Sketch-Based Shape Retrieval via Best View Selection and a Cross-Domain Similarity Measure

Yongzhe Xu, Jiangchuan Hu, Kanoksak Wattanachote, Kun Zeng, and YongYi Gong

***Abstract*—Retrieving 3D shapes from 2D human sketches has received increasing attention in computer vision and computer graphics. Most previous methods projected 3D shapes from numerous viewpoints and then extracted features of 3D shapes from these projections and calculated the similarity with sketches. However, due to the unknown pose of 3D shapes, viewpoints were usually sampled uniformly from a sphere coordinate. Hence, some projections acquired insufficient descriptions of 3D shapes. In this paper, we proposed a view selection algorithm to find the most reasonable viewpoints, which can benefit representation learning for 3D shapes. Additionally, to indicate the apparent discrepancy between sketches and 3D shapes, we leveraged a generalized similarity model to encourage the accuracy of cross-domain feature matching. We first computed line renderings of 3D shapes from an enormous number of viewpoints. Then, we calculated the similarity of shapes between line renderings and sketches. In this vein, we obtained several superior projections. Second, we implemented a sketch network to extract features of the sketch and a shape network to extract features of projections. We combined the features of different projections to secure the compact representation for 3D shapes. Finally, a metric network was constructed using a cross-domain similarity model, and we trained the metric network with triplet loss. Online hard sample mining was leveraged to accelerate the convergence of the network. We evaluated our method on SHREC'13 and SHREC'14 sketch track benchmark datasets. The experimental results demonstrated that both view selection and cross-domain similarity models were able to encourage retrieval performance.**

***Index Terms*—Sketch, 3D shape, similarity model, best view selection.**

I. Introduction

In recent times with the proliferation of 3D shapes, the demand for efficient retrieval of 3D shapes has been increasing both in academia and industry. At present, most 3D shape databases that are frequently used were developed by the *keyword-based* retrieval approach, such as 3D Warehouse and TurboSquid. They require rich and accurate annotations for 3D shapes, which is a considerable workload. Another approach is the *content-based* method, in which users are allowed to provide a similar 3D shape to retrieve what they desired. However, this is impractical since 3D shapes are not easy to obtain. Compared to these methods, *sketch-based* shape retrieval is more convenient for users. Since users without professional skills can easily express their intent by simply drawing sketches, sketch-based shape retrieval has been gaining considerable attention in recent years.

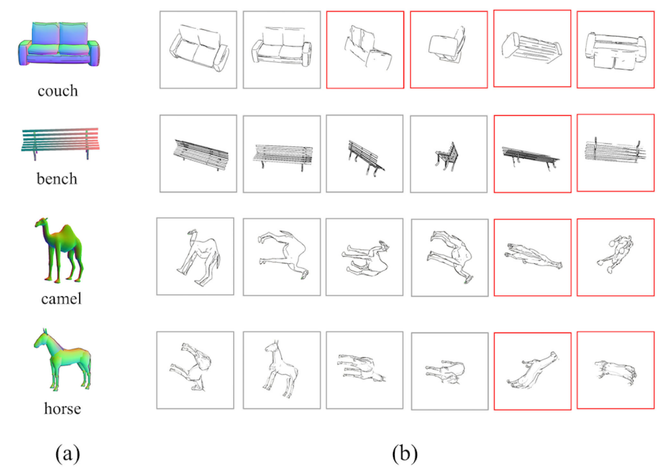


Fig. 1. The line renderings of 3D shapes from uniformly sampled viewpoints: (a) 3D shapes and (b) their corresponding line renderings. The line renderings with the red bounding box are uncommon perspectives.

Sketch-based shape retrieval needs to overcome the apparent discrepancy between images and 3D shapes. To address the problem, most existing works [1]–[4] have attempted to project 3D shapes into 2D line renderings and measure the similarity between sketches and projections. The viewpoints for projection were uniformly sampled on the spherical coordinate system whose origin is the centroid of 3D shapes. However, such methods always suffer from two problems. On the one hand, due to the unknown pose of 3D shapes, some of the projections could not characterize 3D shapes well. We demonstrated some 3D shapes and their line renderings, which were rendered from uniformly sampled viewpoints in Fig. 1. Notice that some projections are difficult to represent as 3D shapes and even lead to ambiguity (e.g., the rightmost projection of a camel sometimes looks similar to a horse). In the design literature [5] and other

Manuscript received June 5, 2019; revised November 2, 2019 and December 27, 2019; accepted January 9, 2020. Date of publication January 15, 2020; date of current version October 23, 2020. This work was supported in part by the National Science Foundation Grant of China under Grants U1711266, 61370160, and 61772149, in part by the Natural Science Foundation of Guangdong Province under Grant 201904010228, and in part by the Guangdong Basic and Applied Basic Research Foundation under Grants 2019A1515011078 and 2015A030313578. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Z. Liu. *(Corresponding author: Kun Zeng.)*

Y. Xu, J. Hu, and K. Zeng are with the School of Software, Sun Yat-sen University, Guangzhou 510006, China (e-mail: xuyzh6@mail2.sysu.edu.cn; hujch3@mail2.sysu.edu.cn; zengkun2@mail.sysu.edu.cn).

K. Wattanachote and Y. Gong are with the School of Informatics, Guangdong University of Foreign Studies, Guangzhou 510006, China (e-mail: kanoksak.wattanachote@gmail.com; gongyongyi@gdufs.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2020.2966882

1520-9210 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Anhui Normal University. Downloaded on September 24,2021 at 03:44:09 UTC from IEEE Xplore. Restrictions apply.

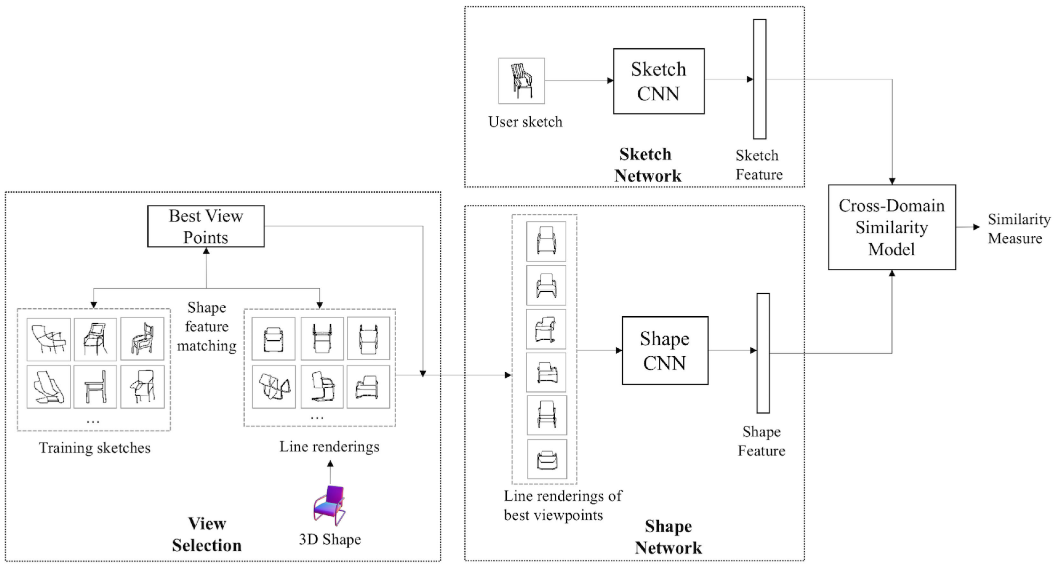


Fig. 2. The framework of our algorithm. First, we proposed a view selection method to select the best viewpoints for 3D shapes. Then, the projections of selected viewpoints and sketches were fed into convolutional networks of corresponding domains to perform feature extraction. Finally, a cross-domain similarity model was applied to calculate the similarity distance between the features of the sketch and the 3D shape.

sketching systems [6]–[8], researchers also recommend the use of informative perspective viewpoints that reduce ambiguity by showing the 3D object with minimal foreshortening on all sides. On the other hand, directly comparing sketches and projections may harm the retrieval since projections can only express one side information of 3D shapes. Recently, Su [9] proposed the MVCNN (multiview convolutional neural network) to fuse different projections into compact representations of 3D shapes, which can effectively solve the problem. Our proposed method leverages MVCNN to improve representation learning for 3D shapes.

Another problem arises from the similarity measure between sketches and 3D shapes, which is insufficiently studied. Most previous works projected the features of sketches and 3D shapes into the same feature space and applied Euclidean distance or cosine similarity to calculate the similarity between them. However, since sketches and 3D shapes lay in different domains, the method for measuring the similarity between them could greatly impact retrieval performance. In addition, because the similarity model is highly related to the property of features, it is nearly impossible to specify the best similarity model by human decision making.

In this paper, we proposed a method to resolve the problem of best view selection and similarity measure between different domains. To solve the viewpoint problem, we proposed a best view selection algorithm inspired by Zhao [10] to select the best viewpoints of 3D shapes, which can reduce unreasonable projections and benefit the representation learning of 3D shapes. The idea is based on the fact that when people draw sketches, they mostly tend to create objects from some common perspectives, which means there hardly exists uncommon perspectives that have never been considered. Thus, we can acquire representative projections of 3D shapes by rendering shapes into line renderings from numerous viewpoints and then compare the contour similarity between line renderings and sketches. By using the prior of sketches, we find some of the best viewpoints for each 3D shape regardless of their pose, and the process can be achieved without any manual annotations. For the similarity measure problem, several works have recently been explored to resolve visual matching or feature adaption for different domains, such as person reidentification [11], [12], sketch-based image retrieval [13]–[16] and feature fusing of different modalities [17], [18]. In this paper, we applied a cross-domain similarity model proposed by Lin [11] to compute the similarity between sketches and 3D shapes. This similarity model was proven to be a generalized form of Mahalanobis distance and cosine similarity. We embedded the similarity model into a neural network and optimized its parameters in a data-driven manner, which is more robust for cross-domain feature matching.

