DeepShape: Deep Learned Shape Descriptor for 3D Shape Matching and Retrieval

Jin Xie†, Yi Fang‡, Fan Zhu†, and Edward Wong‡

†Department of Electrical and Computer Engineering, New York University Abu Dhabi
‡Polytechnic School of Engineering, New York University
{jin.xie, yfang, fan.zhu, ewong}@nyu.edu

Abstract

Complex geometric structural variations of 3D model usually pose great challenges in 3D shape matching and retrieval. In this paper, we propose a high-level shape feature learning scheme to extract features that are insensitive to deformations via a novel discriminative deep auto-encoder. First, a multiscale shape distribution is developed for use as input to the auto-encoder. Then, by imposing the Fisher discrimination criterion on the neurons in the hidden layer, we developed a novel discriminative deep auto-encoder for shape feature learning. Finally, the neurons in the hidden layers from multiple discriminative auto-encoders are concatenated to form a shape descriptor for 3D shape matching and retrieval. The proposed method is evaluated on the representative datasets that contain 3D models with large geometric variations, i.e., McGill and SHREC’10 ShapeGoogle datasets. Experimental results on the benchmark datasets demonstrate the effectiveness of the proposed method for 3D shape matching and retrieval.

1. Introduction

Nowadays there is an explosive growth of 3D meshed surface models in a variety of fields, such as engineering, entertainment and medical imaging [23, 20, 16, 10, 9, 6]. Due to the data-richness of 3D models, shape retrieval for 3D model searching, understanding and analyzing has been receiving 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 preferably that a shape descriptor is deformation-insensitive and invariant to different classes of transformations. Moreover, the shape descriptor should be insensitive to both topological and numerical noise. Once the shape descriptor is formed, the similarity between two shapes is determined for retrieval.

Shape descriptors for shape matching and retrieval have been extensively studied in the geometry community [30, 14, 12, 31, 25]. In the past decades, plenty of shape descriptors have been proposed, such as the $D_2$ shape distribution [12], statistical moments of the model [31, 24], Fourier descriptor [8] and Eigenvalue Descriptor (EVD) [15]. Although these shape descriptors can represent the shape effectively, they are either sensitive to non-rigid transformation or topological changes. To be invariant to isometric transformation, local geometric features are extracted to represent the shape, such as spin images [2], shape context [3] and mesh HOG [32]. However, they are sensitive to local geometric noise and they do not capture the global structure of the shape well.

Apart from the earlier shape descriptors, another popular approaches to shape retrieval are diffusion based methods [27, 7, 23]. Based on the Laplace-Beltrami operator, the global point signature (GPS) [23] was proposed to represent shape. 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 of scaled eigenfunctions of the Laplace-Beltrami operator evaluated at the vertex. The high dimensional vector is called GPS. Another widely used shape signature is heat kernel signature (HKS) [27], where Sun et al. proposed to use the diagonal of the heat kernel as a local descriptor to represent shape. HKS is invariant to isometric deformations, insensitive to the small perturbations on the surface. Both GPS and HKS are point based signatures, that characterize each vertex on the meshed surface by using a vector.

In the aforementioned methods, the shape descriptors are hand-crafted rather than learned from a set of training shapes. In [5], the authors applied the bag-of-features (BOF) paradigm to learn the shape descriptor. The dictio-
2.1. Heat Kernel Signature

The 3D model is represented as a graph $G = (V, E, W)$, where $V$ is the set of vertices, $E$ is the set of edges and $W$ is the set of weights of edges. Given a graph constructed by connecting pairs of data points with weighted edges, the heat kernel $H_t(x, y)$ measures the heat flow across a graph, which is defined to the amount of the heat passing from the vertex $x$ to the vertex $y$ within a certain amount of time. The heat flow across the surface is governed by the heat equation. Provided that there is an initial Dirac delta distribution on meshed surface at $t = 0$, the heat kernel provides the fundamental solution of the heat equation, which is associated with the Laplace-Beltrami operator $L$ by:

$$\frac{\partial H_t}{\partial t} = -LH_t$$  \hspace{1cm} (1)

where $H_t$ denotes the heat kernel and $t$ is the diffusion time. The solution of Eq. (1) can be obtained by the eigenfunction expansion by the Laplace-Beltrami operator described below:

$$H_t = \exp(-tL).$$  \hspace{1cm} (2)

