

DeepShape: Deep-Learned Shape Descriptor for 3D Shape Retrieval

Jin Xie, Guoxian Dai, Fan Zhu, Edward K.Wong, and Yi Fang

Abstract—Complex geometric variations of 3D models usually pose great challenges in 3D shape matching and retrieval. In this paper, we propose a novel 3D shape feature learning method to extract high-level shape features that are insensitive to geometric deformations of shapes. Our method uses a discriminative deep auto-encoder to learn deformation-invariant shape features. First, a multiscale shape distribution is computed and used as input to the auto-encoder. We then impose the Fisher discrimination criterion on the neurons in the hidden layer to develop a deep discriminative auto-encoder. Finally, the outputs from the hidden layers of the discriminative auto-encoders at different scales are concatenated to form the shape descriptor. The proposed method is evaluated on four benchmark datasets that contain 3D models with large geometric variations: McGill, SHREC'10 ShapeGoogle, SHREC'14 Human and SHREC'14 Large Scale Comprehensive Retrieval Track Benchmark datasets. Experimental results on the benchmark datasets demonstrate the effectiveness of the proposed method for 3D shape retrieval.

Index Terms—3D shape retrieval, heat kernel signature, heat diffusion, auto-encoder, Fisher discrimination criterion

1 INTRODUCTION

NOWADAYS with the development of 3D acquisition and printing technology there is an explosive growth of 3D meshed surface models. Due to the data-richness of 3D models, shape retrieval for 3D model searching, understanding and analyzing has received more and more attention. Using a shape as a query, the shape retrieval algorithm aims to find similar shapes. The performance of a shape retrieval algorithm mainly relies on a shape descriptor that can effectively capture the distinctive properties of shape. It is preferable that a shape descriptor should be deformation-insensitive and invariant to different classes of transformations. The shape descriptor should also be insensitive to both topological and numerical noise. Once the shape descriptor is formed, the similarity between two shapes can be determined by comparing their descriptors.

Shape descriptors for shape matching and retrieval have been extensively studied in the geometry community. In the past decades, plenty of shape descriptors have been proposed; these include the $D2$ shape distribution [1], statistical moments of 3D model [2], Fourier descriptor [3], [4], light field descriptor [5] and eigenvalue descriptor [6], etc. Although these shape descriptors can represent shapes effectively, they are either sensitive to non-rigid transformations or topological changes. To be invariant to isometric transformations, local geometric features, such as spin

images [7], shape context [8] and mesh HOG [9], are extracted to represent the shape. However, these features are sensitive to local geometric noise and they do not capture the global structure of the shape very well.

Apart from the above earlier shape descriptors, another popular approach to shape retrieval uses the diffusion-based point signatures [10], [11], [12]. Based on the Laplace-Beltrami operator, the global point signature (GPS) [12] was proposed to represent shapes. Since the eigenfunctions of the Laplace-Beltrami operator are able to robustly characterize the points on a meshed surface, each vertex is represented by a high-dimensional vector (called GPS) of the scaled eigenfunctions of the Laplace-Beltrami operator evaluated at the vertex. Another widely used shape signature is heat kernel signature (HKS) [10], where the diagonal of the heat kernel is used as a local descriptor to represent shapes. HKS is invariant to isometric deformations and insensitive to small perturbations on the surface. Both GPS and HKS are point signatures that characterize vertices on the meshed surface by vectors.

Once the aforementioned point signatures are formed, the global shape descriptors can be learned from a set of training shapes for shape retrieval. For example, in [13], based on the bag-of-features (BOF) paradigm, a dictionary of words is learned by applying K -means clustering to a set of HKSs. A histogram of pairs of spatially-close words over the learned dictionary is then formed as the shape descriptor for retrieval. Using K -means clustering, Lavoué [14] combined the standard and spatial BoF descriptors for 3D shape retrieval. EINagh et al. proposed the compact HKS-based BOF descriptor, i.e., CompactBoFHKS [15]. In the CompactBoFHKS method, feature point detection is employed to select critical points. For each critical point, certain scales of the HKS are selected to form a compact feature vector to describe it. The BoF method is then applied to the feature vectors to learn a shape descriptor for retrieval. Litman et al. [16] employed sparse coding to learn the

- J. Xie, G. Dai, F. Zhu, and Y. Fang are with the Department of Electrical and Computer Engineering, New York University Abu Dhabi, Abu Dhabi 129188, UAE. E-mail: {jin.xie, guoxian.dai, fan.zhu, yfang1@nyu.edu}.
- E.K. Wong is with the Department of Computer Science and Engineering, Tandon School of Engineering, New York University, New York, NY 10012. E-mail: ewong@nyu.edu.

Manuscript received 1 July 2015; revised 4 May 2016; accepted 16 June 2016.
Date of publication 28 July 2016; date of current version 12 June 2017.

Recommended for acceptance by D. Ramanan.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TPAMI.2016.2596722

dictionary of words instead of K -means clustering. The histogram of encoded representation coefficients over the learned dictionary is used to represent shapes for retrieval. Moreover, in order to obtain discriminative representation coefficients, a class-specific dictionary is constructed using supervised learning.

Recently, due to the success of deep neural networks in different application domains, deep learning based 3D shape features have been proposed for 3D shape analysis. Wu et al. [17] proposed to represent 3D shapes as a probability distribution of binary variables on a 3D voxel grid. Then a convolutional deep belief network is developed to learn the joint probabilistic distribution of the voxel data and the category label. Boscaini et al. [18] employed the windowed Fourier transform to points on the meshed surface to form a local frequency representation. These local frequency representations are then passed through a bank of filters to form a deep representation for 3D shapes. The filter coefficients can be learned by using a task-specific cost function. By constructing a geodesic convolution operator, Masci et al. [19] generalized the convolutional neural network to non-euclidean manifolds for 3D shape retrieval and correspondence.

In this paper, we propose a novel discriminative auto-encoder to learn a shape descriptor for shape retrieval. In the proposed discriminative auto-encoder, we impose the Fisher discrimination criterion on the hidden layer so that the neurons in the hidden layer have small within-class scatter but large between-class scatter. To effectively represent shape, we use a multiscale shape distribution as input to the discriminative auto-encoder. We then train a discriminative auto-encoder at each scale and concatenate the outputs of the hidden layers from different scales as the shape descriptor. The proposed shape descriptor is tested on several benchmark shape datasets and shows promising performance.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the HKS and auto-encoder. In Section 3, we present the discriminative auto-encoder based shape descriptor. We describe our experimental results in Section 4 and conclude the paper in Section 5.

2 BACKGROUND

2.1 Heat Kernel Signature

Provided that there is an initial Dirac delta distribution on the meshed surface X at $t = 0$, the heat diffusion process on X can be described with the following heat equation:

$$\frac{\partial H_t}{\partial t} = -\Psi H_t, \quad (1)$$

where H_t denotes the heat kernel, Ψ is the Laplace-Beltrami operator and t is the diffusion time. The heat kernel $H_t(g_0, g_1)$ measures the heat flow across the meshed surface, which is the amount of heat passing from vertex g_0 to vertex g_1 within a certain amount of time. The solution of Eq. (1) can be obtained:

$$H_t = \exp(-t\Psi). \quad (2)$$

With the spectral theorem, the heat kernel can be expressed as

$$H_t(g_0, g_1) = \sum_i e^{-\lambda_i t} \phi_i(g_0) \phi_i(g_1), \quad (3)$$

where λ_i is the i th eigenvalue of the Laplacian-Beltrami operator and ϕ_i is the i th eigenfunction.

The heat kernel signature [10] of vertex g_0 at time t can be defined as the diagonal of the heat kernel of vertex g_0 taken at time t

$$H_t(g_0, g_0) = \sum_i e^{-\lambda_i t} \phi_i(g_0)^2. \quad (4)$$

The HKS $H_t(g_0, g_0)$, is a point signature that can capture the neighborhood information at point g_0 and scale t , and is isometrically invariant.

