Deep Multi-metric Learning for Shape-based 3D Model Retrieval

Jin Xie, Guoxian Dai, and Yi Fang

***Abstract*—Recently feature learning based 3D shape retrieval methods have been receiving more and more attention in the 3D shape analysis community. In these methods, the hand-crafted metrics or the learned linear metrics are usually used to compute the distances between shape features. Since there are complex geometric structural variations with 3D shapes, the single hand-crafted metric or learned linear metric cannot characterize the manifold where 3D shapes lie well. In this paper, by exploring the non-linearity of the deep neural network and the complementarity among multiple shape features, we propose a novel deep multi-metric learning method for 3D shape retrieval. In our method, a novel deep multi-metric network is developed to learn multiple non-linear distance metrics from multiple types of shape features. The developed multi-metric network minimizes a discriminative loss function that for each type of shape feature the outputs of the network from the same class are encouraged to be as similar as possible and the outputs from different classes are encouraged to be as dissimilar as possible. Meanwhile, the Hilbert-Schmid independence criterion (HSIC) is employed to enforce the outputs of different types of shape features to be as complementary as possible. Furthermore, the weights of the learned multiple distance metrics can be adaptively determined in our developed deep metric network. The weighted distance metric is then used as the similarity for shape retrieval. We conduct the experiments with the proposed method on the four benchmark shape datasets, i.e., the Princeton Shape Benchmark (PSB), McGill, SHREC'10 ShapeGoogle and SHREC'14 Human datasets. Experimental results demonstrate that the proposed method can obtain better performance than the learned deep single metric and outperform the state-of-the-art 3D shape retrieval methods.**

***Index Terms*—3D shape retrieval, 3D shape descriptor, multiple shape features, deep neural network, metric learning.**

I. INTRODUCTION

SHAPE based 3D model retrieval is an important research topic in the 3D shape analysis community [1–3]. With the recent advancement of the 3D model acquisition technology, large amounts of 3D shape data were captured and the large-scale 3D shape dataset such as google 3D warehouse [4] was created. Given a query shape, it is preferable to develop an effective shape retrieval algorithm to search similar shapes in a large collection of 3D shapes. Different from 2D images, 3D shapes usually do not have the rich texture and color information, but has the geometric structure information. Based on geometric structures of 3D shapes, discriminative shape features can be extracted to represent shapes. Once shape features are obtained, we can employ a distance metric as the similarity between the shape features for retrieval. Nonetheless, due to large deformations with 3D shapes, how to effectively measure the distance metric between 3D shapes is still a challenging problem in 3D shape retrieval.

In the past decades, various kinds of 3D shape retrieval methods [5–12] have been proposed. These methods mainly focus on extracting novel shape features to represent 3D shapes. Once shape features are extracted, the classic hand-crafted distance metric such as the Euclidean distance, or the learned linear distance metric, is used for retrieval. For example, in [10], the similarity sensitive hashing (SSH) is employed to learn the distance between the extracted bag-of-word (BOW) features for shape retrieval. Due to complex geometric structural variations, single shape features are not discriminative enough to characterize 3D shapes. Compared to the single shape features, multiple shape features can characterize 3D shapes better, where each type of feature may contain some information that other types do not have. In the multiple shape features based shape retrieval methods [13–15], multiple types of shape features are usually concatenated together to form a new feature vector and the existing hand-crafted distance metric or the learned linear distance metric is applied on it for shape retrieval. However, since multiple shape features are not fully independent, the simple concatenation cannot fully exploit the complementary information of the multiple shape features. In addition, since there are usually large deformations with 3D shapes, the hand-crafted distance metric or the learned linear distance metric, cannot characterize the manifold of 3D shapes well.

In this paper, by exploring the non-linearity of the deep neural network and the complementarity of multiple shape features, we propose a novel deep multi-metric network to map multiple shape features to multiple non-linear feature spaces. It is expected that in the non-linear feature spaces the learned multiple deep shape features are discriminative and complementary so that they can characterize the manifold of 3D shapes well. Particularly, we construct a multi-metric network to jointly learn multiple non-linear metrics by minimizing the within-class variations of the learned shape features, maximizing the between-class variations of the learned shape features and employing the Hilbert-Schmidt independence criterion (HSIC) [16] to minimize dependence of the learned multiple shape features, simultaneously. The learned distance metrics are fused as the similarity for shape retrieval. Experimental results on four 3D shape datasets demonstrate the effectiveness of the proposed deep multi-metric learning method for 3D shape retrieval.

The main contribution of our work is that we develop a novel deep multi-metric network to learn multiple non-linear distance metrics by enforcing the outputs of the network to be discriminative and diverse. Moreover, the weights of the learned distance metrics can be obtained by optimizing

Jin Xie, Guoxian Dai, and Yi Fang are with New York University Multimedia and Vision Computing Lab, the Department of Electrical and Computer Engineering, New York University Abu Dhabi, UAE, and the Department of Computer Science and Engineering, Tandon School of Engineering, New York University, New York, USA. (e-mail:{jin.xie, guoxian.dai, yfang}@nyu.edu).

the weighted within-class and between-class variations of the learned multiple shape features. The constructed deep multi-metric network can seek multiple non-linear transformations to map multiple shape features to the non-linear and diverse feature spaces. The transformed non-linear feature spaces can more effectively characterize the manifold of the deformed 3D shapes.

The rest of the paper is organized as follows. Section II introduces related work. In Section III, we present the proposed deep multi-metric learning method for 3D shape retrieval. Section IV presents the experimental results and Section V concludes the paper.

II. Related work

Since 3D shapes can be rendered to a group of 2D depth images at different viewpoints, the classical image features can be extracted on depth images to represent shapes for retrieval. Based on the vector quantization scheme, BOW features have been widely used as shape descriptors. Furuya and Ohbuchi [8] proposed to learn the BOW feature from a collection of SIFT features for shape retrieval. Gao *et al*. [17] employed the BOW feature to describe each region of the depth images. Each 3D shape is then represented by a set of BOW features associated with the selected representative regions and the Earth Mover's distance is used for retrieval. In [18], a polygon of the contour of the projected depth image is first obtained and a set of contour fragments is generated. The BOW features are then learned from a collection of local contour fragment features for shape retrieval. In[19], based on the projected images, query views are incrementally selected for shape matching and retrieval. In [9], Bai *et al*. proposed the two layer coding framework to encode depth images to form a 3D shape descriptor for retrieval. Recently, deep learning has been employed to form deep shape descriptors [20–27] for 3D shape analysis. Leng *et al*. [25] developed a 3D convolutional neural network (CNN) to learn a shape descriptor for retrieval, which can operate on all views of a 3D model. Su *et al*. [23] proposed a multi-view CNN to learn shape descriptors from rendered images at different views. In [22], the authors converted a 3D shape into a panoramic view based image and employed a rotation invariant CNN to learn a deep representation for retrieval. Based on projected images, Bai *et al*. [24] proposed a CNN based real-time 3D shape retrieval algorithm, which can scale up to the large scale shape datasets.

The local shape descriptors such as heat kernel signature (HKS) [28], scale invariant heat kernel signature (SI-HKS) [29], wave kernel signature (WKS) [30] and covariance descriptor [31] can also be used to construct global shape descriptors for retrieval. With the K-means clustering method or the sparse coding method, the BOW feature extraction paradigm is applied to these local shape descriptors to represent 3D shapes [10, 11, 32]. For example, in [32], by selecting some scales of the HKS, EINagh *et al*. proposed the compact HKS based BOW feature for shape retrieval. Bu *et al*. [33] employed deep belief network to learn high-level shape features from the extracted BOW features. Based on the multi-scale shape distribution of the HKS, Xie *et al*. [12] developed a deep multi-scale discriminative auto-encoder and the neurons in the hidden layers are concatenated to form a global shape descriptor for retrieval. Wu *et al*. [34] proposed to represent 3D shape by a probability distribution of binary variables on a 3D voxel grid with a convolutional deep belief network. In [35, 36], based on the local geometry structures of 3D meshes, the circle convolutional Boltzmann machine and mesh convolutional deep belief network are developed to extract shape features for shape retrieval, respectively.

Recent studies show that the combination of multiple types of shape features can improve shape retrieval performance. M. Aono *et al*. [37] employed center-symmetric local binary pattern, entropy descriptor and optional chain code to describe rendered depth images for retrieval. Li *et al*. [13] developed a hybrid shape descriptor, ZFDR, by integrating both visual and geometric information of 3D shapes. The ZFDR descriptor is the combination of Zernike moments and Fourier features of projected 2D images, depth information features and ray-based features. Chen *et al*. [14] combined five types of shape features to form a weighted 3D shape descriptor for shape retrieval: D2 feature, bounding box feature, normal angle area feature, depth buffer-based feature and ray-extent feature. In [15], two types of variants of SIFT features, dense SIFT and one SIFT, are extracted from multi-view rendered depth images. The dense SIFT is a local feature while one SIFT is a global feature. Then, two types of shape descriptors are formed to represent 3D shapes: one is the BOW feature learned from a collection of dense SIFTs and the other is the concatenation of one SIFTs.

In the aforementioned methods, once the shape descriptors are extracted, the Euclidean distance is usually used as the similarity for shape retrieval. Also, the similarity sensitive hashing (SSH) [10] and the Manifold ranking [15] are used to learn distance metrics for 3D shape retrieval. For the combination of multiple types of shape features, by manually setting the weights of multiple shape features, the weighted distance metric is used for shape retrieval. Nonetheless, the simple combination does not consider the complementarity of multiple shape features and the weights between multiple shape features are not adaptively determined.

