IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 16, 2019, accepted July 2, 2019, date of publication July 23, 2019, date of current version August 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2929109

Unsupervised Feature Learning With Graph Embedding for View-Based 3D Model Retrieval

YUTING SU, WENHUI LI, WEIZHI NIE, DAN SONG, AND AN-AN LIU
School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

Corresponding authors: Dan Song (dan.song@tju.edu.cn) and An-An Liu (anan0422@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772359, Grant 61572356, and Grant 61872267, in part by the Tianjin New Generation Artificial Intelligence Major Program under Grant 17ZXRGGX00180 and Grant 18ZXZNGX00150, and in part by the Tianjin University through the Elite Scholar Program under Grant 2019XRX-0035.

ABSTRACT 3D model retrieval is becoming a hot research topic due to its wide applications such as computer-aided design, digital entertainment, and virtual reality. For this challenging task, feature learning and similarity measure are two critical problems. However, existing approaches usually learn discriminative visual features and develop a complex graph matching strategy to measure the similarity independently. In this paper, we propose an unsupervised method which can embed similarity measure into the feature space. The proposed method utilizes both similarity and dissimilarity information to better leverage the unsupervised problem and estimates the labels which are further used for metric learning. With the learned metric, we project the original features to more discriminative feature space and efficiently measure the similarity among models under the new feature space. We conduct extensive evaluations of three popular and challenging datasets. The experimental results demonstrate the superiority and effectiveness of the proposed method, competing against the state of the arts.

INDEX TERMS 3D model retrieval, unsupervised learning, metric learning.

I. INTRODUCTION

With the development of 3D model acquisition and printing technology, there is an explosive growth of 3D models. Due to the huge and ever-increasing 3D data, advanced pattern recognition techniques are becoming fundamental to process these data for many practical problems, for example, digital entertainment, CAD, medical diagnosis and 3D scene understanding [1]–[3]. Due to the success of the 2D image/video retrieval task [4]–[8], 3D model retrieval has attracted more attention and multiple approaches for this task have been developed [9]–[13].

A. MOTIVATION AND OVERVIEW

Given a query model, 3D model retrieval aims to find the relevant models from the 3D model dataset. The existing works on 3D model retrieval can be roughly grouped into two paradigms, model-based methods and view-based methods. In model-based methods, each 3D model is represented by the volume or the point set. These methods mainly extract the graphical features, such as surface distributions [14], voxel-based features [15], shape descriptors [16] and Fourier descriptors [17], to represent the 3D model, which can preserve the spatial structure and geometry information of 3D model. The limitation of these methods is that the performance is seriously restricted by the low-quality of models and expensive computation. Furthermore, it is very difficult to represent the model with these methods when only visual appearance of the model is available.

Recently, extensive works have been done on view-based 3D model retrieval. Benefiting from the deep neural networks, a lot of works [18]–[21] utilize the deep structures to describe the multi-view characteristics of 3D models. In particular, Multi-View Convolutional Neural Networks [18] employed the max-pooling operation for multiple views to generate the model level descriptor. To explore the correlation of multiple views, Feng *et al.* [20] utilized the group based module to exploit the group level descriptors. Wang *et al.* [19] recurrently clustered the views into different sets according to

The associate editor coordinating the review of this manuscript and approving it for publication was Shuhan Shen.

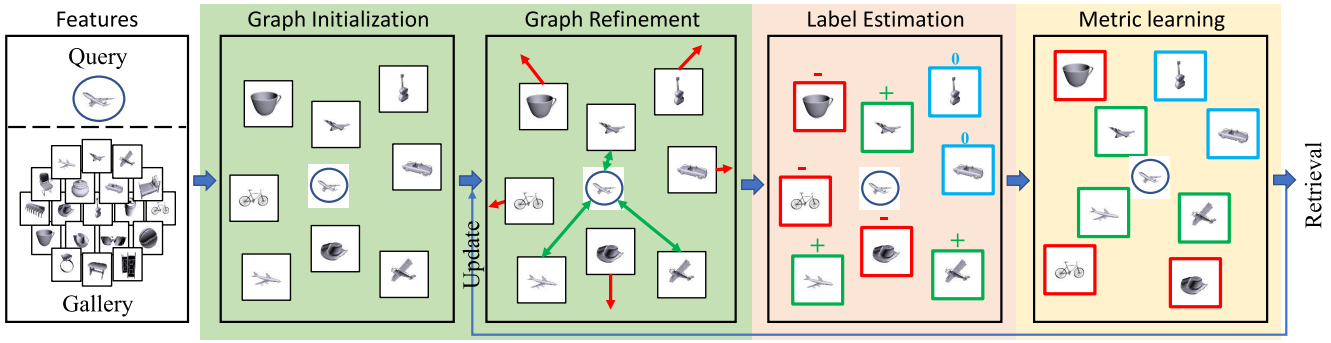


FIGURE 1. Framework of our proposed method. Given the features of query and gallery, we first construct the graph including graph initialization and graph refinement. Graph initialization measures the similarities among models in original feature space and graph refinement utilizes the k-reciprocal information to pull the similar galleries and push the dissimilar galleries for the query. Then, the label estimation procedure uses the similar neighbors and dissimilar neighbors to estimate the positive and negative labels. The metric learning utilizes the label information and the features to learn the projected metric. This metric projects the original features into a new space, which contribute to further refinement. The procedures of graph refinement, label estimation, and metric learning repeat until obtaining stable results.

the similarity of the views and pooled the features in each set to learn the representation of models. These methods highly depend on a large number of labeled samples to ensure the model can learn useful patterns rather than overfitting the data. However, large-scale labeled 3D data are always difficult to obtain for most real applications.

To alleviate the aforementioned problem, multiple researchers have been engaged into the unsupervised methods for this task [22]–[27]. These approaches mainly focus on the similarity measure which can be grouped into distance-based methods [23], statistic-based methods [22], [25] and graph-based methods [24], [26], [27]. The distance-based methods usually directly measure the set-to-set distance in the original feature space, which may reduce the time computation while resulting in low performances. The statistic-based methods learn the statistical model for view representation, which can improve the performance compared against the distance-based methods. The graph-based methods usually formulate the multiple views in a graph structure, where each view is treated as the vertex of the graph. They explicitly leverage the multi-view information of 3D models to solve the many-to-many distance measure, which can enrich model representation and enhance the similarity measure while resulting in high computational complexity.

In light of the above discussions, we develop the unsupervised feature learning with graph embedding (FLGE) for view-based 3D model retrieval. We aim to embed the similarity measure information into the feature space to learn more discriminative features and reduce the complexity of similarity measure simultaneously. Specifically, we utilize the view-level descriptor to initialize the graph and combine the visual similarity and contextual information to refine the graph. Subsequently, we develop a novel algorithm to estimate the label in an unsupervised manner and use the estimated labels to construct the metric learning. After learning the projected metric, we can update the features and conduct graph again. This procedure can be implemented iteratively to improve the projected metric. Finally, we utilize the projected features to measure the similarities among the query model and the gallery models and get the final retrieval results. Our framework is summarized in Fig. 1.

B. CONTRIBUTIONS

The main contributions of this paper are summarized as follows:

- This paper proposes a novel framework to embed the graph information of similarity measure into the feature space, which can enhance the distinctiveness of features from different categories.
- We propose an original label estimation algorithm to estimate the relative labels in an unsupervised manner. The proposed algorithm adopts both similarity and dissimilarity information to estimate the positive and negative labels for metric learning.
- We conduct extensive experiments on three popular 3D model datasets. The experimental results demonstrate the superiority of this method compared against the state of the arts.

The remainder of this paper is organized as follows. In Section II, we introduce the related works on unsupervised view-based 3D model retrieval. Section III presents the proposed approach in details. Experimental results are introduced in Section IV. Finally, we conclude the paper in Section V.

II. RELATED WORK

Unsupervised learning has been widely studied in the machine learning community. For model retrieval, the ways for measuring the similarity between models in an unsupervised manner can be classified into three categories, which are distance-based methods, statistic-based methods and graph matching-based methods.

