Sketch Augmentation-Driven Shape Retrieval Learning Framework Based on Convolutional Neural Networks

Wen Zhou, *Member, IEEE*, Jinyuan Jia, Wenying Jiang, and Chenxi Huang

Abstract—In this article, we present a deep learning approach to sketch-based shape retrieval that incorporates a few novel techniques to improve the quality of the retrieval results. First, to address the problem of scarcity of training sketch data, we present a sketch augmentation method that more closely mimics human sketches compared to simple image transformation. Our method generates more sketches from the existing training data by (i) removing a stroke, (ii) adjusting a stroke, and (iii) rotating the sketch. As such, we generate a large number of sketch samples for training our neural network. Second, we obtain the 2D renderings of each 3D model in the shape database by determining the view positions that best depict the 3D shape: i.e., avoiding self-occlusion, showing the most salient features, and following how a human would normally sketch the model. We use a convolutional neural network (CNN) to learn the best viewing positions of each 3D model and generates their 2D images for the next step. Third, our method uses a cross-domain learning strategy based on two Siamese CNNs that pair up sketches and the 2D shape images. A joint Bayesian measure is used to measure the output similarity from these CNNs to maximize inter-class similarity and minimize intra-class similarity. Extensive experiments show that our proposed approach comprehensively outperforms many existing state-of-the-art methods.

Index Terms—Sketch-based shape retrieval, convolutional neural network, learning framework, sketch augmentation, best view, joint Bayesian fusion

1 Introduction

Sketch-based shape retrieval allows efficient searching and retrieving of relevant 3D models from a large 3D model database, particularly when the models are not annotated. Sketch-based shape retrieval, however, is a challenging problem. Retrieval results can be inaccurate for several reasons. A sketch is often an abstract representation of the model drawn from a view chosen by the user and is often unpredictably deviated from the actual shape due to the limited drawing skill of the user. In contrast to the sketch which is 2D, a shape is a 3D object and can be complex. Projecting a 3D shape to different 2D planes can lead to significantly different images and therefore affecting retrieval accuracy. Furthermore, the response time for retrieval is important for the practicality of the approach.

In the last decade, deep learning has achieved great success in image-related tasks. Many researchers, such as Zhu *et al.* [13], have proposed full learning-based methods that enhance the robustness of shape retrieval results. Its successes in sketch-based research, however, is challenged by the lack of training samples. Currently, in academic circles, one of the largest sketch datasets is the TU Berlin sketch dataset [1], which includes a mere 20,000 sketches. In contrast, there are multiple image datasets with images in the order of millions. The fact is that images are simpler to acquire with cameras while acquiring hand-drawn sketches require more human effort. To improve the effectiveness of deep learning-based methods on sketch-related tasks, it is crucial to increase the number of training samples. Existing methods to increase the number of samples include performing image-based transformation (scale, translation, and rotation) on a training sample to generate new samples. However, this transformation does not sufficiently enrich the sketch samples and have had only very limited effects on the final retrieval result.

In this paper, we proposed a *sketch-augmentation* approach to enrich the training samples for deep learning-based shape retrieval using sketches as inputs. In addition to increasing the number of samples, we also ensure the diversity in the dataset. Diversity is important as significant differences usually exist between sketches in a dataset and actual hand-drawn sketches by users. Our approach is to generate new sketch samples by simulating more hand-drawn sketches. Instead of using solely image-based transformation, we remove and perturb strokes in a sketch to generate new sketch samples. The only image-based transformation we use is rotation. Our approach improves the quantity and diversity of training samples and thus improves the robustness of the model. In addition, for stroke removal, we propose a novel method for identifying strokes rather than removing

- *W. Zhou is with the School of Computer and Information, Anhui Normal University, Wuhu, Anhui 241002, China. E-mail: w.zhou@ahnu.edu.cn.*
- *J. Jia is with the School of Software Engineering, Tongji University, Shanghai 201804, China. E-mail: tjsse17@yeah.net.*
- *W. Jiang is with the School of Computer and Information, Anhui Normal University, Wuhu 241002, China. E-mail: jiangwenying@ahnu.edu.cn.*
- *C. Huang is with the Department of Computer Science, Xiamen University, Xiamen, Fujian 361005, China. E-mail: 909813723@qq.com.*

Manuscript received 8 Feb. 2019; revised 2 Jan. 2020; accepted 16 Feb. 2020. Date of publication 24 Feb. 2020; date of current version 30 June 2021. (Corresponding author: Wen Zhou)
Recommended for acceptance by M. Spagnuolo.
Digital Object Identifier no. 10.1109/TVCG.2020.2975504

strokes based on a hypothesis of hand-drawn sketch habits (Eitz *et al.* [1] and Yu *et al.* [21]).

In addition, our work in this paper also addressed the dimensional mismatch between 2D sketches and 3D shape models. We propose using a convolution neural network (CNN) to obtain the "best" views of the 3D shapes for matching. Here, the best views refer to views that depict the 3D shape that avoid self-occlusion and show the most salient features, following how a human would naturally sketch the 3D shape. Our motivation for this best-view shape approach is as follows. In sketch-based shape retrieval, projecting a 3D model onto multiple 2D views is one of the best solutions to the dimensional mismatch problem between a 2D sketch and a 3D shape. Projecting the 3D shape into poor views would adversely affect the retrieval results. In this paper, we learn the rules of hand-drawn sketches from the sketches themselves; then, these rules are employed to predict the shape projection that is most suitable for retrieval.

Finally, we propose a learning framework that uses Siamese CNNs to complete the retrieval task. The retrieval process can be approached as a matching process between image pairs consisting of the input sketch and every model. Training Siamese networks (where one network is a sketch CNN, and the other network is a shape CNN) can improve the retrieval result considerably. Moreover, we use a joint Bayesian pipeline to measure the similarity between the output features of the Siamese networks, and we adopt a contrastive cost function [33] to evaluate the overall networks.

The remainder of this paper is organized as follows. In Section 2, we present the related research to sketch-based shape retrieval. In Section 3, we present our proposed learning framework and describe it in detail in Section 4. In Section 5, we present the evaluation of our proposed method to illustrate the feasibility and superiority of the proposed framework. Finally, conclusions are drawn in Section 6.

2 Related Works

In this section, we outline the related work in the area of 3D model retrieval.

Funkhouser *et al.* [2] proposed a 3D model retrieval engine that supports switching between 3D and 2D. This model used the 3D spherical harmonic method. Eitz *et al.* [3] realized a 2D/3D-based retrieval algorithm using an approach that combined bag-of-words and HOG models. However, those methods did not preprocess the sketches before performing retrieval, which may have affected the results due to ambiguous strokes in the input sketches or to sketch errors resulting from amateur drawing skills expressing the user's intention poorly. Therefore, Li *et al.* [4] proposed performing a preprocessing operation before starting retrieval; the preprocessing was intended to check user hand-drawn sketch and display a version of the sketch that most closely aligned with the user's intention.

Dalal *et al.* [5] proposed using the histogram of gradients (HOG) descriptor, which captures the edges of gradient structures that are highly characteristic of local shapes. Translations and rotations had minimal effects when they were smaller than the local spatial or orientation bin size. However, because the HOG descriptor follows a pixel-wise strategy and because the sketch was sparse by nature, the sketch representation always produced many zeroes in the final histogram. Saavedra [6] proposed an improved descriptor for the histograms of edge local orientations (HELO), which follows a cell-wise strategy; therefore, it seems to be appropriate for representing sketch-like images. Moreover, Saavedra [6] proposed soft computation for HELO (S-HELO), which computes cell orientations in a soft manner using bilinear and tri-linear interpolation and takes spatial information into account. Then, the method computes an orientation histogram using weighted votes from the estimated cell orientations. Fu *et al.* [7] also improved the HOG descriptor with the binary HOG descriptor (BHOG), which is both faster than the HOG descriptor at computing feature vectors and requires less memory. To enhance the robustness against noise in sketch images, Chatbri *et al.* [8] introduced an adaptation framework based on scale-space filtering. In this approach, the sketches are first filtered by a Gaussian filter to perform smoothing; then, the skeletons of sketches are extracted. Weiss *et al.* [9] proposed the spectral hashing algorithm (SHA), which seeks compact binary codes of feature data and then uses the Hamming distance to measure the correlations of code words by their semantic similarities. Li *et al.* [11] presented a composite features method to conduct the sketch-based retrieval task.

Recently, deep learning has achieved success when applied to many computer vision tasks. Specifically, Chopra *et al.* [14] presented a Siamese convolution neural network that used an architecture consisting of two identical subconvolutional networks and applied it to a weakly-supervised metric learning setting. The goal of the networks was to make the output vectors as similar as possible when the pair of input vectors were labeled as similar and to make them as dissimilar as possible when the input vector pair was labeled as dissimilar. Siamese networks have been applied to text classification [15], speech feature classification [16] and sketch-based 3D shape retrieval [12]. Recently, cross-domain convolution neural network approaches, such as training two Siamese CNNs [12], pyramid cross-domain neural networks (PCDNNs) [13], deep correlated metric learning (DCML) [31] and learning barycentric representations of 3D shapes (LWBR) [32] have been widely adopted for sketch-based shape retrieval.

