Deep Non-linear Metric Learning for 3D Shape Retrieval

Jin Xie, Guoxian Dai, Fan Zhu, Ling Shao, and Yi Fang

***Abstract*—Effective 3D shape retrieval is an important problem in 3D shape analysis. Recently, feature learning based shape retrieval methods have been widely studied, where the distance metrics between 3D shape descriptors are usually hand-crafted. In this paper, motivated by the fact that deep neural network has the good ability to model non-linearity, we propose to learn an effective non-linear distance metric between 3D shape descriptors for retrieval. First, the locality-constrained linear coding method is employed to encode each vertex on the shape and the encoding coefficient histogram is formed as the global 3D shape descriptor to represent the shape. Then, a novel deep metric network is proposed to learn a non-linear transformation to map the 3D shape descriptors to a non-linear feature space. The proposed deep metric network minimizes a discriminative loss function that can enforce the similarity between a pair of samples from the same class to be small and the similarity between a pair of samples from different classes to be large. Finally, the distance between the outputs of the metric network is used as the similarity for shape retrieval. The proposed method is evaluated on the McGill, SHREC'10 ShapeGoogle and SHREC'14 Human shape datasets. Experimental results on the three datasets validate the effectiveness of the proposed method.**

***Index Terms*—3D shape retrieval, 3D shape descriptor, deep metric learning, neural network, heat kernel signature.**

I. INTRODUCTION

SINCE 3D models have been widely applied to industrial design, architectural design and entertainment, etc, modeling, visualizing and analyzing 3D models [1–6] have been receiving more and more attention. Particularly, with the increasing growth of 3D models, content based 3D shape retrieval became an important research topic in the community of computer vision and computer graphics. The objective of 3D shape retrieval is to search 3D shapes similar to a query shape using shape properties from a large collection of 3D shapes. Shape feature extraction and matching are key steps for the content based 3D shape retrieval. Shape feature extraction should capture the distinctive properties of shapes and discriminatively represent shapes. Shape matching is the process of determining how similar two shapes are by calculating the distance metric between the shape features. It is desirable that the within-class distance for shapes is as small as possible while the between-class distance for shapes is as large as possible.

Extensive research efforts have been dedicated to 3D shape retrieval in the past decades. Since a 3D model can be represented as a group of 2D images at different viewpoints, plenty of view based 3D shape retrieval methods have been proposed. The key step in view based 3D shape retrieval is how to perform multiple view matching. Chen *et al*. [7] proposed the light field descriptor (LFD), where the descriptor is computed from a set of contours obtained from the vertices of a dodecahedron. Based on the projected 2D images, Zernike moments and Fourier transform are employed to extract the features of the images. The best matching between two LFDs is used as the similarity between 3D shapes for retrieval. Ansary *et al*. [8] proposed an adaptive view clustering (AVC) method. The representative views are optimally selected with the Bayesian information criteria. A probabilistic method is then employed to calculate the similarity between two 3D shapes for retrieval. In addition, the Bag-of-Words (BOW) methods are also applied on the projected images to extract features for retrieval. In [9], each 3D shape is rendered to a group of depth images. Then the SIFT features are extracted from these depth images and the BOW features are learned from a set of SIFT descriptors to represent 3D shapes. Recently, Bai *et al*. [10] proposed the two layer coding framework to encode the depth images to form the 3D shape descriptor for retrieval.

Apart from the view based shape retrieval methods, the local shape descriptor based retrieval methods mainly focus on learning a global representation from a set of local shape descriptors. These local shape descriptors include global point signature (GPS) [11], heat kernel signature (HKS)[12], scale invariant heat kernel signature (SI-HKS) [13], wave kernel signature (WKS) [14], 3D SIFT [15], 3D shape context [16] and mesh HOG [17], etc. In [18], Bronstein *et al*. proposed to learn the spatially sensitive BOW feature (called shapegoogle descriptor) from a set of HKSs for shape retrieval. Tabia *et al*. [19] extracted the covariances of the patches on the meshed surface and learned a dictionary of words from the Riemannian manifold of the symmetric positive definite matrices. Then the histogram of the words is used as the global descriptor for retrieval. By employing sparse coding, Litman *et al*. [20] constructed a bi-level supervised dictionary to learn encoded representation coefficients from the local shape descriptors HKSs/SI-HKSs for retrieval. EINagh *et al*. [21] proposed the compact HKS-based BOW descriptor for 3D shape retrieval. By selecting the critical points on the shape and the scales of the HKSs of the selected points, a compact HKS-based feature representation can be formed to describe the shape. A global shape descriptor is then learned by applying the BOW method to the compact HKS-based feature vectors. Limberger *et al*. [21] employed the Fisher vector encoding framework to develop a shape descriptor for retrieval, where a Gaussian mixture model is used to fit the distribution of the point signatures (e.g., HKS and WKS) on the shape. The shape

Jin Xie, Guoxian Dai, Fan Zhu and Yi Fang are with NYU Multimedia and Visual Computing Lab, the Department of Electrical and Computer Engineering, New York University Abu Dhabi, UAE and the Department of Electrical and Computer Engineering, New York University Tandon School of Engineering, USA. Ling Shao is with the Department of Computer Science and Digital Technologies, Northumbria University, U.K. (e-mail:{jin.xie, guoxian.dai, fan.zhu, yfang}@nyu.edu. {ling.shao}@northumbria.ac.uk).

descriptor, i.e., the Fisher vector, is formed by stacking the mean and covariance deviation vectors for each of modes in the Gaussian mixture model. Xie *et al*. [22] first developed a multiscale shape distribution to represent shapes. Then a deep discriminative auto-encoder is proposed by imposing the Fisher discrimination criterion on the neurons in the hidden layer and the neurons from multiple discriminative auto-encoders are concatenated to form a global shape descriptor for retrieval.

Nonetheless, these 3D shape retrieval methods mainly focus on shape feature extraction. Once the shape feature is extracted, the hand-crafted distance metric such as the Euclidean distance and the Hausdorff distance, is used as the similarity for retrieval. In [23], the manifold ranking based unsupervised metric learning method is used to compute the distance for retrieval, where the high dimensional shape feature space is represented by a Laplacian graph. In [18], the similarity sensitive hashing (SSH) is employed to learn the distance for shape retrieval. The SSH based metric learning method aims to learn a linear transformation to map the BOW features to the linear feature space. However, due to large deformations of 3D shapes, these methods cannot characterize the manifold of 3D shapes well.

Motivated by the favorable ability of deep neural network to model non-linearity of samples, in this paper, we propose a novel deep non-linear metric learning method for 3D shape retrieval. 3D shapes are usually with large deformations, which results in the large intra-class variations of the shapes. Therefore, it is desirable to develop a non-linear metric to measure the similarity between 3D shapes so that the intra-class variations can be reduced. First, we employ the locality-constrained linear coding (LLC) method [24] to encode each vertex of 3D shapes to form a global 3D shape descriptor. We then develop a deep metric network to learn a non-linear transformation to map the global 3D descriptors to a non-linear feature space. The learned distance metric can minimize a discriminative loss function so that the similarities between the pairs of samples from the same class are as small as possible and the similarities between the pairs of samples from different classes are as large as possible. Furthermore, in order to make the learned distance metric to be more discriminative, we also encourage that the neurons in the hidden layers of the metric network are as close as possible to their means. Experimental results on the three 3D shape datasets demonstrate that the effectiveness of the proposed method for 3D shape retrieval.

The main contribution of this paper is that we develop a novel deep metric network to learn a non-linear distance metric for 3D shape retrieval. Although the linear metric was learned for shape retrieval in [18], little attempt has been made on the deep neural network to learn a non-linear metric for 3D shape retrieval. We exploit the discriminative and non-linear information of the constructed deep metric network to map the global 3D shape descriptor to a non-linear feature space. Compared to the learned linear metric, the learned deep non-linear metric can measure the distances between global 3D shape descriptors for retrieval better.

The rest of the paper is organized as follows. In Section II, we briefly introduce the scale invariant heat kernel signature. In Section III, we present the proposed deep non-linear metric learning method for 3D shape retrieval. Section IV performs extensive experiments and Section V concludes the paper.

II. Background

A. Scale Invariant Heat Kernel Signature

Given an initial Dirac delta distribution defined on the meshed surface at time $t = 0$, the heat diffusion process on the meshed surface X can be described with the following heat equation:

$$\frac{\partial h(x,y,t)}{\partial t} = -Ph(x,y,t) \tag{1}$$

where $h(x,y,t)$ denotes the heat kernel on vertices x and y at diffusion time t, P is the Laplace-Beltrami operator. With the spectral decomposition theorem, the solution of Eq. (1), i.e., heat kernel $h(x,y,t)$, can be obtained with the eigenfunctions and eigenvectors of the Laplace-Beltrami operator P:

$$h(x,y,t) = \sum_i e^{-\lambda_i t}\phi_i(x)\phi_i(y) \tag{2}$$

where λ_i and ϕ_i are the ith eigenvalue and eigenfunction of the Laplace-Beltrami operator P, respectively.