By the spectral theorem, the heat kernel can be further expressed as follows:

$$H_t(x, y) = \sum_i e^{-\lambda_i t} \phi_i(x) \phi_i(y)$$  \hspace{1cm} (3)

where $\lambda_i$ is the $i^{th}$ eigenvalue of the Laplacian, $\phi_i$ is the $i^{th}$ eigenfunction, and $x$ and $y$ denotes the vertex $x$ and $y$, respectively. Heat kernel signature (HKS) $[21]$ of the vertex $x$ at time $t$, $S^t_x$, is defined as the diagonal of the heat kernel of the vertex $x$ taken at time $t$:

$$S^t_x = H_t(x, x) = \sum_{i=0} e^{-\lambda_i t} \phi_i(x)^2.$$  \hspace{1cm} (4)

HKS, $S^t_x$, as a point signature, can capture information of the neighborhood of the point $x$ on the shape at the scale $t$.

2.2. Auto-encoder

An auto-encoder neural network $[13,24]$ usually consists of two parts, i.e., encoder and decoder. The encoder, denoted by $F$, maps the input $x \in \mathcal{R}^{d \times 1}$ to the hidden layer representation, 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 auto-encoder neural network, one neuron in the layer $l$ is connected to all the neurons in the layer $l + 1$. We denote the weight and bias connecting the layer $l$ and the layer $l + 1$ by $W^l$ and $b^l$, respectively. The output of the layer is called the activation function. Usually, the activation function is non-linear, such as sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ or tanh function $\sigma(x) = \frac{e^x-e^{-x}}{e^x+e^{-x}}$. Therefore, the output of the layer $l + 1$ is:

$$f_{i+1}(a^l) = \sigma(W^l a^l + b^l)$$  \hspace{1cm} (5)

where $f_{i+1}(a^l)$ is the activation function in the layer $l + 1$ and $a^l$ is the neurons in the layer $l$. Thus, the encoder $F(x)$ of $k$ hidden layers can be represented as follows:

$$F(x) = f_k(f_{k-1}(\cdots, f_2(x))).$$  \hspace{1cm} (6)

The decoder, denoted by $G$, maps the hidden layer representation $z$ back to the input $x$. It is defined:

$$x = f_L(f_{L-1}(\cdots, f_{k+1}(z)))$$  \hspace{1cm} (7)

where $L$ is the layer number of the auto-encoder neural network. The matrices $W$ and $b$ contain the weights and biases of all layers in the auto-encoder, respectively, where $W = \{W^1, W^2, \ldots, W^{L-1}\}$ and $b = \{b^1, b^2, \ldots, b^{L-1}\}$. To optimize the parameters $W$ and $b$, the standard auto-encoder minimizes the following cost function:

$$<W, b> = \text{argmin}_{W, b} \frac{1}{2} \sum_{i=1}^{N} ||x_i - G(F(x_i))||_2^2 + \frac{1}{2} \lambda \sum_{i=1}^{L-1} ||W^i||_F^2$$

where $x_i$ represents the $i^{th}$ training samples, $N$ is the total number of training samples, and parameter $\lambda$ 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 backpropagation framework. The reader can see [13] for more details.

3. Shape descriptor based on discriminative auto-encoder

We detail the proposed framework of the discriminative auto-encoder based shape descriptor, which comprises three components, namely, multiscale shape distribution, discriminative auto-encoder and 3D shape descriptor. Fig. 1 shows the proposed framework. In the multiscale shape distribution component, the distributions of heat kernel signatures of shape at different scales are extracted as a low-level feature for use as input to the discriminative auto-encoder. Then we train a discriminative auto-encoder to learn a high level feature embedded in the hidden layer of the discriminative auto-encoder component. In the 3D shape descriptor component, we form a descriptor from all hidden layer representations of the multiple discriminative auto-encoders.