The framework of our method is shown in Fig. 2. It is mainly comprised of three modules: a best view selection, feature extraction and a cross-domain similarity model. First, we put the centroid of each 3D shape at the origin of the spherical coordinate system and uniformly sampled a mass of viewpoints to render shapes into line renderings. Then, a view selection method was implemented to select the reasonable projections for each 3D shape. The details of this method are demonstrated in Section III-A. Next, we used two different convolutional networks to extract features of the selected projections and sketches and combined features of projections into compact representations of 3D shapes. Finally, the features of sketches and 3D shapes were fed into a cross-domain similarity model to measure the similarity distances between them. Here, minimal distance guarantees the most similar features. We used triplet loss to train our similarity model and introduced online hard sample mining to accelerate the convergence and increase the performance of the network.

Authorized licensed use limited to: Anhui Normal University. Downloaded on September 24,2021 at 03:44:09 UTC from IEEE Xplore. Restrictions apply.

In summary, the contributions of this paper are three-fold:

- We proposed a view selection method to select the best viewpoints of 3D shapes, which can promote the representation learning of shapes.
- We leveraged a cross-domain similarity model to perform feature matching between sketches and 3D shapes, which can improve the retrieval accuracy.
- We evaluated our method on the SHREC'13 [19] and SHREC'14 [20] benchmarks. The result shows that both the view selection and similarity model can improve the performance.

The rest of this paper is organized as follows. Section II reviews related works. In Section III, we demonstrate our view selection method and similarity model in detail. We present experimental results in Section IV and conclude this paper in Section V.

II. Related Work

In this section, we review some related works of best view selection and sketch-based shape retrieval.

A. Best View Selection

It is challenging to determine the best viewpoints of 3D shapes due to the ill-defined best viewpoints. Eitz *et al.* [1] struggled to learn a view classifier to select the best projections of 3D shapes based on the area of projection and occluding contours. Yasseen *et al.* [21] established the definition of best viewpoints from the perspective of cognitive science that representative projections are supposed to have minimal part occlusion or maximal information and minimal symmetry. They also proposed some methods to quantify these concepts. Lu *et al.* [22] considered the problem from a different perspective and argued that a viewpoint is preferred if people usually draw shapes from it. They selected the preference viewpoints of 3D shapes by measuring the similarity between sketches and line renderings. Zhao *et al.* [10] also considered sketch contours and proposed a method to learn best views of 3D shapes by using the context information of contours of sketches and line renderings. Here, we leveraged sketch contours in our prior process. Moreover, we applied a convolutional network [23] pretrained on edge images to extract the feature of contours, which can preserve the semantics of sketches and is more robust against the large variations of sketches.

B. Sketch Recognition

Since the widespread use of touch screens on portable devices, sketching has been introduced as a means of human-computer interaction, and sketch recognition has become a challenging task in computer vision. In 2012, Eitz *et al.* [24] released the first large-scale TU-Berlin sketch dataset because free-hand sketch recognition has attracted increasing attention. Earlier studies [24]–[26] applied hand-crafted features as feature representations of sketches and used SVM to classify sketches. Due to the success of deep learning, convolutional neural networks were effectively explored for sketch recognition. Omar *et al.* [27], [28] designed a network to extract features from sketches and employed LSTM to investigate the stroke order of sketch drawing. Likewise, Yu *et al.* [29], [30] used CNN as a feature extractor and applied ensemble fusion to improve recognition performance. Recently, Zhang *et al.* [31] proposed hybrid CNN, which consists of two-stream CNNs, to extract powerful features based on sketch appearance and shape. In our work, we used Inception-ResNetv2 [32] as the backbone for our sketch network.

C. 3D Shape Recognition

Our work is also related to 3D shape recognition. A large corpus of previous works computed 3D shape descriptors directly on the native 3D representations of objects, such as polygon meshes, voxel-based discretizations or point clouds. Tabia *et al.* [33] explored the covariance matrices of descriptors of 3D shapes and studied geodesic distances on the Riemannian manifold of matrices as metrics for 3D shape matching and recognition. Xie *et al.* [34] extracted multiple types of hand-crafted shape features and constructed a novel deep multimetric network for 3D shape retrieval. Because of the success of deep learning in feature learning, researchers have increasingly taken interest in applying neural networks to directly extract features of 3D shapes. Volumetric CNNs [35], [36], as an extension of the 2D convolutional network, were explored to extract features on voxelized shapes. Recently, Qi *et al.* [37], [38] first proposed a new network architecture named PointNet to process large point clouds. In this paper, since we aimed to compute the similarity between images (sketch) and 3D shapes, view-based descriptors are preferred to perform feature extraction for 3D shapes. A pioneering work toward exploiting view features of 3D objects is MVCNN [9]. It learns a compact shape descriptor from multiple rendered views of an object using CNN with a view-pooling layer. Bai *et al.* [39] applied CNN to extract features of projective images and sped up the retrieval task by combining GPU acceleration and inverted file. Recently, Ma *et al.* [40] proposed to combine CNN with LSTM to exploit the correlative information from multiple views and achieved promising performance for 3D shape recognition and retrieval. In our work, we first selected the preference viewpoints using the prior of sketches and then applied MVCNN [9] to compute descriptors by extracting features from rendering views of 3D shapes and fusing them into compact descriptors for 3D shapes. Compared to previous works, our method requires fewer projections and can encourage sketch-based shape retrieval.

D. Sketch-Based Shape Retrieval

Sketch-based shape retrieval has received increasing interest in computer vision and computer graphics. Previous works [1], [3], [41], [42] preferred to render 3D shapes into line renderings before extracting hand-crafted features of projections and sketches to measure their similarity. Saavedra *et al.* [3] extracted the external contours of projections and sketches and then applied three kinds of descriptors to describe the contours. Subsequently, the Manhattan distance was used for similarity matching. Eitz *et al.* [1] rendered 3D shapes using occluding contours

Authorized licensed use limited to: Anhui Normal University. Downloaded on September 24,2021 at 03:44:09 UTC from IEEE Xplore. Restrictions apply.

and suggestive contours from numerous viewpoints. They extracted features of projections and sketches by performing the bag-of-features method and computed similarity using cosine similarity. Recently, as the success of deep learning in computer vision, several deep learning-based approaches have been extensively leveraged. Wang *et al.* [2] first adopted a Siamese network to extract features of sketches and projections. They outperformed other state-of-the-art methods based on hand-crafted features. Xie *et al.* [43] proposed to learn barycentric representation of 3D shapes based on Wasserstein distance. They then applied another network to extract features of sketches and calculated the similarity using Euclidean distance. Chen *et al.* [44] adopted the domain adaption method to transfer features of sketches to the feature space of 3D shapes, which compensated for the divergence between different modalities. Their method significantly outperformed the state-of-the-art methods. Therefore, we integrated a deep learning approach in our work. However, we proposed the best view selection method to enhance the representation learning of 3D shapes, which has not been sufficiently studied in this era. Furthermore, we applied a cross-domain similarity model [11] in a sketch-based shape retrieval task, which can encourage retrieval performance.

III. Method

In this section, we devote the overall introduction to the proposed method. Our framework consists of three modules: a view selection module to select reasonable projections of 3D shapes, two networks for feature extraction, and a cross-domain similarity model to measure the similarity of different domains. In subsection A, we demonstrate the process of our view selection algorithm. The feature extraction networks for sketches and 3D shapes are briefly covered in subsection B. Then, we introduce the formulation of the similarity model in subsection C. Finally, we discuss the optimization of our method and online hard sample mining in subsection D.

A. Best View Selection

Representative projections can significantly promote the recognition of 3D shapes. However, due to the unknown pose of 3D shapes and ill-defined best viewpoints, it is still challenging to obtain reasonable projections. Inspired by Zhao [10], we used the prior that sketches may depict the painter's favorite view of the object to select reasonable viewpoints of 3D shapes automatically. Although the drawing style among people may differ greatly, the difference usually occurs in the shape of objects, the richness of contents, and other aspects, while the preference viewpoints are much the same. We show some sketch examples in Fig. 3. The examples illustrate that although large variations exist in different objects, the preference viewpoints are nearly the same (e.g., for airplane sketches, people tend to draw them from side view and frontal view, and rarely choose the bottom view or top view). Hence, the information of reasonable viewpoints may implicitly exist in these sketches, which can further assist in selecting the representative projections of 3D shapes.

The process of our best view selection method can be divided into four steps: (1) to render each 3D shape into line renderings

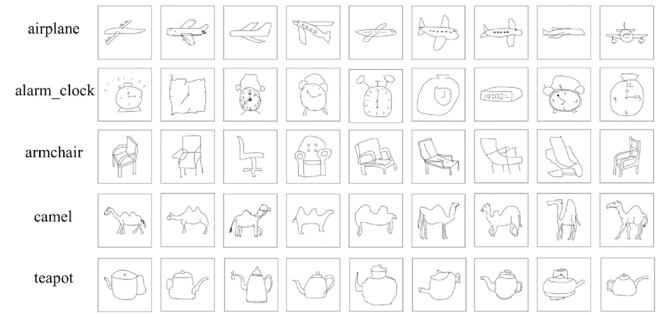


Fig. 3. Some sketch samples drawn from SHREC14 [20].

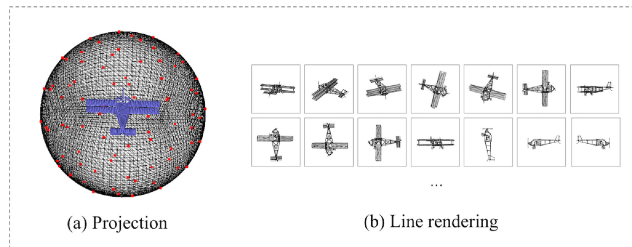


Fig. 4. The rendering effect of 3D shape: (a) the uniformly sampled viewpoints, red points indicate the position of viewpoints, (b) line renderings from different viewpoints.

from densely sampled viewpoints; (2) to obtain the sketches that belong to the same class for each shape; (3) to compute the similarity between sketches and line renderings and obtain the similarity score for each line rendering against corresponding sketches; (4) to rank each line rendering according to similarity scores and select top-K line renderings as the most reasonable projections for each 3D shape.

Specifically, to render 3D shapes, we put the centroid of each shape at the origin of the spherical coordinate system and uniformly sample an enormous number of viewpoints. Virtual cameras are placed at these viewpoints, and all of them point toward the centroid of the mesh. To make the projections of 3D shapes roughly resemble user sketches, we applied exterior silhouettes, occluding contours, suggestive contours [45], apparent ridges [46] and boundary lines to render project views of 3D shapes. After rendering each shape, we obtained its projections $V = \{v_i\}_i^k$, where k represents the number of viewpoints and was set to 80 in our experiment. Fig. 4 presents some rendering results of a 3D shape. The sketches that belong to the same class for each shape are gathered and denoted as $S = \{s_i\}_i^n$, where n is the number of sketches.