Based on the heat kernel, heat kernel signature (HKS) [12] $h(x,t)$, is defined as the diagonal value of the heat kernel of vertex x at time t:

$$h(x,t) = \sum_i e^{-\lambda_i t}\phi_i(x)^2. \tag{3}$$

HKS $h(x,t)$ can be viewed as the remaining heat at vertex x after time interval t. As a point signature, HKS can encode the geometric information of the neighborhood of the shape.

The scale invariant heat kernel signature (SI-HKS) [13] is the scale invariant version of HKS. By sampling the HKS logarithmically with $t = \beta^\tau$, the discrete HKS $h(x,t)$ can be defined:

$$h_\tau = h(x,\beta^\tau). \tag{4}$$

The HKS of the scaled shape with the factor $\beta^{s/2}$, $h_\tau^{'}$, can be represented as:

$$h_\tau^{'} = \beta^2 h_{\tau+s}. \tag{5}$$

The scale factor of the shape can be removed by the derivative $\dot{h_\tau}$ of h_τ:

$$\dot{h}_\tau = log(h_{\tau+1}) - log(h_\tau). \tag{6}$$

Thus, we can obtain:

$$\dot{h}_\tau^{'} = \dot{h}_{\tau+s}. \tag{7}$$

Denote the Fourier transform of $\dot{h}_\tau$ by $H(\omega)$. By taking the Fourier transforms of $\dot{h}_\tau^{'}$ and $\dot{h}_{\tau+s}$, one can see that the absolute values of their Fourier transforms are equivalent. Thus, SI-HKS $\boldsymbol{g}(x)$ on vertex x can be constructed by taking the absolute value of $H(\omega)$ and sampling it at m frequencies:

$$\boldsymbol{g}(x) = (|H(\omega_1)|, \cdots, |H(\omega_m)|). \tag{8}$$

III. Proposed Approach

In this section, we detail the proposed deep non-linear metric learning based shape retrieval method. Fig. 1 shows the proposed shape retrieval framework with the deep metric network. We first employ the LLC method to obtain the encoding coefficient histograms to represent shapes. With the encoding coefficient histograms as input to the developed deep metric network, we then train the deep neural network to learn a non-linear distance metric as the similarity for retrieval.

A. Shape Feature Description

Before we present the deep non-linear metric learning method for 3D shape retrieval, in this subsection, we first extract global 3D shape descriptors to describe 3D shapes. We employ the SI-HKS as a local shape descriptor to describe the neighborhood of each vertex on the shape. Based on the SI-HKSs extracted from the shape, we employ LLC to encode the vertices to represent the shape. Compared to the vector quantization based shapegoogle descriptor [18] and the sparse coding based descriptor [20], LLC has the low computation complexity and low reconstruction error.

Suppose that there are N shapes and we use S_i to index the ith sample, $i = 1, 2, \cdots, N$. For each vertex of the shape, we extract an m-dimensional SI-HKS feature. Thus, shape S_i can be represented by the SI-HKS feature matrix $\boldsymbol{y}_i = [\boldsymbol{g}_{i,1}, \boldsymbol{g}_{i,2}, \cdots, \boldsymbol{g}_{i,T}]$, where $\boldsymbol{g}_{i,j}$ is the m-dimensional SI-HKS feature, $j = 1, 2, \cdots, T$, and T is the number of the vertices on shape S_i. We then can construct a training dataset $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N]$, where $\boldsymbol{y}_i$ is the SI-HKS feature matrix, $i = 1, 2, \cdots, N$. The dictionary $\boldsymbol{D} \in R^{m \times L}$ is to be learned from the training dataset $\boldsymbol{Y}$ via *K*-means clustering, where L is the number of the atoms in the dictionary. The LLC method [24] solves the following problem to encode each SI-HKS feature over $\boldsymbol{D}$:

$$\begin{aligned} & min_{\boldsymbol{u}} \sum_{i=1}^{N} \sum_{j=1}^{T} \|\boldsymbol{g}_{i,j} - \boldsymbol{D}\boldsymbol{u}_{i,j}\|_2^2 + \lambda \|\boldsymbol{d}_{i,j} \bullet \boldsymbol{u}_{i,j}\|_2^2 \\ & s.t. \quad \mathbf{1}^T \boldsymbol{u}_{i,j} = 1, \forall i, j \end{aligned} \tag{9}$$

where $\boldsymbol{u}_{i,j}$ is the encoding coefficient of the local shape descriptor $\boldsymbol{g}_{i,j}$, $\boldsymbol{u} = [\boldsymbol{u}_{1,1}, \boldsymbol{u}_{1,2}, \cdots, \boldsymbol{u}_{N,T}]$ is the encoding coefficient matrix of $\boldsymbol{Y}$ over dictionary $\boldsymbol{D}$, $\boldsymbol{d}_{i,j}$ is the distance vector of $\boldsymbol{g}_{i,j}$ and dictionary $\boldsymbol{D}$, λ is the regularization parameter, and $\bullet$ denotes the element-wise multiplication. By performing the max pooling operator on the encoding coefficient $\boldsymbol{u}_{i,j}$ of shape S_i, $i = 1, 2, \cdots, N$, $j = 1, 2, \cdots, T$, we can obtain an encoding coefficient histogram $\boldsymbol{x}_i$ as a global shape descriptor to describe shape S_i. Fig. 2 shows the global shape descriptor of the hand model with the LLC method.

B. Deep Metric Learning for Shape Retrieval

The traditional Mahalanobis distance metric learning method [25] learns a matrix $\boldsymbol{M} \in \mathbb{R}^{d \times d}$, where the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be computed as

$$d_M(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{M} (\boldsymbol{x}_i - \boldsymbol{x}_j)}. \tag{10}$$

Since $\boldsymbol{M}$ is a positive semi-definite matrix, it can be decomposed as

$$\boldsymbol{M} = \boldsymbol{\psi}^T \boldsymbol{\psi}. \tag{11}$$

Thus, $d_M(\boldsymbol{x}_i, \boldsymbol{x}_j)$ can be rewritten as

$$d_M(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{\psi}\boldsymbol{x}_i - \boldsymbol{\psi}\boldsymbol{x}_j\|_2. \tag{12}$$

From Eq. (12), one can see that the Mahalanobis distance metric learning method can learn a linear transformation $\boldsymbol{\psi}$ to map the samples to a linear feature space. However, the learned linear transformation $\boldsymbol{\psi}$ cannot characterize the non-linear manifold where the samples lie well, particularly when there are usually large intra-class variations and small inter-class variations with the samples. Recent advances [26–30] on the non-linear metric learning demonstrate that learning a non-linear distance metric can obtain better performance. For example, in [27], the authors proposed to learn the non-linear χ^2 histogram distance for the histogram data instead of the Mahalanobis distance.

Since 3D shapes are usually with the complex geometric structural variations such as non-isometric transformations and deformations, it is desirable to develop a non-linear distance metric to characterize the manifold of 3D shapes well. Based on the non-linearity of deep neural network, in this subsection, we propose a deep non-linear metric learning method to seek a non-linear transformation to map the global 3D shape descriptor to a non-linear feature space for retrieval. We construct a deep neural network to map the input encoding coefficient histogram $\boldsymbol{x}_i \in R^{m \times 1}$ to the output $\boldsymbol{z}_i^K \in R^{r \times 1}$, where m and r are the dimensions of the input and output of the deep neural network, K is the number of the layers, $i = 1, 2, \cdots, N$. In the constructed neural network, one neuron in the layer k is connected to all neurons in the layer $k+1$. We denote the weight and bias connecting the layer k and the layer $k+1$ by $\boldsymbol{W}^k$ and $\boldsymbol{b}^k$, respectively. The output of the layer $k+1$, $\boldsymbol{z}_i^{k+1}$, is:

$$\boldsymbol{z}_i^{k+1} = f_{k+1}(\boldsymbol{z}_i^k) = \sigma(\boldsymbol{W}^k \boldsymbol{z}_i^k + \boldsymbol{b}^k) \tag{13}$$

where $f_{k+1}(\boldsymbol{z}_i^k)$ is the activation function in the layer $k+1$, $\boldsymbol{z}_i^k$ is the neuron in the layer k for the input sample $\boldsymbol{x}_i$, $\sigma(\boldsymbol{x})$ is the sigmoid function $\sigma(\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{x}}}$. Thus, the non-linear mapping function $F_K(\boldsymbol{x}_i)$ across K layers can be represented as:

$$F_K(\boldsymbol{x}_i) = f_K(f_{K-1}(\cdots, f_2(\boldsymbol{x}_i))). \tag{14}$$

It is noted that for the input layer we assume that $\boldsymbol{z}_i^1 = \boldsymbol{x}_i$. The weights and biases of all layers in the neural network are $\boldsymbol{W} = \{\boldsymbol{W}^1, \boldsymbol{W}^2, \cdots, \boldsymbol{W}^{K-1}\}$ and $\boldsymbol{b} = \{\boldsymbol{b}^1, \boldsymbol{b}^2, \cdots, \boldsymbol{b}^{K-1}\}$, respectively. Fig. 3 shows an example of the constructed metric network.