2.2 Auto-Encoder

A deep auto-encoder [20], [21] consists of two parts, i.e., encoder and decoder. The encoder, denoted by F , maps the input $h \in \mathcal{R}^{d \times 1}$ to the hidden layer, denoted by $z \in \mathcal{R}^{r \times 1}$, where d is the dimension of the input and r is the number of neurons in the hidden layer. In the deep auto-encoder, a neuron in layer l is connected to all neurons in layer $l + 1$. We denote the weight and bias connecting layers l and $l + 1$ by W^l and b^l , respectively. A non-linear activation function, such as the sigmoid function $\sigma(h) = \frac{1}{1+e^{-h}}$ or tanh function $\sigma(h) = \frac{e^h - e^{-h}}{e^h + e^{-h}}$, is usually used to produce the output at each neuron. The output at layer $l + 1$ can be represented as

$$f_{l+1}(a^l) = \sigma(W^l a^l + b^l), \quad (5)$$

where $f_{l+1}(a^l)$ is the activation function for layer $l + 1$ and a^l is the neurons in layer l . Thus, the encoder $F(h)$ can be represented as

$$F(h) = f_k(f_{k-1}(\dots, f_2(h))). \quad (6)$$

The decoder, denoted by G , maps the hidden layer representation z back to the input h . It is defined as

$$h = f_L(f_{L-1}(\dots, f_{k+1}(z))), \quad (7)$$

where L is the layer number of the auto-encoder. The parameters W and b are the weights and biases of the auto-encoder, where W includes W^1, W^2, \dots, W^{L-1} and b includes b^1, b^2, \dots, b^{L-1} . To optimize parameters W and b , the standard auto-encoder minimizes the following cost function:

$$\begin{aligned} < \hat{W}, \hat{b} > = \underset{W, b}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^M \|h_i - G(F(h_i))\|_2^2 \\ &+ \frac{1}{2} \lambda \sum_{l=1}^{L-1} \|W^l\|_F^2, \end{aligned} \quad (8)$$

where h_i represents the i th training sample, M represents the total number of training samples, and parameter λ is a positive scalar. In Eq. (8), the first term is the reconstruction error and the second term is the regularization term that prevents overfitting. An efficient optimization method can be implemented by the restricted Boltzman machine and

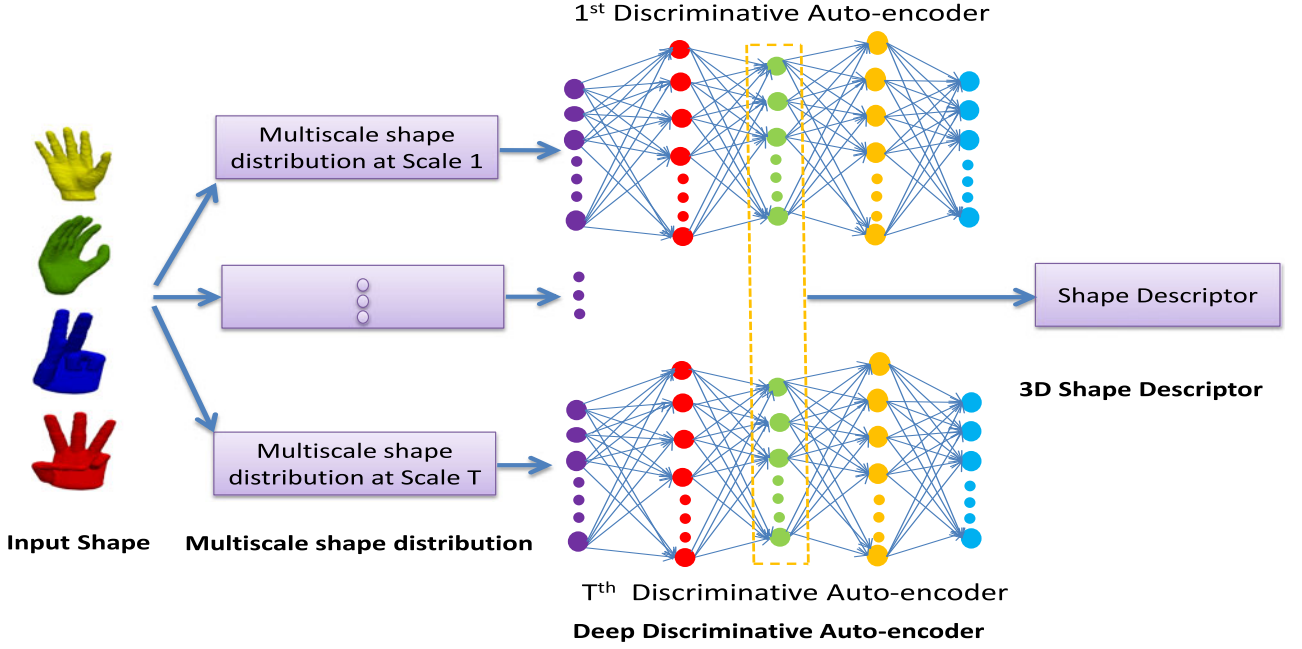


Fig. 1. The framework of the proposed discriminative auto-encoder based shape descriptor.

back-propagation framework. The reader can refer to [20] for more details.

3 PROPOSED APPROACH

In this section, we describe our proposed discriminative auto-encoder based shape descriptor. As depicted in Fig. 1, our proposed framework comprises three components, namely, multiscale shape distribution, discriminative auto-encoder and 3D shape descriptor. In the multiscale shape distribution component, the distributions of heat kernel signatures of shape at different scales are extracted as low-level features and used as input to the discriminative auto-encoder. In the second component, we train a discriminative auto-encoder to learn high-level shape features (embedded in the hidden layer of the discriminative auto-encoder). Finally, the 3D shape descriptor is formed by concatenating the outputs from the hidden layers of the discriminative auto-encoders at different scales.

3.1 Multiscale Shape Distribution

Shape distribution [22] refers to a probabilistic distribution sampled from a shape function describing the 3D model. In this work, we use the HKS as the shape function. The HKS describes the amount of heat that remains at the vertex within a time interval. It is highly related to the curvature of the meshed surface. Therefore, the HKS can characterize the intrinsic geometric structures of neighborhoods of the shape well. It has attractive properties that include invariance to the isometric transformation, robustness against other geometric changes, and multiscale representation with diffusion time [10]. Compared to the voxelization method [17] to represent 3D model, the HKS does not need shape alignment. And unlike the parameterized local surface patch method in [18], it does not need to calculate the complex patch operator by constructing the local geodesic polar coordinates on the surface.

The shape distribution can be formed by using histogram to estimate the probabilistic distribution of the HKSs of the

vertices on the shape. Suppose there are C shape classes, each of which has O samples. We use $y_{i,j}$ to index the j th sample of the i th shape class. For each shape $y_{i,j}$, we extract HKS feature $S_{i,j} \in \mathcal{R}^{N \times T}$, where $S_{i,j} = [S_{i,j}^1, S_{i,j}^2, \dots, S_{i,j}^T]$, $S_{i,j}^t$ is the HKS vector at the t th scale to describe N vertices on shape $y_{i,j}$, $t = 1, 2, \dots, T$, and T is the number of scales. For scale t , we calculate the histogram of $S_{i,j}^t$ to form the shape distribution $h_{i,j}^t$. By considering probabilistic distributions of shape functions derived from the HKS at different scales, a multiscale shape distribution can be obtained. In addition, we normalize the shape distribution, which is centralized by the mean and variance of the shape distributions over all training samples from C classes, namely,

$$h_{i,j}^t = \frac{h_{i,j}^t - \bar{h}^t}{v^t}, \quad (9)$$

where \bar{h}^t and v^t are the mean and variance of all training shape distributions $h_{i,j}^t$.

Fig. 2 shows the multiscale shape distributions of the Centaur and Human models with different poses. From this figure, we can see that the multiscale shape distributions are different for the Centaur and Human shapes. Also, the three centaur models with isometric geometric transformations have consistent multiscale shape distributions. This demonstrates the invariance of the multiscale shape distribution to isometric transformations. For the three human models with structural variations, their multiscale shape distributions can capture their common geometric characteristics despite the inconsistency in their detailed descriptions.