The dimension mismatch between shapes (in 3D) and sketches (in 2D) – how to use 2D view images correctly and compactly in representing a 3D shape – remains one of the key problems in sketch-based shape retrieval. Bai *et al.* [22] presented a named GIFT descriptor, the GIFT generated multiview descriptor for 3D shape, which is an index structure used for multiview matching to achieve fast retrieval. However, poor view images badly hamper the result of retrieval; moreover, how to collect the related multiview of a shape remains a challenging problem. Su *et al.* [23] proposed multiview learning methods based on CNN to complete the sketch-based retrieval; to avoid the best-view-of-shape problem, it uses many different view images to represent a shape. An optimization function was presented to obtain the best result. Shape2Vec [30] has been proposed to solve cross-modal retrieval problems, such as that encountered in sketch-based shape retrieval; this approach represents a novel learning method of semantic-based shape descriptors from training data. To obtain good view images for a shape,

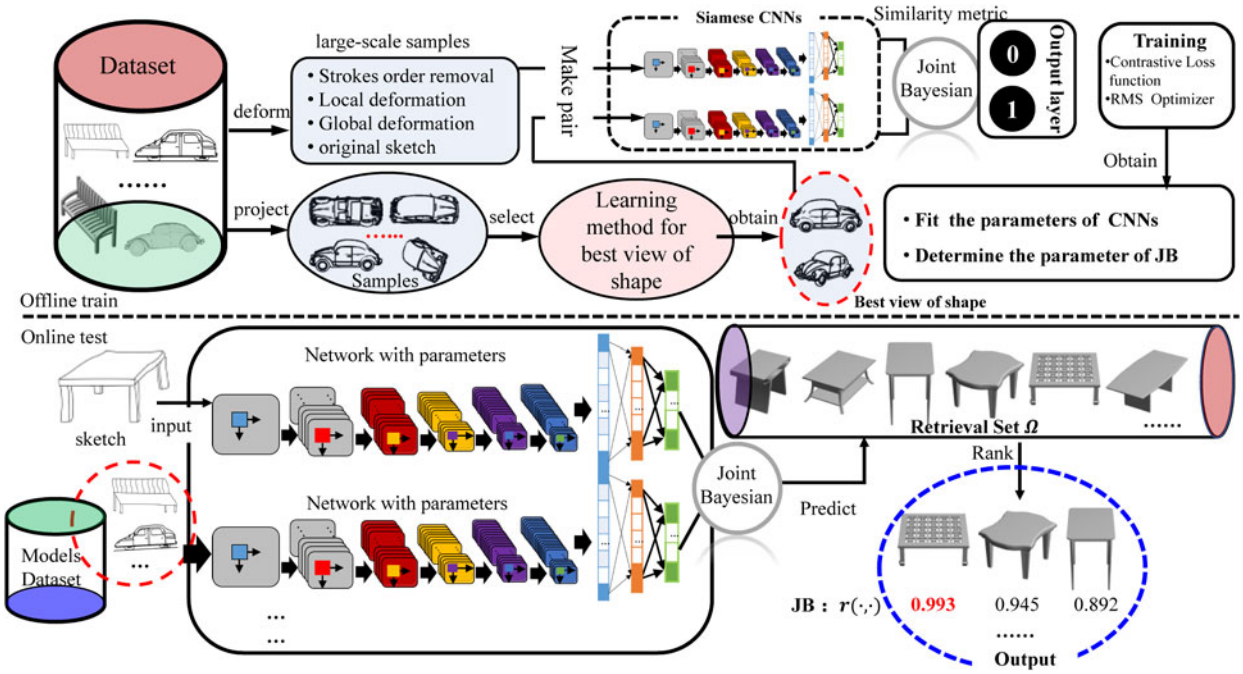

Fig. 1. Overview of the proposed framework. The figure above the dotted line shows the sketch augmentation method used to create more samples, the projection used to obtain the best view of a shape based on learning the semantics of a hand-sketch, and the training of Siamese networks for shape retrieval; The figure below the dotted line shows how the trained Siamese networks are used in a shape retrieval task to rank the retrieved shapes.

it assumes that the shapes in a dataset are consistently aligned. However, the poses of shapes are indeterminate.

In fact, for shape retrieval, the best view selection is an important and difficult task. Many researchers, such as Zhao *et al.* [20], Liu *et al.* [21], and Zhou *et al.* [35], [43], [44], have proposed a variety of methods to select the best views. We present an overview of the methods for identifying the best view of a shape as follows. In previous works, many researchers have conducted related studies, such as Dutagaci *et al.* [25], who proposed a benchmark for evaluating best view selection algorithms. The benchmark consists of the preferred views of 68 3D models provided by 26 human subjects. The evaluation methods include the view area, ratio of visible area [27], surface area entropy [28], silhouette length [27], silhouette entropy [29], curvature entropy [29], and mesh saliency [26]. No method is capable of yielding good results for all models; each is suitable only for certain types of models. Additionally, Zhao *et al.* [20] presented a study of the best view of a shape from sketch contours based on a support vector machine (SVM). Meanwhile, Zhou *et al.* [35] proposed to acquire the best view of a shape by learning standard sketches styles using multilayers perceptrons (MLP), through which better performance compared to the aforementioned methods can be obtained. Relative to other existing methods that can solve the problem involving dimensional mismatch between sketches and shapes, the biggest advantage of using the best view for a shape to represent the 3D shape is faster retrieval and better retrieval accuracy, regardless of the pose of the shapes.

Following earlier studies, great attention has consistently been paid to data augmentation techniques, above all, in sketch recognition [19] and image classification [38]. There are many excellent research studies in the literature, such as Li *et al.* [37], Jia *et al.* [39], Antoniou *et al.* [40], Daniel *et al.* [41] and Cubuk *et al.* [42]. In general, good results can always be obtained by these methods in their respective domains; however, in sketch-based shape retrieval, there exist obvious dimensional discrepancies between sketches and models and smaller intra-class distance between numerous models or sketches. Therefore, for this kind of data augmentation, stringent requirements must be met: Sketch augmentation must carefully preserve tiny discrepancies that exist in intra-class models. Otherwise, these incorrect samples (i.e., augmented sketches) would lead to worse retrieval results.

3 Proposed Framework

Fig. 1 shows an overview of our proposed method, which consists of three main stages: (i) sketch augmentation, where we apply three different types of deformation to increase the number of the sample sketches; (ii) projecting a 3D shape to a 2D view, where we use a CNN-based learning method to identify the best views for the shape; and (iii) shape retrieval using Siamese CNNs and joint Bayesian fusion, where the CNNs use cross-domain learning and learn the similarity between the sketches and the best-projected views of the shapes and joint Bayesian fusion scheme [18] measures the similarity between the output features of the Siamese networks. The joint Bayesian fusion was first proposed to address face verification and was applied by Yu *et al.* [19] to sketch recognition with excellent results.

The details of our proposed framework are presented in the following section.

4 Framework Description

In this section, we present the details of our proposed framework. The proposed framework can be divided into three main parts: sketch augmentation, finding the best views for a shape, and similarity ranking using Siamese networks based

on the joint Bayesian fusion scheme. All the CNN networks used in this paper are based on the AlexNet CNN [10].

4.1 Sketch Augmentation Approach

Ideally, sketch augmentation should have minimal effect on the sketch content but still increase the number of sketches in the training dataset. Yu *et al.* [19] have shown that doing so can improve sketch recognition performance. However, their approach for obtaining the strokes of a sketch is based on an assumption concerning hand-drawn sketch habits (Eitz *et al.* [1]), which assumes that a sketch is initially drawn from an outer-to-inner direction using long strokes to depict the overall contour of the shape, followed by providing details using short strokes. This assumption has a few drawbacks: First, human drawing habits are diverse, and obtaining strokes based on the assumption sometimes leads to incorrect results. Second, since a stroke represented the basic contour of a sketch, the number of strokes was relatively fewer, thereby limiting the degree to which relevant sketch augmentation could be conducted, for instance, via stroke removal or local stroke deformation. As a result, Yu *et al.* [19] approach fails to obtain good results for sketch-based shape retrieval. Sketch-based shape retrieval is a more complex task as there exist bigger intra-class differences but smaller inter-class differences in the samples.

How can one correctly and efficiently augment the sketches? Intuitively, according to human drawing habits, people seldom focus on the key-points of strokes when they are sketching. Thus, stroke removal based on the assumption above can lead to significant change to the sketch. For intra-class sketches, small differences can potentially be unintentionally erased. In this case, extensive incorrect samples (i.e., incorrectly augmented sketches) are may be generated and used for training the proposed learning framework, resulting in wrong retrieval results.

In contrast, in this paper, we propose a novel method to obtain the strokes of a sketch based on image features, such as Harris key-points. Our main objective is to obtain enough additional strokes in a sketch to be able to more accurately describe the key features of the shape in the sketch. Our sketch augmentation method relies on the insights that hand-drawn sketches are rarely identical to the contours of a model projection and often focus on local content details. The accuracy of the sketch in depicting a shape can vary between users. Thus, our sketch augmentation method is based on three deformations: stroke removal, local deformation via stroke perturbation, and global deformation via and re-orientation.

4.1.1 Stroke-Removal Deformation

Here, we present the process for stroke-removal deformation.