For each pair of samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, with the deep neural network of K layers, the distance metric between the samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be measured by the Euclidean distance between the outputs $\boldsymbol{z}_i^K$ and $\boldsymbol{z}_j^K$:

$$\|\boldsymbol{z}_i^K - \boldsymbol{z}_j^K\|_2 = \|F_K(\boldsymbol{x}_i) - F_K(\boldsymbol{x}_j)\|_2. \tag{15}$$

From Eq. (15), one can see that with the non-linear mapping F_K the distance between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$,

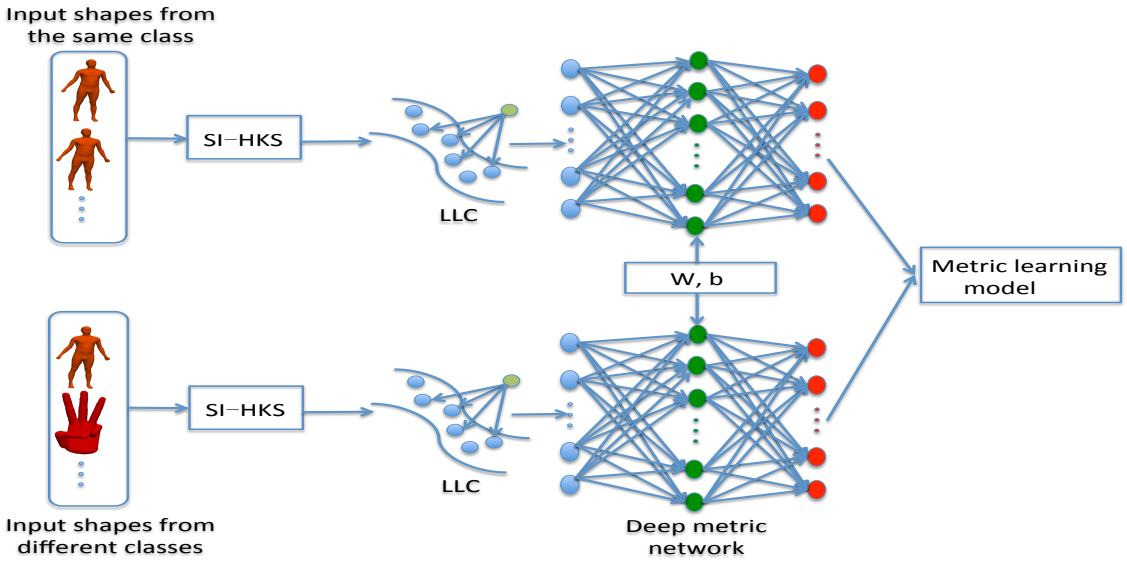


Fig. 1. The proposed deep shape metric learning framework. For the input shapes, we employ the LLC method to encode the extracted SI-HKSs to form the global 3D shape descriptors. The global shape descriptors of the input shapes from the same class and different classes are then fed into the deep metric learning model so that the similarity between the pairs of shapes from the same class are as small as possible and the similarity between the pairs of shapes from different classes are as large as possible.

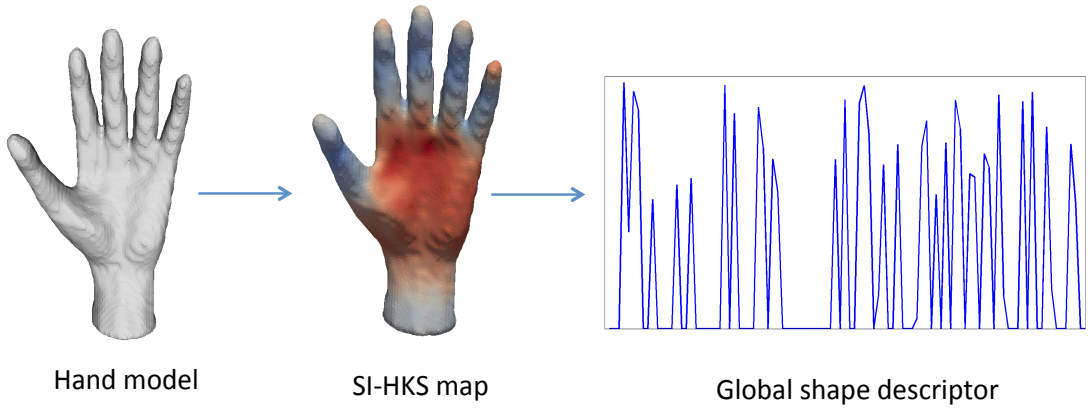


Fig. 2. The global shape descriptor of the hand model. Different colors in the SI-HKS map of the hand model represent different SI-HKS values with the same frequency. The global shape descriptor is obtained by performing the max-pooling operation on the LLC coefficients of the hand model.

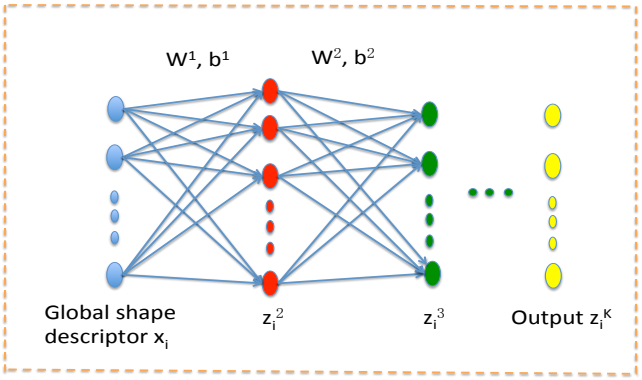


Fig. 3. The constructed metric network in our method. The input to the network is the global 3D shape descriptor $\boldsymbol{x}_i$, $\boldsymbol{z}_i^2$ and $\boldsymbol{z}_i^3$ are the outputs of the hidden layers, respectively, and the output of the network is $\boldsymbol{z}_i^K$. $\boldsymbol{W}$ and $\boldsymbol{b}$ are the parameters to be learned in the network.

can be transformed to $\|F_K(\boldsymbol{x}_i) - F_K(\boldsymbol{x}_j)\|_2$. Compared to $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$ in the original space, the non-linear distance metric $\|F_K(\boldsymbol{x}_i) - F_K(\boldsymbol{x}_j)\|_2$ can characterize the manifold where samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ lie better.

For each output $\boldsymbol{z}_i^K$, our goal is to enforce that the similarity between a pair of samples from the same class, $\|\boldsymbol{z}_i^K - \boldsymbol{z}_j^K\|_2$, $j \in c(i)$, where $c(i)$ denotes the class label of $\boldsymbol{z}_i^K$, is as small as possible while the similarity between a pairs of samples from different classes, $\|\boldsymbol{z}_i^K - \boldsymbol{z}_j^K\|_2$, $j \notin c(i)$, is as large as possible. Let $d_{i,j}^+$ be the loss between a pair of samples from the same class and $d_{i,j}^-$ be the loss between a pair of samples from different classes:

$$\begin{aligned} d_{i,j}^+ &= \|\boldsymbol{z}_i^K - \boldsymbol{z}_j^K\|_2^2 \\ d_{i,j}^- &= max(0, \eta - \|\boldsymbol{z}_i^K - \boldsymbol{z}_j^K\|_2^2) \end{aligned} \tag{16}$$

where η is a constant. The term $d_{i,j}^-$ is a hinge loss function to penalize the similarity between a pair of samples from

different classes that is less than the threshold η. In this work, we propose the following discriminative loss function:

$$J(\boldsymbol{W},\boldsymbol{b}) = \frac{\alpha}{\sum n_i}\sum_{i=1}^{N}\sum_{j\in c(i)}\frac{1}{2}d_{i,j}^{+} + \frac{1-\alpha}{\sum m_i}\sum_{i=1}^{N}\sum_{j\notin c(i)}\frac{1}{2}d_{i,j}^{-} + \frac{\lambda}{N}\sum_{p=2}^{K-1}\sum_{i=1}^{N}\frac{1}{2}\|\boldsymbol{z}_i^p-\boldsymbol{\mu}_i^p\|_2^2 + \frac{1}{2}\gamma\|\boldsymbol{W}\|_F^2 \quad (17)$$

where $0\le\alpha\le 1$ controls the tradeoff between the similarities of the pairs of the training samples from the same class and the similarities of the pairs of the training samples from different classes, n_i is the number of the outputs of the same class label to $\boldsymbol{z}_i^K$, m_i is the number of the outputs of the different class labels to $\boldsymbol{z}_i^K$, $\boldsymbol{\mu}_i^p$ is the mean of the outputs in layer p from the same class $c(i)$, i.e., $\boldsymbol{\mu}_i^p = \frac{\sum_{j\in c(i)}\boldsymbol{z}_j^p}{s_i}$, s_i is the number of the samples associated with $c(i)$, $p=2,3,\cdots,K-1$, parameters λ and γ are the positive scalars.