III. Proposed approach

In this section, we present the proposed deep multi-metric learning method for 3D shape retrieval in detail. Fig. 1 illustrates our proposed deep multi-metric learning framework. We first extract different types of global 3D shape descriptors to form multiple types of shape features with the locality constrained linear coding (LLC) method [38]. We then use multiple shape features as inputs to train multiple deep metric networks. For each metric network, the within-class variations of the outputs of the network are minimized and the between-class variations of the outputs are maximized. Meanwhile, dependence between the outputs from different metric networks is minimized with the HSIC. It is expected that the outputs of multiple metric networks, i.e., multiple kinds of learned shape features, are as discriminative and complementary as possible. For each metric network, the distance between the learned

shape features from two shapes is calculated. Finally, multiple distances are weighted to form the similarity for retrieval.

A. Deep Multi-metric Learning

In our deep multi-metric learning framework, based on the extracted point signatures such as SI-HKS [29], WKS [30] and LDSIFT [39], we use LLC to encode each vertex to form the encoding coefficient histograms as multiple types of shape features. We denote the vth type of shape feature from shape i by $\boldsymbol{x}_{v,i}$, $i = 1, 2, \cdots, N$, $v = 1, 2, \cdots, V$, where N is the number of shapes and V is the type number of shape features. With the input shape feature $\boldsymbol{x}_{v,i} \in R^{m\times 1}$, we can construct a deep neural network to compute the output $\boldsymbol{z}_{v,i}^K \in R^{r\times 1}$ by multiple layers of non-linear transformations, where m and r are the dimensions of the input and output of the deep neural network, K is the layer number. In the constructed network, each neuron in the current layer is connected to all neurons in the next layer. The output of layer $k+1$, $\boldsymbol{z}_{v,i}^{k+1}$, is :

$$\boldsymbol{z}_{v,i}^{k+1} = \sigma(\boldsymbol{W}_v^k \boldsymbol{z}_{v,i}^k + \boldsymbol{b}_v^k) \tag{1}$$

where $\boldsymbol{W}_v^k$ and $\boldsymbol{b}_v^k$ are the weight and bias between layer k and layer $k+1$, respectively, $\boldsymbol{z}_{v,i}^k$ is the neuron in layer k for the input shape feature $\boldsymbol{x}_{v,i}$, $\sigma(\boldsymbol{x})$ is the sigmoid function. For V types of shape features, our deep multi-metric learning framework can learn V non-linear transformations with V metric networks, where for each pair of features, $\boldsymbol{x}_{v,i}$ and $\boldsymbol{x}_{v,j}$, the distance metric between $\boldsymbol{x}_{v,i}$ and $\boldsymbol{x}_{v,j}$ can be converted into the distance metric between the outputs $\boldsymbol{z}_{v,i}^K$ and $\boldsymbol{z}_{v,j}^K$ in the transformed non-linear feature spaces. Moreover, multiple distance metrics are fused in the transformed non-linear feature spaces so that they can characterize the manifold of 3D shapes better.

In our constructed deep multi-metric network, it is desirable that the learned multiple deep shape features are discriminative and complementary so that the learned distance metrics can measure the similarities between shapes well. In order to learn the discriminative shape feature $\boldsymbol{z}_{v,i}^K$, we define the following two loss functions between the samples from the same class and the samples from different classes, respectively:

$$\begin{aligned} &l_+(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K) = \frac{1}{2}\|\boldsymbol{z}_{v,i}^K - \boldsymbol{z}_{v,i_1}^K\|_2^2 \\ &l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K) = max(\eta, \frac{1}{2}\|\boldsymbol{z}_{v,i}^K - \boldsymbol{z}_{v,i_1}^K\|_2^2 - \\ &\frac{1}{2}\|\boldsymbol{z}_{v,i}^K - \boldsymbol{z}_{v,i_2}^K\|_2^2) \end{aligned} \tag{2}$$

where $l_+(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K)$ is the loss between the positive pair $\boldsymbol{z}_{v,i}^K$ and $\boldsymbol{z}_{v,i_1}^K$, $l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K)$ is the loss between the negative pair $\boldsymbol{z}_{v,i}^K$ and $\boldsymbol{z}_{v,i_2}^K$, η is a small constant, $i = 1, 2, \cdots, N$ and $v = 1, 2, \cdots, V$. The loss $l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K)$ can achieve a gap by at least η between the dissimilarity of the positive and negative pairs.

In order to guarantee the complementarity among the multiple types of shape features, we employ the Hilbert-Schmid independence criterion (HSIC) [16] to measure dependence between the multiple types of shape features. For an independent observation (x_n, y_n) drawn from the probability distribution p_{xy}, $n = 1, 2, \cdots, N$, we define the HSIC as the Hilbert-Schmid norm of the cross-variance C_{xy}:

$$HSIC(x_1, y_1, \cdots, x_N, y_N) = \|C_{xy}\|_{HS}^2 \tag{3}$$

where the cross-variance $C_{xy} = E_{xy}[(\phi(x) - \mu_x) \otimes (\phi(y) - \mu_y)]$, $\phi(x)$ is the kernel mapping function, $\mu_x = E(\phi(x))$ and $\mu_y = E(\phi(y))$, $\otimes$ is the tensor product, the Hilbert-Schmid norm $\|\boldsymbol{A}\|_{HS} = \sqrt{\sum_{i,j} a_{ij}^2}$ and a_{ij} is the element of $\boldsymbol{A}$. However, since the joint distribution p_{xy} is usually unknown, we empirically estimate the HSIC [16] as:

$$HSIC(x_1, y_1, \cdots, x_N, y_N) = \frac{1}{(N-1)^2} tr(\boldsymbol{GHLH}) \tag{4}$$

where $\boldsymbol{G}$ and $\boldsymbol{L}$ are the Gram matrices, $\boldsymbol{G}_{i,j} = \boldsymbol{G}(x_i, x_j)$ and $\boldsymbol{L}_{i,j} = \boldsymbol{L}(y_i, y_j)$, $\boldsymbol{H}_{i,j} = \delta_{i,j} - \frac{1}{N}$. In our deep metric network, we use the inner product kernel to compute HSIC($\boldsymbol{z}_v^K$,$\boldsymbol{z}_{v'}^K$), i.e., $\boldsymbol{G}_v = (\boldsymbol{z}_v^K)^T \boldsymbol{z}_v^K$, $\boldsymbol{L}_{v'} = (\boldsymbol{z}_{v'}^K)^T \boldsymbol{z}_{v'}^K$, where $\boldsymbol{z}_v^K = [\boldsymbol{z}_{v,1}^K, \boldsymbol{z}_{v,2}^K, \cdots, \boldsymbol{z}_{v,N}^K]$ and $\boldsymbol{z}_{v'}^K = [\boldsymbol{z}_{v',1}^K, \boldsymbol{z}_{v',2}^K, \cdots, \boldsymbol{z}_{v',N}^K]$.

Based on Eqs. (2) and (4), we propose the following deep multi-metric learning model:

$$\begin{aligned} J(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\theta}) &= \frac{\alpha}{Nt} \sum_{v=1}^{V} \sum_{i=1}^{N} \sum_{i_1 \in c(i)} \theta_v l_+(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K) \\ &+ \frac{1-\alpha}{Nt} \sum_{v=1}^{V} \sum_{i=1}^{N} \sum_{i_1, i_2 \in g(i)} \theta_v l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K) + \\ &\lambda \sum_{v=1; v\neq v'} \frac{1}{2} HSIC(\boldsymbol{z}_v^K, \boldsymbol{z}_{v'}^K) + \frac{1}{2}\gamma \sum_{v=1}^{V} \sum_{k=1}^{K-1} \|\boldsymbol{W}_v^k\|_F^2 + \\ &\frac{1}{2}\tau \sum_{v=1}^{V} \theta_v^2 \quad s.t. \sum_{v=1}^{V} \theta_v = 1, \theta_v \geqslant 0 \end{aligned} \tag{5}$$

where $\boldsymbol{W} = \{\boldsymbol{W}_1^1, \boldsymbol{W}_1^2, \cdots, \boldsymbol{W}_V^{K-1}\}$, $\boldsymbol{b} = \{\boldsymbol{b}_1^1, \boldsymbol{b}_1^2, \cdots, \boldsymbol{b}_V^{K-1}\}$, and $\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_V]$, $0 \leq \alpha \leq 1$ controls the tradeoff between the losses of the positive outputs and the losses of the negative outputs, t is the number of the positive/negative outputs, θ_v is the nonnegative weight for the vth type of shape feature, $c(i)$ is the set of labels of the positive outputs for each output $\boldsymbol{z}_{v,i}^K$, $g(i)$ is the set of label pairs of the positive and negative outputs, $v = 1, 2, \cdots, V$, $v' = 1, 2, \cdots, V$, parameters λ, γ and τ are the positive scalars.

In the proposed deep multi-metric learning model, the first two terms in Eq. (5) enforce that for each type of shape feature the variations of the outputs from the same class are as small as possible and the variations of outputs from different classes are as large as possible. The third term in Eq. (5) minimizes dependence between the outputs from different types of shape features so that the outputs from different types of shape features are as complementary as possible. To furthermore explore the complementarity of multiple types of shape features, we also learn the weight θ_v for the vth type of shape feature.