3.2 Discriminative Auto-Encoder

In this section, we propose a discriminative auto-encoder to extract high-level features for 3D shape retrieval. To boost the discriminative power of the hidden layer features, we

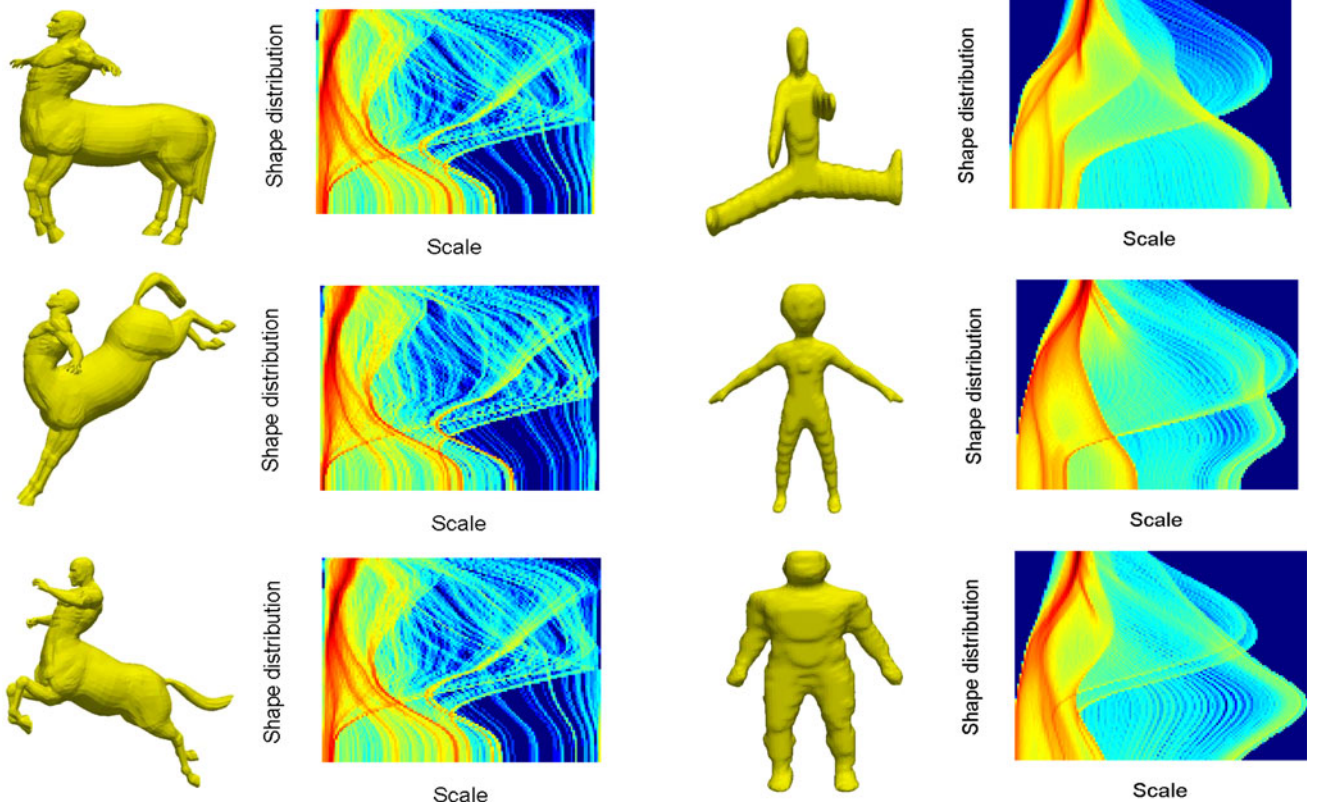


Fig. 2. The multiscale shape distributions of the centaur and human models. The left two columns show the centaur models with isometric transformations and the corresponding multiscale shape distributions. The right two columns show the human models with non-isometric structural variations and the corresponding multiscale shape distributions, respectively.

impose a Fisher discrimination criterion [23] on them. Given the shape distribution input x_i^t of shape class i at scale t , $x_i^t = [h_{i,1}^t, h_{i,2}^t, \dots, h_{i,O}^t]$, we denote by z^t the features in the hidden layer of the auto-encoder. We can write z^t as $z^t = [z_1^t, z_2^t, \dots, z_C^t]$, where $z_i^t = [z_{i,1}^t, z_{i,2}^t, \dots, z_{i,O}^t]$, $z_{i,j}^t$ is the hidden layer feature of the j th sample from class i , for $i = 1, 2, \dots, C$ and $j = 1, 2, \dots, O$. Using the Fisher discrimination criterion, discrimination is achieved by minimizing the within-class scatter of z^t , denoted by $S_w(z^t)$, and maximizing the between-class scatter of z^t , denoted by $S_b(z^t)$. $S_w(z^t)$ and $S_b(z^t)$ are defined as

$$\begin{aligned} S_w(z^t) &= \sum_{i=1}^C \sum_{z_{i,j}^t \in i} (z_{i,j}^t - m_i^t)(z_{i,j}^t - m_i^t)^T \\ S_b(z^t) &= \sum_{i=1}^C n_i (m_i^t - m^t)(m_i^t - m^t)^T, \end{aligned} \quad (10)$$

where m_i^t and m^t are the mean vectors of z_i^t and z^t , respectively, and n_i is the number of samples from class i . We define the discriminative regularization term $tr(S_w(z^t)) - tr(S_b(z^t))$ and incorporate it into the objective function of the discriminative auto-encoder

$$\begin{aligned} J(W^t, b^t) &= \operatorname{argmin}_{W^t, b^t} \sum_{i=1}^C \frac{1}{2} \|x_i^t - G(F(x_i^t))\|_F^2 \\ &+ \frac{1}{2} \lambda \sum_{l=1}^{L-1} \|W^{l,t}\|_F^2 + \frac{1}{2} \gamma (tr(S_w(z^t)) - tr(S_b(z^t))). \end{aligned} \quad (11)$$

For shape distribution $h_{i,j}^t$, we define the following functions:

$$J_0(W^t, b^t, h_{i,j}^t) = \frac{1}{2} \|h_{i,j}^t - G(F(h_{i,j}^t))\|_2^2 \quad (12)$$

$$\begin{aligned} L_0(z_{i,j}^t) &= \frac{1}{2} tr((z_{i,j}^t - m_i^t)(z_{i,j}^t - m_i^t)^T) \\ &- \frac{1}{2} tr((m_i^t - m^t)(m_i^t - m^t)^T). \end{aligned} \quad (13)$$

We adopt the back-propagation method to optimize the objective function Eq. (11). The partial derivatives of the overall cost function $J(W^t, b^t)$ can be computed as

$$\begin{aligned} \frac{\partial J(W^t, b^t)}{\partial W^{l,t}} &= \sum_{i=1}^C \sum_{h_{i,j}^t \in i} \frac{\partial J_0(W^t, b^t, h_{i,j}^t)}{\partial W^{l,t}} + \lambda W^{l,t} \\ &+ \gamma \sum_{i=1}^C \sum_{z_{i,j}^t \in i} \frac{\partial L_0(z_{i,j}^t)}{\partial W^{l,t}} \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{\partial J(W^t, b^t)}{\partial b^{l,t}} &= \sum_{i=1}^C \sum_{h_{i,j}^t \in i} \frac{\partial J_0(W^t, b^t, h_{i,j}^t)}{\partial b^{l,t}} \\ &+ \gamma \sum_{i=1}^C \sum_{z_{i,j}^t \in i} \frac{\partial L_0(z_{i,j}^t)}{\partial b^{l,t}}. \end{aligned} \quad (15)$$

We denote by $\delta_{i,j}^{L,t}$ the error of the output layer L for the input sample $h_{i,j}^t$ in the auto-encoder. For the output layer L , we have

$$\delta_{i,j}^{L,t} = -(\mathbf{h}_{i,j}^t - \mathbf{a}_{i,j}^{L,t}) \bullet \sigma'(\mathbf{u}_{i,j}^{L,t}), \quad (16)$$

where $\mathbf{a}_{i,j}^{L,t}$ is the activation of the output layer for the input sample $\mathbf{h}_{i,j}^t$, $\mathbf{u}_{i,j}^{L,t}$ is the weighted sum of the outputs of layer $L-1$ to the output layer, $\sigma'(\mathbf{u}_{i,j}^{L,t})$ is the derivative of the activation function in the output layer and \bullet denotes the element-wise multiplication. For layers $l = L-1, L-2, \dots, 2$, the error $\delta_{i,j}^{l,t}$ can be recursively obtained by the back-propagation method using the following equation:

$$\delta_{i,j}^{l,t} = ((\mathbf{W}^{l,t})^T \delta_{i,j}^{l+1,t}) \bullet \sigma'(\mathbf{u}_{i,j}^{l,t}). \quad (17)$$

The partial derivatives of the function $J_0(\mathbf{W}^t, \mathbf{b}^t, \mathbf{h}_{i,j}^t)$, can be computed as

$$\begin{aligned} \frac{\partial J_0(\mathbf{W}^t, \mathbf{b}^t, \mathbf{h}_{i,j}^t)}{\partial \mathbf{W}^{l,t}} &= \delta_{i,j}^{l+1,t} (\mathbf{a}_{i,j}^{l,t})^T \\ \frac{\partial J_0(\mathbf{W}^t, \mathbf{b}^t, \mathbf{h}_{i,j}^t)}{\partial \mathbf{b}^{l,t}} &= \delta_{i,j}^{l+1,t}. \end{aligned} \quad (18)$$