Given a sketch s_i, we first conduct an edge-thinning operation and then apply gradient orientation to each pixel in the lines using the Sobel descriptor. In this manner, every pixel p in each line has an edge orientation τ_p. All the lines in a sketch are uniformly sampled to obtain the related pixel seeds. The stroke set $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ is acquired using a simple greedy approach that continually combines eight connected pixels from each seed until the sum of their orientation differences exceeds a threshold ($\pi/2$). For every initial stroke γ_i, we denote its mean position as x_i and its mean orientation as τ_i. Zitnick *et al.* [24] proposed an edge-affinity metric relationship between two edges. Therefore, we can measure two strokes γ_i, γ_j as follows:

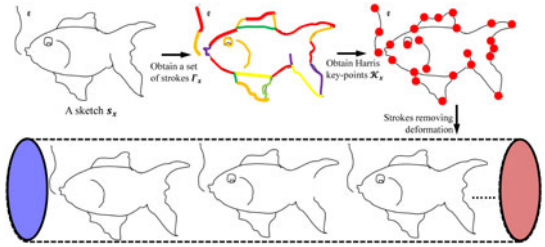

Fig. 2. An overview of the stroke-removing deformation.

$$O(\gamma_i, \gamma_j) = |(\cos\tau_i - \cos\tau_{ij}) \times (\cos\tau_j - \cos\tau_{ij})|^2, \quad (1)$$

where τ_{ij} represents the angle between x_i and x_j. When $O(\gamma_i, \gamma_j) < \delta$, the strokes γ_i and γ_j can be merged. In this paper, we set this threshold to $\delta = 0.8$ through experimentation. Then, we repeatedly apply Equation 1 to obtain the final stroke set Γ.

We employ the Harris corner descriptor [36] to evaluate the importance of every stroke in the stroke set Γ. Given a sketch s_x, we can easily collect its Harris key points. Consequently, it is not difficult to observe that these key points are distributed non-uniformly across many different smooth curve strokes Γ_x. Therefore, some of the unimportant strokes can be removed, generating a new sketch. To finish, we obtain a stroke set Γ_x based on Equation (1).

To remove the strokes, given a sketch s_x that consists of a set of n strokes $\Gamma_x = \{i \in n | \gamma_i\}$, we measure the importance of every stroke γ_i by considering the number of keypoints it contains and its length. For example, to measure the importance of the ith stroke γ_i, the metric equation is as follows:

$$I(\gamma_i) = \frac{e^{\alpha * count(k_i)} / e^{\beta * l_i}}{\sum_{i=0}^{n} e^{\alpha * count(k_i)} / e^{\beta * l_i}}, \quad (2)$$

where k_i and l_i represent a set of Harris keypoints and the length of the ith stroke γ_i, respectively (the number of keypoints in the whole sketch s_i being denoted as $count(K) = m$). Moreover, α and β are two thresholds to adjust the size of $e^{count(k_i)}$ and e^{l_i}, respectively, so that the formula $e^{count(k_i)}/e^{l_i}$ is meaningful (i.e., if $e^{count(k_i)}/e^{l_i} = \infty$, then $I(\gamma_i)$ is meaningless). In this paper, for a sketch s_i, assuming that its size is $x \times y$ and that the number of strokes in its stroke set Γ_i is $|\Gamma_i| = n$, we set $\alpha = \frac{n}{m}$ and $\beta = \frac{n}{\min(x,y)}$. Additionally, the function $count(\cdot)$ returns the size of the set k_i. Therefore, using Equation (2), we can remove a stroke γ_x with a value smaller than $I(\gamma_x)$ to generate a new sketch. This approach greatly increases the number of available sketch samples for training. The entire stroke-removal deformation process is shown in Fig. 2.

4.1.2 Local Deformation

The local deformation step is relatively easy to understand. We first collect some keypoints on a sketch s_x based on Harris corner keypoints. Here, we let $K = \{0 \le i \le m-1 | k_i\}$ represent the keypoint set from a stroke set Γ_x. The local

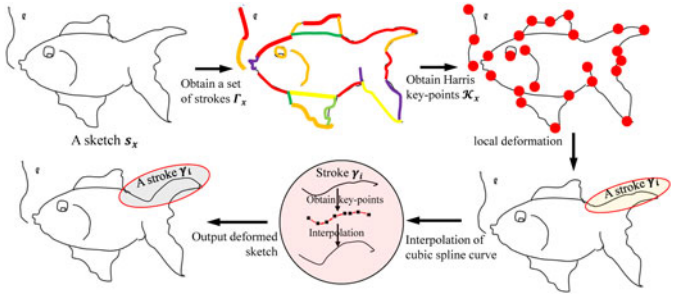

Fig. 3. Overview of the sketch local deformation process.

deformation is based on spline interpolation, in which a key-point k_i can be viewed as a control point. The cubic spline curves γ_i $(0 \leq x \leq 1)$ are represented as in

$$\gamma_i(x) = (1-x)^3 k_0 + 3(1-x)^2 x k_1 + 3(1-x)x^2 k_2 + x^3 k_3, \tag{3}$$

where k_0 and k_3 are the two endpoints of each stroke (i.e., a spline curve). To obtain a new sketch, we move the keypoint k_i in each stroke γ_i. Assuming that k_i is the key-point of the ith stroke γ_i, $\forall k_i \in K$, the new position $k_i^{'}$ of k_i can be represented as follows:

$$k_i^{'} = k_i + J \times e^{-\frac{1}{2\sigma^2}}, \tag{4}$$

where J is an identity matrix. In our work, we set σ to 0.1. Completing the local deformations in this manner results in many different sketches. In practice, when people sketch the same shape repeatedly, there are always local deformations. Fig. 3 shows an overview of the local deformation process.

4.1.3 Global Deformation

Global deformation is performed to facilitate sketch rotation operations. We first obtain a pivot position p_c for the entire ith sketch s_i. For every pixel p_x in sketch s_i (i.e., $\forall p_x \in s_i$) after the rotation operation, the new position of pixel p_{new} is given by:

$$p_{new} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times p_c + p_x, \tag{5}$$

where we set $-\frac{\pi}{6} \leq \theta \leq \frac{\pi}{6}$ to ensure that the overall orientation of the sketch remains.

4.2 Finding The Best View of a Shape

We project the 3D data into many different 2D images to eliminate the dimension mismatch problem between the shape and a sketch. However, determining which views better depict the shape is still a tricky problem. Determining the view positions that cover the full 3D model is a critical problem to solve. To this end, we present a learning-based method to identify the best view of a shape.

While the sketches from the dataset can be viewed as a good or even the best view of a shape, the so-called best shape view should better fit the habits of people drawing sketches by hand. Therefore, obtaining the best view of a shape should depend more on learning from hand-drawn sketches rather than on the geometrical features of the models (e.g., Dutagaci *et al.* [25]).

In our approach, we apply suggestive contours [3] to complete this task. Suggestive contours can be used to project many view images from different viewpoints. We use CNN to both extract features and make classifications to solve the learning problem. The main steps are elaborated below.

Obtain Multiview Images. We can project every model M_i into n view images $V_j = \{v_j^0, v_j^1, \ldots, v_j^{(n-1)}\}$ from different viewpoints. In this paper, we set $n = 400$.

Obtain Features. Several datasets include many models and the corresponding sketches, for instance, the SHREC'13 [17] and SHREC'14 [33] datasets. To obtain the relationship between the shape views and sketches, we perform CNN-based learning tasks to facilitate discriminative training using a similarity metric. The input sizes of the sketches and views should be identical to enable their relationships to be easily measured after extracting features using CNN.

Similarity Metric. We acquire the feature vectors of views and sketches using the pretrained CNN. Note that we use a pretrained CNN so that we can obtain more accurate features. A pretrained AlexNet is included in TensorFlow. For random sketch and view images, via this CNN, their feature vector sizes are identical; therefore, we compare the view features with the sketch features to build a similarity relation. Here, we adopt the approach proposed by Chopra *et al.* [14], as also adopted by Wang *et al.* [12]. We also propose a discriminative loss function that drives the system to make the correct decision. The similarity metric equation is as follows:

$$\Delta(s_i^k; v_j^t; \omega) = \omega \times \eta D_w^2 + (1-\omega) \times \varphi e^{\mu D_w}, \tag{6}$$

where ω is a binary similarity label, i.e., $\omega = 1$ *or* $\omega = 0$. In this paper, ω can only be 1 because s_i^k and v_j^t must be deemed similar, i.e., they are from a pair. The function D_w represents the Manhattan distance between the feature vectors of the sample s_i^k and v_j^t. We follow Chopra *et al.* [14] and set the constant parameters η and φ to 5 and 0.1, respectively.

Obtaining Positive and Negative Samples. The related samples are generated through the discriminative loss function to classify the feature vectors of sketches and views. In particular, to obtain the positive view images and negative view images for the kth view image v_i^k projected from the ith model M_i and each sketch s_j^t $(\forall s_j^t \in S_j)$ from the same category $S_j(|S_j| = N)$, we define a function $0 < p(\cdot) \leq 1$ to determine whether the view image v_i^k is a positive sample:

$$p(v_i^k) = \frac{\Delta(v_i^k; s_j^t; 1) - \min_{0 \leq t \leq N} \Delta(v_i^k; s_j^t; 1)}{\max_{0 \leq t \leq N} \Delta(v_i^k; s_j^t; 1)}, \tag{7}$$

where N is the number of samples. Because there are 80 sketches in each category in the dataset, using the above sketch augmentation method, we can obtain more sketches in each category. In this paper, we set $N = 800$. Equation (7) determines whether a sample v_i^k is positive.

To obtain negative samples to train our CNN network, we must define the decision function Θ to automatically determine whether the input sample (v_i^k) is negative or positive. This decision function is shown in:

$$\Theta(v_i^k) = \begin{cases} 1 & if\ \exists s_j^t \in S_i, p(v_i^k) \geq 0.95 \\ 0 & if\ \forall s_j^t \in S_i, p(v_i^k) \leq 0.05 \\ null & otherwise \end{cases}. \quad (8)$$

Based on Equation (8), we can obtain many different positive and negative samples. Then, the features extracted from the positive samples by CNN are the positive features and vice versa. Overall, we generated one million features from many positive and negative samples.