We then calculated the similarity between each projection and the corresponding sketches. Zhao [10] employed a Harris corner detector to compute the Harris key points of contours, and proposed a key context similarity to compute the similarity between contours of sketch and projection. However, due to the high abstraction of sketches, we found that their approach was not very robust and inevitably led to the selection of unreasonable projections. For this challenge, we applied another shape matching model proposed by Radenovic [23] to perform a similarity measure between sketches and line renderings. Radenovic [23] cast

Authorized licensed use limited to: Anhui Normal University. Downloaded on September 24,2021 at 03:44:09 UTC from IEEE Xplore. Restrictions apply.

shape matching as metric learning with a convolutional network and trained their network using edge maps extracted from natural images. Their experimental results showed that their shape matching model can generalize well to similarity measures of different fields, such as sketches and edge maps. In this paper, we leverage the pretrained model in [23] to extract the features of sketches and line renderings. This feature extractor was denoted as f_{edge}, then the similarity distance between sketch and projection could be calculated as follows:

$$S_{edge}(v_i, s_i) = ||f_{edge}(v_i) - f_{edge}(s_i)||_2 \tag{1}$$

where $||.||_2$ is the Euclidean distance. We further normalized the similarity between the projection and each sketch into [0, 1]:

$$p(v_i, s_i) = \frac{S_{edge}(v_i, s_i) - \min S_{edge}(v_i, s_j)}{\max S_{edge}(v_i, s_j) - \min S_{edge}(v_i, s_j)} \tag{2}$$

where $s_j \in S$. Finally, the similarity score between projection v_i and sketches S can be computed according to the following equation:

$$h(v_i, S) = \exp\left(-\sum_{s_i \in S} p(v_i, s_i)\right) \tag{3}$$

After the similarity scores for each projection are obtained, we can sort them in descending order, and higher scores indicate the more preferred projections. However, since the initial viewpoints are densely sampled, nearby viewpoints may have similar scores due to the tiny difference in their line renderings. Therefore, we employ the ranking algorithm in [10] to increase the diversity of the selected projections. We formulate a new function to calculate the similarity score as follows:

$$\tilde{h}(v_i, S) = \exp\left(-\sum_{s_i \in S} p(v_i, s_i)\right) + \exp\left(-\frac{\phi(v_i)^2}{2\sigma^2}\right) \tag{4}$$

where $\sigma = 0.2$ and $\phi(v_i)$ is a penalty term to suppress those similar projections. It is defined as:

$$\phi(v_i) = \max_{v_j \in \tilde{V}} IoU(v_i, v_j) \tag{5}$$

where $\tilde{V}$ indicates projections that rank higher than v_i according to $h(v_j, S)$. The $Intersection\ over\ Union(IoU)$ is applied to measure the similarity between projections. It is defined as the intersection of projection areas divided by their union. Obviously, the larger IoU is, the higher the similarity between v_i and its previous projections, which means a larger penalty is added on v_i. Note that when $\tilde{V}$ is empty, we define $\tilde{h}(v_i, S) = h(v_i, S)$ by default. The whole process of our ranking method is summarized in Algorithm 1. We show some reasonable projections produced by our approach in Fig. 5. Note that our view selection procedure can filter out most viewpoints from unusual perspectives. This procedure improves the representation learning of 3D shapes. The quantitative justification for our view selection procedure on 3D shape classification is shown in Section IV Table IV.

Our best view selection module is performed offline before training feature extractors. Specifically, we computed the best viewpoints for each 3D shape and saved their preference line renderings offline. These line renderings are further used to train our ShapeCNN, which is described in subsection B. During testing, the preference viewpoints of a 3D shape can be computed by comparing its projections with the training sketches that have the same class label using Algorithm 1. Then, the preference projections can be further leveraged to extract the feature of the 3D shape.

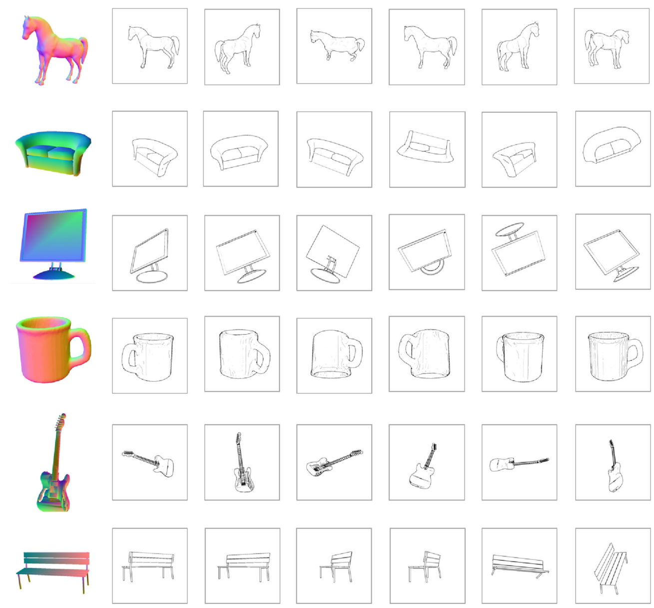


Fig. 5. The top-6 projections selected by our approach.

Algorithm 1: Best View Selection.

Input: line renderings of each viewpoint
Output: U, the set of top-N best line renderings
$U \leftarrow \{\}$
Initial scores $\hat{s_i}$ of each viewpoint $v_i \in V$ using (3)
choose v_i with highest $\hat{s_i}$
$U \leftarrow U + v_i$
$V \leftarrow V - v_i$
repeat
  for all $v_i \in V$ **do**
    $\tilde{s_i} = \hat{s_i} + \exp\left(-\frac{\phi(v_i)^2}{2\sigma^2}\right)$
  end for
  choose v_i with the highest $\tilde{s_i}$
  $U \leftarrow U + v_i$
  $V \leftarrow V - v_i$
until size of U reaches N

B. Feature Extractor

In this module, two different convolutional networks, SketchCNN and ShapeCNN, were employed to extract features of user sketches and 3D shapes, respectively. We applied Inception-ResNetv2 [32] as the backbone of our SketchCNN and used the output of the last avg-pooling layer as the sketch features. We chose Inception-ResNetv2 because the residual

Authorized licensed use limited to: Anhui Normal University. Downloaded on September 24,2021 at 03:44:09 UTC from IEEE Xplore. Restrictions apply.

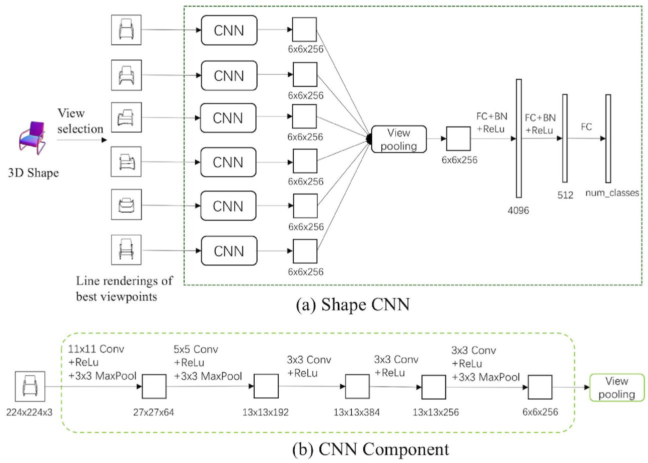


Fig. 6. Our ShapeCNN for feature extraction of 3D shapes: (a) the overall architecture of ShapeCNN, (b) the network architecture of CNN component in ShapeCNN. BN refers to batch normalization [50], and FC refers to fully connected layers. We passed all line renderings from best viewpoints through the CNN component to extract features. These features were fused through the view-pooling layer and further fed into the subsequent layers to obtain a compact shape descriptor.

connections could accelerate the training convergence and had better performance compared with other Inception networks, according to [32]. Following [43], [47], we initialized the parameters of Inception-ResNetv2 by the model pretrained on ImageNet [48]. Since sketches are quite different from the training images in ImageNet [48], we attached a fully connected layer and a softmax classification layer to the end of the last avg-pooling layer to construct a multiclass sketch classifier and fine-tuned the network.

Assume that the sketches can be classified into C classes, given a training sample (s_i, c_i), we trained the SketchCNN using the following classification loss:

$$l_c(s_i, c_i) = -\log\left(\frac{\exp(p(s_i, c_i))}{\sum_{k=1}^{C}\exp(p(s_i, k))}\right) \quad (6)$$

where s_i denotes the i-th training sketch with ground-truth class c_i, and $p(s_i, k)$ denotes the probability of s_i belonging to the k-th class.

For 3D shapes, we applied view-pooling [9] to aggregate features of preference projections and synthesized them into a compact 3D shape descriptor. The architecture of our ShapeCNN is shown in Fig. 6. The convolutional component is the same as in AlexNet [49], and the view-pooling layer is placed close to the last convolutional layer. The feature obtained from view-pooling was fed into subsequent fully connected layers. Similar to SketchCNN, we initialized the convolutional component using a pretrained AlexNet and fine-tuned the network by attaching a classification layer. We used the 512D feature vector before the classification layer as the features of 3D shapes.

C. Cross-Domain Similarity Model

The similarity measure is less investigated in sketch-based shape retrieval. Most existing works projected features of sketches and 3D shapes into the same feature space and computed their similarity based on Euclidean distance or cosine similarity. However, since similarity models are highly related to the properties of features, it is difficult to determine the most appropriate similarity measure. Therefore, we explored a generalized similarity model proposed by Lin [11] to perform cross-domain feature matching. This similarity model combined Mahalanobis distance and cosine similarity under affine transformation, and its parameters could be optimized in a data-driven manner, which encouraged the performance of the cross-domain similarity measure. In [11], the similarity model was applied to compute similarity between images of different modalities. In this paper, we leveraged it to perform a similarity measure between sketches and 3D shapes.