Since $\mathbf{z}_{i,j}^t = \sigma(\mathbf{u}_{i,j}^{k,t}) = \sigma(\mathbf{W}^{k-1,t} \mathbf{a}_{i,j}^{k-1,t} + \mathbf{b}^{k-1,t})$, for $l > k-1$, $\frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{W}^{l,t}} = 0$ and $\frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{b}^{l,t}} = 0$. For $l \leq k-1$, $\frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{W}_{p,q}^{l,t}}$ and $\frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{b}_p^{l,t}}$ can be computed as

$$\begin{aligned} \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{W}_{p,q}^{l,t}} &= \frac{\partial \mathbf{z}_{i,j,p}^t}{\partial \mathbf{W}_{p,q}^{l,t}} \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{z}_{i,j,p}^t} = \mathbf{a}_{i,j,q}^{l,t} \sigma'(\mathbf{u}_{i,j}^{l+1,t})_p \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{z}_{i,j,p}^t} \\ \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{b}_p^{l,t}} &= \sigma'(\mathbf{u}_{i,j}^{l+1,t})_p \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{z}_{i,j,p}^t}, \end{aligned} \quad (19)$$

where $\mathbf{W}_{p,q}^{l,t}$ is the weight associated with the connection between unit p in layer l and unit q in layer $l+1$, $\mathbf{b}_p^{l,t}$ is the bias associated with unit p in layer l , $\mathbf{a}_{i,j,q}^{l,t}$ is the q th component of $\mathbf{a}_{i,j}^{l,t}$ and $\mathbf{z}_{i,j,p}^t$ is the p th component of $\mathbf{z}_{i,j}^t$. The partial derivative of $L_0(\mathbf{z}_{i,j}^t)$ with respect to $\mathbf{z}_{i,j,p}^t$ can be obtained as

$$\begin{aligned} \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{z}_{i,j,p}^t} &= \left(1 - \frac{1}{n_i}\right) \left(\mathbf{z}_{i,j,p}^t - m_{i,p}^t\right) \\ &\quad - \left(\frac{1}{n_i} - \frac{1}{\sum n_i}\right) \left(m_{i,p}^t - m_p^t\right), \end{aligned} \quad (20)$$

where $m_{i,p}^t$ and m_p^t are the p th components of \mathbf{m}_i^t and \mathbf{m}^t , respectively.

Therefore, for $l > k-1$, $\frac{\partial J_0(\mathbf{W}^t, \mathbf{b}^t, \mathbf{h}_{i,j}^t)}{\partial \mathbf{W}^{l,t}} + \gamma \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{W}^{l,t}}$ and $\frac{\partial J_0(\mathbf{W}^t, \mathbf{b}^t, \mathbf{h}_{i,j}^t)}{\partial \mathbf{b}^{l,t}} + \gamma \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{b}^{l,t}}$ can be obtained by Eq. (18). For $l \leq k-1$, $\frac{\partial J_0(\mathbf{W}^t, \mathbf{b}^t, \mathbf{h}_{i,j}^t)}{\partial \mathbf{W}^{l,t}} + \gamma \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{W}^{l,t}}$ and $\frac{\partial J_0(\mathbf{W}^t, \mathbf{b}^t, \mathbf{h}_{i,j}^t)}{\partial \mathbf{b}^{l,t}} + \gamma \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{b}^{l,t}}$ can be computed as:

$$\begin{aligned} \frac{\partial J_0(\mathbf{W}^t, \mathbf{b}^t, \mathbf{h}_{i,j}^t)}{\partial \mathbf{W}^{l,t}} + \gamma \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{W}^{l,t}} &= (\delta_{i,j}^{l+1,t} + \gamma \left(1 - \frac{1}{n_i}\right) \\ (\mathbf{z}_{i,j}^t - \mathbf{m}_i^t) - \left(\frac{1}{n_i} - \frac{1}{\sum n_i}\right) (\mathbf{m}_i^t - \mathbf{m}^t)) \bullet \sigma'(\mathbf{u}_{i,j}^{l+1,t})) (\mathbf{a}_{i,j}^{l,t})^T \end{aligned} \quad (21)$$

$$\begin{aligned} \frac{\partial J_0(\mathbf{W}^t, \mathbf{b}^t, \mathbf{h}_{i,j}^t)}{\partial \mathbf{b}^{l,t}} + \gamma \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{b}^{l,t}} &= \delta_{i,j}^{l+1,t} + \gamma \left(1 - \frac{1}{n_i}\right) \\ (\mathbf{z}_{i,j}^t - \mathbf{m}_i^t) - \left(\frac{1}{n_i} - \frac{1}{\sum n_i}\right) (\mathbf{m}_i^t - \mathbf{m}^t)) \bullet \sigma'(\mathbf{u}_{i,j}^{l+1,t}). \end{aligned} \quad (22)$$

Once the partial derivatives of the objective function $J(\mathbf{W}^t, \mathbf{b}^t)$ with respect to \mathbf{W}^t and \mathbf{b}^t are computed, we can employ the conjugate gradient method to obtain \mathbf{W}^t and \mathbf{b}^t . The training algorithm of our proposed discriminative auto-encoder is summarized in Algorithm 1.

Algorithm 1. Training Algorithm of Our Discriminative Auto-Encoder

Input: training set \mathbf{x}_i^t ; the layer size of the auto-encoder; λ ; γ ; learning rate β .

Output: \mathbf{W}^t and \mathbf{b}^t .

Set $\Delta \mathbf{W}^{l,t} = 0$ and $\Delta \mathbf{b}^{l,t} = 0$ for all l .

For all $\mathbf{h}_{i,j}^t$:

1) Compute $\frac{\partial J_0(\mathbf{W}^t, \mathbf{b}^t, \mathbf{h}_{i,j}^t)}{\partial \mathbf{W}^{l,t}} + \gamma \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{W}^{l,t}}$ and $\frac{\partial J_0(\mathbf{W}^t, \mathbf{b}^t, \mathbf{h}_{i,j}^t)}{\partial \mathbf{b}^{l,t}} + \gamma \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{b}^{l,t}}$: $l > k-1$, compute them with Eq. (18); $l \leq k-1$, compute them with Eqs. (21) and (22).

2) Set $\Delta \mathbf{W}^{l,t}$ to $\Delta \mathbf{W}^{l,t} + \frac{\partial J_0(\mathbf{W}^t, \mathbf{b}^t, \mathbf{h}_{i,j}^t)}{\partial \mathbf{W}^{l,t}} + \gamma \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{W}^{l,t}}$.

3) Set $\Delta \mathbf{b}^{l,t}$ to $\Delta \mathbf{b}^{l,t} + \frac{\partial J_0(\mathbf{W}^t, \mathbf{b}^t, \mathbf{h}_{i,j}^t)}{\partial \mathbf{b}^{l,t}} + \gamma \frac{\partial L_0(\mathbf{z}_{i,j}^t)}{\partial \mathbf{b}^{l,t}}$.

Update $\mathbf{W}^{l,t}$ and $\mathbf{b}^{l,t}$: $\mathbf{W}^{l,t} = \mathbf{W}^{l,t} - \beta(\Delta \mathbf{W}^{l,t} + \lambda \mathbf{W}^{l,t})$, $\mathbf{b}^{l,t} = \mathbf{b}^{l,t} - \beta \Delta \mathbf{b}^{l,t}$.

Output $\mathbf{W}^{l,t}$ and $\mathbf{b}^{l,t}$ until the values of $J(\mathbf{W}^t, \mathbf{b}^t)$ in successive iterations are close enough or the maximum number of iterations is reached.

3.3 3D Shape Descriptor

We use the outputs from the hidden layer of the discriminative auto-encoder to form the shape descriptor. In order to characterize the intrinsic structure of the shape effectively, we train a discriminative auto-encoder at each scale by using a set of training shape distributions, $\mathbf{x}_1^t, \mathbf{x}_2^t, \dots, \mathbf{x}_C^t$, $t = 1, 2, \dots, T$. After training, we concatenate the outputs from the hidden layers at all scales to form the shape descriptor. Denote the t th encoder of the multiple discriminative auto-encoders by F^t . The shape descriptor of the j th shape from class i , $\alpha_{i,j}$, can be represented as

$$\alpha_{i,j} = [F^1(\mathbf{h}_{i,j}^1); F^2(\mathbf{h}_{i,j}^2); \dots; F^T(\mathbf{h}_{i,j}^T)]. \quad (23)$$

4 EXPERIMENTAL RESULTS

In this section, we evaluate our proposed shape descriptor and compare it to state-of-the-art methods on four benchmark datasets: McGill shape dataset [24], SHREC'10 Shape-Google dataset [13], SHREC'14 Human dataset [25] and SHREC'14 Large Scale Comprehensive Retrieval Track Benchmark (SHREC'14 LSCRTB) dataset [26].