In the proposed metric learning model Eq. (17), the first two terms minimize the within-class similarities for the outputs of the deep neural network and simultaneously maximize the between-class similarities so that the within-class variations of the outputs from the same class are as small as possible and the between-class variations of the outputs from the different classes are as large as possible. In Eq. (17), the third term encourages the neurons in the hidden layers from the same class to their means. Since the training samples of the same class share similarities in the original feature space, the transformed features, i.e., the neurons $\boldsymbol{z}_i^k$ in the hidden layers, should also be similar. Moreover, by enforcing the neurons $\boldsymbol{z}_i^k$ in the hidden layers to approach to their means, we can furthermore reduce the within-class variations of the outputs. This can boost the discriminative power of the distance metrics between the outputs of the neural network.

We first compute the output in the top layer of the network with forward propagation. Then from the top layer to the first layer, the partial derivatives of the objective function $J(\boldsymbol{W},\boldsymbol{b})$ can be computed layer by layer with the back-propagation method [31, 32]. We denote $\frac{1}{2}d_{i,j}^{+}$, $\frac{1}{2}d_{i,j}^{-}$ and $\frac{1}{2}\|\boldsymbol{z}_i^p-\boldsymbol{\mu}_i^p\|_2^2$ by $J_1(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)$, $J_2(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)$ and $J_3(\boldsymbol{z}_i^p,\boldsymbol{\mu}_i^p)$, respectively. Thus, the partial derivatives of the objective function $J(\boldsymbol{W},\boldsymbol{b})$ with respect to $\boldsymbol{W}^k$ and $\boldsymbol{b}^k$ can be computed as:

$$\frac{\partial J(\boldsymbol{W},\boldsymbol{b})}{\partial \boldsymbol{W}^k} = \frac{\alpha}{\sum n_i}\sum_{i=1}^{N}\sum_{j\in c(i)}\frac{\partial J_1(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{W}^k} + \frac{1-\alpha}{\sum m_i}\sum_{i=1}^{N}\sum_{j\notin c(i)}\frac{\partial J_2(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{W}^k} + \frac{\lambda}{N}\sum_{p=2}^{K-1}\sum_{i=1}^{N}\frac{\partial J_3(\boldsymbol{z}_i^p,\boldsymbol{\mu}_i^p)}{\partial \boldsymbol{W}^k} + \gamma\boldsymbol{W}^k \quad (18)$$

$$\frac{\partial J(\boldsymbol{W},\boldsymbol{b})}{\partial \boldsymbol{b}^k} = \frac{\alpha}{\sum n_i}\sum_{i=1}^{N}\sum_{j\in c(i)}\frac{\partial J_1(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{b}^k} + \frac{1-\alpha}{\sum m_i}\sum_{i=1}^{N}\sum_{j\notin c(i)}\frac{\partial J_2(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{b}^k} + \frac{\lambda}{N}\sum_{p=2}^{K-1}\sum_{i=1}^{N}\frac{\partial J_3(\boldsymbol{z}_i^p,\boldsymbol{\mu}_i^p)}{\partial \boldsymbol{b}^k}. \quad (19)$$

For layer k, let $\boldsymbol{a}_i^{k+1}$ be the weighted vector in layer $k+1$, $\boldsymbol{a}_i^{k+1} = \boldsymbol{W}^k\boldsymbol{z}_i^k + \boldsymbol{b}^k$, $k=1,2,\cdots,K-1$. $\frac{\partial J_1(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{W}^k}$ can be re-written as the following formula:

$$\frac{\partial J_1(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{W}^k} = \frac{\partial J_1(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{a}_i^{k+1}}(\boldsymbol{z}_i^k)^T + \frac{\partial J_1(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{a}_j^{k+1}}(\boldsymbol{z}_j^k)^T. \quad (20)$$

$\frac{\partial J_2(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{W}^k}$ can be re-written as:

$$\frac{\partial J_2(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{W}^k} = \begin{cases}\frac{-\partial J_1(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{W}^k}, & \|\boldsymbol{z}_i^K-\boldsymbol{z}_j^K\|_2^2<\eta\\ \boldsymbol{0}, & \|\boldsymbol{z}_i^K-\boldsymbol{z}_j^K\|_2^2\ge\eta.\end{cases} \quad (21)$$

It is noted that when $\|\boldsymbol{z}_i^K-\boldsymbol{z}_j^K\|_2^2=\eta$ we chose $\boldsymbol{0}$ as the subgradient of $J_2(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)$ with respect to $\boldsymbol{W}^k$.

Since $\boldsymbol{\mu}_i^p = \frac{\sum_{j\in c(i)}\boldsymbol{z}_j^p}{s_i}$, where s_i denotes the number of the samples associated with $c(i)$, $J_3(\boldsymbol{z}_i^p,\boldsymbol{\mu}_i^p)$ can be represented as:

$$J_3(\boldsymbol{z}_i^p,\boldsymbol{\mu}_i^p) = \frac{1}{2}\|(1-\frac{1}{s_i})\boldsymbol{z}_i^p - \frac{\sum_{j\in c(i),j\neq i}\boldsymbol{z}_j^p}{s_i})\|_2^2. \quad (22)$$

Thus, $\frac{\partial J_3(\boldsymbol{z}_i^p,\boldsymbol{\mu}_i^p)}{\partial \boldsymbol{W}^k}$ can be re-written as:

$$\frac{\partial J_3(\boldsymbol{z}_i^p,\boldsymbol{\mu}_i^p)}{\partial \boldsymbol{W}^k} = \frac{\partial J_3(\boldsymbol{z}_i^p,\boldsymbol{\mu}_i^p)}{\partial \boldsymbol{a}_i^{k+1}}(\boldsymbol{z}_i^k)^T + \sum_{j\in c(i),j\neq i}\frac{\partial J_3(\boldsymbol{z}_i^p,\boldsymbol{\mu}_i^p)}{\partial \boldsymbol{a}_j^{k+1}}(\boldsymbol{z}_j^k)^T. \quad (23)$$

Let $\frac{\partial J_1(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{a}_i^{k+1}}$, $\frac{\partial J_1(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{a}_j^{k+1}}$, $\frac{\partial J_2(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{a}_i^{k+1}}$, $\frac{\partial J_2(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{a}_j^{k+1}}$, $\frac{\partial J_3(\boldsymbol{z}_i^p,\boldsymbol{\mu}_i^p)}{\partial \boldsymbol{a}_i^{k+1}}$ and $\frac{\partial J_3(\boldsymbol{z}_j^p,\boldsymbol{\mu}_j^p)}{\partial \boldsymbol{a}_j^{k+1}}$ be the errors $\boldsymbol{\delta}_{k+1,i}^{1,K}$, $\boldsymbol{\delta}_{k+1,j}^{1,K}$, $\boldsymbol{\delta}_{k+1,i}^{2,K}$, $\boldsymbol{\delta}_{k+1,j}^{2,K}$, $\boldsymbol{\delta}_{k+1,i}^{3,p}$ and $\boldsymbol{\delta}_{k+1,j}^{3,p}$, respectively. For $k=K-1$, $\boldsymbol{\delta}_{K,i}^{1,K}$ and $\boldsymbol{\delta}_{K,j}^{1,K}$ can be represented as:

$$\boldsymbol{\delta}_{K,i}^{1,K} = \frac{\partial(\boldsymbol{Z}_i^K)^T}{\partial \boldsymbol{a}_i^K}\frac{\partial J_1(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{Z}_i^K} = (\boldsymbol{z}_i^K-\boldsymbol{z}_j^K)\bullet\sigma'(\boldsymbol{a}_i^K)$$
$$\boldsymbol{\delta}_{K,j}^{1,K} = \frac{\partial(\boldsymbol{Z}_j^K)^T}{\partial \boldsymbol{a}_j^K}\frac{\partial J_1(\boldsymbol{z}_i^K,\boldsymbol{z}_j^K)}{\partial \boldsymbol{Z}_j^K} = (-\boldsymbol{z}_i^K+\boldsymbol{z}_j^K)\bullet\sigma'(\boldsymbol{a}_j^K) \quad (24)$$

where $\sigma'(\boldsymbol{a}_i^K)$ is the derivative of the activation function in the output layer with respect to a_i^K and $\bullet$ denotes the element-wise multiplication. Also, $\boldsymbol{\delta}_{k+1,i}^{2,K}$ and $\boldsymbol{\delta}_{k+1,j}^{2,K}$ can be represented as:

$$\boldsymbol{\delta}_{K,i}^{2,K} = \begin{cases}(-\boldsymbol{z}_i^K+\boldsymbol{z}_j^K)\bullet\sigma'(\boldsymbol{a}_i^K), & \|\boldsymbol{z}_i^K-\boldsymbol{z}_j^K\|_2^2<\eta\\ \boldsymbol{0}, & \|\boldsymbol{z}_i^K-\boldsymbol{z}_j^K\|_2^2\ge\eta\end{cases}$$
$$\boldsymbol{\delta}_{K,j}^{2,K} = \begin{cases}(\boldsymbol{z}_i^K-\boldsymbol{z}_j^K)\bullet\sigma'(\boldsymbol{a}_j^K), & \|\boldsymbol{z}_i^K-\boldsymbol{z}_j^K\|_2^2<\eta\\ \boldsymbol{0}, & \|\boldsymbol{z}_i^K-\boldsymbol{z}_j^K\|_2^2\ge\eta.\end{cases} \quad (25)$$

For layer $k = K-2, K-3, \cdots, 1$, with the back-propagation algorithm, $\boldsymbol{\delta}_{k+1,i}^{1,K}$, $\boldsymbol{\delta}_{k+1,j}^{1,K}$, $\boldsymbol{\delta}_{k+1,i}^{2,K}$ and $\boldsymbol{\delta}_{k+1,j}^{2,K}$ can be obtained as:

$$\begin{aligned}
\boldsymbol{\delta}_{k+1,i}^{1,K} &= ((\boldsymbol{W}^{k+1})^T \boldsymbol{\delta}_{k+2,i}^{1,K}) \bullet \sigma'(\boldsymbol{a}_i^{k+1}) \\
\boldsymbol{\delta}_{k+1,j}^{1,K} &= ((\boldsymbol{W}^{k+1})^T \boldsymbol{\delta}_{k+2,j}^{1,K}) \bullet \sigma'(\boldsymbol{a}_j^{k+1}) \\
\boldsymbol{\delta}_{k+1,i}^{2,K} &= ((\boldsymbol{W}^{k+1})^T \boldsymbol{\delta}_{k+2,i}^{2,K}) \bullet \sigma'(\boldsymbol{a}_i^{k+1}) \\
\boldsymbol{\delta}_{k+1,j}^{2,K} &= ((\boldsymbol{W}^{k+1})^T \boldsymbol{\delta}_{k+2,j}^{2,K}) \bullet \sigma'(\boldsymbol{a}_j^{k+1}).
\end{aligned} \tag{26}$$

For $\boldsymbol{\delta}_{p,i}^{3,p}$ and $\boldsymbol{\delta}_{p,j}^{3,p}$, we have:

$$\begin{aligned}
\boldsymbol{\delta}_{p,i}^{3,p} &= (1 - \frac{1}{s_i})(\boldsymbol{z}_i^p - \boldsymbol{\mu}_i^p) \bullet \sigma'(\boldsymbol{a}_i^p) \\
\boldsymbol{\delta}_{p,j}^{3,p} &= \frac{1}{s_i}(-\boldsymbol{z}_i^p + \boldsymbol{\mu}_i^p) \bullet \sigma'(\boldsymbol{a}_j^p).
\end{aligned} \tag{27}$$

And for $k = p-2, p-3, \cdots, 1$, $\boldsymbol{\delta}_{k+1,i}^{3,p}$ and $\boldsymbol{\delta}_{k+1,j}^{3,p}$ can be represented:

$$\begin{aligned}
\boldsymbol{\delta}_{k+1,i}^{3,p} &= ((\boldsymbol{W}^{k+1})^T \boldsymbol{\delta}_{k+2,i}^{3,p}) \bullet \sigma'(\boldsymbol{a}_i^{k+1}) \\
\boldsymbol{\delta}_{k+1,j}^{3,p} &= ((\boldsymbol{W}^{k+1})^T \boldsymbol{\delta}_{k+2,j}^{3,p}) \bullet \sigma'(\boldsymbol{a}_j^{k+1}).
\end{aligned} \tag{28}$$

Thus, $\frac{\partial J_1(\boldsymbol{z}_i^K, \boldsymbol{z}_j^K)}{\partial \boldsymbol{W}^k}$, $\frac{\partial J_2(\boldsymbol{z}_i^K, \boldsymbol{z}_j^K)}{\partial \boldsymbol{W}^k}$ and $\frac{\partial J_3(\boldsymbol{z}_i^p, \boldsymbol{\mu}_i^p)}{\partial \boldsymbol{W}^k}$ can be represented as:

$$\begin{aligned}
\frac{\partial J_1(\boldsymbol{z}_i^K, \boldsymbol{z}_j^K)}{\partial \boldsymbol{W}^k} &= \boldsymbol{\delta}_{k+1,i}^{1,K}(\boldsymbol{z}_i^k)^T + \boldsymbol{\delta}_{k+1,j}^{1,K}(\boldsymbol{z}_j^k)^T \\
\frac{\partial J_2(\boldsymbol{z}_i^K, \boldsymbol{z}_j^K)}{\partial \boldsymbol{W}^k} &= \boldsymbol{\delta}_{k+1,i}^{2,K}(\boldsymbol{z}_i^k)^T + \boldsymbol{\delta}_{k+1,j}^{1,K}(\boldsymbol{z}_j^k)^T \\
\frac{\partial J_3(\boldsymbol{z}_i^p, \boldsymbol{\mu}_i^p)}{\partial \boldsymbol{W}^k} &= \boldsymbol{\delta}_{k+1,i}^{3,p}(\boldsymbol{z}_i^k)^T + \sum_{j \in c(i), j \neq i} \boldsymbol{\delta}_{k+1,j}^{3,p}(\boldsymbol{z}_j^k)^T.
\end{aligned} \tag{29}$$

Similarly, we can calculate $\frac{\partial J_1(\boldsymbol{z}_i^K, \boldsymbol{z}_j^K)}{\partial \boldsymbol{b}^k}$, $\frac{\partial J_2(\boldsymbol{z}_i^K, \boldsymbol{z}_j^K)}{\partial \boldsymbol{b}^k}$ and $\frac{\partial J_3(\boldsymbol{z}_i^p, \boldsymbol{\mu}_i^p)}{\partial \boldsymbol{b}^k}$ as:

$$\begin{aligned}
\frac{\partial J_1(\boldsymbol{z}_i^K, \boldsymbol{z}_j^K)}{\partial \boldsymbol{b}^k} &= \boldsymbol{\delta}_{k+1,i}^{1,K} + \boldsymbol{\delta}_{k+1,j}^{1,K} \\
\frac{\partial J_2(\boldsymbol{z}_i^K, \boldsymbol{z}_j^K)}{\partial \boldsymbol{b}^k} &= \boldsymbol{\delta}_{k+1,i}^{2,K} + \boldsymbol{\delta}_{k+1,j}^{2,K} \\
\frac{\partial J_3(\boldsymbol{z}_i^p, \boldsymbol{\mu}_i^p)}{\partial \boldsymbol{b}^k} &= \boldsymbol{\delta}_{k+1,i}^{3,p} + \sum_{j \in c(i), j \neq i} \boldsymbol{\delta}_{k+1,j}^{3,p}.
\end{aligned} \tag{30}$$

By substituting Eqs. (29) and (30) into Eqs. (18) and (19), we can obtain the partial derivatives of the objective function $J(\boldsymbol{W}, \boldsymbol{b})$ with respect to $\boldsymbol{W}^k$ and $\boldsymbol{b}^k$, $\frac{\partial J(\boldsymbol{W}, \boldsymbol{b})}{\partial \boldsymbol{W}^k}$ and $\frac{\partial J(\boldsymbol{W}, \boldsymbol{b})}{\partial \boldsymbol{b}^k}$. Then $\boldsymbol{W}^k$ and $\boldsymbol{b}^k$ can be updated with the gradient descent algorithm. The training algorithm of our proposed deep metric learning model is summarized in Algorithm 1. Fig. (4) plots the curve of the value of the objective function versus the iteration number on the McGill shape dataset. Once weight $\boldsymbol{W}$ and bias $\boldsymbol{b}$ are learned, we can use Eq. (15) to compute the distance metric for retrieval.