Assume x and y are two samples of different domains. We performed feature extraction and obtained feature vectors $\mathbf{x}$ and $\mathbf{y}$, respectively (assume $\mathbf{x}, \mathbf{y} \in R^k$). Then the similarity model can be expressed as:

$$S(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \mathbf{x}^T & \mathbf{y}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{C} & \mathbf{d} \\ \mathbf{C}^T & \mathbf{B} & \mathbf{e} \\ \mathbf{d}^T & \mathbf{e}^T & f \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ 1 \end{bmatrix} \quad (7)$$

where $\mathbf{A}$ represents the self-correlation of sketch samples, $\mathbf{B}$ represents the self-correlation of 3D samples, and $\mathbf{C}$ represents the cross-domain correlation between sketch samples and 3D samples. $\mathbf{A}$ and $\mathbf{B}$ are only dependent on their own domains, while $\mathbf{C}$ is affected by both domains. As derived in [11], $\mathbf{A}$ and $\mathbf{B}$ are restricted as semidefinite matrices, while $\mathbf{C}$ is less restricted ($\mathbf{A}, \mathbf{B}, \mathbf{C} \in R^{k\times k}$), $\mathbf{d}$ and $\mathbf{e}$ are parameter vectors corresponding to $\mathbf{x}$ and $\mathbf{y}$, respectively ($\mathbf{d}, \mathbf{e} \in R^k$), and f is a scalar. We further factorized $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ as:

$$\mathbf{A} = \mathbf{L_A}^T\mathbf{L_A}$$
$$\mathbf{B} = \mathbf{L_B}^T\mathbf{L_B}$$
$$\mathbf{C} = -\mathbf{L_C^x}^T\mathbf{L_C^y} \quad (8)$$

(8) was substituted into (7) to obtain the generalized similarity model:

$$\begin{aligned} \tilde{S}(\mathbf{x}, \mathbf{y}) &= \begin{bmatrix} \mathbf{x}^T & \mathbf{y}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{L_A}^T\mathbf{L_A} & -\mathbf{L_C^x}^T\mathbf{L_C^y} & \mathbf{d} \\ -\mathbf{L_C^x}^T\mathbf{L_C^y}^T & \mathbf{L_B}^T\mathbf{L_B} & \mathbf{e} \\ \mathbf{d}^T & \mathbf{e}^T & f \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ 1 \end{bmatrix} \\ &= \|\mathbf{L_A}\mathbf{x}\|^2 + \|\mathbf{L_B}\mathbf{y}\|^2 + 2\mathbf{d}^T\mathbf{x} \\ &\quad - 2(\mathbf{L_C^x}\mathbf{x})^T(\mathbf{L_C^y}\mathbf{y}) + 2\mathbf{e}^T\mathbf{y} + f \end{aligned} \quad (9)$$

The formulation in (9) represents the final form of the similarity model, and its value is used to obtain the similarity. Note that this model is asymmetric, meaning $\tilde{S}(\mathbf{x}, \mathbf{y}) \neq \tilde{S}(\mathbf{y}, \mathbf{x})$, which is reasonable because $\mathbf{x}$ and $\mathbf{y}$ belong to different domains and thus have different parameters. Since f has no influence on training, we set it to -1.9 throughout this paper as in [11].

To obtain the parameters of the similarity model, we incorporated the similarity measure matrix into a metric network and optimized its parameters using a data-driven manner. We first

Authorized licensed use limited to: Anhui Normal University. Downloaded on September 24,2021 at 03:44:09 UTC from IEEE Xplore. Restrictions apply.

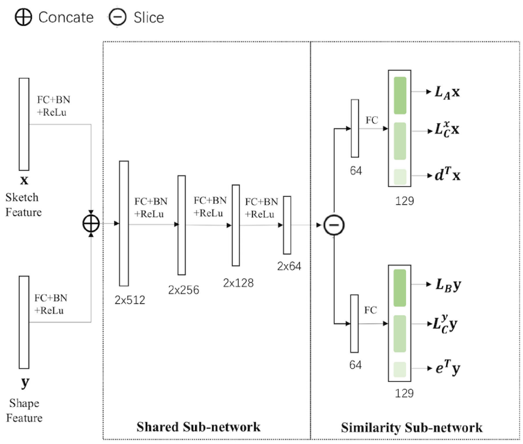


Fig. 7. The architecture of our metric network. All parameters in the shared subnetwork were shared between different domains except batch normalization layer (BN), while the similarity subnetwork had different parameters for different domains. The output of the metric network formed the parameters of the similarity model, and the similarity was further calculated by (9).

projected the features of sketches and 3D shapes into 512D feature vectors and constructed a metric network to compute the parameters of the similarity model. Our metric network is shown in Fig. 7. It is composed of two modules: a shared subnetwork and a similarity subnetwork. The shared subnetwork consisted of four fully connected layers, each of which is followed by ReLU activation and batch normalization [50]. The parameters of all fully connected layers are shared between sketches and 3D shapes except batch normalization [50], since the latter performs normalization according to the mean and variance in samples, which should not be shared between different modalities. The similarity subnetwork has only two fully connected layers for two domains and is aimed at computing similarity components, such as $\mathbf{L_A x}$. Once the outputs from the metric network were obtained, we measured the similarity between sketches and 3D shapes using (9).

Since the similarity components of 3D shapes are independent of sketches, we precomputed the similarity components offline, and the similarity matching was efficient.

D. Optimization

We first fine-tuned two feature networks, as demonstrated in subsection B and employed them to extract features of sketches and 3D shapes. These feature vectors were saved offline and used to train our metric network.

1) Metric Network: Contrastive loss [2], [11], [15], [43], [51], [52] and triplet loss [12], [14], [16], [52]–[55] are two common methods in metric learning. Compared with contrastive loss, triplet loss is more stable and can achieve better results, as shown in previous works [52], [54], [55]. Hence, we used triplet loss to train our metric network.

Given a triplet sample in the form of $(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-)$, in which $\mathbf{x}$ is the sketch feature, $\mathbf{y}^+$ is the shape features of the same class as $\mathbf{x}$ and $\mathbf{y}^-$ is of different class to $\mathbf{x}$, triplet loss can be defined as follows:

$$\tilde{l}_{triplet}(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-) = \max(\tilde{S}(\mathbf{x}, \mathbf{y}^+) - \tilde{S}(\mathbf{x}, \mathbf{y}^-) + C, 0) \tag{10}$$

where C indicates the margin that is enforced between positive and negative pairs. We denote the output of the metric network as:

$$\begin{aligned}\tilde{x} &\triangleq [\mathbf{L_A x} \quad \mathbf{L_C^x x} \quad \mathbf{d}^T\mathbf{x}]^T\\ \tilde{y}^+ &\triangleq [\mathbf{L_B y}^+ \quad \mathbf{L_C^y y}^+ \quad \mathbf{e}^T\mathbf{y}^+]^T\\ \tilde{y}^- &\triangleq [\mathbf{L_B y}^- \quad \mathbf{L_C^y y}^- \quad \mathbf{e}^T\mathbf{y}^-]^T\end{aligned} \tag{11}$$

To incorporate these output vectors into (9), we introduce three auxiliary matrices as in [11]:

$$\begin{aligned}\mathbf{P_1} &= [\mathbf{I}^{r\times r} \quad \mathbf{0}^{r\times(r+1)}]\\ \mathbf{P_2} &= [\mathbf{0}^{r\times r} \quad \mathbf{I}^{r\times r} \quad \mathbf{0}^{r\times 1}]\\ \mathbf{P_3} &= [\mathbf{0}^{1\times 2r} \quad 1^{1\times 1}]^T\end{aligned} \tag{12}$$

where r is the dimension of the output vector from the shared subnetwork, $\mathbf{I}$ indicates the identity matrix and $\mathbf{0}$ indicates the zero matrix. Thus, the original similarity model can be rewritten as:

$$\begin{aligned}\tilde{S}(\mathbf{x}, \mathbf{y}^+) = (\mathbf{P_1}\tilde{x})^T\mathbf{P_1}\tilde{x} + (\mathbf{P_1}\tilde{y}^+)^T\mathbf{P_1}\tilde{y}^+\\ - 2(\mathbf{P_2}\tilde{x})^T\mathbf{P_2}\tilde{y}^+ + 2\mathbf{P_3}^T\tilde{x} + 2\mathbf{P_3}^T\tilde{y}^+ + f\end{aligned} \tag{13}$$

$$\begin{aligned}\tilde{S}(\mathbf{x}, \mathbf{y}^-) = (\mathbf{P_1}\tilde{x})^T\mathbf{P_1}\tilde{x} + (\mathbf{P_1}\tilde{y}^-)^T\mathbf{P_1}\tilde{y}^-\\ - 2(\mathbf{P_2}\tilde{x})^T\mathbf{P_2}\tilde{y}^- + 2\mathbf{P_3}^T\tilde{x} + 2\mathbf{P_3}^T\tilde{y}^- + f\end{aligned} \tag{14}$$

Substitute (13) and (14) into (10), then our objective function can be represented as:

$$\tilde{l}_{triplet}(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-) = \max(L(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-) + C, 0) \tag{15}$$

where

$$\begin{aligned}L(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-) = (\mathbf{P_1}\tilde{y}^+)^T\mathbf{P_1}\tilde{y}^+ - 2(\mathbf{P_2}\tilde{x})^T\mathbf{P_2}\tilde{y}^+ + 2\mathbf{P_3}^T\tilde{y}^+\\ - (\mathbf{P_1}\tilde{y}^-)^T\mathbf{P_1}\tilde{y}^- + 2(\mathbf{P_2}\tilde{x})^T\mathbf{P_2}\tilde{y}^- - 2\mathbf{P_3}^T\tilde{y}^-\end{aligned} \tag{16}$$

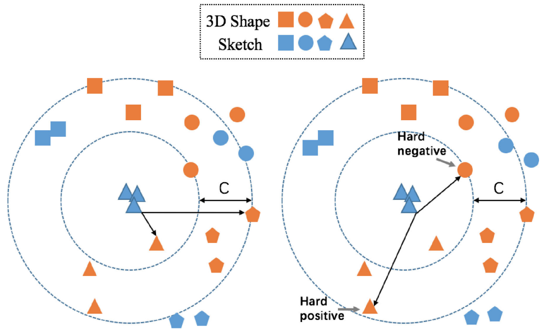
Assume that the parameters of the metric network are Ω, the gradient of the objective function with respect to Ω is

$$\frac{\partial\tilde{l}_{triplet}}{\partial\Omega} = \frac{\partial L}{\partial\Omega} I_{\tilde{S}(\mathbf{x},\mathbf{y}^+) - \tilde{S}(\mathbf{x},\mathbf{y}^-) + C > 0} \tag{17}$$

where $I_{condition} = 1$ if $condition$ is true, otherwise $I_{condition} = 0$. Since there exist two network branches corresponding to sketch and 3D shape, we denote parameters of the sketch branch as Ω_1 and parameters of the 3D shape branch as Ω_2. For the sketch branch, the gradient can be calculated as

$$\begin{aligned}\frac{\partial L}{\partial\Omega_1} &= \frac{\partial L}{\partial\tilde{x}}\frac{\partial\tilde{x}}{\partial\Omega_1}\\ &= \left(2\mathbf{P_2}^T\mathbf{P_2}\tilde{y}^- - 2\mathbf{P_2}^T\mathbf{P_2}\tilde{y}^+\right)\frac{\partial\tilde{x}}{\partial\Omega_1}\end{aligned} \tag{18}$$

Authorized licensed use limited to: Anhui Normal University. Downloaded on September 24,2021 at 03:44:09 UTC from IEEE Xplore. Restrictions apply.