A. DISTANCE-BASED METHODS

For 3D model retrieval, distance-based methods usually first select the representative view of each model for model

representation and then do set-to-set similarity measure. They directly measure the similarity between pairwise models in terms of specific distance metrics [28]. The Euclidean distance [29] and the Hausdorff distance (HAUS) [30] are two commonly used metrics for 3D model retrieval. For example, Gao *et al.* [31] proposed to learn a view-level Mahalanobis distance to estimate the HAUS between pairwise models. Specifically, the two commonly-used distances are defined as follows:

- Nearest Neighbor (NN) [23]: This NN-based method leverages the minimal distance of all view pairs across two models for similarity measure. It can be calculated as follows:

$$NN\ (V_1, V_2) = \min_{v \in V_1, u \in V_2} d(v, u) \tag{1}$$

- Hausdorff Distance (HAUS) [23]: The distance between characteristic views from one model and the closest view from the other model is determined and the HUAS distance is calculated as the maximum of all distances from one characteristic view of one model to the closest view in the other model, which is formulated as follows:

$$HAUS\ (V_1, V_2) = \max \left\{ \begin{matrix} \max_{v \in V_1} \left\{ \min_{u \in V_2} d(v, u) \right\} \\ \max_{v \in V_2} \left\{ \min_{u \in V_1} d(v, u) \right\} \end{matrix} \right\} \tag{2}$$

  where V_1 and V_2 are the characteristic view sets of the two compared models and the $d(v, u)$ is the Euclidean distance between two views.

The benefit of the distance-based methods is that it is very efficient to solve the retrieval problem. However, since the existing distance-based methods usually computed the distance between two models in the original feature space, which can not discover the latent context to enhance the similarity measure.

B. STATISTIC-BASED METHODS

This kind of methods learns a statistical model for each category of 3D models to infer the comparability between the query model and individual category.

- Adaptive view clustering (AVC) [22] 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 View (CCFV) [25] removes the constraint of the setting of the static camera array for view capture. All the views of one model are first grouped to generate view clusters and then the positive matching and negative matching models are trained using positive and negative matched samples.

Generally speaking, the statistic-based methods consider the feature distribution of multi-view images and utilize the statistical models for 3D model learning and inference to augment the discrimination. Compared against the distance-based methods, these methods get superior performance while spending more time on characteristic views selection by view clustering.

C. GRAPH-BASED METHODS

Recently, graph matching was widely leveraged for this task because the multi-view image set of a 3D model conveys the spatial context, which will benefit 3D model retrieval. Different from the aforementioned methods, graph matching-based methods explicitly leverage multi-view information to solve the many-to-many distance measure. Essentially, it aims to construct a graph structure by discovering the salient features of individual view images and/or the visual/spatial relationship between pairwise/high-order view images. Gao *et al.* [32] addressed this task by constructing multiple hyper-graphs for a set of 3D models based on 2D views. This method can explore the higher-order relationship among 3D models. A graph-based characteristic view set extraction and matching method for 3D model retrieval was proposed in [33]. They used the graph clustering method for view grouping and the random-walk algorithm was applied for constructing a view-graph model. In [34], different features were treated as different models to generate multiple graphs and they addressed the feature fusion task by learning the optimized weights of each graph. Zhang *et al.* [35] treated the distortions of view-specific samples as nonlinear noisy mappings of an intact representation of the same object in a latent space, and took into account the non-linearity in estimating the unconformity of view-specific measurements of dissimilarity or similarity. Generally speaking, there are three representative methods:

- Weighted Bipartite Graph Matching (WBGM) [24] first selected the characteristic view of each model by grouping its views into clusters, and used the clusters as the representative view. Each representative view is provided with an initial weight based on the appearance of the selected view and the initial weights are further updated based on the relationship among these representative views. Two groups of views are formulated as two subsets of the weighted bipartite graph and the Hungarian algorithm can be implemented to achieve the optimal matching and similarity measure between pairwise 3D models.
- Multi-Modal Clique-Graph Matching (MCG) [26] replaced individual nodes of the classic graph by one clique, which consists of K nearest neighbors in the specific feature subspace and can convey the local structural attributes in a star model. In the graph, the hyper-edges that link pairwise cliques and an image-set-based clique/edge-wise similarity measure are proposed to address the issue of the set-to-set distance measure, which can preserve the structure characteristics of the graph.
- Hierarchical Graph Structure Learning (HGS) [27] proposed an unsupervised hierarchical graph structure learning method for multi-view 3D model retrieval. It designed two strategies to construct a single-view graph and decomposed the complicated multi-view graph-based similarity measure into multiple single-view graph-based similarity measures to avoid the

difficulty in definition and computation by using the hierarchical structure.

The graph-based methods always perform better with the ability of leveraging the multi-view information and the latent context for 3D model. However, these methods are also time-consuming because of constructing the graph and processing the graph matching procedure.

III. PROPOSED APPROACH

In this section, we first overview the framework of FLGE and then the details of each step will be illustrated.

A. OVERVIEW

Given a probe 3D model p and the gallery set with N 3D models $G = \{g_i | i = 1, 2, \ldots, N\}$, the original distance between two model p and g_i can be measured by Mahalanobis distance,

$$d_M(p, g_i) = (v_p - v_{g_i})^T M (v_p - v_{g_i}) \tag{3}$$

where v_p and v_{g_i} represent the appearance feature of p and g_i, respectively, and M is a positive semi-definite matrix, which aims to enhance the feature representation of 3D model and consequently improves the retrieval performance. In this paper, we propose a novel framework to learn the projected metric, which consists of three modules as follows:

- **Graph Construction.** This module aims to discover the correlation among multiple 3D models and construct the rank graph of the 3D models in the dataset. It contains two parts, graph initialization and graph refinement.
- **Label Estimation.** This module uses the constructed rank graph to estimate labels. Differing from existing methods, which learn the metric with the ground truth labels, we propose to estimate the positive and negative labels in an unsupervised manner.
- **Metric Learning.** This module utilizes the estimated labels and adopts the metric learning strategy to learn the projected metric. Subsequently, we can use the learnt metric to update the rank graph construction, which can iteratively improve the metric further.

B. GRAPH CONSTRUCTION

1) GRAPH INITIALIZATION

Each 3D model is represented by a set of multi-view 2D images. Given the features of paired 3D models, V_i and V_j, we utilize the set-to-set distance to measure the similarity of pairwise models and utilize the similarities to construct the rank graph. Unlike those methods that equally treat all views of each model, we adopt the regularized affine hull (RAH) [36] to reduce the impact of noise views and suppress unnecessary components for the final model representation. For model V_i, its representation with RAH is defined as following:

$$V_i^R = \left\{ \sum_{j=1}^{s} \beta_j v_{i,j} | \sum_{j=1}^{s} \beta_j = 1, \|\beta\|_{l_2} \leq \delta \right\} \tag{4}$$

where s is the view number and $\|\cdot\|_{l_2}$ is the l_2 norm. Equation (4) transforms the original set of view-level features to a single feature vector with the learnt coefficients. The distance between two models under this feature space is $D_R = d_M(V_i^R, V_j^R)$. The final distance between model V_i and model V_j is formulated in a log-logistic form:

$$D(V_i, V_j) = log(1 + e^{D_R}) \tag{5}$$

Subsequently, according to (5), we can sort the distances for each model and obtain the initial rank graph of the model p, which is denoted as $R^0(p, G) = \{g_1, g_2, \ldots, g_N\}$, where $d_M(p, g_i) < d_M(p, g_{i+1})$.