4.1 Experimental Settings

We set time unit $\tau = 0.01$ and take 101 sampled time values (i.e., 101 scales) for the computation of HKS. For multiscale shape distribution, 128 bins are used to form the shape distribution at each scale, which results in a

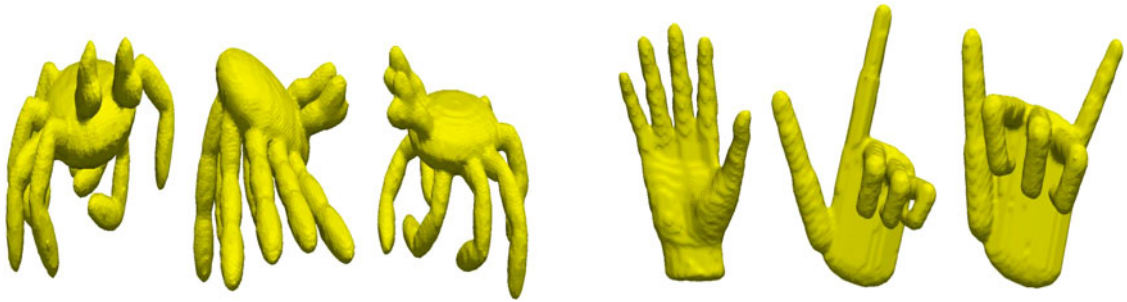


Fig. 3. Example shapes in the McGill dataset. The left three figures show the crab shapes while the right three figures show the Hand shapes with non-rigid transformations.

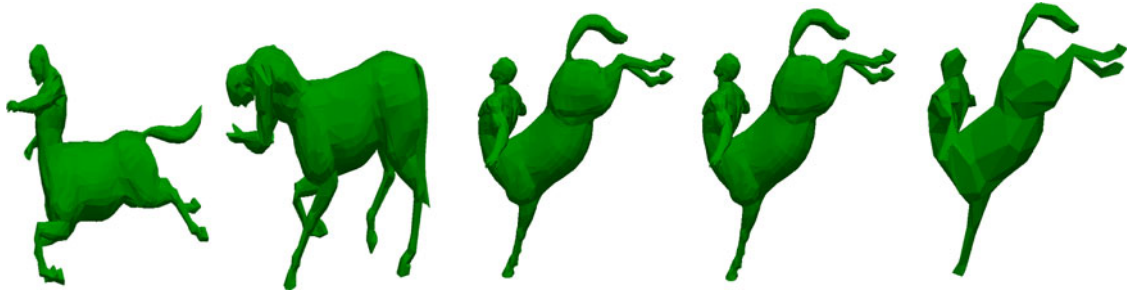


Fig. 4. Example shapes with different transformations in the SHREC'10 ShapeGoogle dataset. From left to right, the centaur shapes with isometry, isometry+topology, topology, partiality and triangulation transformations are shown.

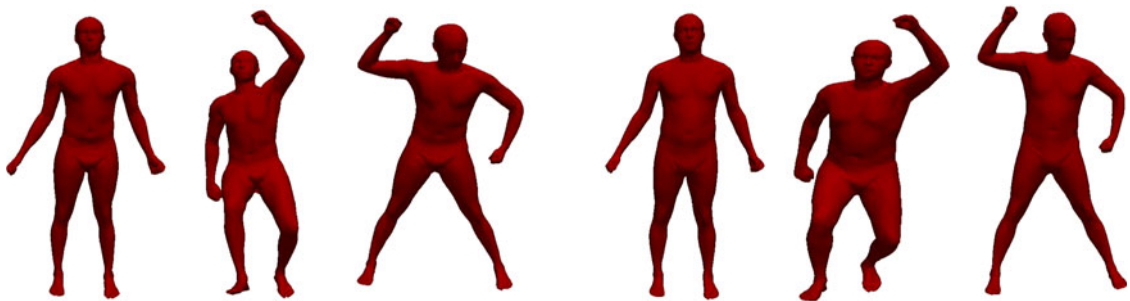


Fig. 5. Two human shapes with different poses in the SHREC'14 human dataset. The left three figures show shapes with pose changes from one person while the right three figures show shapes with different poses from another person.

128-dimensional input for the discriminative auto-encoder. We train an auto-encoder with an encoder of layer size 128-1000-500-250-30 and a symmetric decoder of the same layer size. In Eq. (11), λ and γ are set to 0.001, respectively.

In the McGill 3D shape dataset [24], there are 255 3D meshes with significant part articulations. They come from 10 classes: ant, crab, spectacle, hand, human, octopus, plier, snake, spider, and teddy-bear. Each class contains 3D shapes with large pose changes, which makes the McGill 3D shape dataset very challenging. Fig. 3 shows examples from the McGill shape dataset.

The SHREC'10 ShapeGoogle dataset [13] contains 1,184 synthetic shapes, including 715 shapes from 13 classes generated by five simulated transformations: isometry, topology, isometry+topology, partiality and triangulation, and 456 unrelated shapes. Following the setting in [16], to make the dataset more challenging, all shapes are remeshed to have the same vertices and samples having the same attribute are grouped into the same class. Fig. 4 shows examples from the ShapeGoogle dataset.

The SHREC'14 Human dataset [25] contains two subsets. The first sub-dataset contains 15 synthetic human

models and each model has 20 different poses. The second subset consists of 40 scanned human models, with 10 different poses for each model. Following the setting in [16], all human shapes are re-scaled to 4,500 triangles. The SHREC'14 Human dataset is an extremely challenging one because all human shapes share similar geometry information. Different poses and similar geometric structures will lead to large within-class variations and small inter-class variations. Fig. 5 shows two human shapes with different poses from the SHREC'14 Human dataset.

The SHREC'14 LSCRTB dataset [26] has 8,987 3D shape models from 171 classes. It is a large-scale 3D shape dataset where the shapes are from eight different 3D shape datasets, including the generic models, articulated models, architecture models and CAD models, etc. The average number of models in each class is 53. Most of the shapes are generic models such as bicycle, book, armchair. Moreover, the generic shape models of the same class are not deformed by a template. For example, as shown in Fig. 6, for the armchair model, there are different kinds of armchairs in this dataset.



Fig. 6. Different kinds of armchair shapes in the SHREC'14 LSCRTB dataset.

4.2 Evaluation of the Proposed 3D Shape Descriptor

We evaluate the effectiveness of our proposed shape descriptor on the McGill benchmark dataset [24]. To compare performance, we also generate test results using the multiscale shape distribution descriptor (without auto-encoder) and using the auto-encoder but without the Fisher regularization term. In addition, we investigate the performance of the proposed shape descriptor in terms of robustness to deformations and noise.

4.2.1 Comparison to Multiscale Shape Distribution Descriptor

In our proposed method, we use the multiscale shape distribution as input to the discriminative auto-encoder. Learning deep features from the multiscale shape distribution with the discriminative auto-encoder can be viewed as extracting high-level features from the multiscale shape distribution. To reduce the dimension of the multiscale shape distribution from 101 scales, we concatenate the shape distributions from 26 diffusion time samples to form a 3,328-dimensional feature vector. For a fair comparison, we use 26 discriminative auto-encoders to form the shape descriptor. With a hidden layer size of 30 for each auto-encoder, a 780-dimensional shape descriptor is formed to represent the shape. Fig. 7 shows the precision-recall curves for the multiscale shape distribution descriptor and the proposed shape descriptor. As can be seen in this figure, the proposed shape descriptor with a lower dimension has significantly better retrieval performance.

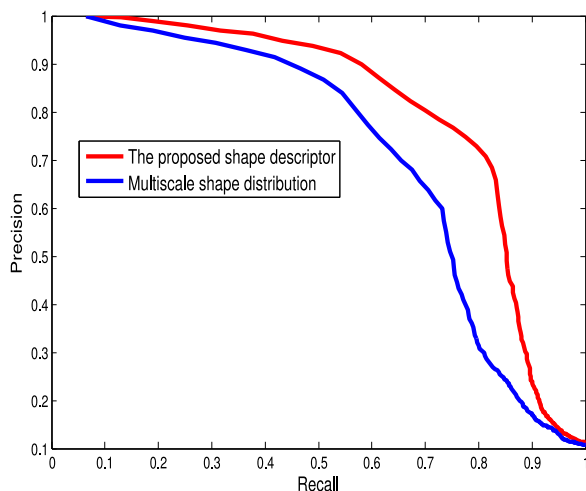


Fig. 7. The precision-recall curves for the multiscale shape distribution descriptor and the proposed shape descriptor on the McGill shape dataset.