(a) Normal triplet sample (b) Hard sample mining

Fig. 8. Illustration of hard sample mining. The orange color indicates the 3D shape sample, and the blue color indicates the sketch sample. Different shapes of icons represent different classes of 3D shapes and sketches. C is the margin in equation (20).

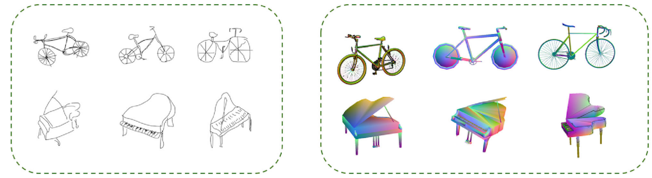
Similarly, the gradient for the shape branch is

$$\begin{aligned}\frac{\partial L}{\partial \Omega_2} &= \frac{\partial L}{\partial \tilde{y}^+}\frac{\partial \tilde{y}^+}{\partial \Omega_2} + \frac{\partial L}{\partial \tilde{y}^-}\frac{\partial \tilde{y}^-}{\partial \Omega_2} \\ &= \left(2\mathbf{P_1}^T\mathbf{P_1}\tilde{y}^+ - 2(\mathbf{P_2}\tilde{x})^T\mathbf{P_2} + 2\mathbf{P_3}^T\right)\frac{\partial \tilde{y}^+}{\partial \Omega_2} \\ &\quad + \left(2(\mathbf{P_2}\tilde{x})^T\mathbf{P_2} - 2\mathbf{P_1}^T\mathbf{P_1}\tilde{y}^- - 2\mathbf{P_3}^T\right)\frac{\partial \tilde{y}^-}{\partial \Omega_2} \quad (19)\end{aligned}$$

Hence, the objective function in (15) can be integrated into the backpropagation.

2) Online Hard Sample Mining: To accelerate convergence of the network, online hard sample mining in [55] was applied to generate triplet samples. As shown in Fig. 8, a normal triplet randomly selects positive and negative samples for the anchor point in a batch. Hard sample mining aims to find the hardest positive (negative) sample that has the maximum (minimum) similarity distance from the anchor point in a batch. Since hard samples can provide larger gradients during training, hard sample mining helps to accelerate convergence and eliminate zero gradients when positive and negative samples already satisfy margin constraints in (15) in the normal triplet sample. In our hard sample mining, we first randomly selected N classes and randomly sampled K sketch features for each class to build sketch batches $\mathbf{X} = \{\mathbf{x}_{1,1}, \ldots, \mathbf{x}_{1,K}, \ldots, \mathbf{x}_{N,1}, \ldots, \mathbf{x}_{N,K}\}$ of size $N \times K$. Similarly, a 3D shape batch $\mathbf{Y} = \{\mathbf{y}_{1,1}, \ldots, \mathbf{y}_{1,K}, \ldots, \mathbf{y}_{N,1}, \ldots, \mathbf{y}_{N,K}\}$ can be created. Then, for each sketch sample $\mathbf{x}_{i,j}$, we select the hardest positive sample $\mathbf{y}_{i,p}$, which has the largest similarity distance to $\mathbf{x}_{i,j}$, from the 3D shape samples of the same class as $\mathbf{x}_{i,j}$. Additionally, we select the hardest negative sample, which has the smallest similarity distance to $\mathbf{x}_{i,j}$ from shape samples of different classes to $\mathbf{x}_{i,j}$. Thus, we obtain a triplet batch of size $N \times K$. By using online hard sample mining, our objective function can be represented as

$$\begin{aligned}\tilde{L}_{triplet}(\mathbf{X}, \mathbf{Y}) = \frac{1}{NK}\sum_{i}^{N}\sum_{j}^{K}\max(\tilde{S}(\mathbf{x}_{i,j}, \mathbf{y}_{i,p}) \\ - \tilde{S}(\mathbf{x}_{i,j}, \mathbf{y}_{k,n}) + C, 0) \quad (20)\end{aligned}$$



(a) sketch (b) 3D shape

Fig. 9. Sketch examples and corresponding 3D shapes from SHREC'13 dataset.

where

$$\mathbf{y}_{i,p} = \operatorname*{argmax}_{i \in \{1,\ldots,N\}, p \in \{1,\ldots,K\}} \tilde{S}(\mathbf{x}_{i,j}, \mathbf{y}_{i,p})$$

$$\mathbf{y}_{k,n} = \operatorname*{argmin}_{k \in \{1,\ldots,N\}, k \neq i, n \in \{1,\ldots,K\}} \tilde{S}(\mathbf{x}_{i,j}, \mathbf{y}_{k,n})$$

E. Training Procedure

We trained our networks with 3D models and human sketches in the following three steps.

- The first step is **best view selection**. Before training, we rendered 3D models into line renderings. Then, we computed the best viewpoints by view selection algorithm (Algorithm 1) with training sketches and line renderings.
- The second step is **feature extraction**. In this step, SketchCNN was used to extract sketch features from user sketches, whereas ShapeCNN was used to extract shape features from line renderings of best viewpoints. Both SketchCNN and ShapeCNN were saved offline.
- The third step is **cross-domain similarity model training**. The features of sketches and 3D shapes were fed into the cross-domain similarity model, which is a metric network to measure the similarity distance, as shown in Fig. 7.

IV. EXPERIMENTS

In this section, we evaluated our method on SHREC'13 [19], [42] and SHREC'14 [19], [20] and compared it to state-of-the-art methods. We further conducted experiments to analyze the benefits of best view selection, generalized similarity models and online hard sample mining.

A. Datasets and Experimental Settings

We tested our method on SHREC'13 and SHREC'14 sketch track benchmark datasets. SHREC'13 was constructed based on a human sketch dataset [24] and Princeton shape benchmark [56]. It contains 7,200 sketches and 1,258 shapes, and a total of 90 classes. Each class consists of 80 sketches: 50 samples for training and 30 samples for testing. The construction of SHREC'14 is similar to SHREC'13. It contains 13,680 sketches and 8,987 shapes, and a total of 171 classes. Following the same configuration as SHREC'13, the sketches for each class were split into 50 training samples and 30 testing samples. We showed some examples of SHREC'13 in Fig. 9. Due to the large variations in sketches, the retrieval task is very challenging.

Authorized licensed use limited to: Anhui Normal University. Downloaded on September 24,2021 at 03:44:09 UTC from IEEE Xplore. Restrictions apply.

In our experiment, we applied the configuration of the SHREC'13 and SHREC'14 datasets. For each class, we used 50 sketches for training and the remaining 30 sketches for testing. In the best view selection procedure, we leveraged all the training sketches to compute preference viewpoints for 3D shapes. We ranked the line renderings of each 3D shape and selected the top 6 preference line renderings to perform feature extraction for 3D shapes. The training sketches and preference line renderings were leveraged to fine-tune SketchCNN and ShapeCNN using the classification loss in (6). The output dimension of the shared subnetwork in our metric network was set to 64, and the margin in (20) was set to 5. We employed the Adam [57] optimizer to train our network. The initial learning rate was 0.01, and decayed exponentially after 5 epochs. To fine-tune SketchCNN and ShapeCNN, their batch sizes were set to 24 and 20, respectively. We adopted the Adam [57] optimizer to minimize classification loss in (6) and triplet loss in (20). Once the classifiers were optimized, we saved the features offline and applied them to perform metric learning. For the metric network, we set N and K in (20) in online hard sample mining to 16 and 4, respectively. Since the number of training sketches is limited, we applied the sketch deformation technique in [29] along with random flip and rotation to augment the train samples.

We reported our results using the following evaluation methods: precision-recall curve, nearest neighbor (NN), first tier (FT), second tier (ST), E-measure (E), discounted cumulative gain (DCG) and mean average precision (mAP).

B. Comparative Results

We compared our method with other state-of-the-art methods on the SHREC'13 and SHREC'14 datasets. We named our method **BV-CDSM** (best view selection and cross-domain similarity model). To reveal the effect of the best view selection, we replaced the best view selection with a uniform viewpoint sampling method, and all other factors remained unchanged, which is denoted as **BV-CDSM(uniform)** in our experiment. We removed our similarity model and used the output of the shared subnetwork to compute similarity based on Euclidean distance and cosine similarity, which were denoted as **BV-CDSM(Euclidean)** and **BV-CDSM(cosine)**, respectively. These comparative settings aimed to explore the differences with other state-of-the-art methods and causes for performance promotion. We also explored the performance of the number of projections and online hard sample mining. To further understand the effect of best view selection, we conducted an experiment for 3D shape recognition.

