effectiveness and efficiency of our method.

The rest of our paper is organized as follows. In Section 2, some related work regarding the representative methods of 3D model retrieval from both model-based and view-based are introduced. Section 3 describes our rank graph generation and rank graph fusion. Section 4 mainly introduce our experimental parameter settings and the results. In Section 5, we make a summary of our paper.

2 Related work

The existing 3D model retrieval methods can be classed into model-based methods and view-based methods.

Model-based methods These methods take a dominant place in early application. The premise of model-based retrieval method is to be accessible to obtain the 3D model of each object and generate the 3D descriptor features, then the recognition algorithms based on the 3D descriptor features can be directly trained. Popular 3D features include geometric moments [32],surface distributions [22], shape descriptors [26], etc. Because of the construction of large-scale 3D model databases like TurboSquid, Shapeways and 3D Warehouse, building the specific classifiers of 3D models from 3D representative descriptors directly became possible and easier. Besides, a variety of distance measure metrics have been introduced to access the resemblance among model descriptors. For example, Euclidean distance is a straight-line distance between two descriptors in Euclidean space. In addition, there are other distance measure, such as the Hausdorff distance [8], Cosine distance and Earth Movers distance [33]. Similarity search that adopted quadratic forms was supported efficiently by a common filter-refinement architecture for query processing, and the architecture was employed to guarantee the particular flexibility. Cheuk et al. [14] described a new method that utilized shape distributions to compare among solid 3D

model. The shape distribution metrics was developed by the growing application to approximate model comparison such as polygonal meshes and Virtual Reality Modeling Language (VRML) models and this method focused on adapting these metrics. In [35], a generic Fourier Descriptor(GFD) was proposed for model retrieval, which overcame the drawbacks of existing model descriptors with non-robust feature and poor generalization performance. Comparing Multiple Resolution Reeb Graphs between polyhedral models was proposed in [12], where a rough-to-detailed strategy is adopted to calculate the resemblance between 3D models and maintains the consistency of the graph structures at the same time. Fang et al. [34] developed techniques to train a deep CNN with the guide of both extracting concise and geometrically informative shape descriptor and redefining existing descriptors.

View-based methods Inspired by the superior performance, the view-based methods have attracted rising attention in recent years. Generally speaking, the key modules of this type retrieval algorithms lie in two main aspects: feature representation and similarity measurement. Chen et al. [3] introduced the Light Field Descriptor(LFD) to extract the representative feature based on the concept that if two models are similar, they should have similar appearance from all angles. In order to capture the views of all angels, an array of cameras were replaced on the vertices of the dodecahedron over a hemisphere. Each set of LFD is defined by ten views. Then the matched result- s are encode by Zernike moments [16] and Fourier descriptors to improve its robustness against rotations, translations and noise. Method proposed by [25] brought about a novel deep network with multiple layers. The input of the network were multiple views rendered from a 3D model, which would be combined into a compact single view. Then the feature learning model was generated by training the network with compact views. The Euclidean distance was leveraged as the metric criterion to calculate the similarity among models. AVC [2] is a typical view selection method that leveraged the statistical model distribution scores plus a probabilistic Bayesian information criteria to cluster representative views. Furthermore, Giorgi et al. [9] proposed an approach to select the best view of the view pool by utilizing geometrical characteristics. The single 2D view is semantically grounded. Combining with the invariant and informative shape descriptor, the method could refine the intra-information and had a higher potential for 3D retrieval. Gao et al. [6] propose a 3D model retrieval algorithm, which releases the camera constraint.

Besides learning discriminative representation, finding a proper way to estimate the similarity between the query model and all candidate models also has a huge impact on the performance of the retrieval method. Nie et al. [30] applied clustering to select exemplar views and then, they leveraged the weighted locality-constrained group sparse coding for similarity measure. Different from traditional similarity estimation method where the distances of view pairs across the two models are integrated, Wang et al. [29] introduced a discriminative probabilistic modeling method. The GMMs of each model was acquired by modeling and the distance between two models was reckoned according to the Kullback-Leibler (KL) divergence. Zhao et al. [36] managed to apply a feature fusion method based on multi-modal graph learning to view-based 3D model retrieval. Each view is described by several visual features and the Hausdorff distance of models are calculated with multiple views. Hong et.al [13] adopted multi-view ensemble manifold regularization (MEMR) to merge multi-view data by constructing the hyper-graph. The graph matching method proposed by [24] acted as a criterion to compute the distance between two sets of views. The graph matching method can help preserve local and global structure. Liu et al. [19] proposed the multi-modal clique graph to solve the 3D model retrieval. They utilized the clique graph to represent the 3D model and measuring the similarity by clique graph matching. In

[31], a boosting multi-model graph learning based method was introduced. The boost was caused by the investigation into the difference of the semantic and discriminative abilities among views, which enhanced the similarity measure.