2) GRAPH REFINEMENT

Only using visual features to conduct graph may restrict the overall performance, as each 3D model could differ significantly from other models even belonging to the same class. Inspired by the works [37], [38], the contextual information of neighbors can enhance the similarity measure and benefit verifying the rank graph. In this part, we utilize the information of k-reciprocal nearest neighbors $R(p, k_1)$ to refine the graph, which can be defined as,

$$R^1(p, k_1) = \left\{ g_i | (g_i \in R^0(p, k_1)) \wedge (p \in R^0(g_i, k_1)) \right\} \tag{6}$$

where R^0 is the initialized rank graph. As the k-reciprocal neighbors are from the k-nearest neighbors and due to the variations in poses and views, the positive samples may be out of the k-nearest neighbors. To enhance the neighbors, we incrementally add the k_1 additional neighbors of each candidate in $R^1(p, k_1)$ into a more robust set $R^2(p, k_1)$:

$$\begin{aligned} R^2(p, k_1) &\leftarrow R^1(p, k_1) \cup R^1(q, \frac{1}{2}k_1), \\ s.t. \left| R^1(p, k_1) \cap R^1(q, \frac{1}{2}k_1) \right| &\geq \frac{1}{2} \left| R^1(q, \frac{1}{2}k_1) \right|, \\ &\forall q \in R^1(p, k_1) \end{aligned} \tag{7}$$

Then, we consider $R^2(p, k_1)$ as contextual knowledge to re-calculate the distance between p and g_i. If two models are similar, their k-reciprocal nearest neighbor sets overlap with each other, i.e. there are some shared samples in the sets. More shared samples, more similar the two models are. The new distance between p and g_i can be calculated by the Jaccard metric of their k-reciprocal sets as $d_J(p, g_i) = 1 - \frac{|R^2(p,k_1) \cap R^2(g_i,k_1)|}{|R^2(p,k_1) \cup R^2(g_i,k_1)|}$, where $|\cdot|$ denotes the number of candidates in the set and we adopt Jaccard distance to recalculate the similarity between p and g_i. Subsequently, we use $d_J(\cdot, \cdot)$ to obtain the final rank graph R^*. We evaluate this parameter k_1 in the Section IV.

C. LABEL ESTIMATION

In this subsection, we discover the pair-label information between the query model and its rank graph by using the k-reciprocal information of the query model. We denote

the k_2^+ as the top-k neighbors and k_2^- as the bottom-k samples. To simplify the parameter setting, we define $|k_2^+| = |k_2^-| = k_2$.

1) POSITIVE LABEL ESTIMATION

In this part, we utilize the neighbor information of the query model to estimate whether the pairwise models belong to the same class. Generally speaking, if neighbors for each query model are accurate, which means the top neighbors are from the same categories, we can directly use the top-k neighbors of the query as the positive samples:

$$y_+(p,q) = \begin{cases} 1, & \text{if } q \in R^*(p, k_2^+) \\ 0, & \text{others} \end{cases} \tag{8}$$

or define a threshold θ to select confident positive samples:

$$y_+(p,q) = \begin{cases} 1, & \text{if } D(p,q) < \theta \\ 0, & \text{others} \end{cases} \tag{9}$$

Actually, there always exist several false positive results in top-k neighbors for query model. If only selecting the top k samples as positive, it will result in too many false positive samples with big k and few positive samples with small k. Meanwhile, it is difficult to define a suitable global threshold to meet all query samples, if we used threshold θ to select samples. To suppress the negative samples with both top-k neighbors and threshold, we introduce a data-driven mechanism to automatically define the threshold to select the positive samples in the top-k neighbors. We use both visual similarity R^0 and contextual information R^* to estimate positive labels. Furthermore, it will be unreasonable if we treat all positive labels equally, *i.e.* $y_+(p,q) = 1$. Therefore, we design a soft label with the Gaussian kernel for the positive labels. Specifically, we modify (8) and (9) as follows,

$$y_+(p,q) = \begin{cases} e^{-D(p,q)}, & \text{if } q \in R^0(p, k_2^+) \cap R^*(p, k_2^+), \\ & \quad D(p,q) < \theta \\ 0, & \text{others} \end{cases} \tag{10}$$

where $\theta = \frac{1}{|k_2^+|} \sum_{i=1}^{|k_2^+|} d_M(p, g_i)$. This setting aims to select positive samples as many as possible and limiting the hard negative samples with the adaptive threshold.

2) NEGATIVE LABEL ESTIMATION

In this part, we describe how to estimate the negative pairs. According to the previous description, the similar models have similar top-k neighbors. Intuitively, if p is similar to q, the bottom-k samples of q dissimilar to the p. If we only use the bottom-k samples of the query model as she negative samples, it will result in easy negative labels because they are far away from the query sample and may have less contribution for metric learning to distinguish the hard negative samples. According to this assumption, we estimate the negative label by using both bottom-k information of the query itself and the bottom-k samples of the similar model g_i for query, which is defined as:

$$y_-(p,q) = \begin{cases} -1, & \text{if } q \in R^*(p, k_2^-) \cup R^*(g_i, k_2^-) \\ 0, & \text{others} \end{cases} \tag{11}$$

where $g_i \in R^*(p, k_2^+)$.

D. METRIC LEARNING

Given the estimated positive pairs and negative pairs in the above subsection, we could design the loss function to learn the discriminative metric and enhance the retrieval task as many supervised works do. Specifically, the log loss function can be designed as follows:

$$J_M(\bar{v}_i, \bar{v}_j) = log(1 + e^{y_{ij}(D_M(\bar{v}_i, \bar{v}_j) - \mu)}) \tag{12}$$

where μ is a constant positive bias and is the average distance between all sample pairs to consider that D_M has a lower bound of zero. y_{ij} is the label value for sample i and j. Under the matrix M, the D_M represents the distance between V_i and V_j, which is denoted by $D_M(\bar{v}_i, \bar{v}_j) = (\bar{v}_i - \bar{v}_j)^T M (\bar{v}_i - \bar{v}_j)$. To simplify the similarity computation, we adopt the first-order statistics $\bar{v}_i$ of model V_i and $\bar{v}_j$ of model V_j, which show the averaged position of the sample set in the high dimensional space, to represent each view set and use them for metric learning.

The logistic function provides a soft margin to separate the two classes, we can obtain the probabilistic metric learning problem by:

$$\min_M E(M) = \min_M \sum_{i=1}^{n} \sum_{j=1}^{m} \omega_{ij} J_M(\bar{v}_i, \bar{v}_j), \quad s.t.\ M \succeq 0. \tag{13}$$

where ω_{ij} is the parameter to handle the imbalanced positive and negative pairs, which is defined by $\frac{1}{N_{pos}}$, if $y(i,j) > 0$ and $\frac{1}{N_{neg}}$, if $y(i,j) < 0$. The N_{pos} and N_{neg} are the number of positive and negative sample pairs. Subsequently, we can use the existing accelerated proximal gradient algorithms [39]–[41] to solve (13) and get the optimal M. As shown in [39], we are able to decompose M as $M = PP^T$. In this way, P is a projection metric and can translate the Mahalanobis distance between model p and model g_i (defined in (3)) to Euclidean distance as follows,

$$d_P(p, g_i) = \left\| P^T v_p - P^T v_{g_i} \right\|_2^2 \tag{14}$$

After learning P, we can recalculate the similarity measure between two models by using Eq. 14 and update the graph construction to select high-confidence pairs. Subsequently, we could utilize the pairs to learn new P to update the graph. By iteratively repeating the whole procedure, the updated rank graph could produce more reliable results, and the previous learnt metric could be further improved. Finally, a stable rank graph and a discriminative distance metric can be achieved after a few iterations.

FIGURE 2. Samples with different views from (a) NTU, (b) MVRED, and (c) ETH.

IV. EXPERIMENT

In this section, we detail the information about the datasets, evaluation criteria, experimental settings and the competing methods. Finally, we illustrate the experimental results and discussion.

A. DATASET

Three popular 3D model datasets are utilized for evaluation, including **NTU** [42], **MV-RED** [2] and **ETH** [43], some samples of which are shown in Fig. 2. The three datasets are briefly introduced as follows:

- **NTU** [42]: The NTU dataset contains 549 models of 47 categories from the World Wide Web pages which are all free downloaded via the Internet. All of the models were converted into Wavefront file format and saved as Obj document format and each object includes 60 different view samples.
- **MV-RED** [2]: The MV-RED consists of 505 objects from 60 categories. Each object was recorded simultaneously by three cameras from three directions. For data acquisition, Camera-45 and Camera-60 captured 36 RGB images every 10 degree by uniformly rotating the table controlled by a step motor respectively. One RGB image in the top-down view is captured by Camera-90. Therefore, each object owns 73 images.
- **ETH** [43]: The ETH dataset contains 8 categories with 80 objects. Each object has 41 different views spaced evenly over the upper viewing hemisphere, and all the positions for cameras are determined by subdividing the faces of an octahedron to the third recursion level.