1) SHREC'13 Dataset: We compared our method with CDMR [4], HOG-SIL [42], SBR-2D-3D [41], SBR-VC [42], Siamese network [2], LWBR [43], DCHML [58] and DCA [44] on the SHREC'13 benchmark. The comparison results of NN, FT, ST, E, DCG and mAP are shown in Table I. It is noticeable that approaches based on deep learning (Siamese, LWBR, DCHML, DCA and ours) are superior to those based on handcrafted features. Compared with other methods utilizing deep learning, our method can also achieve higher accuracy. Compared with DCA, which obtained the best accuracy among

TABLE I
RETRIEVAL RESULTS ON SHREC'13 DATASET

Method	NN	FT	ST	E	DCG	mAP
CDMR[4]	0.279	0.203	0.296	0.166	0.458	0.250
HOG-SIL[42]	0.110	0.069	0.107	0.061	0.307	0.086
SBR-2D-3D[41]	0.132	0.077	0.124	0.074	0.327	0.095
SBR-VC[42]	0.164	0.097	0.149	0.085	0.348	0.116
Siamese[2]	0.405	0.403	0.548	0.287	0.607	0.469
LWBR[43]	0.712	0.725	0.785	0.369	0.814	0.752
DCHML[58]	0.730	0.715	0.773	0.368	0.816	0.744
DCA[44]	0.783	0.796	0.829	0.376	0.856	0.813
BV-CDSM(cosine)	0.764	0.780	0.833	0.387	0.851	0.803
BV-CDSM(Euclidean)	0.757	0.772	0.822	0.390	0.845	0.794
BV-CDSM(uniform)	0.789	0.806	0.854	0.396	0.869	0.828
BV-CDSM	**0.811**	**0.830**	**0.878**	**0.406**	**0.885**	**0.849**

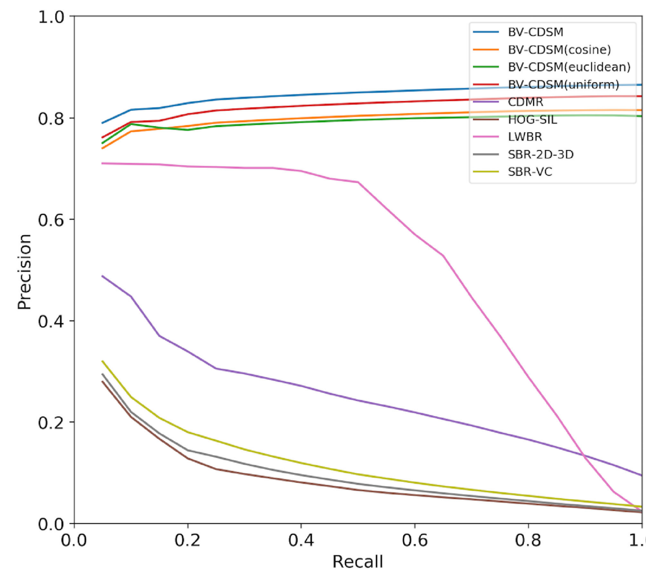


Fig. 10. The precision-recall curve for CDMR, HOG-SIL, SBR-2D-3D, SBR-VC, LWBR and our method on the SHREC'13 dataset.

previous methods, our approach outperformed it by 2.8% in NN, 3.4% in FT, 4.9% in ST, 3.0% in E, 2.9% in DCG and 3.6% in mAP. The precision-recall curve in Fig. 10 demonstrates the advantages of the proposed method since our approach achieved higher precision even when recall increased to 1. For LWBR, it did not take advantage of the sketch prior and the cross-domain similarity measurement. It sampled projections from the 3D shape uniformly and applied Euclidean distance as the similarity measurement. Compared with BV-CDSM(Euclidean) and (BV-CDSM(uniform)), our PR curve performed better than LWBR. This implies that the best view selection model and the cross-domain similarity model are conducive to improving the accuracy of the algorithm. We also reported the results of the uniform viewpoint sampling method and other similarity models. Our best view selection can yield better performance and outperform the uniform viewpoint sampling method by 2.1% in mAP. This is because view selection can promote the feature learning of 3D shapes, which in turn improves the retrieval performance. Regarding the similarity measure, our approach can both advance cosine similarity and Euclidean distance on all evaluation methods, which shows the effect of the cross-domain similarity model. As a result, the view selection model and the

Authorized licensed use limited to: Anhui Normal University. Downloaded on September 24,2021 at 03:44:09 UTC from IEEE Xplore. Restrictions apply.

TABLE II
RETRIEVAL RESULTS ON SHREC'14 DATASET

Method	NN	FT	ST	E	DCG	mAP
CDMR[4]	0.109	0.057	0.089	0.041	0.328	0.054
SBR-VC[42]	0.095	0.050	0.081	0.037	0.319	0.050
SCMR-OPHOG[59]	0.160	0.115	0.170	0.079	0.376	0.131
BOF-JESC[60]	0.086	0.043	0.068	0.030	0.310	0.131
Siamese[2]	0.239	0.212	0.316	0.140	0.496	0.228
LWBR[43]	0.403	0.378	0.455	0.236	0.581	0.401
DCHML[58]	0.403	0.329	0.394	0.201	0.544	0.336
MVPR[47]	0.546	0.506	0.642	0.301	0.715	0.543
DCA[44]	0.770	0.789	0.823	0.398	0.859	0.803
BV-CDSM(cosine)	0.747	0.750	0.795	0.385	0.842	0.766
BV-CDSM(euclidean)	0.743	0.720	0.776	0.379	0.831	0.739
BV-CDSM(uniform)	0.736	0.709	0.790	0.389	0.828	0.739
BV-CDSM	**0.784**	**0.796**	**0.831**	**0.404**	**0.868**	**0.809**

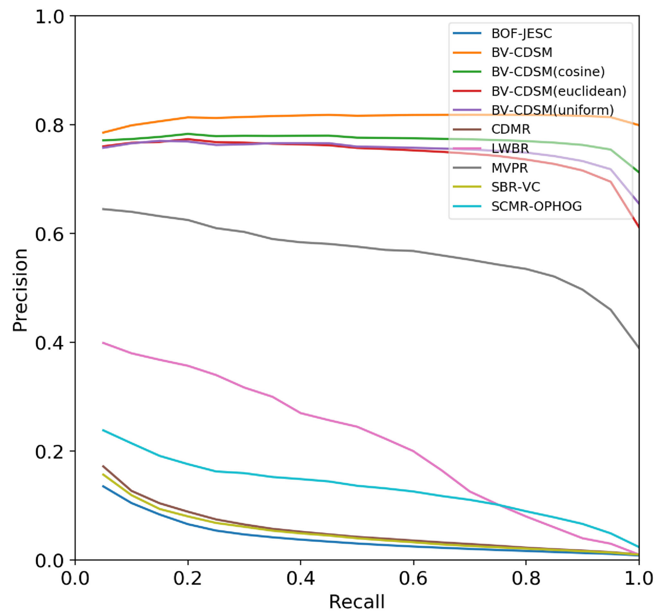
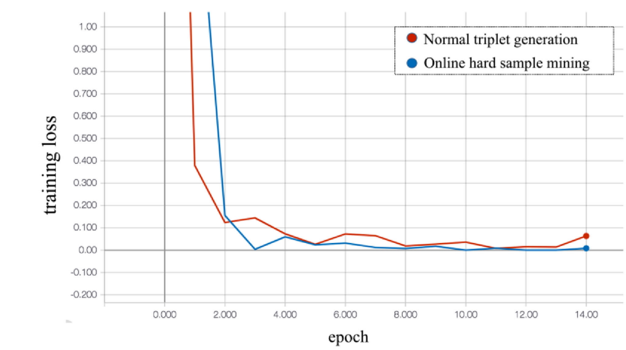


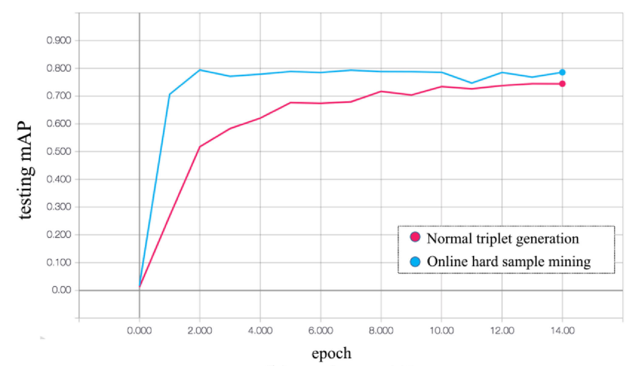
Fig. 11. The precision-recall curve for CDMR, HOG-SIL, SBR-2D-3D, SBR-VC, LWBR, MVPR and our method on SHREC'14 dataset.

TABLE III
THE EFFECT OF DIFFERENT PROJECTION NUMBERS FOR OUR METHOD

Projection	NN	FT	ST	E-Measure	DCG	mAP
1	0.711	0.700	0.769	0.375	0.838	0.717
2	0.732	0.726	0.787	0.385	0.849	0.742
4	0.760	0.774	0.808	0.398	0.865	0.783
6	0.784	0.796	0.831	0.404	0.868	0.809
8	0.796	0.821	0.849	0.415	0.890	0.829
10	0.801	0.826	0.853	0.417	0.894	0.832
12	0.805	0.832	0.859	0.421	0.896	0.834



(a) training loss



(b) testing mAP

Fig. 12. The training loss and testing mAP of online hard sample mining and normal triplet generation on SHREC'14 dataset.

cross-domain similarity model both played significant roles in promoting the retrieval performance.

2) SHREC'14 Dataset: We also evaluated our method on the SHREC'14 dataset and compared it with CDMR [4], SBR-VC [42], SCMR-OPHOG [59], BOF-JESC [60], Siamese [2], LWBR [43], DCHML [58], MVPR [47] and DCA [44]. The results of NN, FT, ST, E, DCG and mAP are shown in Table II. Due to the complexity of the SHREC'14 dataset, most methods, including deep learning-based approaches, could not achieve high performance. DCA significantly outperformed other methods due to its important-aware feature learning and cross-modality transformation. The performance of our method is approximate to DCA. However, due to the view selection, only 6 projections were required in our approach, which is more computationally efficient compared to 12 projections in DCA. We presented the precision-recall curve in Fig. 11 and further validate the robustness of our method since the precision remains nearly unchanged as recall increases to 0.9. Similarly, we also conducted experiments to justify the effectiveness of our best view selection and generalized similarity model. As shown in Table II, both the best view selection and cross-domain similarity model obtained better performance compared with uniform viewpoint sampling and other similarity models.

In addition, we evaluated the effect of the projection number on the SHREC'14 dataset. We ranked the line renderings according to best view selection and selected the top-K renderings for the experiment. The retrieval results are shown in Table III. It can be observed that retrieval performance improved as the projection number increased. This is reasonable since more projections provide richer information about 3D shapes, therefore leading to better retrieval results. However, we found that the retrieval performance did not improve when the projection number reached beyond 8. This is due to the information redundancy of selected viewpoints and the increasing selections of ambiguous projections.