3 Proposed method

In the proposed method, our goal is to design two rank graphs by using the visual and contextual information of the query model to enhance the similarity measure. To achieve this, we exploit the contextual information with the neighbor information of the query model because of the effective context information in the neighbor set. As shown in the framework Fig. 1, our frame- work mainly contains two modules: graph generation and graph fusion. We will detail these two modules in the following subsection.

3.1 Graph generation

For view-based method, each 3D model is translated into a group of view images. Given N models with S views, we use the pre-trained deep model to extract the features for one model. We define the feature of model i by f_i, where $f_i = \{v_1, v_2, ..., v_S\}$ and $v_j \in \mathbb{R}^D$ is the feature for view j, $j \in [1, S]$. According to the features, we can measure the similarity between two models. We use the model as the node of the graph and the similarity between them as the edge of two nodes. In this subsection, we define two methods to measure the similarity. The first method uses set distance as the similarity measure and the other method uses neighbor distance as the similarity measure.

Visual graph We use the set distance as the similarity to construct the visual graph. To simplify the computation of multiple views, we mean the features of multiple views and use the average feature as the model representation. We adopt Euclidean distance between two average features as the similarity, which is defined as following:

$$Sim_V(x_1, x_2) = D(\frac{1}{|V_1|}\sum_{i=1}^{|V_1|} v_i, \frac{1}{|V_2|}\sum_{j=1}^{|V_2|} v_j), \tag{1}$$

where $Sim_V(x_1, x_2)$ represents the similar between model x_1 and x_2. $|V_1|$ and $|V_2|$ represent the view number of model x_1 and x_2, respectively. After getting the similarity $Sim_V(i, \cdot)$ between the query model i and the gallery models, we can rank the similarity and get the visual graph $R_V(i, \cdot)$.

Contextual graph Unlike directly using set distance to construct the visual graph, we utilize the neighbor information to construct the contextual graph to improve the robustness of similarity measure. As one model contains multiple views, we firstly fuse the multiple features of each model and obtain the unified representation for each model. Then, we compute the Euclidean distance between models and obtain the ranked neighbors according to the distance. If model x_1 is similar to model x_2, their neighbors should also be similar to each other. Therefore, we utilize the neighbor information of one model as the model representation by meaning the features of top-k neighbor models, which can improve the robustness of similarity measure. The similarity measure is defined as following:

$$\begin{aligned} &Sim_C(x_1, x_2) = D(\frac{1}{k}\sum_{i=1}^{k} u^i_{x_1}, \frac{1}{k}\sum_{j=1}^{k} u^j_{x_2}) \\ &s.t. \quad u^i_{x_1} \in N^k_{x_1}, u^i_{x_2} \in N^k_{x_2}, \end{aligned} \tag{2}$$

where $Sim_C(x_1, x_2)$ represents the similarity between model x_1 and x_2 with the neighbor information. N_{x_1} is the similar neighbor set of model x_1 and $N_{x_1}^k$ denotes the top-k neighbor set. N_{x_2} is the similar neighbor set of model x_2 and $N_{x_2}^k$ denotes the top-k neighbor set. $u_{x_1}^i$ is one neighbor model from the set $N_{x_1}^k$ and $u_{x_2}^i$ is one neighbor model from the set $N_{x_2}^k$. Similar to the visual graph, we utilize the similarity $Sim_C(i, \cdot)$ computed by Eq. 2 to obtain the contextual graph $R_C(i, \cdot)$.

3.2 Graph fusion

After the rank generation, we can obtain the visual graph R_V and contextual graph R_C. In this subsection, we introduce how to fuse these two graphs. We denote N_V^{k+}, N_C^{k+} as the top-k neighbors in R_V and R_C, and N_V^{k-}, N_C^{k-} as the bottom-k neighbors respectively. Then, we utilize the relation between these two sets to fuse the two graph information. By exploring the importance of the neighbor samples, we can divide the gallery models into four parts: strongly similar models, quasi similar models, potential similar models and dissimilar models. We detail each part as following:

Strongly similar models After we get the two top-k neighbor set N_V^{k+}, N_C^{k+}, we can utilize the relation between these two sets to decide which part the gallery model belongs to. N_V^{k+} and N_C^{k+} are conducted by using different information, so if the same model appears in two sets of query model, it means that this model is strongly similar to the query model. We define the strongly similar model set $SS_+(q)$ for query model q as following:

$$SS_+(q) = N_V^{k+}(q) \cap N_C^{k+}(q) \tag{3}$$

Quasi similar models The N_V^{k+} and N_C^{k+} can not always be completely overlap because of the variation of models. Beside the overlapping samples, there are several unique models in this two sets. We denote the models in the unique sample set of model q as the quasi-similar models $QS_+(q)$, which is defined as following:

$$QS_+(q) = N_V^{k+}(q) \cup N_C^{k+}(q) - SS_+(q) \tag{4}$$

Dissimilar models For the bottom-k models $N_V^{k-}(q)$ and $N_C^{k-}(q)$ of the query model q, they are the most dissimilar models to q. We define the union of $N_V^{k-}(q)$ and $N_C^{k-}(q)$ as the dissimilar models for query model q:

Potential similar models When the models are out of the $N_V^{k+}(q)$ and $N_C^{k+}(q)$, we consider that the models are not so similar to the query model q. We denote these models as potential similar models $PS(q)$:

$$PS(q) = N(q) - N_V^{k+}(q) \cup N_C^{k+}(q) - DS(q) \tag{5}$$

where $N(q)$ denotes the rank neighbor set of the query model q. After dividing the gallery samples into different parts, we can re-calculate the distance between the query model and the gallery model by using the different strategies, which can enhance the similarity measure.