B. Solving the Optimization Problem

In Eq. (5), variables $\boldsymbol{W}$, $\boldsymbol{b}$ and $\boldsymbol{\theta}$ need to be optimized. We employ the alternative optimization method to obtain the

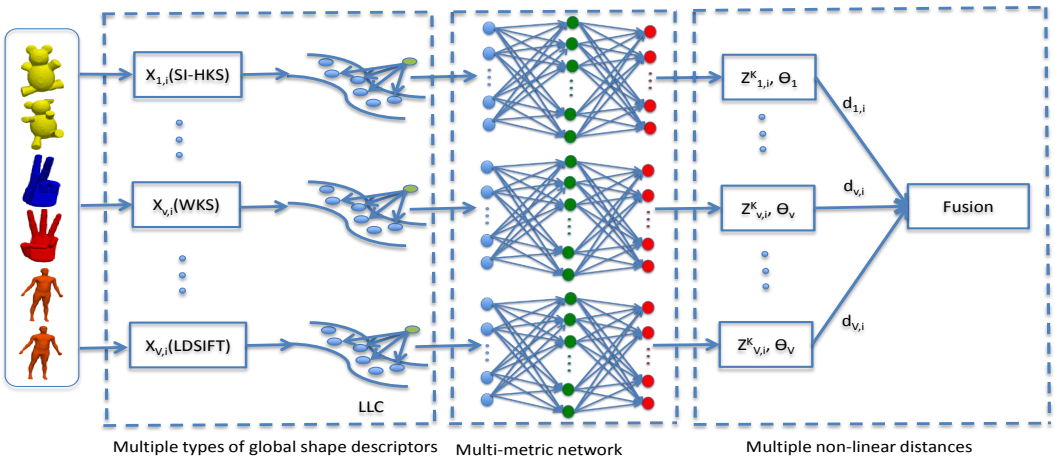


Fig. 1. The proposed deep multi-metric learning framework. Based on different types of 3D shape point signatures, the LLC method [38] is employed to extract the global shape descriptors. The formed multiple types of shape features are then fed into the deep multi-metric network so that the learned deep shape features are discriminative and complementary. The learned multiple non-linear distance metrics are fused with the learned weights for retrieval.

solution by alternately updating one variable and fixing the other variables.

Update $\boldsymbol{W}$. By fixing variables $\boldsymbol{b}$ and $\boldsymbol{\theta}$, we can use the back-propagation method [40, 41] to update variable $\boldsymbol{W}$. In Eq. (5), according to the definition of HSIC, we have:

$$\begin{aligned}&\sum_{v=1;v\neq v'}\frac{1}{2}HSIC(\boldsymbol{z}_v^K,\boldsymbol{z}_{v'}^K)\\&=\frac{1}{(N-1)^2}\sum_{v=1;v\neq v'}\frac{1}{2}tr(\boldsymbol{G}_v\boldsymbol{H}\boldsymbol{L}_{v'}\boldsymbol{H})\\&=\frac{1}{(N-1)^2}\sum_{v=1;v\neq v'}\frac{1}{2}tr((\boldsymbol{z}_v^K)^T\boldsymbol{z}_v^K\boldsymbol{H}(\boldsymbol{z}_{v'}^K)^T\boldsymbol{z}_{v'}^K\boldsymbol{H}).\end{aligned}\tag{6}$$

Denote $\frac{1}{2}tr((\boldsymbol{z}_v^K)^T\boldsymbol{z}_v^K\boldsymbol{H}(\boldsymbol{z}_{v'}^K)^T\boldsymbol{z}_{v'}^K\boldsymbol{H})$ in Eq. (6) by $e(\boldsymbol{z}_v^K,\boldsymbol{z}_{v'}^K)$. The partial derivative of the objective function $J(\boldsymbol{W},\boldsymbol{b},\boldsymbol{\theta})$ with respect to $\boldsymbol{W}_v^k$ can be computed as:

$$\begin{aligned}\frac{\partial J(\boldsymbol{W},\boldsymbol{b},\boldsymbol{\theta})}{\partial\boldsymbol{W}_v^k}&=\frac{\alpha}{Nt}\sum_{i=1}^{N}\sum_{i_1\in c(i)}\theta_v\frac{\partial l_+(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_1}^K)}{\partial\boldsymbol{W}_v^k}+\\&\frac{1-\alpha}{Nt}\sum_{i=1}^{N}\sum_{i_1,i_2\in g(i)}\theta_v\frac{\partial l_-(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_1}^K,\boldsymbol{z}_{v,i_2}^K)}{\partial\boldsymbol{W}_v^k}+\\&\frac{\lambda}{(N-1)^2}\sum_{v'\neq v}\frac{\partial e(\boldsymbol{z}_v^K,\boldsymbol{z}_{v'}^K)}{\partial\boldsymbol{W}_v^k}+\gamma\boldsymbol{W}_v^k.\end{aligned}\tag{7}$$

For layer k, let $\boldsymbol{a}_{v,i}^{k+1}$ be the weighted sum in layer $k+1$, $\boldsymbol{a}_{v,i}^{k+1}=\boldsymbol{W}_v^k\boldsymbol{z}_{v,i}^k+\boldsymbol{b}_v^k$, $k=1,2,\cdots,K-1$. $\frac{\partial l_+(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_1}^K)}{\partial\boldsymbol{W}_v^k}$ can be re-written as the following formula:

$$\begin{aligned}\frac{\partial l_+(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_1}^K)}{\partial\boldsymbol{W}_v^k}&=\frac{\partial l_+(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_1}^K)}{\partial\boldsymbol{a}_{v,i}^{k+1}}\frac{\partial\boldsymbol{a}_{v,i}^{k+1}}{\partial\boldsymbol{W}_v^k}+\\&\frac{\partial l_+(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_1}^K)}{\partial\boldsymbol{a}_{v,i_1}^{k+1}}\frac{\partial\boldsymbol{a}_{v,i_1}^{k+1}}{\partial\boldsymbol{W}_v^k}.\end{aligned}\tag{8}$$

Denote $\frac{1}{2}\|\boldsymbol{z}_{v,i}^K-\boldsymbol{z}_{v,i_2}^K\|_2^2$ in $l_-(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_1}^K,\boldsymbol{z}_{v,i_2}^K)$ by $d(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_2}^K)$. Similar to Eq. (8), we can obtain:

$$\begin{aligned}\frac{\partial d(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_2}^K)}{\partial\boldsymbol{W}_v^k}&=\frac{\partial d(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_2}^K)}{\partial\boldsymbol{a}_{v,i}^{k+1}}\frac{\partial\boldsymbol{a}_{v,i}^{k+1}}{\partial\boldsymbol{W}_v^k}\\&+\frac{\partial d(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_2}^K)}{\partial\boldsymbol{a}_{v,i_2}^{k+1}}\frac{\partial\boldsymbol{a}_{v,i_2}^{k+1}}{\partial\boldsymbol{W}_v^k}.\end{aligned}\tag{9}$$

$\frac{\partial e(\boldsymbol{z}_v^K,\boldsymbol{z}_{v'}^K)}{\partial\boldsymbol{W}_v^k}$ can be re-written as:

$$\frac{\partial e(\boldsymbol{z}_v^K,\boldsymbol{z}_{v'}^K)}{\partial\boldsymbol{W}_v^k}=\sum_i\frac{\partial e(\boldsymbol{z}_v^K,\boldsymbol{z}_{v'}^K)}{\partial\boldsymbol{a}_{v,i}^{k+1}}\frac{\partial\boldsymbol{a}_{v,i}^{k+1}}{\partial\boldsymbol{W}_v^k}.\tag{10}$$

Let $\frac{\partial l_+(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_1}^K)}{\partial\boldsymbol{a}_{v,i}^{k+1}}$, $\frac{\partial l_+(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_1}^K)}{\partial\boldsymbol{a}_{v,i_1}^{k+1}}$, $\frac{\partial d(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_2}^K)}{\partial\boldsymbol{a}_{v,i}^{k+1}}$, $\frac{\partial d(\boldsymbol{z}_{v,i}^K,\boldsymbol{z}_{v,i_2}^K)}{\partial\boldsymbol{a}_{v,i_2}^{k+1}}$ and $\frac{\partial e(\boldsymbol{z}_v^K,\boldsymbol{z}_{v'}^K)}{\partial\boldsymbol{a}_{v,i}^{k+1}}$ be the errors $\boldsymbol{\delta}_{k+1,i}^{l,v}$, $\boldsymbol{\delta}_{k+1,i_1}^{l,v}$, $\boldsymbol{\delta}_{k+1,i}^{d,v}$, $\boldsymbol{\delta}_{k+1,i_2}^{d,v}$ and $\boldsymbol{\delta}_{k+1,i}^{e,v}$, respectively. For $k=K-1$, $\boldsymbol{\delta}_{K,i}^{l,v}$, $\boldsymbol{\delta}_{K,i_1}^{l,v}$, $\boldsymbol{\delta}_{K,i}^{d,v}$, $\boldsymbol{\delta}_{K,i_2}^{d,v}$, $\boldsymbol{\delta}_{K,i}^{e,v}$ can be represented as:

$$\begin{aligned}\boldsymbol{\delta}_{K,i}^{l,v}&=(\boldsymbol{z}_{v,i}^K-\boldsymbol{z}_{v,i_1}^K)\bullet\sigma'(\boldsymbol{a}_{v,i}^K)\\\boldsymbol{\delta}_{K,i_1}^{l,v}&=(-\boldsymbol{z}_{v,i}^K+\boldsymbol{z}_{v,i_1}^K)\bullet\sigma'(\boldsymbol{a}_{v,i_1}^K)\\\boldsymbol{\delta}_{K,i}^{d,v}&=(\boldsymbol{z}_{v,i}^K-\boldsymbol{z}_{v,i_2}^K)\bullet\sigma'(\boldsymbol{a}_{v,i}^K)\\\boldsymbol{\delta}_{K,i_2}^{d,v}&=(-\boldsymbol{z}_{v,i}^K+\boldsymbol{z}_{v,i_2}^K)\bullet\sigma'(\boldsymbol{a}_{v,i_2}^K)\\\boldsymbol{\delta}_{K,i}^{e,v}&=(\boldsymbol{z}_v^K\boldsymbol{H}(\boldsymbol{z}_{v'}^K)^T\boldsymbol{z}_{v'}^K\boldsymbol{H})_i\bullet\sigma'(\boldsymbol{a}_{v,i}^K)\end{aligned}\tag{11}$$