Furthermore, we invalidated the performance of online hard sample mining. We proposed the training loss and testing mAP of online hard sample mining and normal triplet generation on

Authorized licensed use limited to: Anhui Normal University. Downloaded on September 24,2021 at 03:44:09 UTC from IEEE Xplore. Restrictions apply.

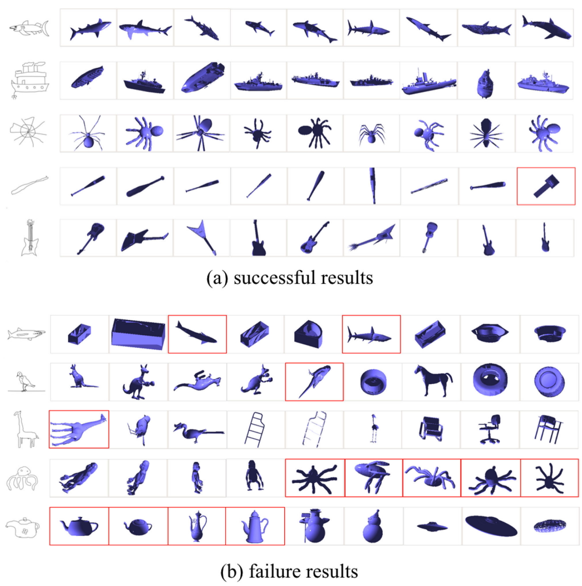


Fig. 13. Some retrieval results of the proposed method: (a) the red bounding box denotes the incorrect retrievals, (b) the red bounding box denotes the correct retrievals.

the SHREC'14 dataset in Fig. 12. Note that the network converged faster in online hard sample mining than normal triplet generation. Additionally, online hard sample mining achieved a higher mAP when the network was finally optimized since the network had a more powerful discriminative ability to those outliers, which is difficult to mine in normal triplet generation.

We present some successful and failed retrieval results of our method in Fig. 13. The first column shows query sketches from different classes, and each row shows the top 9 retrieval results. One can note that our method may have failed to retrieve correct results due to the large within-class variations of sketches. For example, the second row in Fig. 13(b) shows a query sketch of a standing bird; however, since it has a similar look as the kangaroo, our method failed to recognize it and returned 3D shapes of a kangaroo as the top-4 results.

3) 3D Shape Recognition: We designed a 3D shape recognition experiment using the 3D shapes in the SHREC'14 dataset to verify the effectiveness of the proposed best view selection for feature learning of 3D shapes. The number of 3D shapes varies among different classes in the SHREC'14 dataset. We selected those classes with more than (or equal to) 80 shapes, and for each class, we randomly chose 80 samples, among which 60 were used for training and 20 were used for testing. We adopted the same network, as in Fig. 6. The recognition accuracy for our view selection method (**Ours**), uniform viewpoints sampling method (**Uniform**) and Zhao's method [10] (**Zhao**) are shown in Table IV. Note that under the condition of 10 projections, our best view selection method and Zhao's method [10] outperformed the uniform viewpoint sampling method by 9.2% and 7.6%, respectively, which means that view selection can significantly boost the performance of feature learning of 3D shapes. With 4 projections, our method achieved higher accuracy, higher than the uniform viewpoint sampling method (with 10 projections) by 1.5%, which further validates the

TABLE IV
THE ACCURACY OF DIFFERENT VIEW SELECTION METHODS FOR 3D SHAPE RECOGNITION

Accuracy / Method / Projection	Uniform	Zhao[10]	Ours
2	0.738	0.817	**0.821**
4	0.783	**0.863**	0.857
6	0.814	0.876	**0.883**
8	0.825	0.893	**0.912**
10	0.842	0.918	**0.934**

effect of our approach. Our method outperformed Zhao [10] by 1.6% when the projection number was 10. However, we found that when the projection number was reduced to 4, Zhao [10] was slightly higher than ours by 0.6%. A great deal of information about 3D shapes was lost when the projection number was limited. Hence, representation learning is not very robust even for best view selection. In addition, we learned that recognition accuracy increases when more projections are added. This is consistent with the result in Table III, meaning that more projections can preserve more information about 3D shapes, which has more benefits to feature learning.

V. CONCLUSION

In this paper, we proposed a best view selection method to select reasonable projections and applied MVCNN to learn a compact feature for 3D shapes. Because our proposed method can select projections that characterize 3D shapes, we achieved a more robust representation learning for 3D shapes. To indicate the apparent discrepancy between sketches and 3D shapes, a cross-domain similarity model was leveraged to calculate the similarity distance between features of sketches and 3D shapes. We employed triplet loss and online hard sample mining to train our model. Extensive experiments on SHREC'13 and SHREC'14 benchmarks demonstrate that both best view selection and cross-domain similarity measures can achieve promising results.

There are still certain limitations to our work. First, since we did not deeply study the feature representation of sketches, our best view selection algorithm is not very robust to the high abstraction of sketches and could cause some unexpected viewpoints. Second, the training sketches are needed for our best view selection module, which could be impractical for 3D shapes of unknown classes. Additionally, without high-performance equipment support, our system is not end-to-end trainable. For deployment, it needs to be implemented on the server end. All offline 3D features are stored in the server, and the proposed method can be deployed on that server. Furthermore, the retrieval results are highly related to the recognition of sketches; therefore, to achieve high accuracy against the large within-class variations of sketches, we need to integrate both low and high quality of all sketch features to the training dataset. We will investigate the stroke order of sketch drawings to enhance the feature representations of sketches. Moreover, we will study hashing features of sketches and 3D shapes into binary codes [61] to reduce memory storage and boost computational efficiency.

Authorized licensed use limited to: Anhui Normal University. Downloaded on September 24,2021 at 03:44:09 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa, "Sketch-based shape retrieval," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 31:1–31:10, 2012.

[2] F. Wang, L. Kang, and Y. Li, "Sketch-based 3D shape retrieval using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.,* Boston, MA, USA, Jun. 7–12, 2015, pp. 1875–1883.

[3] J. M. Saavedra, B. Bustos, T. Schreck, S. M. Yoon, and M. Scherer, "Sketch-based 3D model retrieval using keyshapes for global and local representation," in *Proc. Eurographics Workshop 3D Object Retrieval*, Cagliari, Italy, May 13, 2012, pp. 47–50.

[4] T. Furuya and R. Ohbuchi, "Ranking on cross-domain manifold for sketch-based 3D model retrieval," in *Proc. Int. Conf. Cyberworlds*, Yokohama, Japan, Oct. 21–23, 2013, pp. 274–281.

[5] K. Eissen and R. Steur, *Sketching: The Basics*. Amsterdam: BIS Publishers, Aug. 2011.

[6] S. Bae, R. Balakrishnan, and K. Singh, "Ilovesketch: As-natural-as-possible sketching system for creating 3d curve models," in *Proc. 21st Annu. ACM Symp. User Interface Softw. Technol.*, Monterey, CA, USA, Oct. 19–22, S. B. Cousins and M. Beaudouin-Lafon, Eds., ACM, 2008, pp. 151–160. [Online]. Available: https://doi.org/10.1145/1449715.1449740

[7] C. Shao, A. Bousseau, A. Sheffer, and K. Singh, "CrossShade: Shading concept sketches using cross-section curves," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 45:1–45:11, 2012. [Online]. Available: https://doi.org/10.1145/2185520.2185541

[8] B. Xu *et al.*, "True2Form: 3D curve networks from 2D sketches via selective regularization," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 131:1–131:13, 2014. [Online]. Available: https://doi.org/10.1145/2601097.2601128

[9] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vision*, Santiago, Chile, Dec. 7–13, 2015, pp. 945–953.

[10] L. Zhao, S. Liang, J. Jia, and Y. Wei, "Learning best views of 3D shapes from sketch contour," *Vis. Comput.*, vol. 31, no. 6–8, pp. 765–774, 2015.

[11] L. Lin, G. Wang, W. Zuo, X. Feng, and L. Zhang, "Cross-domain visual matching via generalized similarity measure and feature learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1089–1102, Jun. 2017.

[12] S. Ding, L. Lin, G. Wang, and H. Chao, "Deep feature learning with relative distance comparison for person re-identification," *Pattern Recognit.*, vol. 48, no. 10, pp. 2993–3003, 2015.

[13] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "Sketch-based image retrieval: Benchmark and bag-of-features descriptors," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 11, pp. 1624–1636, Nov. 2011.

[14] Q. Yu *et al.*, "Sketch me that shoe," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Las Vegas, NV, USA, Jun. 27–30, 2016, pp. 799–807.

[15] Y. Qi, Y. Song, H. Zhang, and J. Liu, "Sketch-based image retrieval via siamese convolutional neural network," in *Proc. IEEE Int. Conf. Image Process.,* Phoenix, AZ, USA, Sep. 25–28, 2016, pp. 2460–2464.

[16] J. Song, Q. Yu, Y. Song, T. Xiang, and T. M. Hospedales, "Deep spatial-semantic attention for fine-grained sketch-based image retrieval," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 5552–5561.

[17] L. Nie *et al.*, "Enhancing micro-video understanding by harnessing external sounds," in *Proc. ACM Multimedia Conf.*, Mountain View, CA, USA, Oct. 23–27, 2017, Q. Liu, R. Lienhart, H. Wang, S. K. Chen, S. Boll, Y. P. Chen, G. Friedland, J. Li, and S. Yan, Eds. ACM, 2017, pp. 1192–1200. [Online]. Available: https://doi.org/10.1145/3123266.3123313

[18] L. Nie, X. Song, and T. Chua, *Learning from Multiple Social Networks*, (Synthesis Lectures on Information Concepts, Retrieval, and Services). Morgan & Claypool Publishers, 2016. [Online]. Available: https://doi.org/10.2200/S00714ED1V01Y201603ICR048

[19] B. Li *et al.*, "SHREC'13 Track: Large scale sketch-based 3D shape retrieval," in *Proc. Eurographics Workshop 3D Object Retrieval*, Girona, Spain, 2013, pp. 89–96.

[20] B. Li *et al.*, "SHREC'14 Track: Extended large scale sketch-based 3D shape retrieval," in *Proc. Eurographics Workshop 3D Object Retrieval*, Strasbourg, France, 2014, pp. 121–130.