When we measure the similarity between model q and model g, we use the rank index as the distance between two models. We denote the $I(q, N_V(g))$ and $I(q, N_C(g))$ as the index of model q in neighbor set $N_V(g)$ of model g and the index of model q in neighbor set $N_C(g)$ of model g, respectively. As the index of model q in $N_V(g)$ might be different to the

index of model g in $N_V(q)$, we combine the $I(q, N_V(g))$ and $I(g, N_V(q))$ to represent the similarity between the model q and g. Considering the visual and contextual graph, we can obtain four indexes, $I(q, N_V(g))$, $I(q, N_C(g))$, $I(g, N_V(q))$ and $I(g, N_C(q))$ to robustly measure the similarity. According to the graph generation, we can define four similarity strategies as following:

$$D^{SS}(q, g) = \min(I(q, N_V(g)), I(q, N_C(g)), I(g, N_V(q)), I(g, N_C(q))) \tag{6}$$

$$D^{QS}(q, g) = \frac{\min(I(q, N_V(g)), I(g, N_V(q))) + \min(I(q, N_C(g))I(g, N_C(q))}{2} \tag{7}$$

$$D^{DS}(q, g) = \frac{4 \times \max(I(q, N_V(g)), I(q, N_C(g)), I(g, N_V(q)), I(g, N_C(q))}{2} \tag{8}$$

$$D^{PS}(q, g) = \frac{I(q, N_V(g)) + I(q, N_C(g)) + I(g, N_V(q)) + I(g, N_C(q))}{2} \tag{9}$$

where $D^{SS}(q, g)$ denotes the similarity between model q and model g when g belong to the strongly similar sets of q. The $D^{QS}(q, g)$ denotes the similarity between model q and model g when g belong to the quasi similar sets of q. The $D^{PS}(q, g)$ denotes the similarity between model q and model g when g belong to the potential similar sets of q and $D^{DS}(q, g)$ denotes the similarity between model q and model g when g belongs to the dissimilar sets of q. Obviously, we can find the four similarities satisfy the condition, $D^{SS}(q, g) \leq D^{QS}(q, g) \leq D^{OS}(q, g) \leq D^{DS}(q, g)$, which is suitable for similarity measure. Then, we use the $D(q, g)$ as the final similarity and rank the similarities between model q and all models in gallery set to get the retrieval list. The proposed algorithm is shown in Algorithm 1.

Algorithm 1 The algorithm for graph generation and graph fusion.

Input: A query model q and a galley model set $G = \{g_i | i = 1, 2, ...N\}$
Output: The ranking list for the query model q.

1: Obtain two ranking lists $N_V(q)$ and $N_C(q)$by (1) and (2) for the query model q.
2: Obtain strong similar models $SS_+(q)$ by using (3).
3: Obtain quasi-strongly similar models $QS_+(q)$ by using (4).
4: Obtain potential similar models $PS(q)$ by using (6).
5: Obtain dissimilar models $DS(q)$ by using (5).
6: **for** $i = 1$ to $|N(q)|$ **do**
7: g_i is one model in gallery set.
8: **if** $g_i \in SS_+(q)$ **then**
9: Measure the distance $D(q, g_i)$ by (7).
10: **else if** $g_i \in QS_+(q)$ **then**
11: Measure the distance $D(q, g_i)$ by (8).
12: **else if** $g_i \in DS(q)$ **then**
13: Measure the distance $D(q, g_i)$ by (9).
14: **else if** $g_i \in PS(q)$ **then**
15: Measure the distance $D(q, g_i)$ by (10).
16: **end if**
17: **end for**
18: Repeat step 6-17 to obtain all the similarities $D(q, g_i | i = 1, 2, ..., N)$.
19: Rank the similarities $D(q, g)$ to obtain the final retrieval result for query model q.

4 Experiment

In this section, we explore several experiments on ETH, MVRED and NTU datasets (Fig. 2) to demonstrate the superiority of the proposed method. For view representation, we feed the views into the AlexNet network [17] which was pre-trained on the ImageNet dataset, and utilize the output of the last second fully connected layer as the visual feature of views. Finally, each view of 3D model is represented by a 4096-dimension vector (Fig. 2).