where $\sigma'(\boldsymbol{a}_{v,i}^K)$ is the derivative of the activation function in the output layer, $\bullet$ denotes the element-wise multiplication and $(\boldsymbol{z}_v^K\boldsymbol{H}(\boldsymbol{z}_{v'}^K)^T\boldsymbol{z}_{v'}^K\boldsymbol{H})_i$ is the ith column of the matrix $\boldsymbol{z}_v^K\boldsymbol{H}(\boldsymbol{z}_{v'}^K)^T\boldsymbol{z}_{v'}^K\boldsymbol{H}$. For layer $k=K-2,K-3,\cdots,1$, with the back-propagation algorithm, $\boldsymbol{\delta}_{k+1,i}^{l,v}$ can be obtained as:

$$\boldsymbol{\delta}_{k+1,i}^{l,v}=((\boldsymbol{W}_v^{k+1})^T\boldsymbol{\delta}_{k+2,i}^{l,v})\bullet\sigma'(\boldsymbol{a}_{v,i}^{k+1}).\tag{12}$$

Similar to Eq. (12), $\boldsymbol{\delta}_{k+1,i_1}^{l,v}$, $\boldsymbol{\delta}_{k+1,i}^{d,v}$, $\boldsymbol{\delta}_{k+1,i_2}^{d,v}$ and $\boldsymbol{\delta}_{k+1,i}^{e,v}$ can also be obtained as:

$$\begin{aligned}\boldsymbol{\delta}_{k+1,i_1}^{l,v}&=((\boldsymbol{W}_v^{k+1})^T\boldsymbol{\delta}_{k+2,i_1}^{l,v})\bullet\sigma'(\boldsymbol{a}_{v,i_1}^{k+1})\\\boldsymbol{\delta}_{k+1,i}^{d,v}&=((\boldsymbol{W}_v^{k+1})^T\boldsymbol{\delta}_{k+2,i}^{d,v})\bullet\sigma'(\boldsymbol{a}_{v,i}^{k+1})\\\boldsymbol{\delta}_{k+1,i_2}^{d,v}&=((\boldsymbol{W}_v^{k+1})^T\boldsymbol{\delta}_{k+2,i_2}^{d,v})\bullet\sigma'(\boldsymbol{a}_{v,i_2}^{k+1})\\\boldsymbol{\delta}_{k+1,i}^{e,v}&=((\boldsymbol{W}_v^{k+1})^T\boldsymbol{\delta}_{k+2,i}^{e,v})\bullet\sigma'(\boldsymbol{a}_{v,i}^{k+1}).\end{aligned}\tag{13}$$

Thus, $\frac{\partial l_+(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K)}{\partial \boldsymbol{W}_v^k}$ can be represented as:

$$\frac{\partial l_+(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K)}{\partial \boldsymbol{W}_v^k} = \boldsymbol{\delta}_{k+1,i}^{l,v}(\boldsymbol{z}_{v,i}^k)^T + \boldsymbol{\delta}_{k+1,i_1}^{l,v}(\boldsymbol{z}_{v,i_1}^k)^T. \quad (14)$$

For $\frac{\partial l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K)}{\partial \boldsymbol{W}_v^k}$, if $\eta \geq \|\boldsymbol{z}_{v,i}^K - \boldsymbol{z}_{v,i_1}^K\|_2^2 - \|\boldsymbol{z}_{v,i}^K - \boldsymbol{z}_{v,i_2}^K\|_2^2$, $\frac{\partial l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K)}{\partial \boldsymbol{W}_v^k} = \boldsymbol{0}$; if $\eta < \|\boldsymbol{z}_{v,i}^K - \boldsymbol{z}_{v,i_1}^K\|_2^2 - \|\boldsymbol{z}_{v,i}^K - \boldsymbol{z}_{v,i_2}^K\|_2^2$, $\frac{\partial l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K)}{\partial \boldsymbol{W}_v^k}$ can be represented as:

$$\begin{aligned}\frac{\partial l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K)}{\partial \boldsymbol{W}_v^k} &= \boldsymbol{\delta}_{k+1,i}^{l,v}(\boldsymbol{z}_{v,i}^k)^T + \boldsymbol{\delta}_{k+1,i_1}^{l,v}(\boldsymbol{z}_{v,i_1}^k)^T \\ &- \boldsymbol{\delta}_{k+1,i}^{d,v}(\boldsymbol{z}_{v,i}^k)^T - \boldsymbol{\delta}_{k+1,i_2}^{d,v}(\boldsymbol{z}_{v,i_2}^k)^T.\end{aligned} \quad (15)$$

$\frac{\partial e(\boldsymbol{z}_v^K, \boldsymbol{z}_{v'}^K)}{\partial \boldsymbol{W}_v^k}$ can be calculated as:

$$\frac{\partial e(\boldsymbol{z}_v^K, \boldsymbol{z}_{v'}^K)}{\partial \boldsymbol{W}_v^k} = \sum_i \boldsymbol{\delta}_{k+1,i}^{e,v}(\boldsymbol{z}_{v,i}^k)^T. \quad (16)$$

With Eqs. (14), (15) and (16), we can obtain $\frac{\partial J(\boldsymbol{W},\boldsymbol{b},\boldsymbol{\theta})}{\partial \boldsymbol{W}_v^k}$. Then $\boldsymbol{W}_v^k$ is updated with the gradient descent method.

Update $\boldsymbol{b}$. By fixing variables $\boldsymbol{W}$ and $\boldsymbol{\theta}$, we can also use the back-propagation method to update variable $\boldsymbol{b}$. The partial derivative of the objective function $J(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\theta})$ with respect to $\boldsymbol{b}_v^k$ can be computed as:

$$\begin{aligned}\frac{\partial J(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\theta})}{\partial \boldsymbol{b}_v^k} &= \frac{\alpha}{Nt}\sum_{i=1}^N \sum_{i_1 \in c(i)} \theta_v \frac{\partial l_+(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K)}{\partial \boldsymbol{b}_v^k} + \\ &\frac{1-\alpha}{Nt}\sum_{i=1}^N \sum_{i_1,i_2 \in g(i)} \theta_v \frac{\partial l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K)}{\partial \boldsymbol{b}_v^k} + \\ &\frac{\lambda}{(N-1)^2}\sum_{v \neq v'} \frac{\partial e(\boldsymbol{z}_v^K, \boldsymbol{z}_{v'}^K)}{\partial \boldsymbol{b}_v^k}.\end{aligned} \quad (17)$$

The partial derivative of $l_+(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K)$ with respect to $\boldsymbol{b}_v^k$, $\frac{\partial l_+(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K)}{\partial \boldsymbol{b}_v^k}$, can be represented as:

$$\begin{aligned}\frac{\partial l_+(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K)}{\partial \boldsymbol{b}_v^k} &= \frac{\partial l_+(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K)}{\partial \boldsymbol{a}_{v,i}^{k+1}} + \frac{\partial l_+(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K)}{\partial \boldsymbol{a}_{v,i_1}^{k+1}} \\ &= \boldsymbol{\delta}_{k+1,i}^{l,v} + \boldsymbol{\delta}_{k+1,i_1}^{l,v}.\end{aligned} \quad (18)$$

For $\frac{\partial d(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_2}^K)}{\partial \boldsymbol{b}_v^k}$, we can calculate as:

$$\begin{aligned}\frac{\partial d(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_2}^K)}{\partial \boldsymbol{b}_v^k} &= \frac{\partial d(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_2}^K)}{\partial \boldsymbol{a}_{v,i}^{k+1}} + \frac{\partial d(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_2}^K)}{\partial \boldsymbol{a}_{v,i_2}^{k+1}} \\ &= \boldsymbol{\delta}_{k+1,i}^{d,v} + \boldsymbol{\delta}_{k+1,i_2}^{d,v}.\end{aligned} \quad (19)$$

Thus, for $\frac{\partial l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K)}{\partial \boldsymbol{b}_v^k}$, if $\eta \geq \|\boldsymbol{z}_{v,i}^K - \boldsymbol{z}_{v,i_1}^K\|_2^2 - \|\boldsymbol{z}_{v,i}^K - \boldsymbol{z}_{v,i_2}^K\|_2^2$, $\frac{\partial l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K)}{\partial \boldsymbol{b}_v^k} = \boldsymbol{0}$; if $\eta < \|\boldsymbol{z}_{v,i}^K - \boldsymbol{z}_{v,i_1}^K\|_2^2 - \|\boldsymbol{z}_{v,i}^K - \boldsymbol{z}_{v,i_2}^K\|_2^2$, $\frac{\partial l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K)}{\partial \boldsymbol{b}_v^k}$ can be represented as:

$$\frac{\partial l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K)}{\partial \boldsymbol{b}_v^k} = \boldsymbol{\delta}_{k+1,i}^{l,v} + \boldsymbol{\delta}_{k+1,i_1}^{l,v} - \boldsymbol{\delta}_{k+1,i}^{d,v} - \boldsymbol{\delta}_{k+1,i_2}^{d,v}. \quad (20)$$

Algorithm 1 Training algorithm of the proposed deep multi-metric learning model.