4.2.2 Comparison to Standard Auto-Encoder

In order to demonstrate the effectiveness of the proposed discriminative auto-encoder, we also compare the proposed shape descriptor to the shape descriptor obtained by employing a standard auto-encoder without the Fisher discrimination term. We concatenate 101 auto-encoders to form both shape descriptors of the same dimension. Fig. 8 shows the precision-recall curves for the two shape descriptors. One can see that the proposed shape descriptor with the discriminative auto-encoder performs better than the shape descriptor obtained by using the standard auto-encoder without the Fisher discrimination term. It implies that by imposing the Fisher discrimination constraint on the hidden layers the learned shape descriptor can reduce within-class variations and increase between-class variations, therefore improving the retrieval performance.

4.2.3 Robustness to Deformations and Noise

A good shape descriptor should be robust to pose changes and noise corruptions. We evaluate robustness of the proposed shape descriptor against pose changes and noise. We choose Teddy-bear and Human models with different poses from the McGill dataset [24] in our experiment. The shape descriptors of the deformed shapes are illustrated in Fig. 9. From the figure, we can see that the descriptors for the model with different poses are very similar. On the other hand, the shape descriptors for different models are distinctive. This shows that the hidden-layer features in the

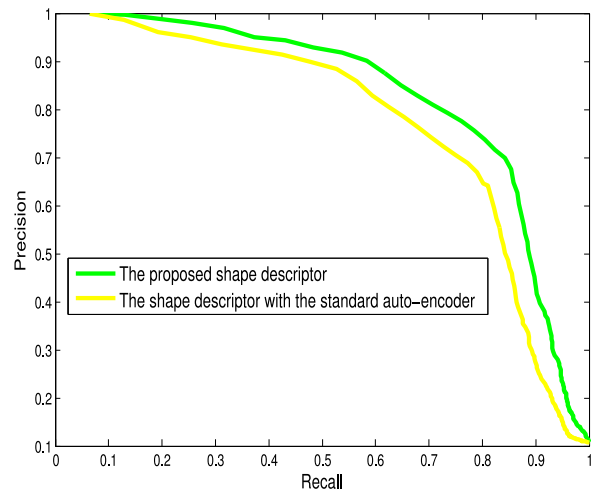


Fig. 8. The precision-recall curves for the shape descriptor using standard auto-encoder (without the Fisher discrimination criterion) and the proposed shape descriptor on the McGill dataset.

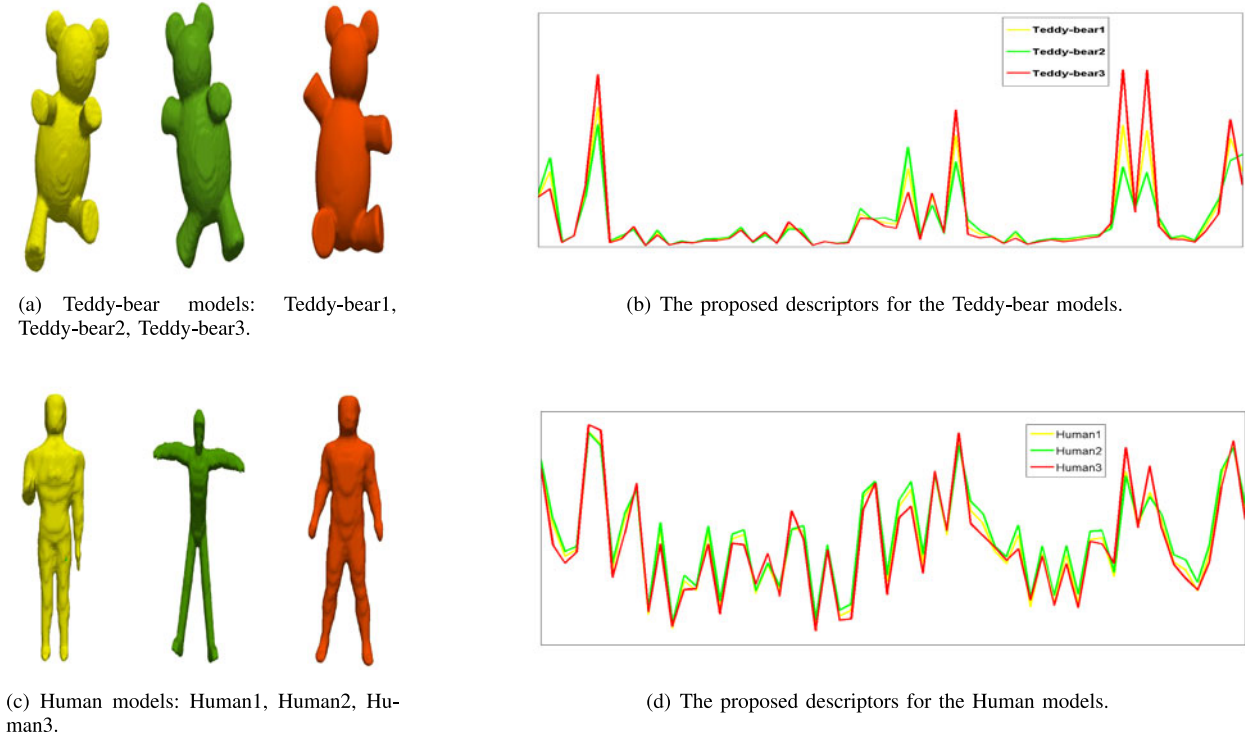


Fig. 9. The proposed descriptors for the teddy-bear model and the human model.

proposed discriminative auto-encoder have small within-class variations but large between-class variations.

By perturbing the vertices of the mesh with various levels of numerical noise, we also demonstrate that the proposed shape descriptor is robust to noise. The noise, represented as a three-dimensional vector, is randomly generated from a multivariate normal distribution, $N_3(\mu, R \times \Sigma)$,

where μ is the three-dimensional mean vector of the coordinates of the vertices, Σ is the 3×3 covariance matrix of the vertices, and R denotes the ratio between the variance of noise and the variance of the coordinates of the vertices.

Fig. 10 shows the clean Crab and Hand models, and their noisy models. In (a) and (c), the green and red noisy models are generated by noise of $R = 0.01$ and $R = 0.04$, respectively.

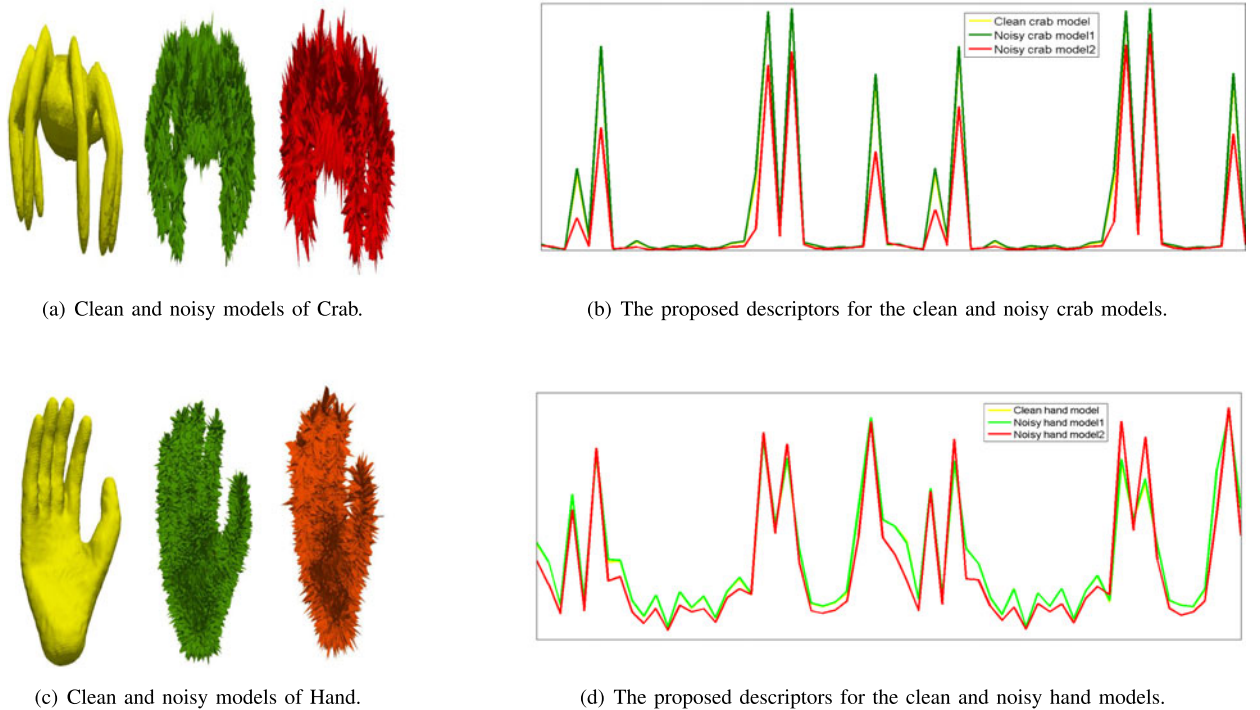


Fig. 10. The proposed descriptors for the clean and noisy models of crab and hand. In (a) and (c), the green and red shapes were generated with noise of $R = 0.01$ and $R = 0.04$, respectively. In (b) and (d), the descriptors for the shapes are represented by the yellow, green and red curves, corresponding to the clean model, the noisy model with noise of $R = 0.01$, and the noisy model with noise of $R = 0.04$, respectively.