4.1 Dataset

- ETH [18]: The ETH database is composed of eight categories that contains 80 objects in total. Each object contains 41 dissimilar views equally distributed over the upper viewing hemisphere, the way that all cameras are placed is directed by subdividing the faces of an octahedron to the third recursion level. In this database, all views will be utilized as both the images in database and the image for querying.
- MV-RED [21]: The MV-RED contains 505 objects being classified to 60 categories. For each object, it was rendered into 2D view images by three cameras from three different locations. In order to acquire the 2D data of 3D models, the table controlled by a step motor would be rotated uniformly so that the Camera-45 and the Camera-60 could capture 36 RGB view images every 10 degree. Additionally, Camera-90 would capture a single RGB image in the top-down view. Thus, each object has 73 view images in total.
- NTU [3]: The unprecedented development of the Internet facilitate the sharing and accessing of all kinds of information. The NTU database consists of 549 object from 47 categories, which are all free downloaded from the Internet. All 3D objects are transformed into Wavefront file format and saved as Obj document format. Each 3D object contains 60 sample images captured from different views.

4.2 Evaluation criteria

For the evaluation on each dataset, each 3D model is selected as the query once for retrieval. To evaluate the 3D model retrieval performance, the following popular criteria are employed as the measures of the retrieval performance.

- Nearest Neighbor (NN): NN is used to assess the performance of the nearest neighbor result.

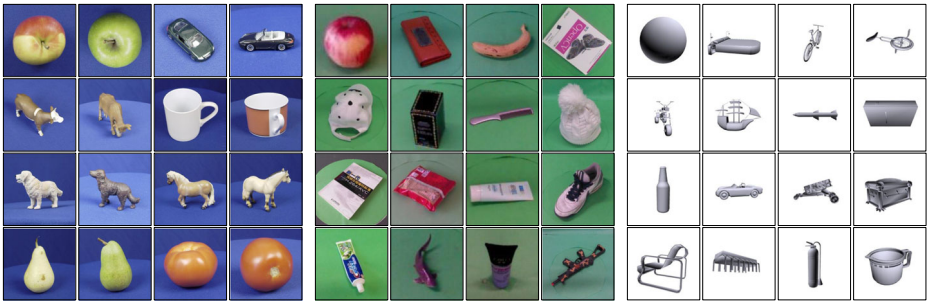

Fig. 2 View examples from three datasets. **a** ETH, **b** MVRED and **c** NTU

- First Tier (FT): Given the specific query, the retrieval process will return a group of results in order from relevant to irrelevant. Defining κ as the number of most related results of the query, First Tier (FT) is a criterion utilized to calculate the recall of the top κ results.
- Second Tier (ST) is defined as the recall of the top 2κ results.
- F-measure: F-measure is defined to jointly evaluate the PR values (precision and recall) of top relevant results. For each query, it takes the top 20 returned results into consideration.
- Discounted Cumulative Gain (DCG) [2]: DCG measures the importance of different positions where relevant results appear. This method assigns relevant results at the top ranking positions with higher weights.
- Average Normalized Modified Retrieval (ANMRR) [23]: ANMRR is defined to access the ranking performance by considering the ranking order. Additionally, in order to measure the retrieval result, the ranking information of relevant models among all the returned retrieved models is used. In contrast to the other criterion, the lower ANMRR value is, the performance is better.

4.3 Comparison against the state-of-the-art methods

In this section, we compare our approach with six methods, NN [7], HAUS [7], AVC [2], CCFV [6], WBGM [5] ,MCG [19]. For our approach, we develop three settings, visual graph(w/o R_C), contextual graph(w/o R_V) and two graphs fusion(R_C+R_V). The experimental results under different evaluation criteria are show in Fig. 3. According to the results, we have several pivotal observations:

- On ETH dataset, which is shown in Fig. 3a, using the two type information obtains the best results. Specifically, our method outperforms the competing methods by 2.5%-16.3% under NN criteria (except MCG), 3.9%-23.9% under FT criteria, 1.5%-14.5% under ST criteria, 1.2%-9.4% under F-measure criteria, 3.1%-23.1% under DCG criteria and observe the decline of 3.2%-22.0% under ANMRR criteria, respectively. Comparing to the MCG methods on NN criteria, our method get same performance. While on other criteria, our method outperforms MCG by 5.6%, 1.5%, 1,2%, 5.5%, 5.1% with the respect to other five criteria. When only using the single visual information RV , we get 93.5%, 70.0%, 80.9%, 60.9%, 81.0%, 18.0 % under six evaluation criteria, which still outperforms AVC by 12.3%, 19.1%, 11.9%, 7.1%, 17.3%, 17.4%. When using the contextual information to generate rank graph RC, the performance is increased by 17.5%, 10.0%, 2.2%, 0.8%, 1.65%, 0.50% under NN, FT, ST, F-measure, DCG and ANMRR which demonstrates the contextual information enhance the similarity measure. Comparing to the contextual information RC, the performance with fusing visual and contextual information is increased by 2.3%, 3.8%, 2.4%, 2.3%, 4.1%, 4.1%, in terms of NN, FT, ST, F-measure, DCG and ANMRR, which confirms the superiority of our four-stages similarity measure.
- On MVRED dataset, which is shown in Fig. 3b, our method outperforms the competing methods by 1.0%-28.4%, 1.8%-20.5%, 1.2%-21.2%, 3.3%-19.3%, 1.6%-24.4%, 1.7%-21.0% with respect to the six evaluation criteria, respectively. When using the contextual information R_C, it outperforms the RV by 1.8%, 4.1%, 2.3%, 5.8%, 3.2%, 4.4% with respect to the NN, FT, ST, F-measure, DCG and ANMRR, which shows the superiority of contextual information comparing to visual information. When we fuse the two types information, the experiment of $R_C + R_V$ outperforms R_C by 5.0%, 2.3%,