3.1. Multiscale shape distribution

Shape distribution [20] refers to a probability distribution sampled from a shape function describing the 3D model. We can consider HKS at each scale as a shape function defined on the surface of a 3D model. Then the shape distribution can be defined as the probability distribution of the heat kernel signed distance function. In this work, we use histogram to estimate the probability distribution. Suppose there are $C$ shape classes, each of which has $J$ 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 $S_{i,j} \in R^{N \times T}$, where $S_{i,j} = [S_{i,j}^1, S_{i,j}^2, \ldots, S_{i,j}^T]$, $S_{i,j}^t$ denotes HKS of the shape $y_{i,j}$ at the $t^{th}$ scale, $t = 1, 2, \ldots, T$. $N$ is the number of vertices of shape $y_{i,j}$ and $T$ is the number of scales. For the scale $t$, we calculate the histogram of $S_{i,j}^t$ of $N$ vertices of the shape $y_{i,j}$ to form the shape distribution $h_{i,j}^t$. By considering probability distributions of shape functions derived from HKS at different scales, a multiscale shape distribution can be developed.

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, $h_{i,j}^t = \frac{h_{i,j}^t - \mu}{\sigma}$

3.2. Discriminative auto-encoder

In this subsection, we propose a discriminative auto-encoder to extract discriminative high-level feature for shape retrieval. In order to boost the discriminative power of the hidden layer features, we impose a Fisher discrimination criterion [11] on them. Given the shape distribution input $x_i^t$ of the shape class $i$ at the scale $t$, $x_i^t = [h_{i,1}^t, h_{i,2}^t, \ldots, h_{i,C}^t]$, we denote by $z^t$ the features of the hidden layer $k$ in the auto-encoder from all classes. We can write $z^t$ as $z^t = [z_1^t, z_2^t, \ldots, z_C^t]$, where $z_1^t = [z_{1,1}^t, z_{1,2}^t, \ldots, z_{1,J}^t]$, $z_{i,j}^t$ is the hidden layer feature of the $j^{th}$ sample from the class $i$, $i = 1, 2, \ldots, C$, $j = 1, 2, \ldots, J$. Based on the Fisher discriminative criterion, the discrimination can be 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:

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

where $m_i^t$ and $m^t$ are the mean vector of $z_i^t$ and $z^t$, respectively, and $n_i$ is the number of samples of class $i$. Intuitively, we can define the discriminative regularization term $L(z^t)$ as $tr(S_w(z^t)) - tr(S_b(z^t))$. Thus, by incorporating the discriminative regularization term into the standard auto-encoder model, we can form the following objective function of the discriminative auto-encoder:

$$J(W^t, b^t) = \sum_{i=1}^{C} \frac{1}{2} ||x_i^t - G(F(x_i^t))||_2^2 + \frac{1}{2} \lambda ||W^t||_F^2 + \frac{1}{2} \gamma (tr(S_w(z^t)) - tr(S_b(z^t)))$$

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

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

(12)
$L_0(z_{i,j}^t) = \frac{1}{2}tr((z_{i,j}^t - m_i^t)(z_{i,j}^t - m_i^t)^T) - \frac{1}{2}tr((m_i^t - m_i^t)(m_i^t - m_i^t)^T).$

(13)

To optimize the objective function of the discriminative auto-encoder, we adopt the back-propagation method of the error. We denote by $W_{p,q}^{l,t}$ by the weight associated with the connection between the unit $p$ in the layer $l$ and the unit $q$ in the layer $l + 1$. Also, $b^l_{i,j}$ is the bias associated with the connection with the unit $p$ in the layer $l$. The partial derivatives of the overall cost function $J(W^l, b^l)$ can be computed as:

$$\frac{\partial J(W^l, b^l)}{\partial W^{l,t}} = \sum_{i=1}^{C} \sum_{h_{i,j}' \in i} \frac{\partial J_0(W^l, b^l, h_{i,j}^t)}{\partial W^{l,t}} + \lambda W^{l,t}$$

$$+ \gamma \sum_{i=1}^{C} \sum_{z_{i,j}^t \in i} \frac{\partial L_0(z_{i,j}^t)}{\partial W^{l,t}}$$