Training the CNN to Perform Classification. By following the steps described above, we obtain many different positive and negative samples. Next, we need to conduct training to fit the related parameters. We only need to determine whether a view image is good or bad among the many different view images; then, we remove the bad images. Therefore, the goal is to train a binary classifier to classify the related view images. Previously, the pretrained CNN was used to extract features, but the parameters must be retrained to provide a correct prediction when CNN is used to determine whether a view image is good. This new CNN has a learning rate of $\eta = 0.001$ and an added softmax function after the last layer of the CNN network that computes the scores of the two perceptrons of the output layer (with one perceptron being used to output good samples and the other being used to output bad samples). The final output of this classification CNN depends on which perceptron has the highest score.

Predicting & View Ranking. At the testing stage, based on the network trained as described above, we can predict the label of every view image projected by the same model because the best view images are always selected as those with the highest scores. However, to preserve the diversity of the final result, we need to conduct view ranking.

For a 3D shape M_i, we assume that we can obtain its relevant good view images set V_i based on preceding network predication. Furthermore, to keep the diversity of best view images, we need to further measure their relationship. These view images projected by nearby positions are often very similar. If we directly rank these view images according to their predication scores, obviously, we maybe gain many similar view images. It is meaningless for many similar best view images to be collected for training in the next step; it is also very time-consuming. Therefore, we attempt to avoid the situation by not generating multiple view images projecting from nearby positions that are similar to each other, to increase the diversity of the projected best view images. The sketches drawn by users are often diverse; for instance, when people sketch to represent a 3D desk, they may draw it from the front side or a lateral side. Thus, for 3D shape retrieval, we have to consider the diversity of input sketches as well.

Consider the diversity, the IoU (Intersection of Union) criterion is utilized to evaluate the relationship of these view images, including those projected from nearby positions. More specifically, to increase diversity, we penalize images with similar views and reduce their scores. To accomplish this, Equation 9 is employed to change their original scores:

$$t_i(v_i^k) = s_i(v_i^k) + \Psi\left(\max_{v_i^w \in V_i}^{v_i^w \neq v_i^k} IOUs(v_i^k, v_i^w)\right), \quad (9)$$

where the function Ψ is a monotonically decreasing function that penalizes similar views. The function Ψ is given by:

$$\Psi(x) = e^{-\frac{x^2}{2\sigma}}, \quad (10)$$

where σ is an experimental value that controls the penalty. In this paper, we set $\sigma = 0.15$. An overview of our proposed method is shown in Fig. 4.

Finally, a mean-shift algorithm is used to rank the scores of every view image; in this way, the best view for a shape can be obtained. In this paper, to preserve the diversity for every shape, the number of best views of a shape is set to 3, i.e., we collect the top 3 view images as the best view images for a shape in the final ranked list; more details on view ranking are shown in Algorithm 1.

Algorithm 1. View Ranking Algorithm

Input: The good view images set V_i , n
Output: The best view set T, which includes the n best view images of 3D shape M_i
Initialize: $T \leftarrow \emptyset$
For all $v_i^c \in V_i$ do
  Computing score t_c of v_i^c (Equation 9)
End for
while $V_i \neq \emptyset$ **do**
  Obtaining the toppest score $t = \max_{j \in |V_i|} t_j$
  Based on t, getting the corresponding v
  Starting from v, according to mean-shift algorithm,
  acquiring the candidate v_{can}
  if $v_{can} \notin T$ **then**
    $V_i \leftarrow V_i - v_{can}$
    $T \leftarrow T + v_{can}$
  else
    repeat
  end
  if T *is not changed or* $|T|$ *equals to* n **then**
    Iteration over
    return T
  else
    repeat
  end
end

4.3 Siamese Networks for Shape Retrieval

4.3.1 CNN Architecture

In this section, we present the architecture of the CNNs used for shape retrieval. The task is to extract feature vectors from the best view images and sketches. Therefore, we use Siamese networks, and we use the same design for both networks, even though they are trained separately. The size of the input patch is 100×100 for both sources. Each CNN has five convolutional layers, three max pooling layers, and three fully connected layers to generate the features.

The first convolutional layer follows a 3×3 pooling layer generating 96 response maps, each pooled to a size of 3×3. The 256 features generated by the final pooling operation are linearly transformed to 1000×1 features in the last fully connected layer. In the learning pipeline for obtaining the best view of a shape, the softmax function is used in the output layer to obtain a binary classification result. However, in the

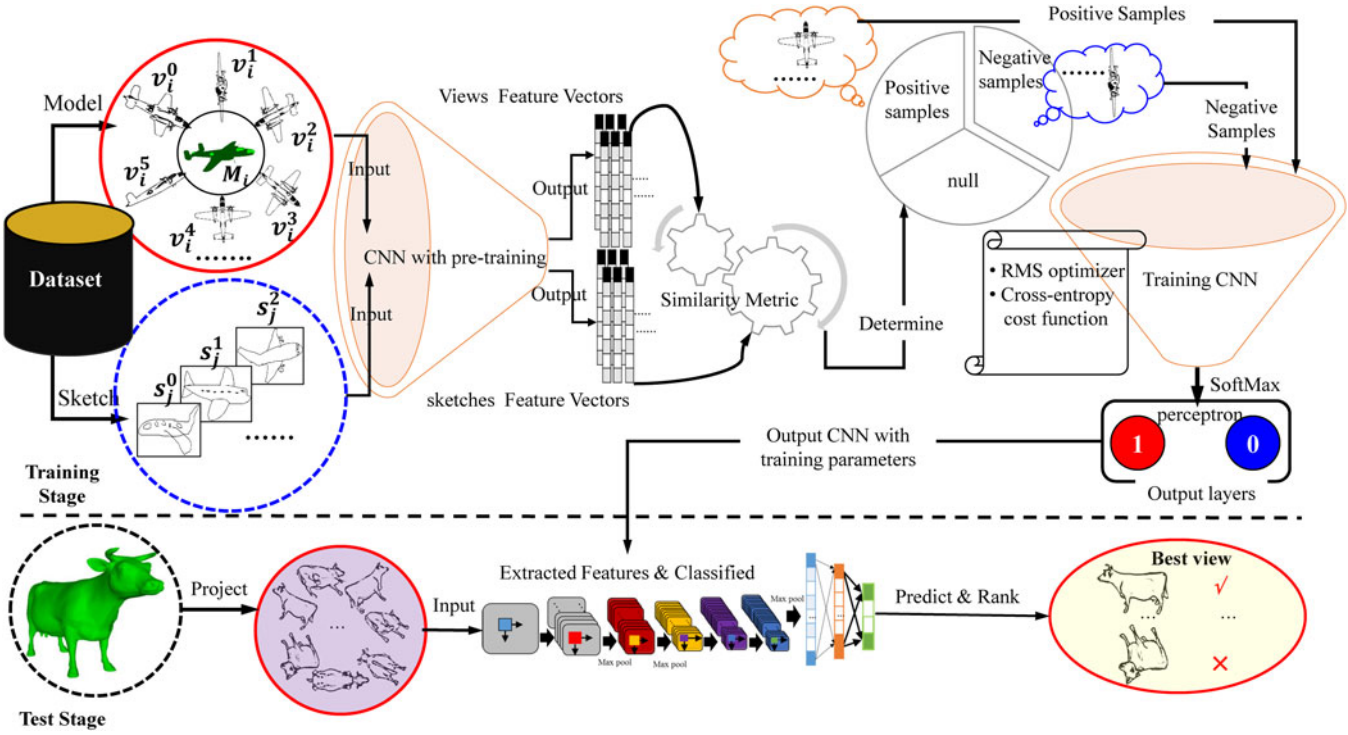

Fig. 4. The overview of the proposed best view method for shapes based on CNNs. The whole pipeline can be divided into two parts, i.e., the training stage and the test stage. The training stage consists of making pairs between sketches and models, collecting related positive and negative samples through the utilization of feature selection based on CNNs with pretraining and training our CNNs using positive and negative samples. In the test stage, we depend on the trained CNNs to predict the good and bad view images. Finally, we propose a rank algorithm to acquire the best view based on view image diversity.

Siamese learning pipeline for sketch-based retrieval, the joint Bayesian function is utilized to output the final binary classification result. Additionally, a rectified linear unit (ReLU) is used in all the layers. To train the samples, we employ a learning rate of $\eta = 0.005$. SGD with mini-batch, the RMS optimizer, and the cross-entropy cost function are also utilized.

4.3.2 Joint Bayesian

Chen *et al.* [18] proposed a joint Bayesian approach to test and verify face features and reduce the separability between classes. Furthermore, Yu *et al.* [19] adopted this method for sketch recognition tasks. In this paper, we also adopt this method to measure the interclass relations as follows:

$$r(x_1, x_2) = log\frac{P(x_1, x_2|H_I)}{P(x_1, x_2|H_E)} = x_1^T A x_1 + x_2^T A x_2 - 2x_1^T G x_2, \tag{11}$$

where the terms H_I, H_E represent the hypotheses that the terms x_1, x_2 belong to intra-class and inter-class relations, respectively. Moreover, the vector x_1 represents the sketch features, and the vector x_2 represents the model features. Let each x_1 represent the $1000 \times 1 = 1000D$ concatenated feature vector from our network ensemble. The terms A and G are two negative semidefinite matrices. Simply, they can be represented as follows.

$$A = (C_\mu + C_\xi)^{-1} - \frac{C_\mu + C_\xi}{(2 \times C_\mu + C_\xi)^2} \tag{12}$$

$$G = -\frac{C_\mu}{(2 \times C_\mu + C_\xi)^2}, \tag{13}$$

where x_1, x_2 belongs to intra-class, the term $C_\xi = 0$, C_μ is the covariance between x_1 and x_2, and vice versa. if $A = G$, Equation (11) becomes a metric of Mahalanobis distance. Therefore, in essence, joint Bayesian metric is an improved Mahalanobis distance based on the intra-class hypothesis.