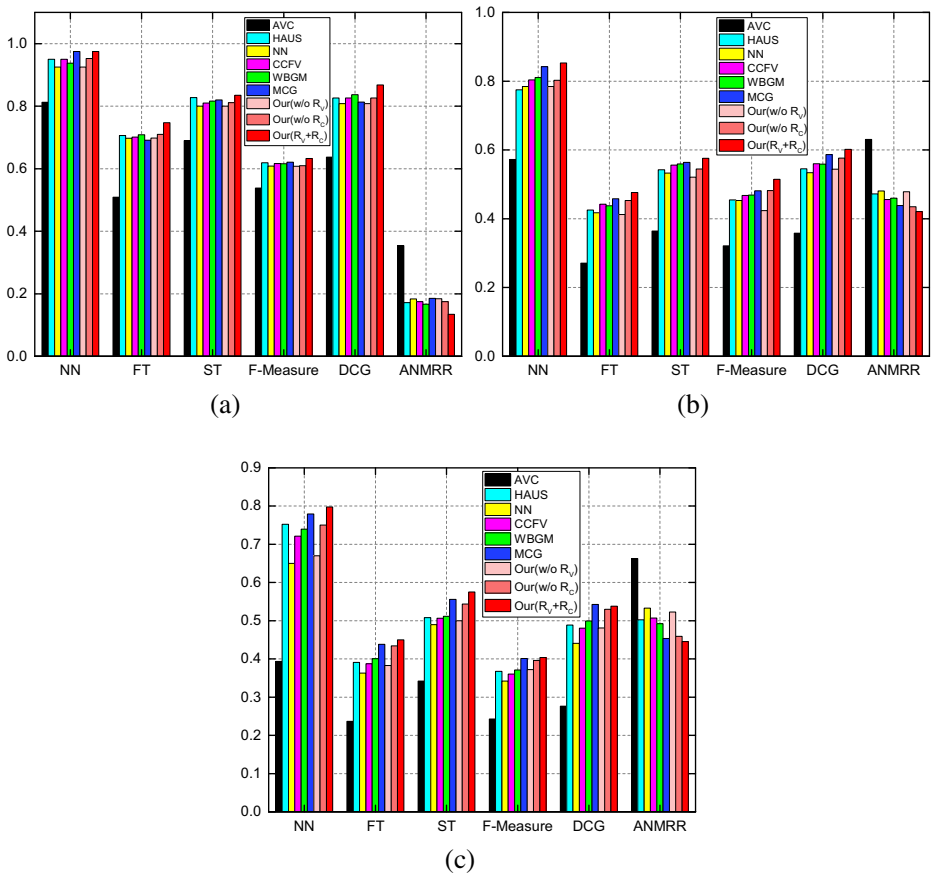

Fig. 3 Performance with different methods on (**a**) ETH, (**b**) MVRED and (**c**) NTU

3.1%, 3.2%, 2.5%, 1.4% in terms of six evaluation criteria, which demonstrates the effectiveness of proposed fusion strategy.

– On NTU dataset, which is shown in Fig. 3c, we can get similar performance comparing to the ETH and MVRED dataset. Our methods outperform all the competing methods, Specifically, we outperform them by 1.8%-40.4%, 1.2%-21.3%, 2.0%-23.3%, 0.2%-16.1%, 0.6%-27.2% and 0.8%-21.8% under the NN, FT, ST, F-measure, DCG and ANMRR, respectively. When using contextual information to generate graph, our R_C outperforms R_V by 8.1%, 5.1%, 4.4%, 2.4%, 4.9%, 6.4% under six criteria. The performance is improved by fusing these two information. Our $R_C + R_V$ outperforms R_C by 4.7%, 1.6%, 3.2%, 0.7%, 1.8%, 1.4% in terms of six evaluation criteria, which demonstrates superiority of the fusion strategy again.