(14)

$$\frac{\partial J(W^l, b^l)}{\partial b^{l,t}} = \sum_{i=1}^{C} \sum_{h_{i,j}' \in i} \frac{\partial J_0(W^l, b^l, h_{i,j}^t)}{\partial b^{l,t}}$$

$$+ \gamma \sum_{i=1}^{C} \sum_{z_{i,j}^t \in i} \frac{\partial L_0(z_{i,j}^t)}{\partial b^{l,t}}.$$ 

Denote by $\delta^{l,t}$ the error of the output layer $L$ in the auto-encoder. For the output layer (the layer $L$), we have:

$$\delta^{L,t} = - (h_{i,j}^t - a^{L,t}) \cdot \sigma'(u^{L,t})$$

(16)

where $a^{L,t}$ is the activation of the output layer, $u^{L,t}$ is the total weighted sum of the activations of the layer $L - 1$ to the output layer, $\sigma'(u^{L,t})$ is the derivative of the activation function in the output layer and $\cdot$ denotes the element-wise production. For other layers $l = L - 1, L - 2, \cdots, 2$, with the back-propagation method in [13], the error $\delta^{l,t}$ can be recursively obtained by the following equation:

$$\delta^{l,t} = ((W^{l,t})^T \delta^{l+1,t}) \cdot \sigma'(u^{L,t}).$$

(17)

Therefore, the partial derivatives of the function $J_0(W^l, b^l, h_{i,j}^t)$, $\frac{\partial J_0(W^l, b^l, h_{i,j}^t)}{\partial W^{l,t}}$ and $\frac{\partial J_0(W^l, b^l, h_{i,j}^t)}{\partial b^{l,t}}$ can be computed:

$$\frac{\partial J_0(W^l, b^l, h_{i,j}^t)}{\partial W^{l,t}} = \delta^{l+1,t}(a^{l,t})^T$$

(18)

and $\frac{\partial L_0(z_{i,j}^t)}{\partial W_{p,q}}$ can be computed as follows:

$$\frac{\partial L_0(z_{i,j}^t)}{\partial W_{p,q}} = \frac{\partial z_{i,j}^t}{\partial W_{p,q}} \frac{\partial L_0(z_{i,j}^t)}{\partial z_{i,j}^t}. (19)$$

where $z_{i,j}^t$ is the $p^{th}$ component of $z_{i,j}^t$. The partial derivative of $L_0(z_{i,j}^t)$ with respect to $z_{i,j}^t$ can be obtained:

$$\frac{\partial L_0(z_{i,j}^t)}{\partial z_{i,j}^t} = (1 - \frac{1}{n_i})(z_{i,j}^t - m_{i,p})$$

(20)
Algorithm 1 Algorithm of discriminative auto-encoder.

Input: training set $x_{i}^{t}$; the layer size of the auto-encoder; $\gamma$; $\lambda$; $\alpha$.

Output: $W^{t}$ and $b^{t}$.

Initialize $\Delta W^{l,t}$ and $\Delta b^{l,t}$ with the restricted Boltzmann machine for all $l$.

For all $h_{i,j}^{l}:

1. Compute $\frac{\partial J_{0}(W^{t},W^{t},h_{i,j}^{l})}{\partial W^{l,t}} + \gamma \frac{\partial L_{0}(z_{i}^{l})}{\partial W^{l,t}} + \frac{\partial J_{0}(W^{t},W^{t},h_{i,j}^{l})}{\partial b^{l,t}} + \gamma \frac{\partial L_{0}(z_{i}^{l})}{\partial b^{l,t}}$; $l \neq k - 1$, compute them with Eq. (18); $l = k - 1$, compute them with Eqs. (21) and (22).