Finally, we train the joint Bayesian model, thus learning a good metric that exploits the intra-ensemble correlations. Note that when using this approach, each feature dimension is fused, implicitly giving more weight to the more important features as well as finding the optimal combination of different features from different models.

4.3.3 Training Networks for Shape Retrieval

In a learning method, the training samples are a critical component. Our goals in training the Siamese networks are as follows: fitting the parameters of the Siamese networks and determining the parameters (i.e., the matrix A, G). The input to the Siamese networks should be a pair of samples consisting of a sketch and a best view image. The training process is as follows.

1) *Create pairs*: We need to create pairs consisting of every sketch and every best view image for the Siamese networks. A positive pair (i.e., a sketch and a best view image are from the same category) can be viewed as a positive sample and vice versa. Using this approach, a pair is created for every deformed sketch and every best view image.
2) *Determine the parameters of the joint Bayesian pipeline*: Initially, we can utilize these pairs to determine the parameters of the joint Bayesian pipeline. In this way,

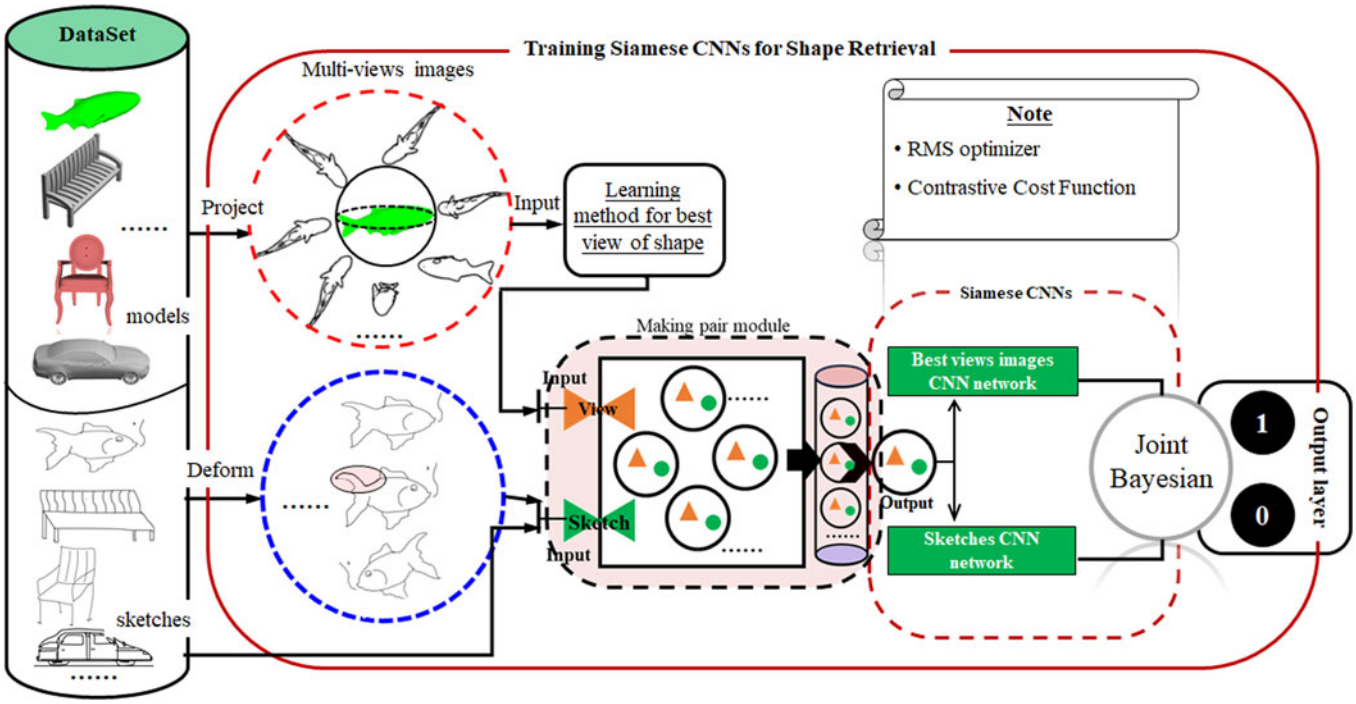

Fig. 5. Overview of the training samples based on the joint Bayesian model. First, the proposed sketch augmentation method can provide more samples to train the networks. Second, based on hand-sketch inherent semantics, the best views of a shape can be obtained using CNN as a classifier. Third, by utilizing the assignment of pairs between sketches and best views, these pairs are treated as samples with which to train our Siamese networks. These samples include positive and negative pairs, i.e., for every positive and negative pair, its two members are from the same and different categories, respectively.

we can obtain the scores of every perceptron of the output layers of the pretrained Siamese networks. The network parameters can be confirmed by training using these samples.

3) *Fit the parameters of Siamese networks*: Intuitively, the cross-entropy cost function is used to measure the difference between the predicted and actual values. Utilizing the RMS optimizer to obtain the minimum of the cost function across the entire network, we can consistently fit the parameters of the networks. Using this approach, we can complete the training task to obtain the parameters of the Siamese networks.

Finally, the Siamese CNN networks with trained parameters are obtained. An overview of the training samples based on the joint Bayesian model is shown in Fig. 5.

4.3.4 Testing for Shape Retrieval

In the above section, by training the Siamese networks, we obtain the Siamese networks with parameters and the joint Bayesian model with parameters. For the Siamese network, completing a retrieval task involves performing N prediction operations (where N is the number of best view images in the retrieval dataset). In general, the number of models in the dataset is smaller than the number of best view images (with every model having at least two best view images). The whole sketch-based retrieval process based on Siamese networks is as follows:

1) *Make pairs*: An input sketch s_x must be paired with its N best view images, forming N pairs P ($|P| = N$), which are the input to the Siamese network.

2) *Obtain the retrieval result*: Initially, the retrieval result set R should be null. Intuitively, the Siamese network predicts each pair; the result is a label value. When the label value is positive (i.e., 1), the model corresponding to that best view image becomes a retrieval result. More specifically, $\forall$ pairs $p_j = (s_x, v_i^k), 0 \leq j \leq N-1$, let the model corresponding to the view image v_i^k be M_i. If $argmax(r(F_{s_x}, F_{v_i^k})) = 1$, then model M_i is added to the retrieval result (i.e., $M_i \longrightarrow R$). Similar operations on all N pairs completes the above operation, and we achieve the final retrieval result R.

3) *Rank the retrieval results*: Better retrieval results should be placed in a better position. Therefore, a ranking operation is required. In this paper, the output value of the joint Bayesian pipeline is used as a sort criterion (i.e., $\forall M_x \in R$, when the model M_x has $2 \leq K \leq 5$ best view images $V_i = \{0 \leq k \leq K-1 | v_i^k\}$, the rank score $\Xi(M_i)$ of model M_i is as in:

$$\Xi(M_i) = \max_{0 \leq k \leq K-1} r(F_{s_x}, F_{v_i^k}), \tag{14}$$

where F is the sketch or the view image feature vector. The function $r(\cdot)$ is shown in Equation (11).

An overview of the proposed method is shown in Fig. 6.

5 Evaluation

In this section, we present the validation and evaluation results of our proposed framework. We performed these evaluations on the SHREC'13 dataset [17], one of the most well known 3D model datasets and includes all the TU Berlin

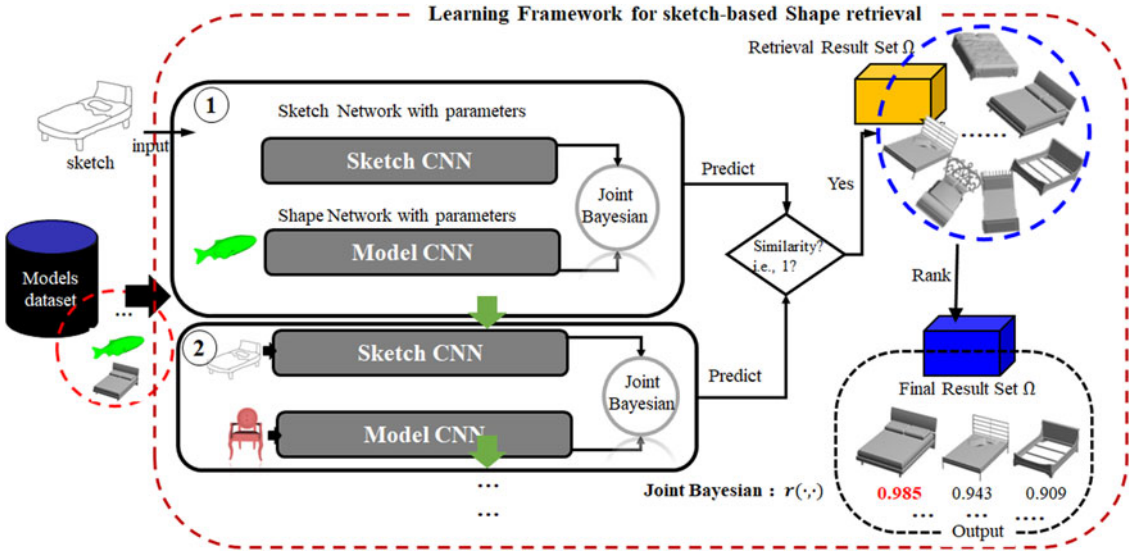

Fig. 6. An overview of shape retrieval based on our proposed framework. First, by utilizing trained Siamese networks for an input sketch, we pair up the sketch and all the best views of the shapes in the dataset. Each pair, based on the Siamese network, was treated as a sample to predict whether they were similar. Assuming that they are deemed similar, the best view of a shape can be put into a retrieval result set (denoted as a yellow cube in the figure). Finally, based on the value of the joint Bayesian model, a rank algorithm is completed to obtain the final retrieval result (represented as a blue cube in the figure).

dataset sketches [1], as well as the SHREC'14 dataset [34], which includes more sketches and 3D models.