TABLE 1
Retrieval Results on the McGill Dataset

Methods	NN	1-Tier	2-Tier	DCG
Covariance method [30]	0.977	0.732	0.818	0.937
Graph-based method [29]	0.976	0.741	0.911	0.933
PCA-based VLAT [28]	0.969	0.658	0.781	0.894
Hybrid BOW [27]	0.957	0.635	0.790	0.886
Hybrid 2D/3D [14]	0.925	0.557	0.698	0.850
CBoFHKS [15]	0.901	0.778	0.876	0.891
DASD	0.988	0.782	0.834	0.955

Particularly, in the noisy model with noise of $R = 0.04$, geometric structures of the mesh have moderately deteriorated. As shown in Fig. 10, the variations in the proposed shape descriptors for the clean and noisy models (plotted with yellow, green and red curves) are small. Since the level of noise for $R = 0.01$ is low, the yellow and green curves basically overlap. The experimental results demonstrate that the proposed shape descriptor is robust to noise.

4.3 Comparison with State-of-the-Art Methods

We tested our proposed shape descriptor on four benchmark datasets – McGill [24], SHREC'10 ShapeGoogle [13], SHREC'14 Human [25] and SHREC'14 LSCRTB [26]– and compare the results with several state-of-the-art methods. Each shape is represented by a compact 1D shape descriptor and L_2 norm is used to compute the distance between two shape descriptors in our retrieval experiments.

4.3.1 McGill Shape Dataset

For the McGill 3D shape dataset, we compare our method to the Hybrid BOW method [27], the PCA-based VLAT method [28], the graph-based method [29], the hybrid 2D/3D approach [14], the covariance descriptor [30] and the CompactBoFHKS method (CBoFHKS) [15]. We denote our proposed discriminative auto-encoder-based shape descriptor by DASD. In the CompactBoFHKS method, 21 scales are chosen every five scales from 101 scales for the HKS. The size of the bag-of-words is set to 64. In our experiments, 10 shapes per class are randomly chosen to train the discriminative auto-encoder and the remaining shapes in each class are used for testing. We use different performance measures in our evaluation, namely, Nearest Neighbor (NN), the First Tier (1-Tier), the Second Tier (2-Tier) and the Discounted Cumulative Gain (DCG). The retrieval performance of our method and other state-of-the-art methods is illustrated in Table 1. From this table, we can see that the proposed method achieves the best performance with the NN, 1-Tier, and DCG measures. There are large non-rigid deformations with the objects in the McGill shape dataset, which results in large within-class variations. Nonetheless, due to the discriminative feature representation of our method, as shown earlier in Fig. 9, DASD is robust to large non-rigid deformations.

4.3.2 SHREC'10 ShapeGoogle Dataset

We also compared our proposed DASD method to the bag-of-feature (BOF) descriptors with standard vector quantization (VQ) [13], unsupervised dictionary learning (UDL) [16], supervised dictionary learning (SDL) [16] and the

TABLE 2
Retrieval Results (Mean Average Precision)
on the SHREC'10 ShapeGoogle Dataset

Transformation	VQ [13]	UDL [16]	SDL [16]	CBoFHKS [15]	DASD
Isometry	0.988	0.977	0.994	0.966	0.998
Topology	1.000	1.000	1.000	1.000	0.996
Isometry+Topology	0.933	0.934	0.956	0.915	0.982
Partiality	0.947	0.948	0.951	0.968	0.973
Triangulation	0.954	0.950	0.955	0.891	0.955

CompactBoFHKS method (CBoFHKS) [15] on the SHREC'10 ShapeGoogle dataset [13]. We used the mean average precision criterion in our evaluation. Evaluation results are summarized in Table 2. From this table, one can see that our proposed DASD is superior to the BOF descriptors with standard VQ, UDL, SDL and CBoFHKS in the cases of isometry, isometry+topology, partiality and triangulation. Since deep auto-encoder has good ability to model non-linearity, DASD can characterize the low-dimensional manifold embedded in the high-dimensional shape space better and therefore achieve better performance. For example, in the cases of isometry+topology and partiality, the supervised dictionary learning based shape descriptor can achieve accuracies of 0.956 and 0.951, while our proposed DASD can achieve accuracies of 0.982 and 0.973, respectively.

4.3.3 SHREC'14 Human Dataset

For the synthetic and scanned human sub-datasets, we compare the proposed DASD method to several recently proposed shape retrieval methods: Histogram of area projection transform (HAPT) [31], intrinsic pyramid matching (ISPM) [32], reduced Bi-harmonic distance matrix (RBiHDM) [33], deep belief network (DBN) [25], the bag-of-feature descriptors with standard vector quantization (VQ) [13], and unsupervised dictionary learning (UDL) [16]. For the synthetic sub-dataset, 11 shapes in each class are used to train the discriminative auto-encoder and the remaining shapes in each class are used for testing. For the scanned sub-dataset, six shapes in each class are used for training and the remaining shapes are used for testing. The mean average precisions are reported in Table 3. The scanned sub-dataset is an extremely challenging dataset, as can be seen in the table, our method can achieve better performance. For the synthetic sub-dataset, the mean average precision of our method is lower than the ISPM method [32], and slightly lower than the DBN [25] and UDL [16] methods.

TABLE 3
Retrieval Results (Mean Average Precision)
on the SHREC'14 Human Dataset

Method	Synthetic model	Scanned model
HAPT [31]	0.817	0.637
ISPM [32]	0.92	0.258
RBiHDM [33]	0.642	0.640
DBN [25]	0.842	0.304
VQ [13]	0.813	0.514
UDL [16]	0.842	0.523
DASD	0.823	0.657

TABLE 4
Retrieval Results on the SHREC'14 LSCRTB Dataset

Method	NN	1-Tier	2-Tier	E	DCG
CSLBP [26]	0.840	0.353	0.452	0.197	0.736
HSR-DE [26]	0.837	0.381	0.490	0.203	0.752
KVLAD [26]	0.605	0.413	0.546	0.214	0.746
DBNAA_DERE [26]	0.817	0.355	0.464	0.188	0.731
BF-DSIFT [26]	0.824	0.378	0.492	0.201	0.756
VM-1SIFT [26]	0.732	0.282	0.380	0.158	0.688
ZFDR [26]	0.838	0.386	0.501	0.209	0.757
DBSVC [26]	0.868	0.438	0.563	0.234	0.790
DASD	0.897	0.401	0.503	0.243	0.774

4.3.4 SHREC'14 LSCRTB Dataset

For the SHREC'14 LSCRTB dataset, we compared our proposed DASD method to the state-of-the-art methods [26]: CSLBP, HSR-DE, KVLAD, DBNAA_DERE, BF-DSIFT, VM-1SIFT, ZFDR and DBSVC. We used the NN, 1-Tier, 2-Tier, E-Measures (E) and DCG for performance evaluation. The evaluation results are listed in Table 4. From this table, one can see that the proposed DASD method can obtain the best performance with the NN and E measures. In terms of the 1-Tier and 2-Tier measures, our proposed DASD method is lower than the DBSVC method. Nonetheless, our proposed DASD method is comparable to the CSLBP, HSR-DE, DBNAA_DERE, BF-DSIFT, VM-1SIFT and ZFDR methods. Note that we did not compare our method to the LCDR-DBSRC method in [26]. This is because that we only calculated the Euclidean distance between the learned 3D descriptors while in the LCDR-DBSRC method the manifold ranking algorithm [34] is employed on the learned 3D shape features.

In the compared methods, the CSLBP, HSR-DE, DBNAA_DERE and ZFDR methods use a combination of hand-crafted features while the KVLAD, BF-DSIFT, VM-1SIFT and DBSVC methods use learned features. In all these methods, the features are extracted on the rendered images by projecting the 3D shape from different viewpoints. This type of approaches suffer from the following two drawbacks: (1) the rendered images are sensitive to the deformation of the 3D model; (2) pre-alignment of 3D models in the same category is usually employed prior to 3D surface parameterization in order to normalize the models for learning. However, normalization of 3D models may be difficult due to different types of structural variations.