IV. Experimental Results

In this section, we first evaluate our proposed deep nonlinear metric learning based shape retrieval method, and then compare it with the state-of-the-art 3D shape retrieval methods on three benchmark datasets, i.e., McGill shape dataset [33], SHREC'10 ShapeGoogle dataset [18] and SHREC'14 Human dataset [34].

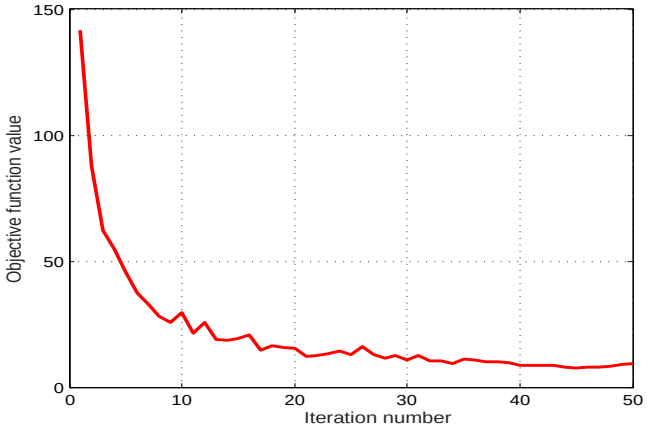


Fig. 4. Convergence curve of the proposed objective function on the McGill shape dataset.

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

Input: training set $\boldsymbol{x}_i$; layer size K of the neural network; weight α; regularization parameters λ and γ; threshold η; learning rate θ.

Output: $\boldsymbol{W}$ and $\boldsymbol{b}$.

For $q = 1, 2, \cdots, Q$:

1) Compute the outputs of the neural network with forward propagation for all input training samples $\boldsymbol{x}_i$;
2) For $k = K-1, K-2, \cdots, 1$
   Compute $\frac{\partial J(\boldsymbol{W}, \boldsymbol{b})}{\partial \boldsymbol{W}^k}$ with Eqs. (24-28), (29), (18);
   Compute $\frac{\partial J(\boldsymbol{W}, \boldsymbol{b})}{\partial \boldsymbol{b}^k}$ with Eqs. (24-28), (30), (19);
3) Update $\boldsymbol{W}^k$ and $\boldsymbol{b}^k$ for $k = 1, 2, \cdots, K-1$:
   $\boldsymbol{W}^k = \boldsymbol{W}^k - \theta \frac{\partial J(\boldsymbol{W}, \boldsymbol{b})}{\partial \boldsymbol{W}^k}$;
   $\boldsymbol{b}^k = \boldsymbol{b}^k - \theta \frac{\partial J(\boldsymbol{W}, \boldsymbol{b})}{\partial \boldsymbol{b}^k}$.

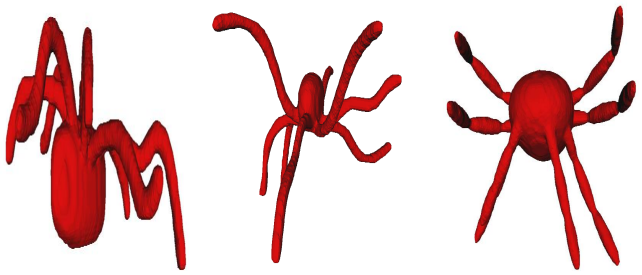
Output $\boldsymbol{W}$ and $\boldsymbol{b}$ until the values of $J(\boldsymbol{W}, \boldsymbol{b})$ in adjacent iterations are close enough or the maximum number of iterations is reached.

A. Experimental Settings

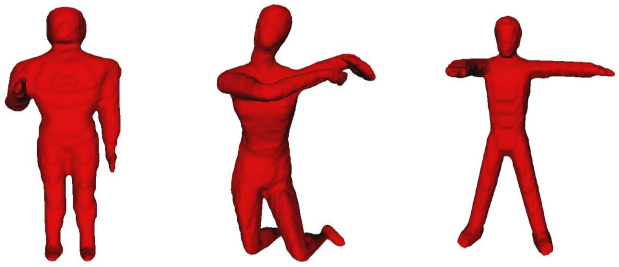
We take 19 frequency components to compute the SI-HKS to form a 19-dimensional feature vector for each vertex of the shape. In the LLC method, the size of the learned dictionary is 2000 and 5 atoms are selected to form the sub-dictionary for each SI-HKS feature vector. Thus, for each shape, a 2000-dimensional global 3D shape descriptor is used as input to the deep neural network. In the proposed deep metric learning model, the neural network with layers of 2000-1000-300-100 is used. Moreover, in Eq. (17), parameters α, λ and γ are set to 0.6, 0.06 and 0.0001, respectively. The threshold η is set to 5.0.

In the McGill 3D shape dataset, there are 255 3D shapes with complex geometric structural variations, which are from 10 classes. Since for each class there are large deformations with the shapes, the shape retrieval task on this dataset is challenging. Fig. 5 shows the example shapes with the large deformations in the McGill shape dataset.

The SHREC'10 ShapeGoogle dataset contains 1184 synthetic shapes, where 715 shapes from 13 classes are with



(a) Octopuse models.



(b) Human models.

Fig. 5. The example shapes in the McGill 3D shape dataset.

the five simulated transformations, i.e., isometry, topology, isometry+topology, partiality and triangulation. And there are 456 unrelated shapes in this dataset. All shapes are re-meshed to have the same number of vertices and keep the same vertex-wise correspondence. Fig. 6 shows the five kinds of transformations in the SHREC'10 ShapeGoogle dataset.

The SHREC'14 Human dataset contains two sub-datasets, including 300 synthetic human shapes and 400 scanned human models. For each human shape model, there are 20 different poses and 10 different poses in the two sub-datasets, respectively. Since there are large pose changes with the human shapes from the same class and similar geometric structures with the shapes from different classes, the SHREC'14 Human dataset is an extremely challenging one. Fig. 7 shows different human shapes in the SHREC'14 Human dataset.

B. Evaluation of The Proposed Method

In order to demonstrate the effectiveness of the proposed method, we compare the proposed method to the LLC based 3D shape descriptor without employing deep non-linear metric learning on the McGill dataset.

1) Comparison to the LLC based shape descriptor: As described in Section III. A, we employed LLC to encode the SI-HKSs of vertices on the shape. The resulting encoding coefficient histogram, i.e., the LLC based shape descriptor, is used to represent the shape globally. In our proposed deep metric learning based shape retrieval method, we use the LLC based global 3D shape descriptor as input to the deep metric network. With the deep metric network, the LLC based global 3D shape descriptor can be mapped to a non-linear feature space. Thus, the distance between the outputs of the deep metric network can be viewed as the non-linear transformation of the distance between the original 3D shape descriptors. And the within-class variations of the 3D shape descriptors are minimized and the between-class variations of the 3D shape descriptors are maximized. We denote our proposed shape retrieval method with the deep non-linear metric learning model by DNML. In order to demonstrate the effectiveness of the proposed DNML method, we compare the proposed DNML method to the LLC based 3D shape descriptor without using the proposed deep metric learning model on the McGill shape dataset.

For the LLC based 3D shape descriptor, since the size of the learned dictionary is 2000, we form a 2000-dimensional global shape descriptor to describe the shape. The Euclidean distance between the 2000-dimensional shape descriptors is used as the similarity for retrieval. In the proposed method, we use the proposed neural network to train our deep metric learning model. The Euclidean distance between the outputs of the metric network is used for retrieval. Fig. 8 shows the precision-recall curves for the LLC based shape descriptor and the proposed method. As can be seen in this figure, compared to the LLC based shape descriptor without applying metric learning, the learned distance metric can significantly improve the retrieval performance.