The proposed framework was implemented using the C++ and Python 3.6 and was executed on a PC running Windows 10, an Intel Core I7-7700HQ CPU, 8 GB of memory and an NVIDIA GeForce GTX 1060 GPU. Moreover, Google TensorFlow, a popular open-source framework for deep learning was used in this study.

5.1 Best View of a 3D Shape

In this section, we report on the experiments used to validate our learning method to obtain the best view of a 3D shape. Existing methods to obtain the best view of a 3D shape includes SVM-based learning methods [20] and web-image-driven methods [21]. The AUC (Area Under the Receiver Operating Characteristic (ROC) Curve) indicator, which computes the area under the precision-recall curve for a retrieval result, was applied to evaluate the retrieval performance. A comparison of our proposed method with other tested methods is shown in Fig. 7.

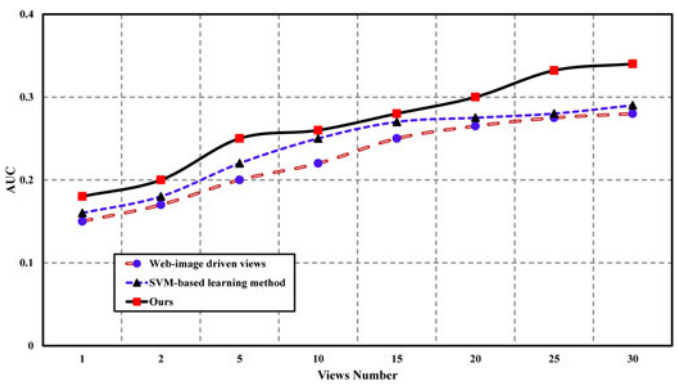

Fig. 7. AUC comparison of our proposed method and other tested methods.

Fig. 7 shows that our method is superior to the other methods. This result can be explained as follows. Poor view images adversely affect the retrieval results. Methods such as those of Polonsky *et al.* [27], Vazquez *et al.* [28], Page *et al.* [29], and Lee *et al.* [26] obtain the best views of a shape considering saliency and entropy but do not match to the goal of sketch-based shape retrieval, where the best view is one that is most consistent with the habits of users drawing sketches by hand. Therefore, a learning approach performs better at this task. In practice, the number of collected best views of a 3D shape is often less than 15. In this case, the results of our method and those of SVM-based learning methods are highly similar.

To further validate the importance of the proposed best-view approach, we perform an experiment whereby we remove the best-view approach from our proposed framework. We substitute it with a multiple views method where we uniformly sample view images from different viewpoints along the bounding sphere of the 3D shape to represent the shape. As there are millions of different 3D shapes whose poses are almost always unpredictable, it is a challenge to determine how many view images would suffice to meet the requirements of the retrieval system. Having more view images also increases the computational load, leading to long waiting time for users who interact with the retrieval system.

Fig. 8 compares the time consumed to retrieve the matching 3D shapes from the two datasets used when we use our best view approach versus the multiviews approach. We see that with multiviews approach, the time needed to complete the retrieval greatly increase for both of SHREC'13 and SHREC'14 datasets. Moreover, because the size of SHREC'14 is larger, more time is spent achieving a single retrieval.

Fig. 9 shows the average number of view images required to yield the required precision of the retrieval. With multi-views approach, We need close to an average of 100 views

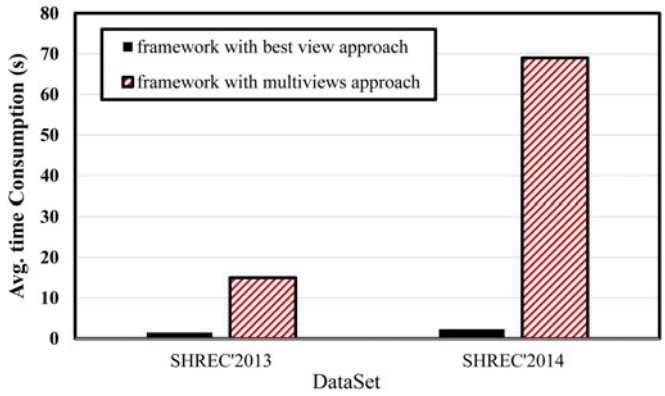

Fig. 8. Avg. retrieval time comparison on the SHREC'13 and SHREC'14 datasets in the retrieval test stage, once the retrieval procedure is completed.

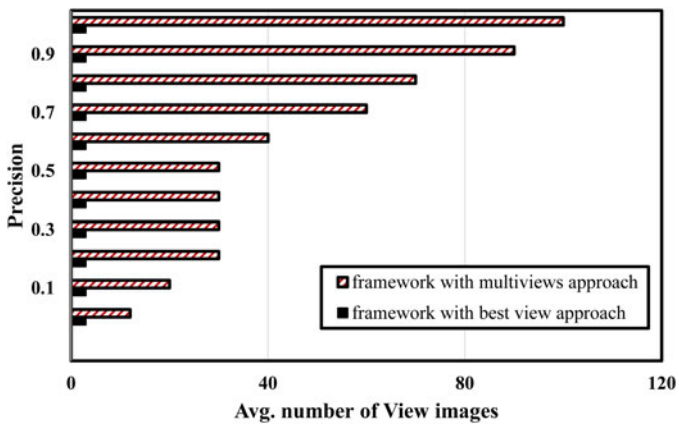

Fig. 9. Comparison of the avg. number of view images required to yield relevant precision for the SHREC'13 dataset.

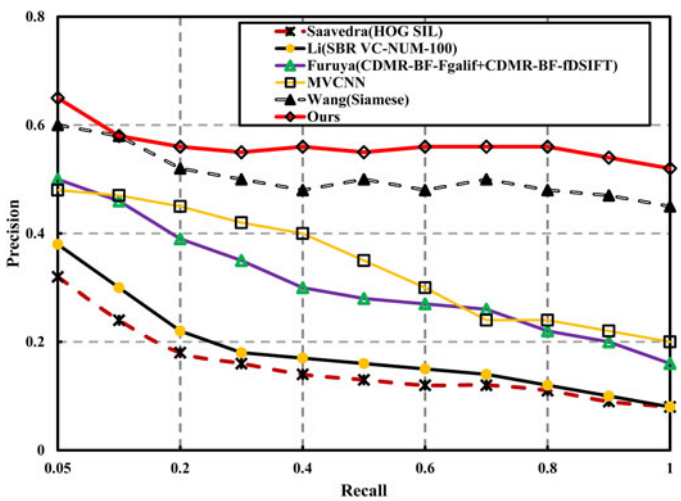

Fig. 10. Performance comparison on the SHREC'13 dataset based on the PR criterion.

per shape to reach a precision of 0.9. On the other hand, our approach selects the best views for matching and does not require a large number of views.

TABLE 1
Comparisons on the SHREC'13 Dataset Based on 6 Criteria

Methods \ Criterion	NN	FT	ST	E	DCG	AP
Saavedra [17]	0.11	0.06	0.1	0.06	0.3	0.08
Li *et al.* [17]	0.16	0.09	0.14	0.08	0.34	0.11
Furuya *et al.* [17]	0.27	0.2	0.29	0.16	0.45	0.25
Wang *et al.* [13]	0.4	0.4	0.55	0.28	0.6	0.46
Ours	0.68	0.63	0.73	0.42	0.72	0.7

TABLE 2
Comparisons on the SHREC'14 Dataset Based on 6 Indicators

Methods \ Criterion	NN	FT	ST	E	DCG	AP
Saavedra [17]	0.08	0.045	0.06	0.03	0.28	0.04
Li *et al.* [17]	0.09	0.05	0.08	0.03	0.3	0.05
Furuya *et al.* [17]	0.1	0.05	0.08	0.04	0.32	0.05
Wang *et al.* [13]	0.23	0.2	0.3	0.15	0.5	0.22
Ours	0.34	0.377	0.431	0.205	0.504	0.31

5.2 Sketch-Based Retrieval

In this section, we use the SHREC'13 benchmark [17] to evaluate our method. We also show the retrieval results within the same domain. Furthermore, we utilize the relations between models and sketches in SHREC'13 to train our network and the fusion model. Our CNN is based on AlexNet [10], which was initially designed to classify images. All the sketches in the dataset have been resampled to a size of 100×100 using the Python PIL library. To avoid overfitting, we use regularization. In addition, we divide the dataset into a training set (70 percent) and a test set (30 percent).

We present the evaluation results on the SHREC'13 dataset in this section. First, we compare the precision-recall curves of our method to the state-of-the-art methods, which include Saavedra (HOG SIL) [17], Li (SBR VC-NUM-100) [17], Furuya *et al.* [17], MVCNN (Su *et al.*) [23] and Wang (Siamese) *et al.* [12]. The results are shown in Fig. 10 for the SHREC'13 dataset.

In addition, to emphasize the advantages of our method, we perform related experiments using six different criteria: Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-Measure (E), Discounted Cumulative Gain(DCG), and Average Precision (AP). The results are listed in Tables 1 and 2.