2. Set $\Delta W^{l,t}$ to $\Delta W^{l,t} + \frac{\partial J_{0}(W^{t},W^{t},h_{i,j}^{l})}{\partial W^{l,t}} + \gamma \frac{\partial L_{0}(z_{i}^{l})}{\partial W^{l,t}}$.

3. Set $\Delta b^{l,t}$ to $\Delta b^{l,t} + \frac{\partial J_{0}(W^{t},W^{t},h_{i,j}^{l})}{\partial b^{l,t}} + \gamma \frac{\partial L_{0}(z_{i}^{l})}{\partial b^{l,t}}$.

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

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

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

Therefore, based on Eqs. (18), (19) and (20), for $l \neq k - 1$, $\frac{\partial J_{0}(W^{t},W^{t},h_{i,j}^{l})}{\partial W^{l,t}} + \gamma \frac{\partial L_{0}(z_{i}^{l})}{\partial W^{l,t}}$ and $\gamma \frac{\partial L_{0}(z_{i}^{l})}{\partial b^{l,t}}$ can be obtained by Eq. (18). For $l = k - 1$, $\frac{\partial J_{0}(W^{t},W^{t},h_{i,j}^{l})}{\partial W^{l,t}} + \gamma \frac{\partial L_{0}(z_{i}^{l})}{\partial W^{l,t}}$ and $\gamma \frac{\partial L_{0}(z_{i}^{l})}{\partial b^{l,t}}$ can be computed:

\[
\frac{\partial J_{0}(W^{t},b^{t},h_{i,j}^{l})}{\partial W^{l,t}} + \frac{\partial L_{0}(z_{i}^{l})}{\partial W^{l,t}} = (b^{l+1,t} + \gamma(1 - \frac{1}{n_{i}})
\]

\[
(z_{i,j}^{l} - m_{i}^{t}) - (\gamma \frac{1}{n_{i}} - \frac{1}{\sum n_{i}})(m_{i}^{t} - m^{t})(a^{l,t})^{T}
\]

\[
\frac{\partial J_{0}(W^{t},b^{t},h_{i,j}^{l})}{\partial b^{l,t}} + \frac{\partial L_{0}(z_{i}^{l})}{\partial b^{l,t}} = b^{l+1,t} + \gamma(1 - \frac{1}{n_{i}})
\]

\[
(z_{i,j}^{l} - m_{i}^{t}) - (\gamma \frac{1}{n_{i}} - \frac{1}{\sum n_{i}})(m_{i}^{t} - m^{t}).
\]

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

3.3. 3D Shape Descriptor

In this subsection, we use the activations of the hidden layer of the discriminative auto-encoder to form the shape descriptor. In order to characterize the intrinsic structure of the shape more effectively, we train multiple discriminative auto-encoders by setting multiscale shape distributions to the inputs of the discriminative auto-encoder. That is, for each scale $t$, we can learn $W^{t}$ and $b^{t}$ from a set of training shape distributions, i.e., $x_{i}^{1}, x_{i}^{2}, \ldots, x_{i}^{T}$, $t = 1, 2, \ldots, T$. Thus, $T$ discriminative auto-encoders can be formed by $T$ groups of shape distributions. Once the multiple discriminative auto-encoders are trained, we can concatenate the activations of all hidden layers to form a shape descriptor.

Denote the $t^{th}$ encoder of the multiple discriminative auto-encoders by $F^{t}$, which corresponds to the input of the multiscale shape distribution at the scale $t$. The shape descriptor of the $j^{th}$ shape from the class $i$, i.e., activations in the hidden layers of the multiple discriminative auto-encoders, can be represented:

\[
\alpha_{i,j} = [F^{1}(h_{i,j}^{1}); F^{2}(h_{i,j}^{2}); \ldots; F^{T}(h_{i,j}^{T})].
\]

4. Experimental Results

We conducted the experiments for shape matching and retrieval to evaluate performance of the proposed 3D shape descriptor. We define a universal time unit $\tau = 0.01$ and take 101 sampled time values (i.e., 101 scales) for the computation of the HKS descriptor. And 128 bins are used to form the histogram of HKS at each scale, which results in the 128-dimensional input of the discriminative auto-encoder. We train an auto-encoder, which consists of an encoder with layers of size 128-1000-500-30 and a symmetric decoder. Moreover, in Eq. (11), $\lambda$ and $\gamma$ are set to 0.001, respectively.