B. EVALUATION CRITERIA

For the evaluation of each dataset, each 3D model is selected once as the query for retrieval. To evaluate the performance of 3D model retrieval, we employ seven popular criteria, including AUC, NN, FT, ST, F-Measure, DCG and ANMRR.

- Precision-Recall Curve [44] is able to comprehensively demonstrate the retrieval performance, which illustrates the precision and recall measures by changing the threshold for distinguishing relevance and irrelevance in model retrieval. The area under curve (AUC) of PR-curve can be calculated for quantitative evaluation.
- Nearest Neighbor (NN) is defined to evaluates the retrieval accuracy of the nearest neighbor returned result.
- First Tier (FT) is used to compute the recall of the top κ results, where κ is the number of the most relevant objects for the query.
- Second Tier (ST) is defined as the recall of the top 2κ results.
- F-measure (F) jointly evaluates the precision and the recall of top relevant results. It considers the top 20 returned results for each query.
- Discounted Cumulative Gain (DCG) [22] discounts the value of relevant results according to their ranked position, which assigns relevant results at the top ranking positions with higher weights because of the supposition that the user considers lower results less.
- Average Normalized Modified Retrieval Rank (ANMRR) [45] evaluates the ranking performance by considering the ranking order and uses the ranking information of relevant objects among the retrieved objects to measure the retrieval result. The lower ANMRR value indicates the better performance.

C. EXPERIMENT SETTING AND COMPETING METHODS

For feature representation of individual view image, we adopted the AlexNet model [46], which was pre-trained on the ImageNet dataset, to extract the visual feature. All view images were first resized to 256x256. We utilized the output of the second last fully-connected layers as the visual representation, which generated a 4096 dimensional vector for each view. Totally, seven baseline methods (including two distance-based method, Nearest Neighbor (NN) [23] and Hausdorff Distance (HAUS) [23], and two statistical-based methods, Adaptive view clustering (AVC) [22] and Camera

Constraint Free View (CCFV) [25], and three graph-based methods, Weighted Bipartite Graph Matching (WBGM) [24], Multi-Modal Clique-Graph Matching (MCG) [26] and Hierarchical Graph Structure Learning (HGS) [27] were implemented for comparison. The competing methods are addressed in the section of related work.

D. EXPERIMENTAL RESULTS AND DISCUSSION

Extensive experiments were conducted to evaluate the effectiveness of the proposed method on NTU, MVRED and ETH. We first analyze the sensitivity of our method to four important hyper-parameters, *i.e.*, the view number s in (4), the neighbor number k_1 for graph refinement, k_2 for label estimation, and the iteration number T. By default, we vary the value of one parameter and keep the others fixed. Then, we evaluate the performance with different modules in our framework. Furthermore, we visualize the visual feature before and after our method. Finally, we compare our method against the state-of-the-arts.

1) PARAMETER ANALYSIS

a: SENSITIVITY ANALYSIS ON VIEW NUMBER

For most real applications, it is always expected that 3D model retrieval is conducted with as few view images as possible. Therefore, we evaluate the retrieval performance by varying the number of views used on MVRED, which is most challenging 3D dataset for real applications. To further verify the robustness of our method, we compare it with other representative methods. Specially, we tune the view number from 10 to 70 with the step size of 10. We averaged 10 random trials with respect to the specific view number. From the comparison results shown in Fig. 3, we have following observations:

- All methods can get consistent improvements by increasing the view numbers, which is reasonable since more views can convey more appearances and structural characteristics of 3D models. We use all views of each 3D model in the remaining experiments.
- Our method can consistently outperform the competing methods in terms of all evaluation criteria. When increasing the view number from 10 to 70, our method can achieve the gain of 12.1%, 6.1%, 14.8%, 6.7%, 4.9% in terms of AUC, FT, ST, F-measure, DCG and the decline of 4.6% in terms of ANMRR. In particular, our method with 40 views can outperform the second best method with the gain of 6.0%, 5.5%, 3.8%, 4.4%, 3.6% in terms of AUC, FT, ST, F-measure, DCG and the decline of 5.7% in terms of ANMRR, respectively.

b: SENSITIVITY ANALYSIS ON NEIGHBOR NUMBER

The impacts of the neighbor numbers k_1 and k_2 were evaluated on the MVRED dataset. The results are shown in Fig. 4. We tune k_1 and k_2 from 5 to 30. We empirically set k_2 to 20 when we varied k_1 and then tuned k_2 by fixing k_1 with the optimal value. As shown in Fig. 4 (a), the performance is improved by increasing k_1 and the best result is obtained

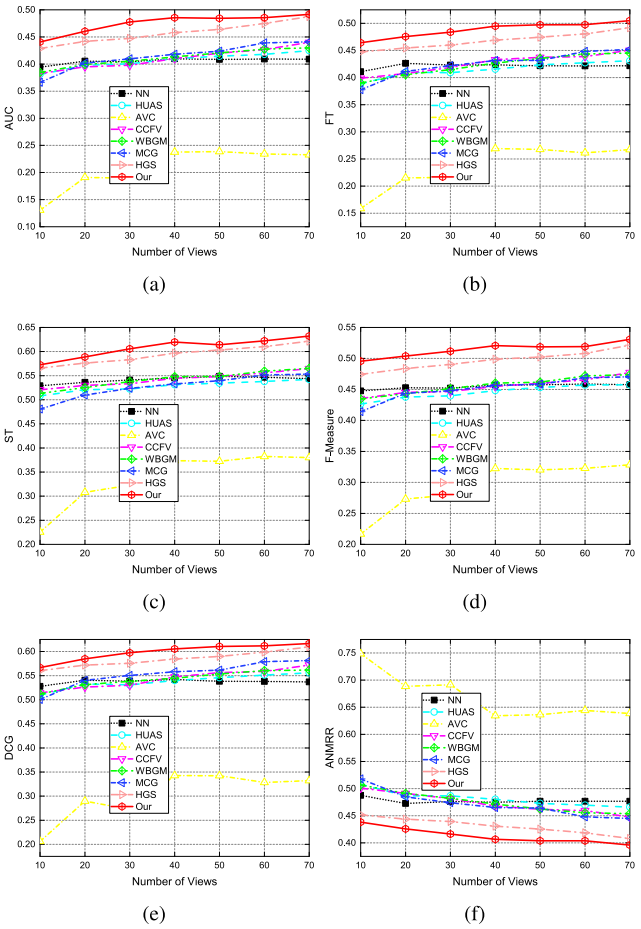


FIGURE 3. Comparison by varying view numbers on MVRED. (a) AUC, (b) FT, (c) ST, (d) F-Measure, (e) DCG, and (f) ANMRR.

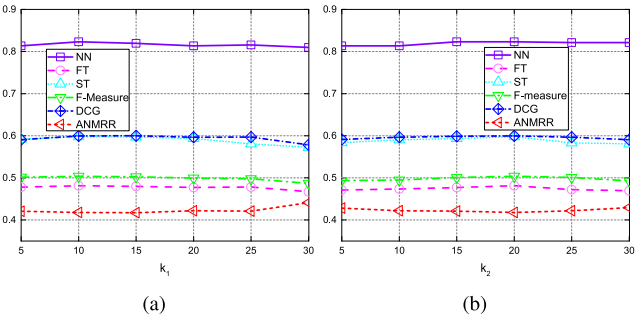


FIGURE 4. Performance by varying the neighbor numbers k_1 (a) and k_2 (b) on MVRED.

when $k_1 = 10$. We observed that the performance decreased when assigning a large value to k_1 after the peak arrived. Since there will be more negative samples in the neighbor set, too many neighbors will have negative influence on similarity measure. The similar observation can be found for k_2, as illustrated in Fig. 4 (b). According to the above observation, the best results can be obtained by setting $k_1 = 10$ and $k_2 = 20$. In all experiments, we used 10 as the neighbor size to construct the rank graph and used 20 as the neighbor size to predict the positive and negative labels.