4.4 Performance with different view number

As the limitation in real application, we cannot always get as many views as we need to represent 3D models. To study the impact of view number, we give an empirical analysis on

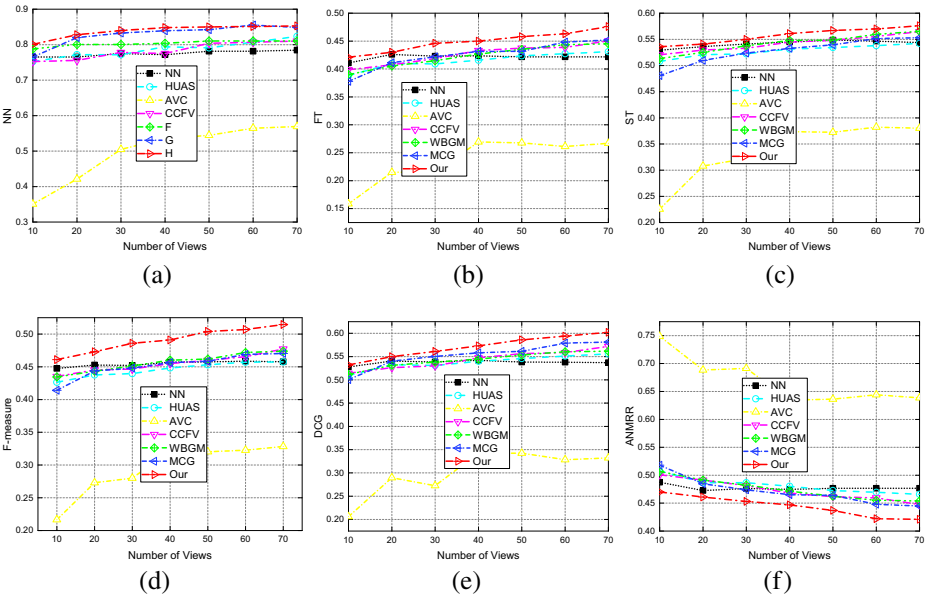

Fig. 4 Performance comparison with different view numbers on MVRED. **a** NN, **b** FT, **c** ST, **d** F-Measure, **e** DCG, **f** ANMRR

MVRED dataset. The experimental results with changing the view number under six evaluation criteria are shown in Fig. 4. Following the setting, we change the view number from 10 to 70 with the interval of 10. From the results, we can get the following observations:

- The performance is increased with adding more views. Intuitively, the 3D model with more views will get richer sample representation. The discriminative representation enhances the performance. As the Fig. 4 shown, the performance of all methods is improved with increasing the view number.
- Our method gets best performance with all view number settings in terms of all evaluation criteria. Specifically, when the view number is set to 70, our method outperforms the second best method(MCG) by 0.3%, 2.4%, 2.3%, 4.4%, 2.1% and 2.4% under the NN, FT, ST, F-measure, DCG and ANMRR evaluation criteria. When we set the view number to 10, our method outperforms the MCG methods by 3.6%, 4.3%, 5.5%, 4.7%, 3.2%, 4.8%, which demonstrates the superiority of our method.
- Even using little views, our method can still obtain better results than other competing methods. For example, our method with 50 views outperforms WBGM with 70 views by 4.0%, 1.3%, 0.2%, 3.0%, 2.4%, 1.7% under six evaluation criteria.

4.5 Performance with different neighbor number

In this subsection, we explore the influence of the neighbor samples on three datasets. As shown in Fig. 5, the neighbor number is varied from 5 to 30 with the step size of 5. From the Figs. 5a, b and c, we can find that the increasing the neighbor number can improve the performance on all the datasets. With more neighbor number, the more contextual information is used during the process of similarity measure. Therefor, the performance is increased with increasing neighbor number. If the neighbor number is bigger than the optimal one, the