4.1. Shape Matching Performance

The shape matching is a key step in 3D model retrieval. A good shape descriptor should be robust to represent the 3D model with pose changes, topological changes and noise corruption. The models used in the experiment were chosen from the McGill dataset [26]. We evaluate performance of the proposed shape descriptor from the two aspects.

Consistency over deformable shapes In this experiment, we test the performance of the proposed shape descriptor on the deformed shape models. We choose the Teddy-bear and Human models with different poses. The shape descriptors of the deformed shapes are illustrated in Fig. 2. From the figure one can see that the descriptors of the model with different pose changes are very similar, which demonstrates that the proposed shape descriptor has the potential to consistently represent the shapes with pose changes. On the other hand, the shape descriptors of different models are distinctive. This verifies that the hidden layer features in the proposed discriminative auto-encoder have small within-class variations but large between-class variations.
Resistance to noise  By perturbing the vertices of the mesh with various levels of the numerical noise, we will demonstrate that the proposed shape descriptor is robust to noise. The noise, a 3-dimensional vector, is randomly generated from a multivariate normal distribution,

\[ \text{Noise} \sim N_3(\mu, NR \times \Sigma), \]

where \( \mu = [E[X_1], E[X_2], \cdots, E[X_k]] \) is the 3-dimensional mean vector of the coordinates of all vertices, \( \Sigma = [\text{Cov}[X_i, X_j]] \) is the 3 \times 3 covariance matrix of all vertices, \( i = 1, 2, \cdots, k \), \( j = 1, 2, \cdots, k \), and \( NR \) denotes the ratio between the variance of noise and variance of the coordinates of the vertices.

Fig. 3 shows the clean Crab and Hand models, and their noisy models, respectively. In (a) and (c), the green and red noisy models are generated by noise of \( NR = 0.01 \) and \( NR = 0.04 \), respectively. Particularly, in the noisy model with noise of \( NR = 0.04 \), geometric structures of the mesh have been moderately deteriorated. As indicated in Fig. 3, the variations of the proposed shape descriptors of the clean and noisy models (plotted with the yellow, green and red curves, respectively) are small. Since the level of noise of \( NR = 0.01 \) is low, we can see that the difference between the shape descriptors of the clean model and the noisy model of \( NR = 0.01 \) is very small. Therefore, the yellow and green curves are basically overlapped. The test demonstrates that the proposed shape descriptor formed by the deep discriminative auto-encoder is robust to noise.

4.2. 3D Shape Retrieval Performance

In order to demonstrate effectiveness of our method, we test the proposed shape descriptor on two benchmark datasets of 3D models, i.e., McGill [26], SHREC’10 Shape- Google [5] datasets. Each shape is represented by a compact 1D shape descriptor and \( L_2 \) norm is used to compute the distance between the two shape descriptors for retrieval.

4.2.1 McGill Shape Dataset

The McGill 3D shape dataset is a challenging dataset, which contains 255 objects with significant part articulations. They are from 10 classes: ant, crab, spectacle, hand, human, octopus, plier, snake, spider and teddy-bear. Each class contains one 3D shape with a variety of pose changes. Fig. 4 shows some examples in the McGill shape dataset.

We compare our proposed method to the state-of-the-art methods: the Hybrid BOW [21], the PCA based VLAT method [29], the graph-based method [1], the hybrid 2D/3D approach [17] and covariance descriptor [28]. We denote our proposed discriminative auto-encoder based shape descriptor by DASD. In our proposed DASD method, 10 shapes per class are randomly chosen to train the discriminative auto-encoder and the other shapes per class are used
Figure 3. Descriptors of the clean and noisy models of Crab and Hand. In (a) and (c), the green and red shapes are with noise of $NR = 0.01$ and $NR = 0.04$, respectively. In (b) and (d), the descriptors of the shapes plotted by the yellow, green and red curves correspond to the clean model, the noisy model with noise of $NR = 0.01$ and the noisy model with noise of $NR = 0.04$, respectively.