Input: multiple shape features $\boldsymbol{x}_{v,i}$; layer number K of the network; weight α; constant η; regularization parameters λ, γ and τ; learning rate β.
Output: $\boldsymbol{W}$, $\boldsymbol{b}$ and $\boldsymbol{\theta}$.
For $s = 1, 2, \cdots, S$:

1) Compute the forward outputs of the neural network for all shape features $\boldsymbol{x}_{v,i}$, $v = 1, 2, \cdots, V$;
2) For $k = K-1, K-2, \cdots, 1$
   Compute $\frac{\partial J(\boldsymbol{W},\boldsymbol{b},\boldsymbol{\theta})}{\partial \boldsymbol{W}_v^k}$ with Eqs. (14), (15) and (16) and update $\boldsymbol{W}_v^k$: $\boldsymbol{W}_v^k = \boldsymbol{W}_v^k - \beta\frac{\partial J(\boldsymbol{W},\boldsymbol{b},\boldsymbol{\theta})}{\partial \boldsymbol{W}_v^k}$;
   Compute $\frac{\partial J(\boldsymbol{W},\boldsymbol{b},\boldsymbol{\theta})}{\partial \boldsymbol{b}_v^k}$ with Eqs. (18), (20) and (21) and update $\boldsymbol{b}_v^k$: $\boldsymbol{b}_v^k = \boldsymbol{b}_v^k - \beta\frac{\partial J(\boldsymbol{W},\boldsymbol{b},\boldsymbol{\theta})}{\partial \boldsymbol{b}_v^k}$;
3) Update $\boldsymbol{\theta}$ with the interior-point algorithm to solve the quadratic programming problem Eq. (22).

Output $\boldsymbol{W}$, $\boldsymbol{b}$ and $\boldsymbol{\theta}$ until the difference between the values of $J(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\theta})$ in adjacent iterations is smaller than the threshold or the setting iteration number is reached.

$\frac{\partial e(\boldsymbol{z}_v^K, \boldsymbol{z}_{v'}^K)}{\partial \boldsymbol{b}_v^k}$ can be calculated as:

$$\frac{\partial e(\boldsymbol{z}_v^K, \boldsymbol{z}_{v'}^K)}{\partial \boldsymbol{b}_v^k} = \sum_i \boldsymbol{\delta}_{k+1,i}^{e,v}. \quad (21)$$

With Eqs. (18), (20) and (21), we can obtain $\frac{\partial J(\boldsymbol{W},\boldsymbol{b},\boldsymbol{\theta})}{\partial \boldsymbol{b}_v^k}$.

Update $\boldsymbol{\theta}$. By fixing variables $\boldsymbol{W}$ and $\boldsymbol{b}$, Eq. (5) can be converted into a standard quadratic programming problem:

$$\begin{aligned}\hat{\boldsymbol{\theta}} = argmin_{\boldsymbol{\theta}} &\sum_{v=1}^V \theta_v \sum_{i=1}^N \Big(\sum_{i_1 \in c(i)} \frac{\alpha}{Nt} l_+(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K) + \\ &\sum_{i_1,i_2 \in g(i)} \frac{1-\alpha}{Nt} l_-(\boldsymbol{z}_{v,i}^K, \boldsymbol{z}_{v,i_1}^K, \boldsymbol{z}_{v,i_2}^K)\Big) + \frac{1}{2}\tau\sum_{v=1}^V \theta_v^2 \\ s.t. &\sum_{v=1}^V \theta_v = 1, \theta_v \geqslant 0.\end{aligned} \quad (22)$$

Variable $\boldsymbol{\theta}$ can be solved with the classical interior-point method [42]. The training algorithm of the proposed deep multi-metric learning model is summarized in Algorithm. 1.

Once $\boldsymbol{W}$, $\boldsymbol{b}$ and $\boldsymbol{\theta}$ are learned, in order to exploit the complementarity of the multiple shape features, we can use the fused distance metric $\sum \theta_v \|\boldsymbol{z}_{v,i}^K - \boldsymbol{z}_{v,j}^K\|_2$ for retrieval.

IV. Experimental results

In this section, we first evaluate our proposed deep multi-metric learning model for retrieval, and then compare it with the state-of-the-art 3D shape retrieval methods on four benchmark datasets, i.e., Princeton Shape Benchmark (PSB) [43], McGill shape dataset [44], SHREC'10 ShapeGoogle dataset [10] and SHREC'14 Human dataset [45].

A. Experimental Settings

For the four 3D shape datasets, we extract different types of point signatures to form multiple shape features. The point signatures used in our paper is shown as follows.

- **SI-HKS.** Scale invariant heat kernel signature (SI-HKS) [29] is the scale invariant version of heat kernel signature (HKS) [28]. SI-HKS can be constructed by taking the absolute values of the Fourier transform of the derivative of HKS at different frequencies.
- **WKS.** In wave kernel signature (WKS) [30], the behavior of a quantum particle on the meshed surface is modeled by the Schrödinger equation. Based on the solution of the Schrödinger equation, the average probability to measure the particle at the point is used to construct WKS.
- **LDSIFT.** For each interested point on the meshed surface, the depth map of the point is computed by projecting a neighborhood to its dominant plane. Then we can compute the SIFT descriptor on the projected local depth map to form the local depth SIFT (LDSIFT) descriptor [39].

For SI-HKS, we take 19 frequency components to compute SI-HKS and form a 19-dimensional feature vector. We also choose 100-dimensional WKS and 128-dimensional LDSIFT to describe shapes. In the LLC method, for each type of point signature, the size of the learned dictionary is 2000 and 5 atoms are selected to form the sub-dictionary. Thus, for each shape, three types of 2000-dimensional global 3D shape descriptors are formed as multiple shape features. In the proposed deep multi-metric learning model, the neural network with layers of 2000-1500-1000-500 is used. Moreover, in Eq. (5), parameters η, α, λ, γ and τ are set to 4.5, 0.6, 0.06, 0.001 and 0.2, respectively. Moreover, for each training sample, we choose 3 positive/negative samples to form the positive/negative pairs.

The PSB dataset consists of two subsets: training subset and testing subset. The training subset contains 907 3D shapes from 90 classes while the testing subset contains 907 3D shapes from 92 classes. Fig. 2 shows example shapes in the PSB dataset.

The McGill 3D shape dataset includes ten objects: ant, crab, spectacle, hand, human, octopus, plier, snake, spider and teddy-bear, which contains 255 3D shapes. There are significant part articulations with the shapes. Fig. 3 shows the ten objects in the McGill shape dataset.

The SHREC'10 ShapeGoogle dataset consists of 715 shapes from 13 classes of objects. In order to make the dataset challenging, 456 shapes unrelated to 13 classes of objects are also included in this dataset. Five simulated transformations, isometry, topology, isometry+topology, partiality and triangulation, are applied to 715 3D shapes (55 transformations per class). The shapes are represented by triangular meshes with different numbers of vertices ranging from 300 and 30000. Fig. 4 shows the example model with the five simulated transformations and five unrelated example shapes in the SHREC'10 ShapeGoogle dataset.

The SHREC'14 Human dataset contains two sub-datasets. One is created by DAZ studio, called the synthetic sub-dataset. The other is obtained by scanning real human body shapes, called the scanned sub-dataset. The synthetic sub-dataset includes 300 human shapes from 15 different human models, where there are five male, five female and five child body models. The scanned sub-dataset is composed of 400 scanned human shapes from 40 human models, where half the models are male and half female. Different from the McGill 3D shape dataset, there are only human shapes in this dataset. Compared to the generic objects in the McGill 3D shape dataset, differences between human bodies are much more subtle, which makes this dataset very challenging. Fig. 5 shows the example human shapes in this dataset.

B. Evaluation of the Proposed Method

1) Comparison to the learned single distance metric: In order to demonstrate the effectiveness of the proposed deep multi-metric learning model for shape retrieval, we compare the proposed model to the learned single distance metric on the McGill shape dataset.

In our evaluation, the single distance metric is learned with the model:

$$argmin_{\boldsymbol{W},\boldsymbol{b}} \frac{\alpha}{Nt} \sum_{i=1}^{N} \sum_{i_1 \in c(i)} l_+(\boldsymbol{z}_i^K, \boldsymbol{z}_{i_1}^K) + \frac{1-\alpha}{Nt} \sum_{i=1}^{N} \sum_{i_1, i_2 \in g(i)} l_-(\boldsymbol{z}_i^K, \boldsymbol{z}_{i_1}^K, \boldsymbol{z}_{i_2}^K) + \frac{1}{2}\gamma \sum_{k=1}^{K-1} \|\boldsymbol{W}^k\|_F^2. \tag{23}$$

Once $\boldsymbol{W}$ and $\boldsymbol{b}$ are learned, we can use the Euclidean distance between the outputs of the network as the similarity for shape retrieval.

We denote the proposed deep multi-metric learning model by MMLM. We denote the deep single metric learning model with SI-HKS, WKS and LDSIFT by SI-HKS_SMLM, WKS_SMLM and LDSIFT_SMLM, respectively. The mean average precision (MAP) is used to evaluate these methods. The comparison results are illustrated in Table I. One can see that the proposed MMLM is superior to the single metric learning model. Compared to the single metric learning model, the deep multi-metric learning model can exploit the complementary information of multiple types of shape features to characterize 3D shapes better. In addition, for the learned single distance metric, we can see that WKS_SMLM is better than SI-HKS_SMLM and LDSIFT_SMLM. This implies that WKS is more discriminative to characterize 3D shapes.