[21] Z. Yasseen, A. Verroust-Blondet, and A. H. Nasri, "View selection for sketch-based 3D model retrieval using vis. part shape description," *Vis. Comput.*, vol. 33, no. 5, pp. 565–583, 2017.

[22] D. Lu, H. Ma, and H. Fu, "Efficient sketch-based 3D shape retrieval via view selection," in *Proc. Advances Multimedia Inf. Process. - PCM - 14th Pacific-Rim Conf. Multimedia* (Lecture Notes in Computer Science, 8294), Nanjing, China, Springer, Dec. 13–16, 2013, pp. 396–407.

[23] F. Radenovic, G. Tolias, and O. Chum, "Deep shape matching," in *Proceedings of the 15th European Conference on Computer Vision* (Lecture Notes in Computer Science, 11209). Munich, Germany: Springer, Sep. 8–14, 2018, Part V, pp. 774–791.

[24] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM Trans. Graph.*, vol. 31, no. 4, pp. 44:1–44:10, 2012.

[25] R. G. Schneider and T. Tuytelaars, "Sketch classification and classification-driven analysis using fisher vectors," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 174:1–174:9, 2014.

[26] Y. Li, T. M. Hospedales, Y. Song, and S. Gong, "Free-hand sketch recognition by multi-kernel feature learning," *Comput. Vision Image Understanding*, vol. 137, pp. 1–11, 2015.

[27] O. Seddati, S. Dupont, and S. Mahmoudi, "DeepSketch: Deep convolutional neural networks for sketch recognition and similarity search," in *Proc. IEEE 13th Int. Workshop Content-Based Multimedia Indexing, CBMI*, Prague, Czech Republic, Jun. 10–12, 2015, pp. 1–6.

[28] S. Dupont, O. Seddati, and S. Mahmoudi, "DeepSketch 2: Deep convolutional neural networks for partial sketch recognition," in *Proc. IEEE 14th Int. Workshop Content-Based Multimedia Indexing*, CBMI 2016, Bucharest, Romania, Jun. 15–17, 2016, pp. 1–6.

[29] Q. Yu *et al.*, "Sketch-a-Net: A deep neural network that beats humans," *Int. J. Comput. Vision*, vol. 122, no. 3, pp. 411–425, 2017.

[30] Q. Yu, Y. Yang, Y. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-Net that beats humans," in *Proc. Brit. Mach. Vision Conf.*, Swansea, U.K., BMVA Press, Sep. 7–10, 2015, pp. 7.1–7.12.

[31] X. Zhang *et al.*, "A hybrid convolutional neural network for sketch recognition," *Pattern Recognit. Lett.*, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865519300078

[32] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA. AAAI Press, Feb. 4–9, 2017, pp. 4278–4284.

[33] H. Tabia and H. Laga, "Covariance-based descriptors for efficient 3D shape matching, retrieval, and classification," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1591–1603, Sep. 2015. [Online]. Available: https://doi.org/10.1109/TMM.2015.2457676

[34] J. Xie, G. Dai, and Y. Fang, "Deep multimetric learning for shape-based 3D model retrieval," *IEEE Trans. Multimedia*, vol. 19, no. 11, pp. 2463–2474, Nov. 2017. [Online]. Available: https://doi.org/10.1109/TMM.2017.2698200

[35] Z. Wu *et al.*, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 7–12, 2015, pp. 1912–1920.

[36] C. R. Qi *et al.*, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 5648–5656.

[37] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 77–85.

[38] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst. 30: Annu. Conf. Neural Inf. Process. Syst.*, 4–9 Dec., Long Beach, CA, USA, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5105–5114.

[39] S. Bai *et al.*, "GIFT: towards scalable 3D shape retrieval," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1257–1271, Jun. 2017. [Online]. Available: https://doi.org/10.1109/TMM.2017.2652071

[40] C. Ma, Y. Guo, J. Yang, and W. An, "Learning multi-view representation with LSTM for 3-D shape recognition and retrieval," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1169–1182, May 2019. [Online]. Available: https://doi.org/10.1109/TMM.2018.2875512

[41] B. Li and H. Johan, "Sketch-based 3D model retrieval by incorporating 2D-3D alignment," *Multimedia Tools Appl.*, vol. 65, no. 3, pp. 363–385, 2013.

[42] B. Li *et al.*, "A comparison of methods for sketch-based 3D shape retrieval," *Comput. Vision Image Understanding*, vol. 119, pp. 57–80, 2014.

[43] J. Xie, G. Dai, F. Zhu, and Y. Fang, "Learning barycentric representations of 3D shapes for sketch-based 3D shape retrieval," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 3615–3623.

Authorized licensed use limited to: Anhui Normal University. Downloaded on September 24,2021 at 03:44:09 UTC from IEEE Xplore. Restrictions apply.

[44] J. Chen and Y. Fang, “Deep cross-modality adaptation via semantics preserving adversarial learning for sketch-based 3D shape retrieval,” in *Proceedings of the 15th European Conference on Computer Vision* (Lecture Notes in Computer Science, 11217). Munich, Germany, Springer, Sep. 8–14, 2018, Part XIII, pp. 624–640.

[45] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella, “Suggestive contours for conveying shape,” *ACM Trans. Graph.*, vol. 22, no. 3, pp. 848–855, 2003.

[46] T. Judd, F. Durand, and E. H. Adelson, “Apparent ridges for line drawing,” *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007. [Online]. Available: https://doi.org/10.1145/1276377.1276401

[47] H. Li *et al.*, “Multi-view pairwise relationship learning for sketch based 3D shape retrieval,” in *Proc. IEEE Int. Conf. Multimedia Expo,* Hong Kong, China, Jul. 10–14, 2017, pp. 1434–1439.

[48] J. Deng *et al.*, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, Miami, FL, USA. 2009, pp. 248–255.

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst. 25: 26th Annu. Conf. Neural Inf. Process. Syst. Proc. a Meeting held*, Lake Tahoe, NV, USA, Dec. 3–6, 2012, pp. 1106–1114.

[50] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.,* 2015, pp. 448–456, vol. 37.

[51] S. Chopra, R. Hadsell, and Y. LeCun, “Learning a similarity metric discriminatively, with application to face verification,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, San Diego, CA, USA., 2005, pp. 539–546.

[52] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, “The sketchy database: learning to retrieve badly drawn bunnies,” *ACM Trans. Graph.*, vol. 35, no. 4, pp. 119:1–119:12, 2016.

[53] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.,* Boston, MA, USA, Jun. 2015, pp. 815–823.

[54] E. Hoffer and N. Ailon, “Deep metric learning using triplet network,” in *Proc. Similarity-Based Pattern Recognit. - Third Int. Workshop, SIMBAD*, Copenhagen, Denmark, Oct. 12–14, Proc., ser. Lecture Notes in Computer Science, A. Feragen, M. Pelillo, and M. Loog, Eds., vol. 9370. Springer, 2015, pp. 84–92. [Online]. Available: https://doi.org/10.1007/978-3-319-24261-3_7

[55] A. Hermans, L. Beyer, and B. Leibe, “In defense of the triplet loss for person re-identification,” *CoRR*, 2017, *abs/1703.07737*.

[56] P. Shilane, P. Min, M. M. Kazhdan, and T. A. Funkhouser, “The princeton shape benchmark,” in *Proc. Int. Conf. Shape Model. Appl.,* Genova, Italy, Jun. 7–9, 2004, pp. 167–178.

[57] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, 2014, *abs/1412.6980*.

[58] G. Dai, J. Xie, and Y. Fang, “Deep correlated holistic metric learning for sketch-based 3D shape retrieval,” *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3374–3386, Jul. 2018.

[59] B. Li *et al.*, “A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries,” *Comput. Vision Image Understanding*, vol. 131, pp. 1–27, 2015.

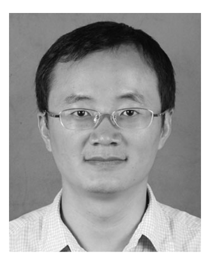
[60] C. Zou, C. Wang, Y. Wen, L. Zhang, and J. Liu, “Viewpoint-aware representation for sketch-based 3D model retrieval,” *IEEE Signal Process. Lett.*, vol. 21, no. 8, pp. 966–970, Aug. 2014.

[61] J. Chen *et al.*, “Deep sketch-shape hashing with segmented 3D stochastic viewing,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 791–800. [Online]. Available: http://openaccess.thecvf.com/CVPR2019.py

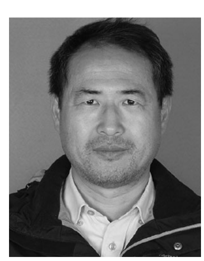
Yongzhe Xu received the B.S. and M.S. degrees in software engineering from Sun Yat-sen University, Guangzhou, China, in 2017 and 2019, respectively. His research interests include deep learning and computer vision.

Jiangchuan Hu received the B.S. degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2017. He is currently working toward the M.S. degree with the School of Data and Computer Science, Sun Yat-sen University. His research interests include 3-D image processing, stereoscopic image retargeting, and computer vision.

Kanoksak Wattanachote received the B.S. degree in statistics from Srinakharinwirot University, Bangkok, Thailand, in 1995, the M.S. degree in information management from the School of Advanced Technologies, Asian Institute of Technology, Khlong Nueng, Thailand, in 2001, and the Ph.D. degree in computer science and information engineering from National Central University, Taoyuan City, Taiwan (ROC), in 2016. He was a Postdoctoral Research Fellow with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, Guangdong, China, from 2016 to 2017. He is currently working as an Assistant Professor with the School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, China. His research interests include computer vision, image processing, video editing, interactive multimedia, big data technologies, and machine learning.



Kun Zeng received the Ph.D. degree from the National Laboratory of Pattern Recognition Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2008. He is currently an Associate Professor with Sun Yat-Sen University, Guangzhou, China. His current research interests include multimedia, computer vision, and machine learning.



Yongyi Gong received the M.S. and Ph.D. degrees in computer software and theory from Sun Yet-sen University, Guangzhou, China, in 1996 and 2007, respectively. He has been a Faculty Member with the School of Information Science and Technology, Guangdong University of Foreign Studies, since 2010, where he is currently a Full Professor. His research interests include 3-D image processing, stereoscopic image retargeting, and computer vision.

Authorized licensed use limited to: Anhui Normal University. Downloaded on September 24,2021 at 03:44:09 UTC from IEEE Xplore. Restrictions apply.