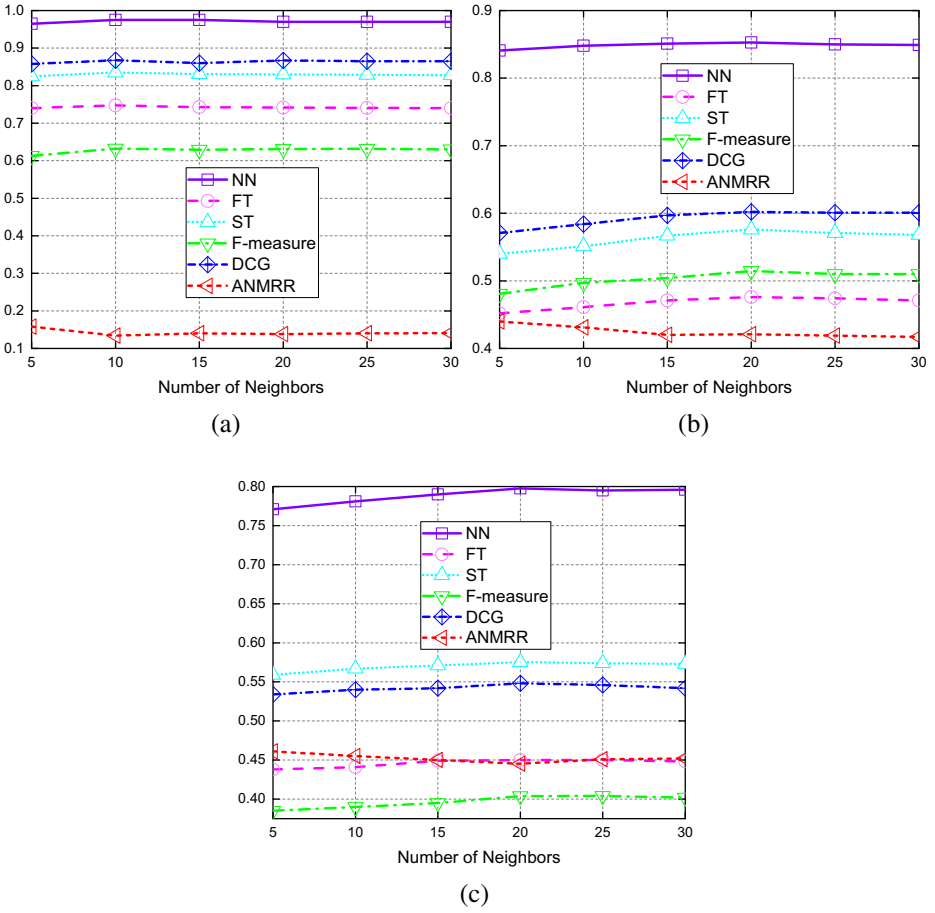

Fig. 5 Performance comparison with different neighbor numbers on (**a**) ETH, (**b**) MVRED and (**c**) NTU

performance is decreased because there exist more noise samples when using the neighbor information. Moreover, when the number of neighbor samples is bigger or smaller than the optimal value, the small change in performance proves the robustness of our method to this value. The best results are achieved by setting the neighbor number k to 10, 20 and 20 for ETH, MVRED and NTU dataset, respectively.

5 Conclusion

In this paper, we proposed a novel method to address the problem of rankling list optimization by incorporating the graph generation and the graph fusion via the investigation on the relation between models. In graph generation stage, visual similarity, which is defined as the visual information of all 12 views of the query model, is combined with the contextual similarity between the query and all models in the gallery pool for generating graph. In graph fusion stage, the similar samples are further subdivided into the strong similar set and the quasi-similar set based on the mining of top-k results. Moreover, in order to explore the

hidden information of the ranking list, the ordinary samples are leveraged to offer the ordinary set while the dissimilar set are mined according to the bottom-k dissimilar samples. All samples will be reused for re-query to yield their new ranking list. In the final stage, the ultimate list of the initial query model will be generated via aggregation. Compared to the existing methods, the experimental results show the superiority of our proposed method on three challenging 3D model datasets.

Acknowledgments 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).

References

1. Ankerst M, Kastenmüller G, Kriegel H-P, Seidl T (1999) 3d shape histograms for similarity search and classification in spatial databases. In: Advances in Spatial Databases, 6th International Symposium, pp 207–226
2. Ansary TF, Daoudi M, Vandeborre J-P (2007) A bayesian 3-d search engine using adaptive views clustering. IEEE Trans Multimed 9(1):78–88
3. Chen D-Y, Tian X-P, Shen Y-T, Ouhyoung M (2003) On visual similarity based 3d model retrieval. Comput Graph Forum 22:223–232
4. Feng Y, Zhang Z, Zhao X, Ji R, Gao Y (2018) GVCNN: group-view convolutional neural networks for 3d shape recognition. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, pages 264–272. IEEE Computer Society
5. Gao Y, Dai Q, Wang M, Zhang N (2011) 3d model retrieval using weighted bipartite graph matching. Sig Proc Image Comm 26(1):39–47
6. Gao Y, Tang J, Hong R, Yan S, Dai Q, Zhang N, Chua T-S (2012) Camera constraint-free view-based 3-d object retrieval. IEEE Trans Image Process 21(4):2269–2281
7. Gao Y, Dai Q (2014) View-based 3d object retrieval: Challenges and approaches. IEEE MultiMedia 21(3):52–57
8. Gao Y, Wang M, Ji R, Wu X, Dai Q (2014) 3-d object retrieval with hausdorff distance learning. IEEE Trans Ind Electron 61(4):2088–2098
9. Giorgi D, Mortara M, Spagnuolo M (2010) 3d shape retrieval based on best view selection. In: Proceedings of the ACM workshop on 3D object retrieval, 3DOR. ACM, pp 9–14
10. Guo H, Wang J, Gao Y, Li J, Lu H (2016) Multi-view 3d object retrieval with deep embedding network 25:5526–5537
11. He X, Zhou Y, Zhou Z, Bai S, Bai X (2018) Triplet-center loss for multi-view 3d object retrieval. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pages 1945–1954. IEEE Computer Society
12. Hilaga M, Shinagawa Y, Komura T, Kunii TL (2001) Topology matching for fully automatic similarity estimation of 3d shapes. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp 203–212
13. Hong C, Yu J, You J, Chen X, Tao D (2015) Multi-view ensemble manifold regularization for 3d object recognition. Inf Sci 320:395–405
14. Ip CY, Lapadat D, Sieger L, Regli WC (2002) Using shape distributions to compare solid models. In: Seventh ACM Symposium on Solid Modeling and Applications. ACM, pp 273–280
15. Kim W-Y, Kim Y-S (2000) A region-based shape descriptor using zernike moments. Proc Sig Image Comm 16(1-2):95–102
16. Khotanzad A, Hong YH (1990) Invariant image recognition by zernike moments. IEEE Trans Pattern Anal Mach Intell 12(5):489–497
17. Krizhevsky A, Sutskever I, Hinton GE. (2012) Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp 1106–1114
18. Leibe B, Schiele B (2003) Analyzing appearance and contour based methods for object categorization. In: 2003 IEEE, Computer Society Conference on Computer Vision and Pattern Recognition, pp 409–415
19. Liu A, Nie W, Gao Y, Su Y (2016) Multi-modal clique-graph matching for view-based 3d model retrieval. IEEE Trans Image Process 5:2103–2116