We also compare our method with the state-of-the-art methods using the SHREC'14 dataset [34] against Saavedra (HOG SIL) [17], Li (SBR VC-NUM-100) [17], Furuya *et al.* [17], Wang *et al.* (Siamese) [12], Dai *et al.* (DCML) [31] and Xie *et al.* [32]. The results are shown in Fig. 11.

Fig. 11 shows that our proposed method is better than the others. In particular, our method achieves much better precision when recall approaches 1.

Meanwhile, in Fig. 12, poor performance is shown for our method based on the Euclidean distance. The reason is that the Euclidean distance equates the differences between the different attributes of the sample. This ignores the key effect that the great gaps within a category have on the final result.

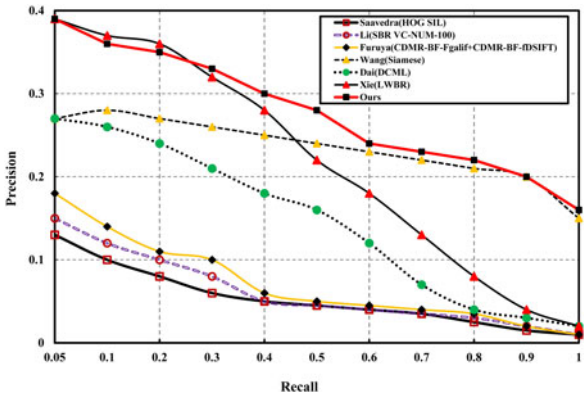

Fig. 11. Performance comparisons on the SHREC'14 dataset.

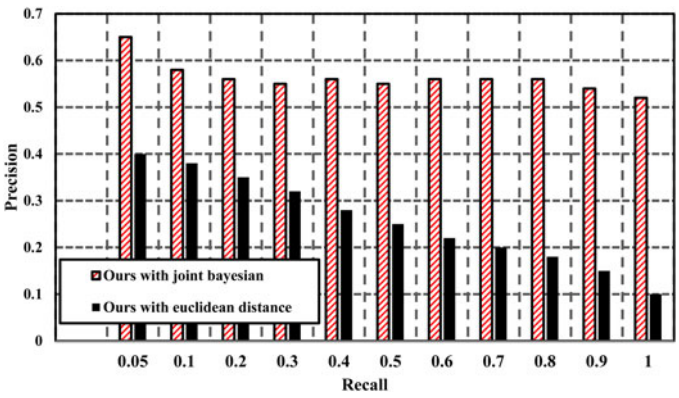

Fig. 12. Comparisons of sketch-based retrieval with and without the joint Bayesian approach on the SHREC'13 dataset.

Therefore, joint Bayesian approach is obviously an essential part among our proposed learning framework.

5.3 Sketch Augmentation

To understand the impact of sketch augmentation, we compare the results of our method with and without sketch augmentation. This experiment also uses the SHREC'13 dataset. The final results are shown in Fig. 13. It shows that with sketch augmentation, we can achieve up to 0.1 higher in precision.

To further validate our analysis on sketch augmentation, we replace our sketch augmentation method with that used by Yu *et al.* [19] and perform retrievals on the SHREC'13 dataset. The experiment effectively verifies the poor performance of retrieval on intra-class models while supporting our analysis. We complete the shape retrieval task only for these models that come from the same class. We collected 100 models and their related sketches from the SHREC'13 dataset. In fact, for these models, incorrect retrieval result is likely to occur, even for our proposed framework, because they are too similar, as exemplified by shark and dolphin models.

The result is shown as Fig. 14. For every method, in general, the overall performance becomes worse, no matter which method is adopted to achieve this special retrieval task. Nevertheless, our sketch augmentation is relatively superior to others. In particular, Yu *et al.* method yield the worst result as the few intra-class differences that exist for these models seem to be removed. Following the stroke-removal operation, these incorrect samples are used to train our learning framework, leading to the worst retrieval performance.

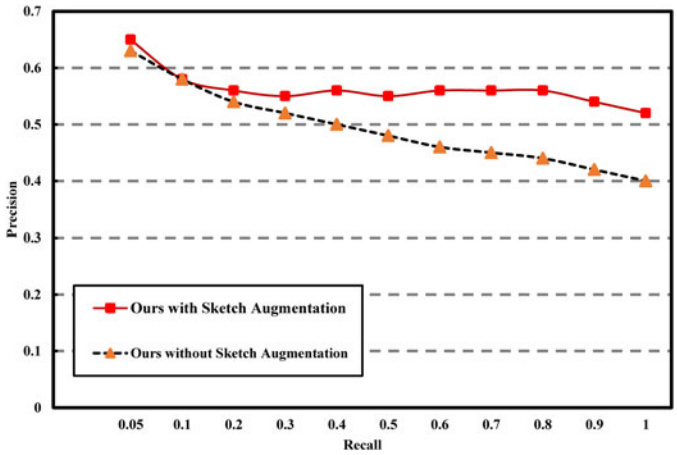

Fig. 13. Comparisons of sketch-based retrieval with and without sketch augmentation on the SHREC'13 dataset.

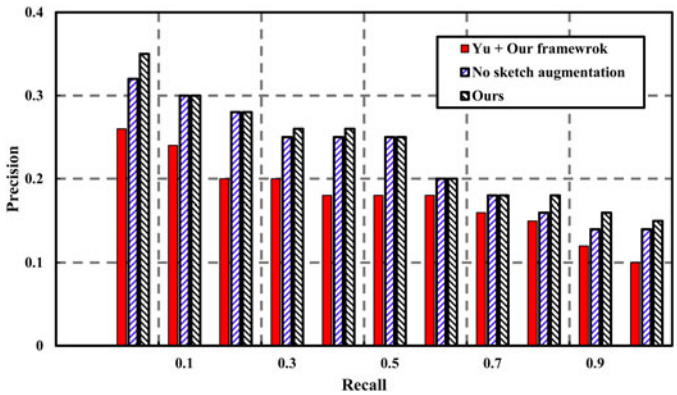

Fig. 14. Performance comparison of retrieval results based on collected intraclass models (SHREC'13).

5.4 Limitations

We conducted additional experiments on the SHREC'13 dataset. The results (see Fig. 15) show that when retrieved objects have similar shapes and differ only in detail, some incorrect retrieval results still occur. For instance, the dolphin model and the shark model have similar shapes. Moreover, when the input object is a subcomponent or a part of an object, such as a tire, incorrect retrievals can also occur.

Why do these conditions generate incorrect retrieval results? We believe one key reason is the quality of the learning samples. In this paper, we placed greater emphasis on pursuing a large number of samples rather than on assessing the quality of the samples. In the future, we plan to design a scheme to remove poor samples and propose a feasible approach or indicator to evaluate and measure these augmented sketches on whether they are qualified to be a training sample and therefore reduces incorrect retrieval results.

6 Conclusion

In this paper, we proposed a novel learning framework for conducting sketch-based shape retrieval. To obtain sufficient large-scale learning samples, we presented a novel

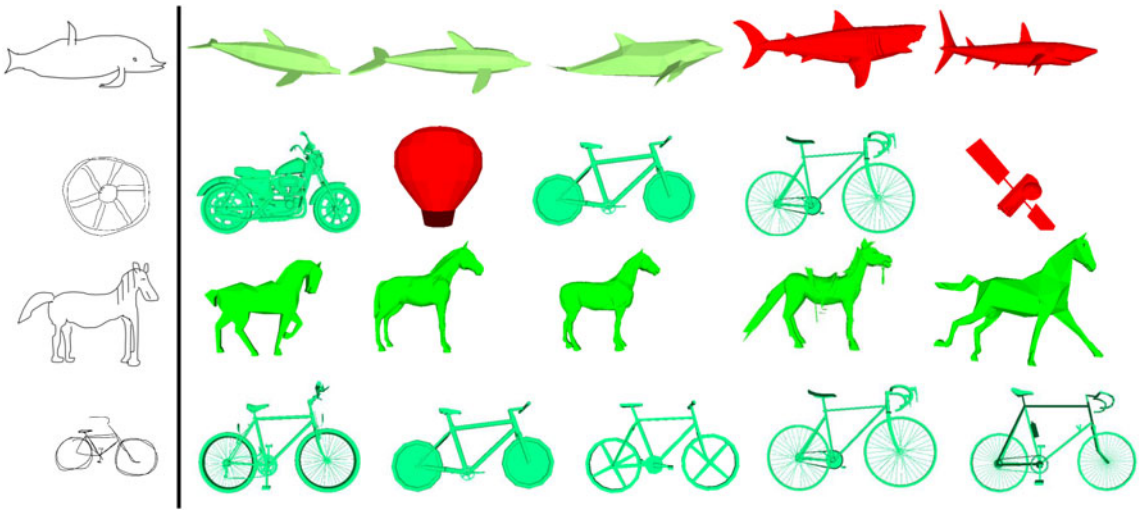

Fig. 15. Retrieval examples for some of the samples in the SHREC'13 dataset. Green denotes correct retrievals, and red represents incorrect retrievals.

sketch augmentation approach to increase the diversity and the number of the samples. In addition, we proposed a new learning-based method to identify the best views of a shape that determines the view positions that best cover the full 3D model for retrieval. Additionally, we used a Siamese CNN to learn from these samples toward the aim of retrieving better results and a joint Bayesian pipeline to measure the similarity between the output features of the Siamese networks. Using these combined techniques, we improve the performance of the final retrieval system. The extensive experiments showed that our proposed framework is comprehensively superior and more robust than existing state-of-the-art sketch-based shape retrieval approaches.