TABLE I
EVALUATION OF THE PROPOSED MMLM AND THE LEARNED SINGLE DISTANCE METRIC.

Methods	Mean average precision (MAP)
SI-HKS_SMLM	0.831
WKS_SMLM	0.883
LDSIFT_SMLM	0.745
MMLM	**0.937**

2) Comparison to the fused multiple shape features: We also compare our proposed MMLM method to the fused multiple shape features without applying the developed deep multi-metric network. In this evaluation, we employ the LLC method to encode the SI-HKS, WKS and LDSIFT features of each vertex on 3D shapes. The formed multiple global shape features are denoted by SI-HKS_LLC, WKS_LLC and LDSIFT_LLC, respectively. The three types of shape features are fused by the weighted Euclidean distances, $\lambda_1 d_1 + \lambda_2 d_2 + (1-\lambda_1-\lambda_2)d_3$, where λ_1 and λ_2 are the weights, d_1, d_2 and d_3 are the

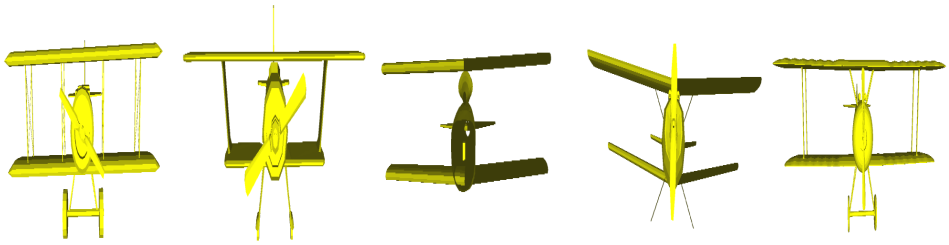


Fig. 2. The airplane models in the PSB dataset.

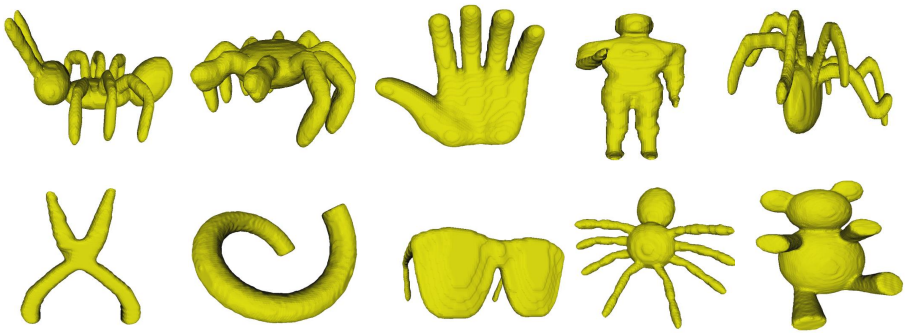
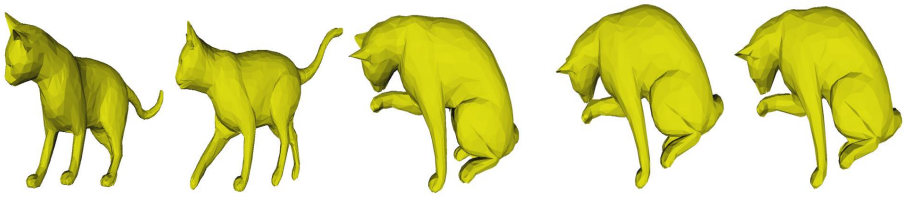
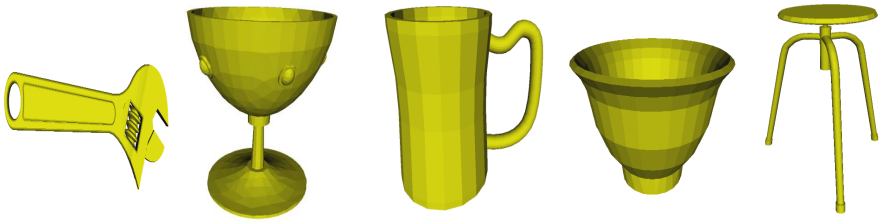


Fig. 3. Ten classes of 3D models in the McGill shape dataset.



(a)



(b)

Fig. 4. (a) shows the simulated isometry, isometry+topology, topology, partiality and triangulation transformations while (b) shows the unrelated example shapes in the SHREC'10 ShapeGoogle dataset.

Euclidean distances between the SI-HKS_LLC, WKS_LLC and LDSIFT_LLC features, respectively. Parameters λ_1 and λ_2 are determined from 0.001 to 1 with step 0.003 on the training dataset. Fig. 6 shows the precision-recall curves for the fused multiple shape features without using the deep multi-metric network and the proposed MMLM method. As can be seen in this figure, the proposed MMLM method can significantly improve the retrieval performance, which implies that by mapping the multiple shape features to the non-linear feature spaces with the deep metric network, they can characterize 3D shapes better.

C. Comparison Evaluation

1) PSB dataset: For the PSB shape dataset, the training subset is used to train the proposed multi-metric network and the testing subset is used to evaluate the MMLM method. We compare our proposed MMLM method to the following 3D shape retrieval methods: the hybrid 2D/3D approach [46], the Bag of Visual Feature method (BoVF) [8], Compact multiview descriptor (CMVD) [47], 3D CNN [25] and GIFT [24]. We

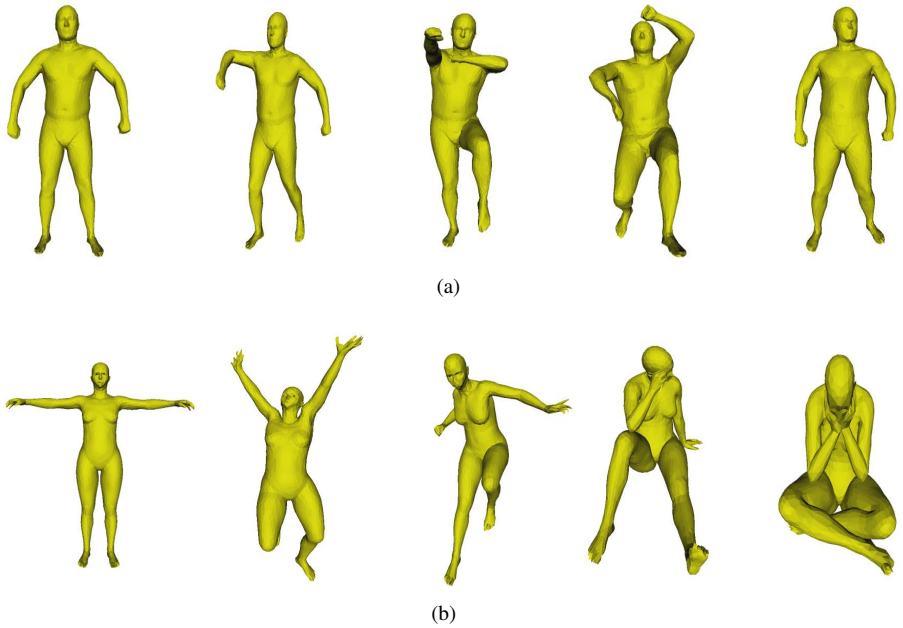


Fig. 5. (a) and (b) show the synthetic and scanned example shapes of human bodies with different poses in the SHREC'14 Human dataset, respectively.

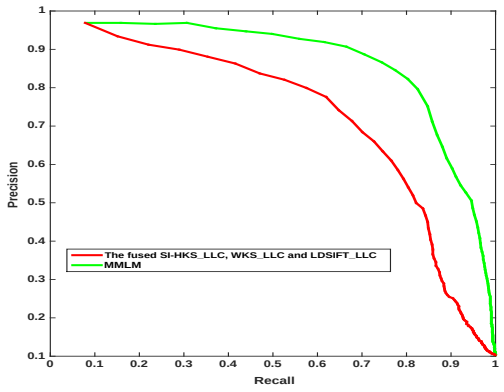


Fig. 6. The precision-recall curves for the fused multiple shape features without using the deep multi-metric network and the proposed MMLM method on the McGill shape dataset.

use the Nearest Neighbor (NN), the First Tier (FT), the Second Tier (ST) and the Discounted Cumulative Gain (DCG) to evaluate these methods. The comparison results are listed in Table II. From this table, one can see that the proposed MMLM method can yield good performance. Particularly, by fusing multiple learned shape features, the proposed MMLM method is slightly higher than the deep learning based shape retrieval methods [24, 25] in terms of the NN and FT criteria.

TABLE II
RETRIEVAL RESULTS ON THE PSB DATASET.

Methods	NN	FT	ST	DCG
BoVF [8]	0.481	0.253	0.345	0.527
Hybrid 2D/3D [46]	0.742	0.473	0.606	-
CMVD [47]	0.566	0.286	0.367	0.564
3D CNN [25]	0.901	0.639	**0.849**	0.841
GIFT [24]	0.849	0.712	0.830	-
MMLM	**0.911**	**0.720**	0.831	**0.863**

2) McGill shape dataset: For the McGill shape dataset, 10 shapes per class are chosen as the training samples to train the proposed deep multi-metric network. The remaining samples per class are used to test. All experiments are repeated over 20 times to report the retrieval performance. We compare our proposed MMLM method to the state-of-the-art shape retrieval methods: learning based covariance descriptor [31], Graph based method [48], the PCA based VLAT method [49], the Hybrid BOW [50], the hybrid 2D/3D approach [46], the manifold ranking method (MR) [15], the discriminative auto-encoder method (DA) [12]. Particularly, in [15, 31, 46, 50], multiple shape features are employed to represent 3D shapes. The Euclidean distance is used as the similarity between the shape features for retrieval in [31, 46, 50] while in [15] the manifold ranking based metric learning method is employed to learn the distance metric for retrieval.