C. Comparison Evaluation

1) McGill shape dataset: In our proposed DNML method, 10 shapes per class are chosen as the training samples to train the proposed deep metric network and the remaining samples per class are used to test. We compare our proposed method to the current shape retrieval methods: learning based covariance descriptor [19], Graph-based method [35], the PCA based VLAT method [36], the Hybrid BOW [37], the hybrid 2D/3D approach [38], the manifold ranking method [23]. Following the evaluation criteria in [19], Nearest Neighbor (NN), the First Tier (FT), the Second Tier (ST) and the Discounted Cumulative Gain (DCG) are used to evaluate these methods. The retrieval performance of these methods is illustrated in Table I. As can be seen in this table, in terms of the evaluation criteria FT, ST and DCG, compared to these methods [19, 23, 35–38], the proposed DNML method can achieve the best performance.

Particularly, in [23], with the BOW method, the global shape descriptor is learned from a set of SIFT features of the rendered depth images. Then, the high dimensional space of the global shape descriptors is represented by a Laplacian graph and the manifold ranking based metric learning method is employed to compute the distance for retrieval. Compared to the manifold ranking based distance metric learning method [23], our proposed DNML method can learn a deep non-linear transform to obtain better retrieval performance.

2) SHREC'10 ShapeGoogle dataset: For the SHREC'10 ShapeGoogle dataset, we compare the proposed DNML method to the vector quantization (VQ) based BOW method [18], the unsupervised dictionary learning (UDL) method [20] and the supervised dictionary learning (SDL) method [20]. Comparison results with the mean average precision (MAP) are listed in Table II. From this table, one can see that in the cases of the isometry, topology, isometry+topology

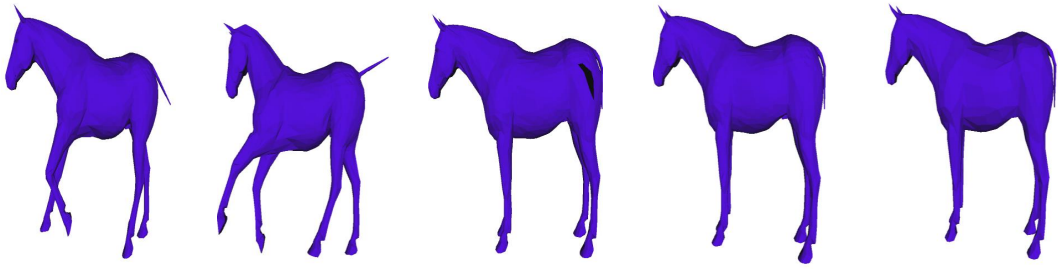
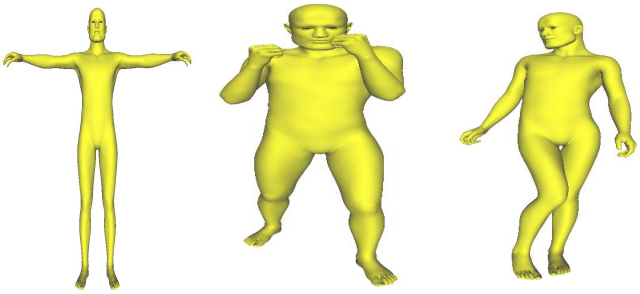


Fig. 6. The five simulated transformations in the SHREC'10 ShapeGoogle dataset: isometry transform, isometry+topology transform, partiality transform, topology transform and triangulation transform.

(a) Synthetic human models.



(b) Scanned human models.

Fig. 7. The example shapes in the SHREC'14 Human dataset.

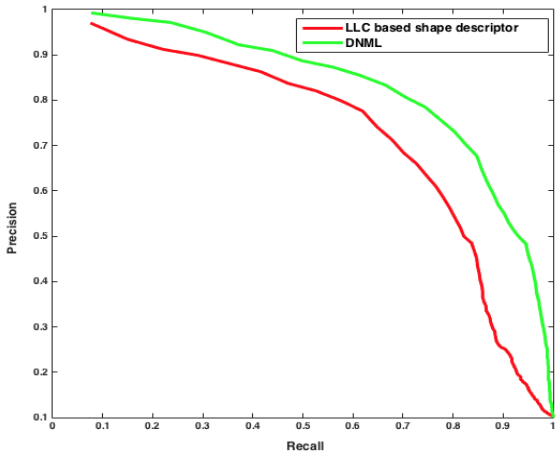


Fig. 8. The precision-recall curve for the LLC based shape descriptor and the proposed DNML method on the McGill shape dataset.

TABLE I
RETRIEVAL RESULTS ON THE MCGILL DATASET.

Methods	NN	FT	ST	DCG
Covariance descriptor [19]	**0.977**	0.732	0.818	0.937
Graph-based method [35]	0.976	0.741	0.911	0.933
PCA based VLAT [36]	0.969	0.658	0.781	0.894
Hybrid BOW [37]	0.957	0.635	0.790	0.886
Hybrid 2D/3D [38]	0.925	0.557	0.698	0.850
Manifold ranking [23]	-	0.761	-	-
Proposed DNML	0.962	**0.906**	**0.969**	**0.967**

and partiality transformations, our proposed DNML method is comparable or superior to the vector quantization (VQ) based BOW method [18], the unsupervised dictionary learning (UDL) method [20] and the supervised dictionary learning (SDL) method [20]. For example, in the cases of isometry+topology and partiality transformations, our proposed DNML method can obtain the accuracies of 0.979 and 0.983 while the SDL method [20] can obtain the accuracies of 0.956 and 0.951. However, in the case of the triangulation transformation, the accuracy of our proposed DNML method is slightly lower than those of the three methods.

It is noted that in the VQ based BOW method [18], the authors employed the similarity sensitive hashing (SSH) to learn the distance metric between the BOW features. The SSH based metric learning can be viewed to learn a linear distance metric between the shape descriptors for retrieval. Since there are the large non-rigid deformations with the shapes, in the proposed DNML method, we employ the deep neural network to non-linearly map the 3D shape descriptors to a non-linear feature space. Compared to the learned linear distance metric, the learned non-linear distance metric in our proposed DNML method can characterize the manifold of the deformable shapes better. Therefore, the proposed DNML method can obtain better performance in the most cases. As can be seen in Table II, in the cases of the isometry+topology and partiality transformations, the proposed DNML method can obtain the accuracies of 0.979 and 0.983 while the VQ based BOW method [18] can obtain the accuracies of 0.933 and 0.947, respectively.

3) SHREC'14 Human dataset: We compare the proposed DNML method to the following shape retrieval methods: Histogram of area projection transform (HAPT) [39], intrinsic pyramid matching (ISPM) [40], reduced Bi-harmonic distance matrix (RBiHDM) [41], deep belief network (DBN) [34], the standard vector quantization (VQ) based BOW method

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

Transformation	VQ [18]	UDL [20]	SDL [20]	Proposed DNML
Isometry	0.988	0.977	0.994	**1.000**
Topology	**1.000**	**1.000**	**1.000**	**1.000**
Isometry+Topology	0.933	0.934	0.956	**0.979**
Partiality	0.947	0.948	0.951	**0.983**
Triangulation	0.954	0.950	**0.955**	0.943

[18], the unsupervised dictionary learning (UDL) method [20] and the supervised dictionary learning (SDL) method [20]. For the synthetic and scanned sub-datasets, 10 shapes and 5 shapes per class are used to train the proposed deep non-linear metric network and the other shapes per class are used for testing, respectively. The mean average precision is also used to evaluate these methods. The experimental results are listed in Table III. As can be seen in this table, for the synthetic sub-dataset, in comparison with the methods [18, 20, 34, 39–41], our proposed DNML method can obtain better performance. Nonetheless, for the scanned sub-dataset, the mean average precision of our proposed DNML method is slightly higher than that of the SDL method [20].

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

Method	Synthetic model	Scanned model
HAPT[39]	0.817	0.637
ISPM[40]	0.92	0.258
RBiHDM[41]	0.642	0.640
DBN[34]	0.842	0.304
VQ [18]	0.813	0.514
UDL [20]	0.842	0.523
SDL [20]	0.95.1	0.791
Proposed DNML	**0.973**	**0.801**

D. Computational Time Evaluation

For each shape with T vertices, the computational complexity of SI-HKS is $\mathcal{O}(T^3)$, dominated by the computation of the eigenvectors and eigenvalues of the Laplace-Beltrami operator. The computational complexity of the global shape feature representation with *K*-means clustering and LLC coding is $\mathcal{O}(ImLN)$ and $\mathcal{O}(L + L_a^2)$, where I is the iteration number of *K*-means clustering, m is the dimension of SI-HKS, N is the number of training samples, L is the size of the learned dictionary and L_a is the number of selected atoms. In the stage of training the proposed deep metric learning model, the computational complexity of updating $\boldsymbol{W}$ is $\mathcal{O}(Q\sum_{k=1}^{K-1} l_k l_{k+1})$ while the computational complexity of updating $\boldsymbol{b}$ is $\mathcal{O}(Q\sum_{k=2}^{K-1} l_k l_{k+1})$, where Q is the iteration number of the back-propagation algorithm, l_k is the size of the kth layer of the neural network. In the testing stage, the computational complexity of forming the deep shape descriptor is $\mathcal{O}(\sum_{k=1}^{K-1} l_k l_{k+1})$.