20. Liu A, Nie W, Gao Y, Su Y (2017) 3-d model retrieval: View-based A benchmark. IEEE Trans Cybern PP(99):1–13
21. Liu A, Nie W, Gao Y, Su Y (2017) View-based 3-d model retrieval: A benchmark. IEEE Transactions on Cybernetics
22. Lu K, Wang Q, Xue J, Pan W (2014) 3d model retrieval and classification by semi-supervised learning with content-based similarity. Inf Sci 281:703–713
23. Mủller H, Mủller W, Squire D, Marchand-maillet S, Pun T (2001) Performance evaluation in content-based image retrieval: overview and proposals. Pattern Recogn Lett 22(5):593–601
24. Nie W, Liu A, Gao Zx, Su Y (2015) Clique-graph matching by preserving global & local structure. In: IEEE, Conference on Computer Vision and Pattern Recognition, pp 4503–4510
25. Nie W, Xiang S, Liu A (2018) Multi-scale cnns for 3d model retrieval. Multimed Tools Appl 77(17):22953–22963
26. Persoon E, Fu K-S (1977) Shape discrimination using fourier descriptors. IEEE Trans Syst Man Cybern 7(3):170–179
27. Qi CR, Yi L, Su H, Guibas LJ (2017) Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, pp 5099–5108
28. Su H, Maji S, Kalogerakis E, Learned-Miller EG. (2015) Multi-view convolutional neural networks for 3d shape recognition. In: IEEE, International Conference on Computer Vision, pp 945–953
29. Wang M, Gao Y, Lu K, Rui Y (2013) View-based discriminative probabilistic modeling for 3d object retrieval and recognition. IEEE Trans Image Process 22(4):1395–1407
30. Wang X, Nie W (2015) 3d model retrieval with weighted locality-constrained group sparse coding. Neurocomputing 151:620–625
31. Wang D, Wang B, Zhao S, Yao H, Liu H (2017) View-based 3d object retrieval with discriminative views. Neurocomputing 252:58–66
32. Yang L, Albregtsen F (1996) Fast and exact computation of cartesian geometric moments using discrete green's theorem. Pattern Recogn 29(7):1061–1073
33. Yi L, Wang X, Wang H-y, Zha H, Qin H (2010) Learning robust similarity measures for 3d partial shape retrieval. Int J Comput Vis 89(2-3):408–431
34. Yi F, Xie J, Dai G, Wang M, Zhu F, Xu T, Wong EK (2015) 3d deep shape descriptor. In: IEEE, Conference on Computer Vision and Pattern Recognition, pp 2319–2328
35. Zhang D, Lu G (2002) Generic fourier descriptor for shape-based image retrieval. In: Proceedings of the 2002 IEEE, International Conference on Multimedia and Expo, pp 425–428
36. Zhao S, Yao H, Zhang Y, Wang Y, Liu S (2015) View-based 3d object retrieval via multi-modal graph learning. Signal Process 112:110–118
37. Zhu Z, Wang X, Bai S, Yao C, Bai X (2016) Deep learning representation using autoencoder for 3d shape retrieval. Neurocomputing 204:41–50

Wen-Hui Li received the M.S. and PH.D degrees in the School of Electrical and Information Engineering, Tianjin University. He was an intern student with the SeSaMe Center of National University of Singapore. His research interests are in the field of computer vision, machine learning, and 3D model retrieval.

Yu-Ting Su received the M.S. and PH.D degrees in electronic engineering from Tianjin University of China. His research interests include multiple object tracking, computer vision, location-based social network, and 3D model retrieval.

Zhen-Lan Zhao is currently a Master student at the School of Electrical and Information Engineering, Tianjin University. His research interests are in the field of computer vision, unsupervised learning, and 3D model retrieval.

Tong Hao is with the College of Life Sciences, Tianjin Normal University, Tianjin 300387, China. Her research interests include bioinformatics, machine learning, multimedia analysis.

Yang-Yang Li received the B.S. degree from the Nanjing University of Information and Technology, in 2009, and the Ph.D. degree in computer science from the Beijing University of Posts and Telecommunications, in 2015. He is currently a Senior Engineer at the National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data (PSRPC). His research interests include the mobile internet, social networks, and edge computing.