The retrieval performance of these methods is illustrated in Table III. As can be seen in this table, in terms of the evaluation criteria FT, ST and DCG, compared to these methods, the proposed MMLM method can achieve the best performance. It is noted that, although in the Hybrid BOW method [50], the hybrid 2D/3D approach [46] and the manifold ranking method [15], different types of shape features are fused for shape retrieval, the discriminative information and the complementary information of multiple shape features are not fully exploited. In our proposed MMLM method, multiple shape features are mapped to the non-linear feature spaces with the developed multi-metric network, where for each type of learned deep shape feature the within-class variation is required to be as small as possible and the between-class variation to be as large as possible while multiple types of learned deep shape features are enforced to be as complementary as possible. Therefore, the distances between multiple types of learned deep shape features is more effective for retrieval.

TABLE III
RETRIEVAL RESULTS ON THE MCGILL DATASET.

Methods	NN	FT	ST	DCG
Covariance descriptor [31]	0.977	0.732	0.818	0.937
Graph based method [48]	0.976	0.741	0.911	0.933
PCA based VLAT [49]	0.969	0.658	0.781	0.894
Hybrid BOW [50]	0.957	0.635	0.790	0.886
Hybrid 2D/3D [46]	0.925	0.557	0.698	0.850
MR [15]	-	0.903	-	-
DA [12]	**0.988**	0.812	0.934	-
MMLM	0.971	**0.916**	**0.991**	**0.973**

3) SHREC'10 ShapeGoogle dataset: For the SHREC'10 ShapeGoogle dataset, the dictionary learning and discriminative auto-encoder based shape retrieval methods are involved to compare: the vector quantization based BOW method (VQ) [10], the unsupervised dictionary learning method (UDL) [11], the supervised dictionary learning method (SDL) [11] and the discriminative auto-encoder method (DA) [12]. All experiments are repeated over 20 times. Comparison results with the mean average precision are listed in Table IV. From this table, one can see that in the cases of the five simulated transformations, our proposed MMLM method is comparable or superior to these methods. For example, in the cases of isometry+topology, partiality and triangulation transformations, our proposed MMLM method can obtain the accuracies of 0.988, 0.985 and 0.963 while the DA method [12] can obtain the accuracies of 0.982, 0.973 and 0.955.

In the vector quantization based BOW method [10], the similarity sensitive hashing (SSH) method is used to learn a linear distance metric between the shape descriptors for retrieval, which maps the BOW shape feature to a linear feature space. Nonetheless, the transformed linear feature space cannot handle the large non-rigid deformations of 3D shapes. In the proposed MMLM method, we employ the deep neural network to non-linearly map the 3D shape features to the non-linear feature space. Different from the learned linear distance metric, with the developed multi-metric network, we can learn multiple non-linear distance metrics from multiple types of shape features. Moreover, the learned multiple non-linear distance metrics are adaptively fused for retrieval.

TABLE IV
RETRIEVAL RESULTS ON THE SHREC'10 SHAPEGOOGLE DATASET.

Transformation	VQ [10]	UDL [11]	SDL [11]	DA [12]	MMLM
Isometry	0.988	0.977	0.994	0.998	**1.000**
Topology	**1.000**	**1.000**	**1.000**	0.996	**1.000**
Isometry+Topology	0.933	0.934	0.956	0.982	**0.988**
Partiality	0.947	0.948	0.951	0.973	**0.985**
Triangulation	0.954	0.950	0.955	0.955	**0.963**

4) SHREC'14 Human dataset: We evaluate our proposed MMLM method on the synthetic sub-dataset and the scanned sub-dataset. For the synthetic sub-dataset, 10 shapes per class are used to train and the other shapes per class are used for testing. For the scanned sub-dataset, 5 shapes per class are used to train and the rest of shapes are used to test. The experiments are repeated over 20 times. The mean average precision is used as the evaluation criterion. The recent shape retrieval methods on the two sub-datasets are involved to compare: reduced Bi-harmonic distance matrix (RBiHDM) [51], intrinsic pyramid matching (ISPM) [52], Histogram of area projection transform (HAPT) [53], deep belief network (DBN) [45], the standard vector quantization method (VQ) [10], the unsupervised dictionary learning method (UDL) [11] and the supervised dictionary learning method (SDL) [11]. The experimental results are listed in Table V. As can be seen in this table, for the synthetic sub-dataset, compared to the state-of-the-art methods [10, 11, 45, 51–53], our proposed MMLM method can obtain the best performance. Compared to the synthetic sub-dataset, the scanned human shape sub-dataset consists of more human models and is more challenging. On this sub-dataset, the retrieval performance of our proposed MMLM is slightly higher than that of the SDL method and is far more superior to the other methods [10, 45, 51–53].

TABLE V
RETRIEVAL RESULTS ON THE SHREC'14 HUMAN DATASET.

Method	Synthetic model	Scanned model
HAPT [53]	0.817	0.637
ISPM [52]	0.92	0.258
RBiHDM [51]	0.642	0.640
DBN [45]	0.842	0.304
VQ [10]	0.813	0.514
UDL [11]	0.842	0.523
SDL [11]	0.951	0.791
MMLM	**0.983**	**0.815**

D. Discussion

In this subsection, we perform the sensitivity analysis of our proposed MMLM method with respect to parameter λ in Eq. (5). Parameter λ controls the balance between discrimination and complementarity of the learned shape features. We conduct experiments on the McGill shape dataset in the cases of different λ. For each class, 10 shapes are chosen as the training samples and the remaining shapes are used as the testing samples. The MAP is used to evaluate the proposed MMLM method. The MAPs of the proposed MMLM method in the cases of different λ are shown in Fig. 7, where parameter λ is empirically selected from 0.01 to 0.13 with step 0.005. From this figure, one can see that λ ranging from 0.06 to 0.09 has few effects on the final retrieval performance. Nonetheless, if λ is too small or large, discrimination and complementarity of the outputs of multiple metric networks cannot be kept simultaneously. Thus, the learned multiple shape features cannot represent 3D shapes well. Therefore, in our comparison evaluation, λ is set to 0.06 to train our proposed deep multi-metric learning model.

V. CONCLUSIONS

In this paper, based on multiple types of shape features, we proposed a deep multi-metric learning method for 3D shape retrieval. We exploited the non-linearity of the deep neural network and the complementarity of multiple shape features to develop a multi-metric network. With the developed multi-metric network, multiple non-linear distance metrics are learned so that for each type of shape feature the variations of outputs from the same class are minimized and the variations of outputs from different classes are maximized while dependence of multiple shape features is minimized. The

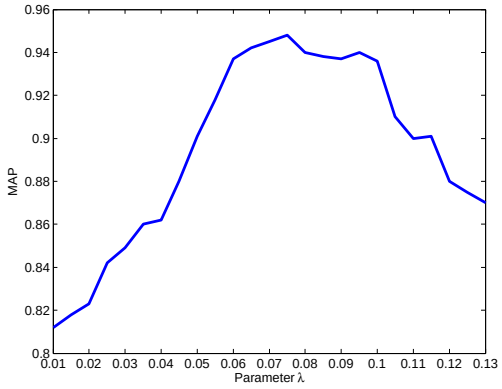


Fig. 7. The MAPs of the proposed MMLM method in the cases of different λ on the McGill shape dataset.

fused distance metric with the learned weights is used as the similarity for shape retrieval. Experiments on the PSB, McGill, SHREC'10 ShapeGoogle and SHREC'14 Human datasets demonstrate that the proposed method can yield good retrieval performance.

REFERENCES

[1] T. F. Ansary, M. Daoudi, and J. Vandeborre, "A Bayesian 3-D search engine using adaptive views clustering," *IEEE Trans. Multimedia*, vol. 9, no. 1, pp. 78–88, 2007.

[2] J. Chen, C. Lin, P. Hsu, and C. Chen, "Point cloud encoding for 3D building model retrieval," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 337–345, 2014.

[3] H. Wong, B. Ma, Z. Yu, P. F. Yeung, and H. H. Ip, "3-D head model retrieval using a single face view query," *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 1026–1036, 2007.

[4] https://3dwarehouse.sketchup.com/.

[5] D. Saupe and D. V. Vranic, "3D model retrieval with spherical harmonics and moments," *DAGM Symposium on Pattern Recognition*, pp. 392–397, 2001.

[6] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Computer Graphics Forum*, vol. 22, pp. 223–232, 2003.

[7] J. Assfalg, M. Bertini, A. D. Bimbo, and P. Pala, "Content-based retrieval of 3D objects using spin image signatures," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 589–599, 2007.

[8] T. Furuya and R. Ohbuchi, "Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features," in *ACM International Conference on Image and Video Retrieval, Santorini Island, Greece*, 2009.

[9] X. Bai, S. Bai, Z. Zhu, and L. J. Latecki, "3D shape matching via two layer coding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2361–2373, 2015.

[10] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov, "Shape google: Geometric words and expressions for invariant shape retrieval," *ACM Transactions on Graphics*, vol. 30, p. 1, 2011.

[11] R. Litman, A. M. Bronstein, M. M. Bronstein, and U. Castellani, "Supervised learning of bag-of-features shape descriptors using sparse coding," *Computer Graphics Forum*, vol. 33, no. 5, pp. 127–136, 2014.

[12] J. Xie, Y. Fang, F. Zhu, and E. Wong, "Deepshape: Deep learned shape descriptor for 3D shape matching and retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition, Boston, USA*, 2015.