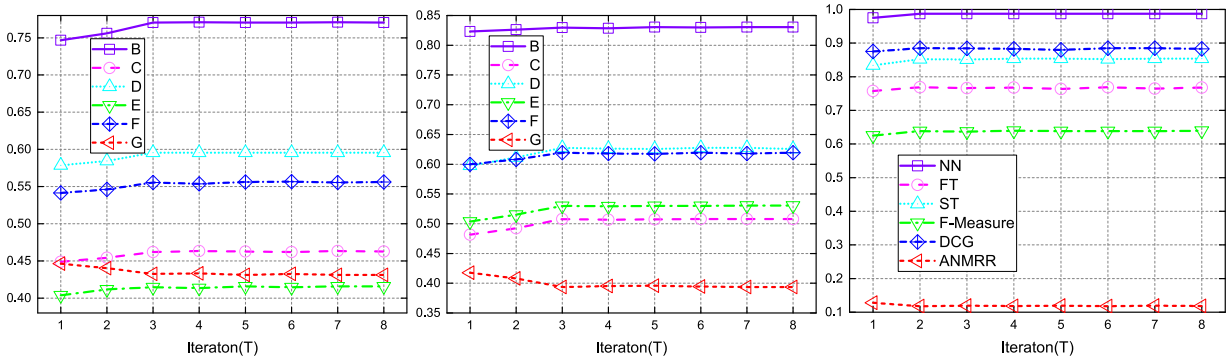


FIGURE 5. Comparison by varying the iteration numbers on (a) NTU, (b) MVRED, and (c) ETH.

TABLE 1. Methods comparison on MVRED dataset. Baseline: Use NN to measure the similarity between pairwise view sets. GI: Graph initialization. GR: Graph refinement. LM: Label estimation and metric learning.

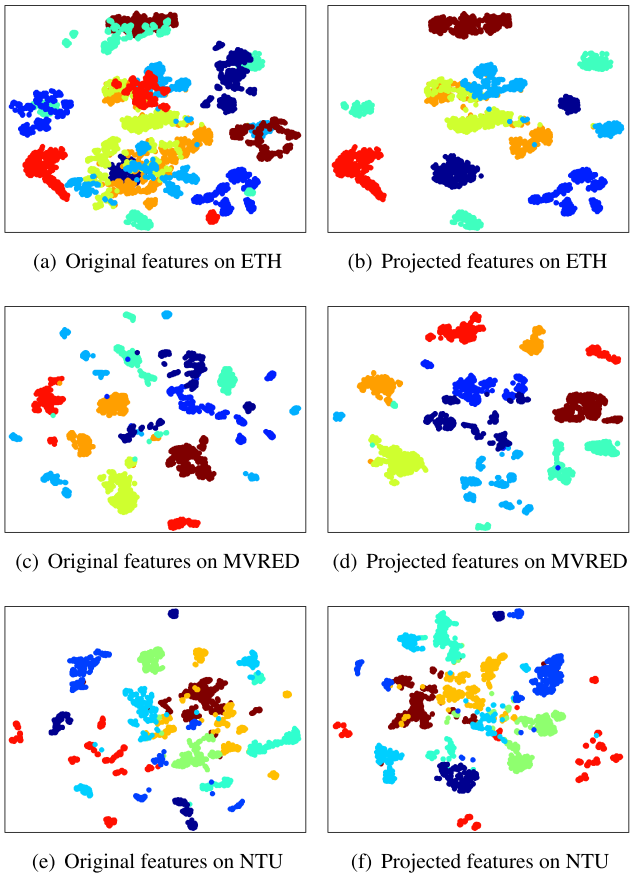
MVRED	NN	FT	ST	F	DCG	AUC	ANMRR
Baseline	0.785	0.418	0.533	0.453	0.534	0.404	0.481
Ours w/GI	0.804	0.444	0.554	0.468	0.563	0.429	0.454
Ours w/GI+LM	0.806	0.459	0.565	0.480	0.569	0.441	0.441
Ours w/GI+GR	0.810	0.465	0.580	0.490	0.578	0.450	0.435
Ours w/GI+GR+LM	0.823	0.481	0.598	0.504	0.600	0.474	0.418

c: SENSITIVITY ANALYSIS ON ITERATION NUMBER

After learning the projected metric, we can use (3) to update the initial rank graph and learn the projected metric iteratively. We varied the iteration number from 1 to 8. The performances on NTU, MVRED and ETH are shown in Fig. 5. From the results we can observe that the performances can be improved by iteration. Specifically, we can get 98.8% on ETH, 83.0% on MVRED and 77.0% on NTU in terms of NN. Moreover, we can achieve stable results only after a few iterations. Therefore, this method is robust to achieve high performances. Considering the performance and computational cost, we utilized T=3 in our experiments.

2) ABLATION EXPERIMENT ON DIFFERENT MODULES

To investigate the effectiveness of the proposed method, we conduct ablation studies on MVRED (Tab. 1). Firstly, we show the effect of graph initialization module. As shown in Tab. 1, "Ours w/GI" outperforms the baseline method and obtain the gain of 2.4%, 6.2%, 3.9%, 3.3%, 5.4%, 6.2% in terms of NN, FT, ST, F-measure, DCG, AUC and the decline of 5.6% in terms of ANMRR. Next, we evaluate the effect of the label estimation and metric learning. As reported in Tab. 1, "Ours w/GI+LM" improves the performance of "Ours w/GI", which demonstrates the effectiveness of label estimation and metric learning. We also evaluate the performance of graph refinement by adding the graph refinement module after the graph initialization (Ours w/GI+GR). "Ours w/GI+GR" consistently improves results over baseline and "Ours w/GI". For example, "Ours w/GI+GR" obtains the gain of 4.7%, 4.7%, 4.7%, 2.7%, 4.9% in terms of FT, ST, F-measure, DCG, AUC and the decline of 4.2% in terms of ANMRR comparing against without the graph refinement. Furthermore, when integrating the three modules together, our method gains more improvement in performance and "Ours w/GI+GR+LM" obtains the gain of 4.8%, 15.1%, 12.2%, 11.3%, 12.4%, 17.3% in terms of NN, FT, ST, F-measure, DCG, AUC and the decline of 13.1% in terms of ANMRR comparing against the baseline method.



(a) Original features on ETH (b) Projected features on ETH

(c) Original features on MVRED (d) Projected features on MVRED

(e) Original features on NTU (f) Projected features on NTU

FIGURE 6. Visualization of features on ETH (a) (b), MVRED (c) (d), and NTU (e) (f).

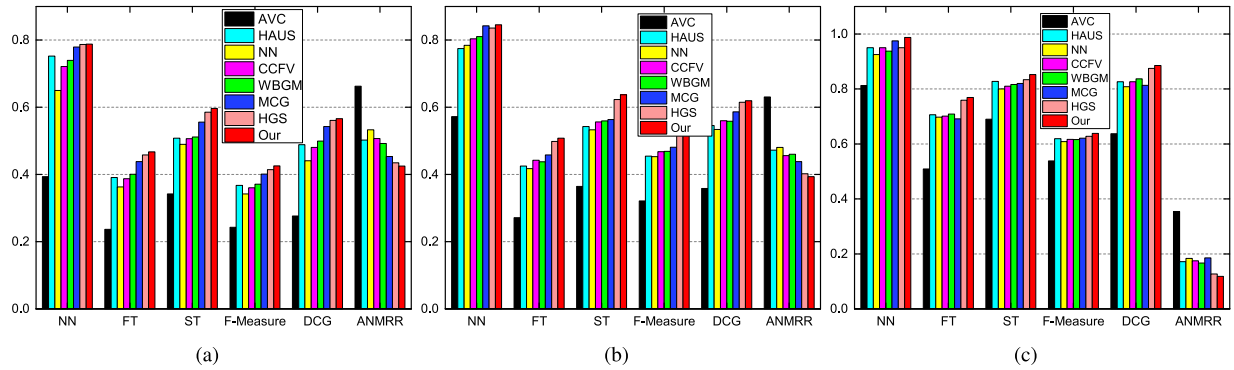


FIGURE 7. Comparison against the state of the arts on (a) NTU, (b) MVRED, and (c) ETH.