Nonetheless, in our proposed DASD method, based on the HKS, the formed multiscale shape distribution is a statistic of the local geometric structure information of the shape, which is robust to the deformations. Then, by imposing the Fisher discrimination criterion on the neurons in the hidden layer of the neural network, the proposed discriminative auto-encoder can minimize within-class variations and maximize between-class variations of the shape features. Furthermore, the developed discriminative auto-encoder can improve robustness of the shape features to large deformations. And it does not need shape alignment.

5 CONCLUSIONS

In this paper, we have proposed a novel deep shape descriptor for 3D shape retrieval. We first computed the multiscale shape distribution features and then trained a set

of discriminative auto-encoders to extract high-level shape features at different scales. By imposing the Fisher discrimination criterion on the hidden layers of the auto-encoders, our feature representation results in small within-class scatter and large between-class scatter. Our shape descriptor is formed by concatenating the high-level features from different scales. Experimental results demonstrated the superior performance of our proposed descriptor.

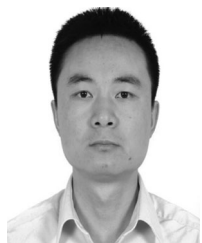
ACKNOWLEDGMENTS

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] R. Osada, T. A. Funkhouser, B. Chazelle, and D. P. Dobkin, "Matching 3D models with shape distributions," in *Proc. Int. Conf. Shape Modeling Appl.*, 2001, pp. 154–166.
- [2] M. Novotni and R. Klein, "Shape retrieval using 3D Zernike descriptors," *Comput. Aided Des.*, vol. 36, no. 11, pp. 1047–1062, 2004.
- [3] D. V. Vranic, D. Saupe, and J. Richter, "Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics," in *Proc. IEEE 4th Workshop Multimedia Signal Process.*, 2001, pp. 293–298.
- [4] D. Saupe and D. V. Vranic, "3D model retrieval with spherical harmonics and moments," in *Proc. 23rd DAGM Symp. Pattern Recog.*, 2001, pp. 392–397.
- [5] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 223–232, 2003.
- [6] V. Jain and H. Zhang, "A spectral approach to shape-based retrieval of articulated 3D models," *Comput. Aided Des.*, vol. 39, no. 5, pp. 398–407, 2007.
- [7] J. Assfalg, M. Bertini, A. D. Bimbo, and P. Pala, "Content-based retrieval of 3D objects using spin image signatures," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 589–599, Apr. 2007.
- [8] M. Krtgen, M. Novotni, and R. Klein, "3D shape matching with 3D shape contexts," presented at the 7th Central Eur. Seminar Comput. Graph., Budmerice, Slovakia, 2003.
- [9] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud, "Surface feature detection and description with applications to mesh matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 373–380.
- [10] J. Sun, M. Ovsjanikov, and L. J. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.
- [11] A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro, "A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching," *Int. J. Comput. Vis.*, vol. 89, pp. 266–286, 2010.
- [12] R. M. Rustamov, "Laplace-Beltrami eigenfunctions for deformation invariant shape representation," in *Proc. Eurographics Symp. Geometry Process.*, 2007, pp. 225–233.
- [13] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov, "Shape Google: Geometric words and expressions for invariant shape retrieval," *ACM Trans. Graph.*, vol. 30, no. 1, pp. 1:1–1:20, 2011.
- [14] G. Lavoué, "Combination of bag-of-words descriptors for robust partial shape retrieval," *Visual Comput.*, vol. 28, no. 9, pp. 931–942, 2012.
- [15] 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. 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. Unal, and R. C. Wilson, "SHREC'15 track: Non-rigid 3D shape retrieval," in *Proc. Eurographics Workshop 3D Object Retrieval*, 2015, pp. 107–120.
- [16] R. Litman, A. M. Bronstein, M. M. Bronstein, and U. Castellani, "Supervised learning of bag-of-features shape descriptors using sparse coding," *Comput. Graph. Forum*, vol. 33, no. 5, pp. 127–136, 2014.

- [17] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao "3D shapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1912–1920.
- [18] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, "Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks," *Comput. Graph. Forum*, vol. 34, no. 5, pp. 13–23, 2015.
- [19] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on Riemannian manifolds," in *Proc. IEEE Workshop 3D Representation Recognition*, 2015, pp. 37–45.
- [20] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [21] Y. Bengio, "Learning deep architectures for AI," *Foundations Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
- [22] R. Osada, T. Funkhouser, B. Chazelle, and D. Dokin, "Shape distributions," *ACM Trans. Graph.*, vol. 33, pp. 133–154, 2002.
- [23] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: Wiley, 2001.
- [24] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. J. Dickinson, "Retrieving articulated 3D models using medial surfaces," *Mach. Vision Appl.*, vol. 19, no. 4, pp. 261–275, 2008.
- [25] D. Pickup, X. Sun, P. L. Rosin, R. R. Martin, Z. Cheng, Z. Lian, M. Aono, A. B. Hamza, A. M. Bronstein, M. 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 *Proc. Eurographics Workshop 3D Object Retrieval*, 2014, pp. 101–110.
- [26] 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 3D shape retrieval," in *Proc. Eurographics Workshop 3D Object Retrieval*, 2014, pp. 131–140.
- [27] P. Papadakis, I. Pratikakis, T. Theoharis, G. Passalis, and S. J. Perantonis, "3D object retrieval using an efficient and compact hybrid shape descriptor," in *Proc. Eurographics Workshop 3D Object Retrieval*, 2008, pp. 9–16.
- [28] H. Tabia, D. Picard, H. Laga, and P. H. Gosselin, "Compact vectors of locally aggregated tensors for 3D shape retrieval," in *Proc. Eurographics Workshop 3D Object Retrieval*, 2013, pp. 17–24.
- [29] 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 *Proc. Eurographics Workshop 3D Object Retrieval*, 2009, pp. 29–36.
- [30] H. Tabia, H. Laga, D. Picard, and P. H. Gosselin, "Covariance descriptors for 3D shape matching and retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 4185–4192.
- [31] A. Giachetti and C. Lovato, "Radial symmetry detection and shape characterization with the multiscale area projection transform," *Comput. Graph. Forum*, vol. 31, no. 5, pp. 1669–1678, 2012.
- [32] C. Li and A. B. Hamza, "A multiresolution descriptor for deformable 3D shape retrieval," *Vis. Comput.*, vol. 29, no. 6–8, pp. 513–524, 2013.
- [33] J. Ye, Z. Yan, and Y. Yu, "Fast nonrigid 3D retrieval using model space transform," in *Proc. Int. Conf. Multimedia Retrieval*, 2013, pp. 121–126.
- [34] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," in *Proc. Advances Neural Inf. Process. Syst.*, 2003, pp. 169–176.



Jin Xie received the PhD degree from the Department of Computing, Hong Kong Polytechnic University. He is a postdoctoral associate with 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 the master's degree from Fudan University, China. He is working toward the PhD degree in the Department of Computer Science and Engineering, 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 PhD degree in the Visual Information Engineering Group from the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, United Kingdom, in 2011 and 2015, respectively. He is currently a post-doctoral associate with 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 more than 10 papers in well-known journals/conferences such as the *International Journal of Computer Vision*, the *IEEE Transactions on Neural Networks and Learning Systems*, *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 the *IEEE Transactions on Cybernetics*.



Edward K. Wong received the PhD degree in electrical engineering from Purdue University. He is currently an associate professor and the director of MS CS program in the Department of Computer Science and Engineering, New York University Tandon School of Engineering. His research interests lie in the general areas of computer vision, pattern recognition, and machine learning. His current research focus is on developing novel machine-learning-based techniques for video surveillance applications. He had previously worked on funded projects in document image analysis and security, video scene segmentation and classification, fingerprint verification, morphological image processing, infrared target classification, three-dimensional object recognition, pavement image analysis, and optical character recognition, among others. He had published extensively in image processing and multimedia conferences and journals. He is currently an associate editor of two international journals in multimedia and security, and he had served on the organizing committee and technical program committee of several major IEEE and ACM technical conferences in image processing and multimedia.



Yi Fang received the PhD degree from Purdue University with research focus on computer graphics and vision. Upon one year industry experience as a research intern in Siemens, Princeton, New Jersey and a senior research scientist in Riverain Technologies, Dayton, Ohio, and a half-year academic experience as a senior staff scientist in the 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.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.