[13] B. Li and H. Johan, "3D model retrieval using hybrid features and class information," *Multimedia Tools Application.*, vol. 62, no. 3, pp. 821–846, 2013.

[14] Q. Chen, B. Fang, Y. Yu, and Y. Tang, "3D CAD model retrieval based on the combination of features," *Multimedia Tools Application.*, vol. 74, no. 13, pp. 4907–4925, 2015.

[15] R. Ohbuchi and T. Furuya, "Distance metric learning and feature combination for shape-based 3D model retrieval," in *Proceedings of the ACM Workshop on 3D Object Retrieval, Firenze, Italy*, 2010, pp. 63–68.

[16] A. Gretton, O. Bousquet, A. J. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *Algorithmic Learning Theory, 16th International Conference, Singapore*, 2005, pp. 63–77.

[17] Y. Gao, Y. Yang, Q. Dai, and N. Zhang, "3D object retrieval with bag-of-region-words," in *ACM Conference on Multimedia, Firenze, Italy*, 2010, pp. 955–958.

[18] X. Bai, C. Rao, and X. Wang, "Shape vocabulary: A robust and efficient shape representation for shape matching," *IEEE Trans. Image Processing*, vol. 23, no. 9, pp. 3935–3949, 2014.

[19] Y. Gao, M. Wang, Z. Zha, Q. Tian, Q. Dai, and N. Zhang, "Less is more: Efficient 3-D object retrieval with query view selection," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 1007–1018, 2011.

[20] Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai, "Deep learning representation using autoencoder for 3D shape retrieval," in *International Conference on Security, Pattern Analysis, and Cybernetics, China*, 2014, pp. 279–284.

[21] S. Bai, X. Bai, W. Liu, and F. Roli, "Neural shape codes for 3D model retrieval," *Pattern Recognition Letters*, vol. 65, pp. 15–21, 2015.

[22] B. Shi, S. Bai, Z. Zhou, and X. Bai, "Deeppano: Deep panoramic representation for 3D shape recognition," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015.

[23] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *International Conference on Computer Vision, Santiago, Chile*, 2015, pp. 945–953.

[24] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. Jan Latecki, "Gift: A real-time and scalable 3D shape search engine," in *International Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, USA*, June 2016.

[25] B. Leng, Y. Liu, K. Yu, X. Zhang, and Z. Xiong, "3D object understanding with 3D convolutional neural networks," *Information Sciences*, 2015.

[26] B. Leng, X. Zhang, M. Yao, and Z. Xiong, "A 3D model recognition mechanism based on deep boltzmann machines," *Neurocomputing*, vol. 151, pp. 593–602, 2015.

[27] B. Leng, S. Guo, X. Zhang, and Z. Xiong, "3D object retrieval with stacked local convolutional autoencoder," *Signal Processing*, vol. 112, pp. 119–128, 2015.

[28] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Proceedings of the Symposium on Geometry Processing*, pp. 1383–1392, 2009.

[29] M. M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA*, 2010, pp. 1704–1711.

[30] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *IEEE International Conference on Computer Vision Workshops, Barcelona, Spain*, 2011, pp. 1626–1633.

[31] H. Tabia, H. Laga, D. Picard, and P. H. Gosselin, "Covariance descriptors for 3D shape matching and retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA*, 2014, pp. 4185–4192.

[32] Z. Lian, J. Zhang, S. Choi, H. ElNaghy, J. El-Sana, T. Furuya, A. Giachetti, R. A. Güler, L. Lai, C. Li, H. Li, F. A. Limberger, R. R. Martin, R. U. Nakanishi, A. Neto, L. G. Nonato, R. Ohbuchi, K. Pevzner, D. Pickup, P. L. Rosin, A. Sharf, L. Sun, X. Sun, S. Tari, G. B. Ünal, and R. C. Wilson, "SHREC'15 track: Non-rigid 3D shape retrieval," in *Eurographics Workshop on 3D Object Retrieval, Zurich, Switzerland*, 2015, pp. 107–120.

[33] S. Bu, Z. Liu, J. Han, J. Wu, and R. Ji, "Learning high-level feature by deep belief networks for 3-D model retrieval and recognition," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2154–2167, 2014.

[34] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapenets: A deep representation for volumetric shapes," in *IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA*, 2015, pp. 1912–1920.

[35] Z. Han, Z. Liu, J. Han, C. Vong, S. Bu, and X. Li, "Unsupervised 3D local feature learning by circle convolutional restricted boltzmann machine," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5331–5344, 2016.

[36] Z. Han, Z. Liu, J. Han, C. Vong, S. Bu, and C. L. P. Chen, "Mesh convolutional restricted boltzmann machines for unsupervised learning of features with structure preservation on 3-D meshes," *IEEE Transactions on Neural Network and Learning Systems*, vol. 99, no. 11, pp. 1–14, 2016.

[37] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, Q. Chen, N. K. Chowdhury, B. Fang, T. Furuya, H. Johan, R. Kosaka, H. Koyanagi, R. Ohbuchi, and A. Tatsuma, "SHREC'14 track: Large scale comprehensive retrieval track benchmark," in *Eurographics Workshop on 3D Object Retrieval, Strasbourg, France*, 2014.

[38] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA*, 2010, pp. 3360–3367.

[39] T. Darom and Y. Keller, "Scale-invariant features for 3D mesh models," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2758–2769, 2012.

[40] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504 – 507, 2006.

[41] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[42] J. Nocedal and S. J. Wright, *Numerical Optimization, second edition*. World Scientific, 2006.

[43] P. Shilane, P. Min, M. M. Kazhdan, and T. A. Funkhouser, "The princeton shape benchmark," in *International Conference on Shape Modeling and Applications, Genova, Italy*, 2004, pp. 167–178.

[44] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. J. Dickinson, "Retrieving articulated 3D models using medial surfaces," *Machine Vision Application*, vol. 19, no. 4, pp. 261–275, 2008.

[45] D. Pickup, X. Sun, P. L. Rosin, R. R. Martin, Z. Cheng, Z. Lian, M. Aono, A. Ben Hamza, A. Bronstein, M. Bronstein, S. Bu, U. Castellani, S. Cheng, V. Garro, A. Giachetti, A. Godil, J. Han, H. Johan, L. Lai, B. Li, C. Li, H. Li, R. Litman, X. Liu, Z. Liu, Y. Lu, A. Tatsuma, and J. Ye, "SHREC'14 track: Shape retrieval of non-rigid 3D human models," in *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval*, 2014.

[46] G. Lavoué, "Combination of bag-of-words descriptors for robust partial shape retrieval," *The Visual Computer*, vol. 28, no. 9, pp. 931–942, 2012.

[47] P. Daras and A. Axenopoulos, "A 3D shape retrieval framework supporting multimodal queries," *International Journal of Computer Vision*, vol. 89, no. 2-3, pp. 229–247, 2010.

[48] A. Agathos, I. Pratikakis, P. Papadakis, S. J. Perantonis, P. N. Azariadis, and N. S. Sapidis, "Retrieval of 3D articulated objects using a graph-based representation," in *Eurographics workshop on 3D object retrieval, Munich, Germany*, 2009, pp. 29–36.

[49] H. Tabia, D. Picard, H. Laga, and P. H. Gosselin, "Compact vectors of locally aggregated tensors for 3D shape retrieval," in *Eurographics Workshop on 3D Object Retrieval, Girona, Spain*, 2013, pp. 17–24.

[50] P. Papadakis, I. Pratikakis, T. Theoharis, G. Passalis, and S. J. Perantonis, "3D object retrieval using an efficient and compact hybrid shape descriptor," in *Eurographics workshop on 3D object retrieval, Crete, Greece*, 2008, pp. 9–16.

[51] J. Ye, Z. Yan, and Y. Yu, "Fast nonrigid 3D retrieval using modal space transform," in *International Conference on Multimedia Retrieval, Dallas, TX, USA*, 2013, pp. 121–126.

[52] C. Li and A. B. Hamza, "A multiresolution descriptor for deformable 3D shape retrieval," *The Visual Computer*, vol. 29, no. 6-8, pp. 513–524, 2013.

[53] A. Giachetti and C. Lovato, "Radial symmetry detection and shape characterization with the multiscale area pro-

jection transform," *Computer Graphics Forum*, vol. 31, no. 5, pp. 1669–1678, 2012.

Jin Xie received his Ph.D. degree from the Department of Computing, The Hong Kong Polytechnic University. He is a research scientist at New York University Abu Dhabi and New York University Tandon School of Engineering. His research interests include computer vision and machine learning. Currently he is focusing on 3D computer vision with convex optimization and deep learning methods.

Guoxian Dai received his master degree from Fudan University, China. He is a Ph.D. candidate in the Department of Computer Science and Engineering at the New York University Tandon School of Engineering. His current research interests focus on 3D shape analysis such as 3D shape retrieval and cross-domain 3D model retrieval.

Yi Fang received his Ph.D. degree from Purdue University with research focus on computer graphics and vision. Upon one year industry experience as a research intern in Siemens in Princeton, New Jersey and a senior research scientist in Riverain Technologies in Dayton, Ohio, and a half-year academic experience as a senior staff scientist at Department of Electrical Engineering and Computer science, Vanderbilt University, Nashville, he joined New York University Abu Dhabi as an Assistant Professor of Electrical and Computer Engineering. He is currently working on the development of state-of-the-art techniques in large-scale visual computing, deep visual learning, deep cross-domain and cross-modality model, and their applications in engineering, social science, medicine and biology.