TABLE 2. Speed comparison on different datasets (s/query).

Dataset	Distance-based		Stastic-based		Graph-based			Our
	NN	HAUS	AVC	CCFV	WBGM	CGM	HGS	
NTU	5.0	5.2	12.9	13.9	25.2	26.4	31.6	**0.25**
MVRED	10.4	10.4	19.5	19.8	27.0	27.5	35.5	**0.29**
ETH	3.5	3.8	5.7	5.9	12.9	14.3	18.8	**0.22**

3) FEATURE VISUALIZATION

We utilized 3280 samples (each view of one 3D model is treated as a sample) across 8 categories from ETH and the other 3280 samples of 8 categories from MVRED and NTU respectively to visualize their visual features before and after our method by t-SNE [47]. The results are shown in the Fig. 6. The original feature visualization on ETH, MVRED and NTU are shown in Fig. 6(a), (c), (e) and the projected features by the proposed method are shown in Fig. 6(b), (d), (f), respectively. From Fig. 6, we can have following observations: (1) As shown in Fig. 6(a), the original features are not discriminated very well and a lot of samples from different categories collide into a mess on ETH, whereas our method can obviously separate the samples from different classes very well (Fig. 6(b)). (2) As illustrated in Fig. 6(c)(e), the original features belonging to same classes are dispersive and those of different classes are confused. However, our method can still succeed in separating the samples from different categories and align them to the corresponding clusters, which can compact the samples belonging to the same class (Fig. 6(d)(f)). These in-depth results show the effectiveness of our strategy to embed the graph information into feature space.

4) COMPARISON AGAINST THE STATE OF THE ARTS

a: RETRIEVAL PERFORMANCE

The comparison against the existing methods are shown in Fig. 7. Generally, the graph-based methods can outperform the distance-based and statistical-based methods on three dataset, which indicates the graph-based methods use graph structure and graph matching to learn the model spatial characteristics of 3D models and consequently benefit similarity measure. Our method achieves competing performances compared against the three kinds of methods on all evaluated datasets. Specifically, we can get several observations:

- Compared with the distance-based methods, the proposed method outperforms NN and HUAS on all three datasets. Specifically, our method outperforms the distance-based methods by the gain of 4.8%-21.2%, 19.6%-28.8%, 17.4%-21.8%, 15.9%-24.5% in terms of NN, FT, ST, F-measure, DCG and the decline of 15.8%-28.4% in terms of ANMRR on NTU dataset (Fig. 7(a)). On MVRED (Fig. 7(b)), we observe the gain of 5.2%-47.7%, 14.8%-87.3%, 14.6%-75.0%, 14.8%-67.1%, 10.6%-72.9% and the decline of 13.7%-37.6% in terms of ANMRR. On ETH (Fig. 7(c)), we can achieve the gain of 3.9%-21.5%, 9.6%-51.1%, 5.2%-23.6%, 3.5%-18.6%, 7.1%-39.0% and achieve the decline of 32.4%-66.7% in terms of ANMRR.
- Compared with the statistical-based methods, our method can achieve the gain of 9.2%-100.3%, 20.6%-97.3%, 17.8%-74.4%, 18.1%-75.3%, 17.8%-104.9% and the decline of 16.1%-35.9% on NTU. On MVRED, we can achieve the gain of 5.2%-47.7%, 14.8%-87.3%, 14.6%-75.0%, 14.8%-67.1%, 10.6%-72.9% and the decline of 13.7%-37.6%. On ETH, we can observe the gain of 3.9%-21.5%, 9.6%-51.1%, 5.2%-23.6%, 3.5%-18.6%, 7.1%-39.0% and the decline of 32.4%-66.7%, in terms of NN, FT, ST, F-measure, DCG and ANMRR.
- Compared with the graph-based methods, our method can outperform them by the gain of 0.1%-6.6%, 2.0%-16.6%, 1.9%-16.6%, 2.8%-14.8%, 0.9%-13.3% in terms of NN, FT, ST, F-measure, DCG and the decline of 2.2%-13.6% in terms of ANMRR on NTU. In Fig. 7(b), we get the improvement by the gain of 0.3%-1.1%, 16.0%-1.9%, 14.0%-2.3%, 14.5%-2.1%, 10.9%-0.7% and the decline of 14.5%-2.1% on MVRED. On ETH, as shown in Fig. 7(c), we can observe the gains by 1.3%-5.3%, 1.3%-11.2%, 2.2%-4.4%, 1.7%-3.7%, 1.2%-8.8% and the decline of 6.9%-36.3%, in terms of NN, FT, ST, F-measure, DCG and ANMRR, respectively.

TABLE 3. Comparison of label estimation with existing methods on NTU.

Type	Method	NN ↑	FT ↑	ST ↑	F-measure ↑	DCG ↑	AUC ↑	ANMRR ↓
Distance	NN [23]	0.650	0.363	0.490	0.342	0.441	0.310	0.533
	NN [23]+Our	0.769	0.491	0.609	0.424	0.5	0.422	0.405
	Gain	**18.3%**	**35.3%**	**24.3%**	**24.0%**	**32.2%**	**36.1%**	**24.0%**
	HAUS [23]	0.752	0.391	0.508	0.367	0.489	0.334	0.502
	HAUS [23]+Our	0.763	0.472	0.601	0.418	0.562	0.415	0.423
	Gain	**1.5%**	**20.7%**	**18.3%**	**13.9%**	**14.9%**	**24.3%**	**15.7%**
Statistic	AVC [22]	0.393	0.237	0.342	0.243	0.276	0.191	0.663
	AVC [22]+Our	0.754	0.465	0.596	0.410	0.555	0.403	0.430
	Gain	**91.9%**	**96.2%**	**74.3%**	**68.7%**	**101.1%**	**111.0%**	**35.1%**
	CCFV [25]	0.721	0.387	0.506	0.361	0.481	0.329	0.507
	CCFV [25]+Our	0.727	0.439	0.552	0.383	0.527	0.371	0.457
	Gain	**0.8%**	**13.4%**	**9.1%**	**6.1%**	**9.6%**	**12.8%**	**9.9%**
Graph	WBGM [24]	0.740	0.401	0.512	0.371	0.500	0.341	0.492
	WBGM [24]+Our	0.750	0.479	0.603	0.424	0.568	0.417	0.416
	Gain	**1.4%**	**19.5%**	**17.8%**	**14.3%**	**13.6%**	**22.3%**	**15.4%**
	MCG [26]	0.780	0.438	0.556	0.401	0.543	0.378	0.453
	MCG [26]+Our	0.790	0.468	0.599	0.419	0.556	0.413	0.427
	Gain	**1.4%**	**6.8%**	**7.7%**	**4.5%**	**2.4%**	**9.3%**	**5.7%**
	HGS [27]	0.787	0.458	0.586	0.414	0.561	0.406	0.435
	HGS [27]+Our	0.800	0.488	0.602	0.429	0.582	0.426	0.406
	Gain	**1.7%**	**6.6%**	**2.7%**	**3.6%**	**3.7%**	**4.9%**	**6.7%**

TABLE 4. Comparison of label estimation with existing methods on MVRED.