Figure 4. Example shapes in the McGill dataset. The left three columns show the shapes of Crab while the right three columns show the shapes of Hand with nonrigid transformations.

as the testing samples. The proposed method is evaluated with different performance measures, namely, Nearest Neighbor (NN), the First Tier (1-Tier), the Second Tier (2-Tier). The retrieval performance of these methods is illustrated in Table 1. From this table, compared to the state-of-the-art methods [21, 29, 11, 17, 28], we can see that the proposed method can achieve the best performance on the 4 performance measures. There are large nonrigid deformations with the objects in the McGill shape dataset, which results in large within-class variations of the shape descriptors. Nonetheless, due to the discriminative feature representation in the hidden layer of the discriminative autoencoder, as shown in Fig. 2, DASD is still robust to nonrigid deformations.

Table 1. Retrieval results on the McGill dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>NN</th>
<th>1-Tier</th>
<th>2-Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Covariance method [28]</td>
<td>0.977</td>
<td>0.732</td>
<td>0.818</td>
</tr>
<tr>
<td>Graph-based method [11]</td>
<td>0.976</td>
<td>0.741</td>
<td>0.911</td>
</tr>
<tr>
<td>PCA-based VLAT [29]</td>
<td>0.969</td>
<td>0.658</td>
<td>0.781</td>
</tr>
<tr>
<td>Hybrid BOW [21]</td>
<td>0.957</td>
<td>0.635</td>
<td>0.790</td>
</tr>
<tr>
<td>Hybrid 2D/3D [17]</td>
<td>0.925</td>
<td>0.557</td>
<td>0.698</td>
</tr>
<tr>
<td>DASD</td>
<td><strong>0.988</strong></td>
<td><strong>0.782</strong></td>
<td><strong>0.834</strong></td>
</tr>
</tbody>
</table>
4.2.2 SHREC’10 ShapeGoogle Dataset

SHREC’10 ShapeGoogle dataset [5] contains 1051 synthetic shapes. In this dataset, there are 518 shapes from 13 classes are generated with the five simulated transformations, i.e., isometry, topology, isometry+topology, partiality and triangulation, and there are 455 unrelated distractor shapes. Following the setting in [19], in order to make the dataset more challenging, all shapes are re-scaled to have the same size and the samples in the dataset which have the same attribute are considered to be of the same class. For example, male and female shapes are considered to be from the same class. Fig. 5 shows some examples of the ShapeGoogle dataset.

Table 2. Retrieval results on the SHREC’10 ShapeGoogle dataset.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Isometry</td>
<td>0.988</td>
<td>0.977</td>
<td>0.994</td>
<td>0.998</td>
</tr>
<tr>
<td>Topology</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>0.996</td>
</tr>
<tr>
<td>Isometry+Topology</td>
<td>0.933</td>
<td>0.934</td>
<td>0.956</td>
<td>0.982</td>
</tr>
<tr>
<td>Partiality</td>
<td>0.947</td>
<td>0.948</td>
<td>0.951</td>
<td>0.973</td>
</tr>
<tr>
<td>Triangulation</td>
<td>0.954</td>
<td>0.950</td>
<td>0.955</td>
<td>0.955</td>
</tr>
</tbody>
</table>

5. Conclusions

In this paper, we propose a deep shape descriptor with the discriminative auto-encoder for shape matching and retrieval, which is insensitive to geometric structure variations. By imposing the Fisher discrimination criterion on the feature representation in the hidden layer of the auto-encoder, we develop a discriminative auto-encoder so that the feature representation in the hidden layer have small within-class scatter but large between-class scatter. Then, with the multiscale shape distribution, we train multiple discriminative auto-encoders to extract all features in the hidden layers to form the deep shape descriptor. The deep shape descriptor demonstrates its performance in various tests for matching and retrieving 3D shapes.
References