However, several problems remain in our proposed framework. For example, the learning approach for finding the best view of a 3D shape depends too heavily on the samples; when the learning samples are lacking, our approach makes it very difficult to obtain good results. Moreover, there exist many incorrectly augmented sketches that utilize training, but we lack an effective approach by which to measure whether these samples are suitable for use in the learning framework.

Considering the existing problems in our framework described above, in the future, we plan to further improve the performance of our method by employing more complex networks, such as Google-Net. Recently, end-to-end methods have attracted greater attention; it is necessary to devise a better scheme to minimize the discrepancies between sketches and models. Moreover, it is very imperative and urgent that a feasible metric criterion is designed to evaluate these augmented sketches to avoid these potentially unqualified samples from exerting negative effects on final retrieval results.

Acknowledgments

The authors would like to thank the comments and suggestions of all the anonymous reviewers, whose comments helped us to significantly improve this paper. We thank Wei Tsang Ooi from the National University of Singapore, who provided valuable inputs to our manuscript. This work was supported in part by the National Natural Science Foundation of China (NSFC) (Grants. 61902003, 61976006), the Key Project of Natural Science Foundation of China (Grant no. U19A2063) and the Doctoral Scientific Research Foundation of Anhui Normal University.

References

[1] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–10, 2012.
[2] T. Funkhouser *et al.*, "A search engine for 3D models," *ACM Trans. Graph.*, vol. 22, no. 1, pp. 83–105, 2003.
[3] M. Eitz *et al.*, "An evaluation of descriptors for large-scale image retrieval from sketched feature lines," *Comput. Graph.*, vol. 34, no. 5, pp. 482–498, 2010.
[4] B. Li, Y. Lu, and R. Fares, "Semantic sketch-based 3d model retrieval," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, 2015, pp. 555–558.
[5] N. Dalal and B. Triggs, "Semantic sketch-based 3D model retrieval," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 886–893.
[6] J. M. Saavedra "Sketch based image retrieval using a soft computation of the histogram of edge local orientations," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 2998–3002.
[7] H. Fu, H. Zhao, X. Kong, and X. Zhang, "Bhog: Binary descriptor for sketch-based image retrieval, multimedia systems," *Multimedia Syst.*, vol. 22, no. 1, pp. 127–136, 2016.
[8] H. Chatbri and K. Kameyama, "Towards making thinning algorithms robust against noise in sketch images," in *Proc. 21th Int. Conf. Pattern Recognit.*, 2012, pp. 3030–3033.
[9] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. IEEE Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.
[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[11] Y. Li, H. Lei, S Lin, and G. Luo, "A new sketch-based 3D model retrieval method by using composite features," *Multimedia Tools Appl.*, vol. 77, no. 2, pp. 2921–2944, 2018.
[12] F. Wang, L. Kang, and Y. Li, "Sketch-based 3D shape retrieval using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1875–1883.
[13] F. Zhu, J. Xie, and Y. Fang, "Learning cross-domain neural networks for sketch-based 3D shape retrieval," in *Proc. Nat. Conf. Artif. Intell.*, 2016, pp. 3683–3689.
[14] S. Chopra, R. Hadsell, and Y. Lecun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 539–546.

[15] W. T. Yih, K. Toutanova, J. C. Platt, and C. Meek, "Learning discriminative projections for text similarity measures," in *Proc. 15th Conf. Comput. Natural Lang. Learn.*, 2011, pp. 247–256.
[16] K. Chen and A. Salman, "Extracting speaker-specific information with a regularized siamese deep network," in *Proc. Advances Neural Inf. Process. Syst.*, 2011, pp. 298–306.
[17] B. Li, Y. Lu, and H. Johan, "Sketch-based 3D model retrieval by viewpoint entropybased adaptive view clustering," in *Proc. Eurographics Workshop 3D Object Retrieval*, 2013, pp. 49–56.
[18] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 566–579.
[19] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-net: A deep neural network that beats humans," *Int. J. Comput. Vis.*, vol. 122, no. 3, pp. 411–425, 2017.
[20] L. Zhao, S. Liang, J. Jia, and Y. Wei, "Learning best views of 3D shapes from sketch contour," *Vis. Comput.*, vol. 31, no. 6, pp. 765–774, 2015.
[21] H. Liu, L. Zhang, and H. Huang, "Web-image driven best views of 3D shapes," *Vis. Comput.*, vol. 28, no. 3, pp. 279–287, 2012.
[22] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. Jan Latecki "GIFT: A real-time and scalable 3D shape search engine," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5023–5032.
[23] 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.*, 2015, pp. 945–953.
[24] C. L. Zitnick, P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.
[25] H. Dutagaci, C. P. Cheung, and A. Godil, "A benchmark for best view selection of 3D objects," in *Proc. ACM Workshop 3D Object Retrieval*, 2010, pp. 45–50.
[26] C. H. Lee, H. Chang, A. Varshney, and D. W. Jacobs, "Mesh saliency," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 659–666, 2005.
[27] O. Polonsky, G. Patane, S. Biasotti, C. Gotsman, and M. Spagnuolo, "Whats in an image?," *Vis. Comput.*, vol. 21, no. 8–10, pp. 840–847, 2005.
[28] P. Vazquez, M. Feixas, M. Sbert, and W. Heidrich, "Viewpoint selection using viewpoint entropy," in *Proc. Vis. Model. Visualization*, 2001, pp. 273–280.
[29] D. L. Page, A. F. Koschan, S. R. Sukumar, B. Roui-Abidi, and M. A. Abidi, "Shape analysis algorithm based on information theory," in *Proc. Int. Conf. Image Process.*, 2003, pp. 229–232.
[30] F. Tasse and N. Dodgson, "Shape2Vec: Semantic-based descriptors for 3D shapes, sketches and images," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 208–216, 2016.
[31] G. Dai, J. Xie, and F. Zhu, "Deep correlated metric learning for sketch-based 3D shape retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3374–3382, Jul. 2018.
[32] 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. Vis. Pattern Recognit.*, 2017, pp. 3615–3623.
[33] R. Hadsell, S. Chopra, and Y. Lecun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1735–1742.
[34] B. Li *et al.*, "Extended large scale sketch-based 3D shape retrieval," in *Proc. Eurographics Workshop 3D Object Retrieval*, 2014, pp. 121–130.
[35] W. Zhou and J. Jia, "A learning framework for shape retrieval based on multilayer perceptrons," *Pattern Recognit. Lett.*, vol. 1, no. 117, pp. 119–130, 2019.
[36] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, vol. 15, pp. 44–52.
[37] Y. Li, Y.-Z. Song, and S. Gong, "Sketch recognition by ensemble matching of structured features," in *Proc. British Mach. Vis. Conf.*, 2013, vol. 1, pp. 2–12.
[38] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard, "Adaptive data augmentation for image classification," in *Proc. IEEE Int. Conf. Image Process.*, 2016, pp. 3688–3692.
[39] J. Shijie, W. Ping, J. Peiyi, and H. Siping, "Research on data augmentation for image classification based on convolution neural networks," in *Proc. Chinese Autom. Congress*, 2017, pp. 4165–4170.
[40] A. Antoniou, A. Storkey, and H. Edwards, "Augmenting image classifiers using data augmentation generative adversarial networks," in *Proc. Int. Conf. Artif. Neural Netw.*, 2018, pp. 594–603.
[41] D. Ho, E. Liang, I. Stoica, P. Abbeel, and X. Chen, "Population based augmentation: Efficient learning of augmentation policy schedules," in *Proc. 32th Int. Conf. Mach. Learn.*, 2019, pp. 2731–2741.
[42] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 113–123.
[43] W. Zhou and J. Jia, "Training deep convolutional neural networks to acquire the best view of a 3D shape," *Multimedia Tools Appl.*, vol. 79, no. 1, pp. 581–601, 2020.
[44] W. Zhou, J. Jia, C. Huang, and Y. Cheng, "Web3D learning framework for 3D shape retrieval based on hybrid convolutional neural networks," *Tsinghua Sci. Technol.*, vol. 25, no. 1, pp. 93–102, 2020.

Wen Zhou (Member, IEEE) received the PhD degree from the School of Software Engineering, Tongji University, in 2018. Since November 2018, he has been affiliated with the School of Computer and Information, Anhui Normal University, Wuhu, China, where he is currently a lecturer. He is also an a Member of the Chinese Computer Federation (CCF). His research interests include WebVR visualization, virtual reality, sketch-based retrieval and machine learning, among other areas.

Jinyuan Jia received the PhD degree from the Hong Kong University of Science and Technology, in 2004. Since 2007, he has been affiliated with the School of Software Engineering, Tongji University, Shanghai, China, where he is currently a professor. He is an ACM member, a senior member of the Chinese Computer Federation (CCF) and a senior member of the Chinese Steering Committee on Virtual Reality. His research interests include computer graphics, CAD, geometric modeling, Web3D, mobile VR, game engine, digital entertainment, computer simulation, and peer-to-peer distributed virtual environments.

Wenying Jiang received the BSc degree in digital media technology from Huaibei Normal University, in 2019. She is currently working toward the master's degree in computer science and technology at Anhui Normal University, China. Her research interests include machine learning and 3D visualization.

Chenxi Huang received the BSc and PhD degrees from the Department of Computer Science, Tongji University, in 2015 and 2019, respectively. Since 2019, he has been affiliated with the Department of Computer Science, Xiamen University, where he is currently an assistant professor. His research interests include image processing, image reconstruction, data fusion and three-dimensional visualization, and machine learning.