Type	Method	NN ↑	FT ↑	ST ↑	F-measure ↑	DCG ↑	AUC ↑	ANMRR ↓
Distance	NN [23]	0.785	0.418	0.533	0.453	0.534	0.404	0.481
	NN [23]+Our	0.804	0.499	0.613	0.520	0.613	0.484	0.402
	Gain	**2.4%**	**19.4%**	**15.0%**	**14.8%**	**14.8%**	**19.8%**	**16.4%**
	HAUS [23]	0.775	0.425	0.542	0.454	0.545	0.416	0.472
	HAUS [23]+Our	0.842	0.505	0.637	0.526	0.624	0.503	0.394
	Gain	**8.6%**	**18.8%**	**17.5%**	**15.9%**	**14.5%**	**20.9%**	**16.6%**
Statistic	AVC [22]	0.572	0.271	0.364	0.321	0.358	0.242	0.631
	AVC [22]+Our	0.833	0.509	0.623	0.531	0.619	0.495	0.394
	Gain	**45.6%**	**87.8%**	**71.2%**	**65.4%**	**72.9%**	**104.5%**	**37.6%**
	CCFV [25]	0.804	0.442	0.556	0.468	0.560	0.429	0.456
	CCFV [25]+Our	0.814	0.520	0.621	0.537	0.623	0.497	0.385
	Gain	**1.2%**	**17.6%**	**11.7%**	**14.7%**	**11.3%**	**15.9%**	**15.6%**
Graph	WBGM [24]	0.810	0.438	0.559	0.469	0.559	0.427	0.460
	WBGM [24]+Our	0.820	0.520	0.629	0.535	0.630	0.501	0.383
	Gain	**1.2%**	**18.7%**	**12.5%**	**14.1%**	**12.7%**	**17.3%**	**16.7%**
	MCG [26]	0.842	0.458	0.564	0.481	0.586	0.446	0.438
	MCG [26]+Our	0.860	0.514	0.627	0.535	0.630	0.507	0.387
	Gain	**2.1%**	**12.2%**	**11.2%**	**11.2%**	**7.5%**	**13.7%**	**11.6%**
	HGS [27]	0.836	0.498	0.623	0.526	0.615	0.492	0.402
	HGS [27]+Our	0.842	0.533	0.642	0.548	0.642	0.514	0.371
	Gain	**0.7%**	**7.0%**	**3.0%**	**4.2%**	**4.4%**	**4.5%**	**7.7%**

b: SPEED ANALYSIS

For real application, the speed is an important factor to evaluate the retrieval performance. To show the efficiency of our method, we illustrate the speeds of different methods on three datasets in Tab. 2. For fair comparison, all the methods were tested on a Windows 7 ultimate x64 with single Core (CPU:3.3 GHz; RAM: 8GB). The experimental results shows our algorithm is much faster than the other methods. Specifically, our proposed method only costs 0.25s for one query on NTU dataset, while HGS, which achieved the second best results on NTU, costs 31.63s. Considering the speed, the second fast method is NN (5.04s), while its performance is much lower than ours as shown in Fig. 7.

c: LABEL ESTIMATION WITH EXISTING METHODS

We evaluated the propose method by initializing the rank graph with the existing methods on three datasets. The results are listed in Tab 3, 4 and 5, respectively. From the results, we have several observations:

- Our method can consistently outperform all the distance-based, statistic-based and graph-based methods on three datasets in terms of all evaluation criteria. For example, our method improves performance of the best method HGS by the gain of 1.7%, 6.6%, 2.7%, 3.6%, 3.7%,4.9% in terms of NN, FT, ST, F-measure, DCG, AUC and the decline of 6.7% in terms of ANMRR on NTU as shown in Tab 3. On MVRED, our method improves performance of HGS by the gain of 0.7%, 7.0%, 3.0%, 4.2%, 4.4%, 4.5% in terms of NN, FT, ST, F-measure, DCG, AUC and the decline of 7.7% in terms of ANMRR as shown in Tab 4. On ETH as shown in Tab 5, our method improves it by the gain of 2.6%, 0.3%, 1.4%, 0.3%, 0.6%, 0.1% in terms of NN, FT, ST, F-measure, DCG, AUC and the decline of 4.7% in term of ANMRR.

TABLE 5. Comparison of label estimation with existing methods on ETH.

Type	Method	NN ↑	FT ↑	ST ↑	F-measure ↑	DCG ↑	AUC ↑	ANMRR ↓
Distance	NN [23]	0.925	0.698	0.800	0.608	0.808	0.721	0.184
	NN [23]+Our	0.988	0.727	0.812	0.615	0.848	0.754	0.155
	Gain	**6.8%**	**4.2%**	**1.5%**	**1.2%**	**5.0%**	**4.6%**	**15.8%**
	HAUS [23]	0.950	0.706	0.828	0.619	0.826	0.747	0.172
	HAUS [23]+Our	0.960	0.730	0.830	0.622	0.839	0.750	0.156
	Gain	**1.1%**	**3.4%**	**0.2%**	**0.5%**	**1.6%**	**0.4%**	**9.3%**
Statistic	AVC [22]	0.813	0.509	0.690	0.538	0.637	0.567	0.354
	AVC [22]+Our	0.975	0.741	0.814	0.618	0.863	0.769	0.139
	Gain	**19.9%**	**45.6%**	**18.0%**	**14.9%**	**35.5%**	**35.6%**	**60.7%**
	CCFV [25]	0.950	0.701	0.810	0.617	0.826	0.743	0.175
	CCFV [25]+Our	0.963	0.724	0.820	0.620	0.846	0.753	0.157
	Gain	**1.4%**	**3.3%**	**1.2%**	**0.5%**	**2.4%**	**1.4%**	**10.3%**
Graph	WBGM [24]	0.938	0.709	0.816	0.616	0.837	0.751	0.167
	WBGM [24]+Our	0.988	0.780	0.846	0.638	0.901	0.806	0.105
	Gain	**5.3%**	**10.1%**	**3.7%**	**3.6%**	**7.7%**	**7.3%**	**37.1%**
	MCG [26]	0.975	0.691	0.820	0.621	0.813	0.734	0.185
	MCG [26]+Our	0.988	0.747	0.831	0.628	0.867	0.774	0.135
	Gain	**1.3%**	**8.1%**	**1.3%**	**1.1%**	**6.6%**	**5.5%**	**27.0%**
	HGS [27]	0.950	0.759	0.834	0.628	0.875	0.779	0.127
	HGS [27]+Our	0.975	0.761	0.846	0.630	0.880	0.780	0.121
	Gain	**2.6%**	**0.3%**	**1.4%**	**0.3%**	**0.6%**	**0.1%**	**4.7%**

- Even initializing the rank graph by the methods with low performances, our method can still improve the performances. Specially, for distance-based methods NN, we can get the gain of 6.8%, 4.2%, 1.5%, 1.2%, 5.0%, 4.6% in terms of NN, FT, ST, F-measure, DCG, AUC and the decline of 15.8% in terms of ANMRR on ETH dataset as shown in Tab. 5. The similar performance can be obtained on NTU and MVRED as shown in Tab. 3 and Tab. 4, respectively.

V. CONCLUSION

In this paper, we propose a novel unsupervised method of metric learning for 3D model retrieval. We utilize the visual feature to initialize graph and refine the graph by combining the visual and contextual information. To improve the quality of the estimated labels, we adopt both similarity and dissimilarity to handle the noisy label information and learn an improved metric with iteratively updating the whole procedure. Furthermore, when initializing the rank graph by existing distance-based methods, statistic-based methods and graph-based methods, our method improves the performances of these methods without any annotations. Extensive experiments on three 3D datasets have proven the effectiveness and efficiency of our method.

REFERENCES

[1] J. Xie, G. Dai, F. Zhu, L. Shao, and Y. Fang, "Deep nonlinear metric learning for 3-D shape retrieval," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 412–422, Jan. 2018.

[2] A.-A. Liu, W.-Z. Nie, Y. Gao, and Y.-T. Su, "View-based 3-D model retrieval: A benchmark," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 916–928, Mar. 2018.

[3] J. Xie, G. Dai, F. Zhu, E. K. Wong, and Y. Fang, "Deepshape: Deep-learned shape descriptor for 3D shape retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1335–1345, Jul. 2017.

[4] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, "Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding," *Pattern Recognit.*, vol. 48, no. 10, pp. 3102–3112, Dec. 2015.

[5] S. Jia, L. Ma, and D. Qin, "Research on scene understanding-based encrypted image retrieval algorithm," *IEEE Access*, vol. 7, pp. 6587–6596, 2018.

[6] Z. Chen, S. Ai, and C. Jia, "Structure-aware deep learning for product image classification," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 15, no. 1s, pp. 4:1–4:20, 2019.

[7] A. Mazaheri, B. Gong, and M. Shah, "Learning a multi-concept video retrieval model with multiple latent variables," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 14, no. 2, pp. 46:1–46:21, 2018.