The proposed DNML method was implemented in Matlab and tested on a Dell workstation with an Intel Xeon E5 CPU and 32 GB RAM. We evaluate computational time of the proposed DNML method on the McGill shape dataset. For constructing the training dataset, we choose 10 shapes for each class as the training samples. The computation of SI-HKS, the global shape descriptor and training the proposed deep metric network takes 6.5 min, 11 min and 3.5 min, respectively. Thus, the total training time on the 100 training samples is about 21 min. For testing, the average computational time of the learned non-linear distance metric between a pair of samples is about 2.1 sec.

V. CONCLUSIONS

In this paper, we proposed a deep non-linear metric learning method for 3D shape retrieval. We developed a metric network by minimizing a discriminative loss function that can enforce the similarity between a pair of samples from the same class to be small, the similarity between a pair of samples from different classes to be large and the neurons in the hidden layers to approach to their means. Based on the proposed metric network, we can non-linearly map the global 3D shape descriptors to a non-linear feature space. The distance between the outputs of the metric network is used as the similarity for retrieval. The proposed deep non-linear metric learning method demonstrates its retrieval performance on the McGill, SHREC'10 ShapeGoogle and SHREC'14 Human datasets.

ACKNOWLEDGMENT

The authors would like to thank the editor and the anonymous reviewers for their constructive comments on this paper. This work was supported by New York University Abu Dhabi under Grants AD131 and REF131.

REFERENCES

[1] A. Barmpoutis, "Tensor body: Real-time reconstruction of the human body and avatar synthesis from RGB-D," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1347–1356, 2013.

[2] H. P. H. Shum, E. S. L. Ho, Y. Jiang, and S. Takagi, "Real-time posture reconstruction for microsoft kinect," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1357–1369, 2013.

[3] O. Lopes, M. Reyes, S. Escalera, and J. Gonzàlez, "Spherical blurred shape model for 3D object and pose recognition: Quantitative analysis and HCI applications in smart environments," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2379–2390, 2014.

[4] B. B. Amor, H. Drira, S. Berretti, M. Daoudi, and A. Srivastava, "4D facial expression recognition by learning geometric deformations," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2443–2457, 2014.

[5] M. Ovsjanikov, J. Sun, and L. J. Guibas, "Global intrinsic symmetries of shapes," *Computer Graphics Forum*, vol. 27, no. 5, pp. 1341–1348, 2008.

[6] R. Litman and A. M. Bronstein, "Learning spectral descriptors for deformable shape correspondence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 171–180, 2014.

[7] 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.

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

[9] 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, July 8-10*, 2009.

[10] 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*, to appear.

[11] R. M. Rustamov, "Laplace-beltrami eigenfunctions for deformation invariant shape representation," in *Eurographics Symposium on Geometry Processing, Barcelona, Spain, July 4-6*, 2007, pp. 225–233.

[12] J. Sun, M. Ovsjanikov, and L. J. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Computer Graphics Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.

[13] 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, USA*, 2010, pp. 1704–1711.

[14] 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, November 6-13*, 2011, pp. 1626–1633.

[15] 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.

[16] I. Kokkinos, M. M. Bronstein, R. Litman, and A. M. Bronstein, "Intrinsic shape context descriptors for deformable shapes," in *IEEE International Conference on Computer Vision and Pattern Recognition, Providence, RI, USA*, 2012, pp. 159–166.

[17] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud, "Surface feature detection and description with applications to mesh matching," in *IEEE International Conference on Computer Vision and Pattern Recognition, Miami, Florida, USA*, 2009, pp. 373–380.

[18] 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, no. 1, p. 1, 2011.

[19] 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.

[20] 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.

[21] Z. Lian, J. Zhang, S. Choi, H. ElNaghy, J. El-Sana, T. Furuya, A. Giachetti, R. A. Guler, L. Lai, C. Li, H. Li, F. A. Limberger, R. Martin, R. U. Nakanishi, A. P. Neto, L. G. Nonato, R. Ohbuchi, K. Pevzner, D. Pickup, P. Rosin, A. Sharf, L. Sun, X. Sun, S. Tari, G. Unal, 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.

[22] 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*, June 2015.

[23] R. Ohbuchi and T. Furuya, "Distance metric learning and feature combination for shape-based 3D model retrieval," in *ACM Workshop on 3D Object Retrieval*, New York, 2010, pp. 63–68.

[24] 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, USA, June 13-18*, 2010, pp. 3360–3367.

[25] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell, "Distance metric learning with application to clustering with side-information," in *Advances in Neural Information Processing Systems, Vancouver, Canada*, 2002, pp. 505–512.

[26] X. Cai, C. Wang, B. Xiao, X. Chen, and J. Zhou, "Deep nonlinear metric learning with independent subspace analysis for face verification," in *ACM Multimedia Conference, Nara, Japan, October 29 - November 02*, 2012, pp. 749–752.

[27] D. Kedem, S. Tyree, K. Q. Weinberger, F. Sha, and G. R. G. Lanckriet, "Non-linear metric learning," in *Advances in Neural Information Processing Systems, Nevada, USA.*, 2012, pp. 2582–2590.

[28] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Joint learning of discriminative prototypes and large margin nearest neighbor classifiers," in *IEEE International Conference on Computer Vision, Sydney, Australia, December 1-8*, 2013, pp. 3112–3119.

[29] J. Hu, J. Lu, and Y. Tan, "Discriminative deep metric learning for face verification in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, June 23-28*, 2014, pp. 1875–1882.

[30] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou, "Multi-manifold deep metric learning for image set classification," in *IEEE Conference on Computer Vision and Pattern Recognition, Boston, USA*, June 2015.

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

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

[33] 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.

[34] 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 *Eurographics Workshop on 3D Object Retrieval*, 2014.

[35] 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.

[36] 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.

[37] 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.

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

[39] A. Giachetti and C. Lovato, "Radial symmetry detection and shape characterization with the multiscale area projection transform," *Computer Graphics Forum*, vol. 31, no. 5, pp. 1669–1678, 2012.

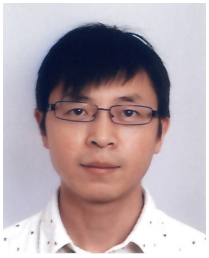
[40] C. Li and A. B. Hamza, "A multiresolution descriptor for deformable 3D shape retrieval," *The Visual Computer*, vol. 29, no. 6-8, pp. 513–524, 2013.

[41] 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.

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. His research interests include image forensics, computer vision and machine learning. Currently he is focusing on 3D computer vision with the 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.

Fan Zhu received the MSc degree with distinction in Electrical Engineering and the Ph.D. degree at the Visual Information Engineering group from the Department of Electronic and Electrical Engineering, the University of Sheffield, Sheffield, U.K, in 2011 and 2015, respectively. He is currently a post-doctoral associate at New York University Abu Dhabi. His research interests include submodular optimization for computer vision, sparse coding, 3D feature learning, dictionary learning and transfer learning. He has authored/co-authored over 10 papers in well-known journals/conferences such as IJCV, IEEE TNNLS, CVPR, CIKM and BMVC, and two China patents. He has been awarded the National Distinguished Overseas Self-funded Student of China prize in 2014. He serves as a reviewer of IEEE Transactions on Cybernetics.



Ling Shao is Chair in Computer Vision and Head of the Computer Vision and Artificial Intelligence Group with the Department of Computer and Information Sciences at Northumbria University, Newcastle upon Tyne and an Advanced Visiting Fellow with the Department of Electronic and Electrical Engineering at the University of Sheffield. He received the B.Eng. degree in Electronic and Information Engineering from the University of Science and Technology of China (USTC), the M.Sc. degree in Medical Image Analysis and the Ph.D. (D.Phil.) degree in Computer Vision at the Robotics Research Group from the University of Oxford. Previously, he was a Senior Lecturer (2009-2014) with the Department of Electronic and Electrical Engineering at the University of Sheffield and a Senior Scientist (2005-2009) with Philips Research, The Netherlands. His research interests include Computer Vision, Image/Video Processing, Pattern Recognition and Machine Learning. He has authored/co-authored over 200 papers in refereed journals/conferences such as IEEE TPAMI, TIP, TNNLS, IJCV, ICCV, CVPR, ECCV, IJCAI and ACM MM, and holds over 10 EU/US patents.

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.