[8] L. Zhang, L. Zhang, B. Du, J. You, and D. Tao, "Hyperspectral image unsupervised classification by robust manifold matrix factorization," *Inf. Sci.*, vol. 485, pp. 154–169, Jun. 2019.

[9] Y. Zhen, Y. Gao, D.-Y. Yeung, H. Zha, and X. Li, "Spectral multimodal hashing and its application to multimedia retrieval," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 27–38, Jan. 2016.

[10] R. Hong, Z. Hu, R. Wang, M. Wang, and D. Tao, "Multi-view object retrieval via multi-scale topic models," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5814–5827, Dec. 2016.

[11] F. Nie, J. Li, and X. Li, "Convex multiview semi-supervised classification," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5718–5729, Dec. 2017.

[12] X. Zheng, R. Ji, X. Sun, Y. Wu, F. Huang, and Y. Yang, "Centralized ranking loss with weakly supervised localization for fine-grained object retrieval," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, Stockholm, Sweden, 2018, pp. 1226–1233.

[13] H. Zeng, Q. Wang, and J. Liu, "Multi-feature fusion based on multi-view feature and 3D shape feature for non-rigid 3D model retrieval," *IEEE Access*, vol. 7, pp. 41584–41595, 2019.

[14] K. Lu, Q. Wang, J. Xue, and W. Pan, "3D model retrieval and classification by semi-supervised learning with content-based similarity," *Inf. Sci.*, vol. 281, pp. 703–713, Oct. 2014.

[15] A. D. P. Papoiu, N. M. Emerson, T. S. Patel, R. A. Kraft, R. Valdes-Rodriguez, L. A. Nattkemper, R. C. Coghill, and G. Yosipovitch, "Voxel-based morphometry and arterial spin labeling fMRI reveal neuropathic and neuroplastic features of brain processing of itch in end-stage renal disease," *J. Neurophysiol.*, vol. 112, no. 7, pp. 1729–1738, 2014.

[16] P. Polewski, W. Yao, P. Krzystek, M. Heurich, and U. Stilla, "Detection of fallen tree segments in airborne LiDAR point clouds of a temperate forest by combining point/primitive-level shape descriptors," Gemeinsame Tagung, Zürich, Switzerland, Tech. Rep. 23, 2014, pp. 1–12.

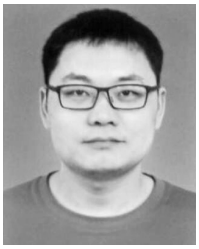
[17] E. Persoon and K.-S. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 3, pp. 388–397, May 1986.

[18] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*. Dec. 2015, pp. 945–953.

[19] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant set clustering and pooling for multi-view 3D object recognition," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2017, pp. 1–12.
[20] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "Gvcnn: Group-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 264–272.
[21] T. Yu, J. Meng, and J. Yuan, "Multi-view harmonized bilinear network for 3D object recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 186–194.
[22] T. F. Ansary, M. Daoudi, and J. P. Vandeborre, "A Bayesian 3-D search engine using adaptive views clustering," *IEEE Trans. Multimedia*, vol. 9, no. 1, pp. 78–88, Jan. 2007.
[23] Y. Gao and Q. Dai, "View-based 3D object retrieval: Challenges and approaches," *IEEE Multimedia Mag.*, vol. 21, no. 3, pp. 52–57, Jul. 2014.
[24] Y. Gao, Q. Dai, M. Wang, and N. Zhang, "3D model retrieval using weighted bipartite graph matching," *Signal Process., Image Commun.*, vol. 26, no. 1, pp. 39–47, 2011.
[25] Y. Gao, J. Tang, R. Hong, S. Yan, Q. Dai, N. Zhang, and T.-S. Chua, "Camera constraint-free view-based 3-D object retrieval," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2269–2281, Apr. 2012.
[26] A.-A. Liu, W.-Z. Nie, Y. Gao, and Y.-T. Su, "Multi-modal clique-graph matching for view-based 3D model retrieval," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2103–2116, May 2016.
[27] Y. Su, W. Li, A. Liu, and W. Nie, "Hierarchical graph structure learning for multi-view 3D model retrieval," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 913–919.
[28] J. Yu, Y. Rui, Y. Y. Tang, and D. Tao, "High-order distance-based multiview stochastic learning in image classification," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2431–2442, Dec. 2014.
[29] P.-E. Danielsson, "Euclidean distance mapping," *Comput. Graph. Image Process.*, vol. 14, no. 3, pp. 227–248, 1980.
[30] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993.
[31] Y. Gao, M. Wang, R. Ji, X. Wu, and Q. Dai, "3-D object retrieval with Hausdorff distance learning," *IEEE Trans. Ind. Electron.*, vol. 61, no. 4, pp. 2088–2098, Apr. 2014.
[32] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai, "3-D object retrieval and recognition with hypergraph analysis," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4290–4303, Sep. 2012.
[33] A. Liu, Z. Wang, W. Nie, and Y. Su, "Graph-based characteristic view set extraction and matching for 3D model retrieval," *Inf. Sci.*, vol. 320, pp. 429–442, Nov. 2015.
[34] S. Zhao, H. Yao, Y. Zhang, Y. Wang, and S. Liu, "View-based 3D object retrieval via multi-modal graph learning," *Signal Process.*, vol. 112, pp. 110–118, Jul. 2015.
[35] Z. Zhang, Z. Zhai, and L. Li, "Uniform projection for multi-view learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1675–1689, Aug. 2017.
[36] P. Zhu, L. Zhang, W. Zuo, and D. Zhang, "From point to set: Extend the learning of distance metrics," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2664–2671.
[37] S. Bai and X. Bai, "Sparse contextual activation for efficient visual re-ranking," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1056–1069, Mar. 2016.
[38] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3652–3661.
[39] S. Liao and S. Z. Li, "Efficient PSD constrained asymmetric metric learning for person re-identification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3685–3693.
[40] Z. Wang, R. Hu, C. Chen, Y. Yu, J. Jiang, C. Liang, and S. Satoh, "Person reidentification via discrepancy matrix and matrix metric," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 3006–3020, Oct. 2018.
[41] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
[42] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 223–232, Sep. 2003.
[43] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *Proc. CVPR*, Jun. 2003, pp. 409–415.
[44] F. Lu, I. Sato, and Y. Sato, "Uncalibrated photometric stereo based on elevation angle recovery from BRDF symmetry of isotropic materials," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 168–176.
[45] H. Müller, W. Müller, D. M. Squire, S. Marchand-Maillet, and T. Pun, "Performance evaluation in content-based image retrieval: Overview and proposals," *Pattern Recognit. Lett.*, vol. 22, no. 5, pp. 593–601, 2001.
[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.
[47] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

YUTING SU received the M.S. and Ph.D. degrees in electronic engineering from Tianjin University, China, where he is currently a Professor with the School of Electrical and Information Engineering. His research interests include multimedia content analysis and security, multiple object tracking, and multimedia content analysis and security.

WENHUI LI is currently pursuing the Ph.D. degree with the School of Electrical and Information Engineering, Tianjin University. He was an Intern Student with the SeSaMe Center, National University of Singapore. His research interests include computer vision, machine learning, and 3D model retrieval.



WEIZHI NIE received the Ph.D. degree from Tianjin University, Tianjin, China, where he is currently an Assistant Professor with the School of Electrical and Information Engineering. He was a Visiting Scholar with the NExT Center, National University of Singapore, where he was with Prof. T.-S. Chua. His research interests include computer vision, machine learning, and social networks.

DAN SONG received the Ph.D. degree in computer science and technology from Zhejiang University, China. Her research interests include computer graphics, computer vision, 3D human body reconstruction, and virtual fitting.

AN-AN LIU received the B.Eng. and Ph.D. degrees from Tianjin University, Tianjin, China, where he is currently a Professor with the School of Electrical and Information Engineering. He was a Visiting Scholar with the Robotics Institute, Carnegie Mellon University, where he was with Prof. T. Kanade. His research interests include computer vision and machine learning.