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

Anonymous CVPR submission

Paper ID 795

Abstract

Complex geometric structural variations of 3D models 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, 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 [26, 22, 18, 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 by the similarity between the shape descriptors for retrieval.

Shape descriptors for shape matching and retrieval have been extensively studied in the geometry community [33, 14, 12, 34, 28]. In the past decades, plenty of shape descriptors have been proposed, such as the $D2$ shape distribution [12], statistical moments of the model [34, 27], Fourier descriptor [8], Light Field Descriptor [15], Eigenvalue Descriptor (EVD)[16]. 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 [36]. 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 [30, 7, 26, 24]. Based on the Laplace-Beltrami operator, the global point signature (GPS) [26] 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) [30], 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 [24], the authors applied the bag-of-features (BOF) paradigm to learn the shape descriptor. The dictionary of words is learned by the K-means clustering method from a set of HKSs of shapes. Then a histogram of pairs of spatially-close words over the learned dictionary is formed as the shape descriptor for retrieval. Based on K-means clustering, Lavouè*et al*. [20] combined the standard and

spatial BOF descriptors for shape retrieval. Since *K*-means clustering can be viewed as a special case of sparse coding, Litman *et al*. [21] employed sparse coding to learn the dictionary of words instead of *K*-means clustering. The histogram of encoded representation coefficients over the learned dictionary is used to represent shape for retrieval. Moreover, in order to obtain the discriminative representation coefficients, a class-specific dictionary is constructed in a supervised way.

Recently, due to the favorable ability of modeling the nonlinearity by mapping the high dimensional feature to the low dimensional discriminative feature in the hidden layer of the network, the deep auto-encoder [13, 4] has been widely used in many challenging tasks such as image denoising [35], image classification [19] and face recognition [17]. Inspired by great success of the deep auto-encoder in computer vision and pattern recognition, in this paper, we develop a novel auto-encoder based shape descriptor for retrieval, which imposes the Fisher discrimination criterion on the hidden layer to make the hidden layer features discriminative and insensitive to geometric structure variations. It is expected that the neurons in the hidden layer have small within-class scatter but big between-class scatter. Moreover, in order to much more effectively represent shape, by using the multiscale shape distribution as the input of the auto-encoder, we train multiple discriminative auto-encoders and concatenate all neurons in the hidden layers as the high-level learning shape descriptor for retrieval. The proposed shape descriptor is verified on the representative and benchmark shape datasets, showing very promising performance.

The rest of the paper is organized as follows. Section 2 briefly introduces HKS and auto-encoder. Section 3 presents the proposed shape descriptor with the discriminative auto-encoder. Section 4 performs extensive experiments and Section 5 concludes the paper.

2. Background

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 weigh value for each edge. 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 $u(x, t)$, where x denotes one vertex on the meshed surface and t denotes the diffusion time. Provided that there is an initial heat 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:

$$\frac{\partial H_t}{\partial t} = -LH_t \tag{1}$$

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.

$$H_t = exp(-tL) \tag{2}$$

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

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

where λ_i is the i^{th} eigenvalue of the Laplacian, ϕ_i is the i^{th} eigenfunction, and x and y denotes the vertex x and y, respectively. Heat kernel signature (HKS) [30] of the vertex x at time t, S_x^t, is defined as the diagonal of the heat kernel of the vertex x taken at time t:

$$\begin{aligned} S_x^t &= H_t(x, x) \\ &= \sum_{i=0} e^{-\lambda_i t}\phi_i(x)\phi_i(x) \end{aligned} \tag{4}$$

HKS, as a point signature, can capture information of the neighborhood of a point x on the shape at a scale defined by t. In the following section, without the specific instruction, we use t to represent the scale of HKS, where $t = 1, 2, \cdots, T$.

2.2. Auto-encoder

An auto-encoder neural network [13, 4] usually consists of two parts, i.e., encoder and decoder. The encoder, denoted by F, maps the input $\boldsymbol{x} \in \mathcal{R}^{d\times 1}$ to the hidden layer representation, denoted by $\boldsymbol{z} \in \mathcal{R}^{r\times 1}$, 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 $\boldsymbol{W}^l$ and $\boldsymbol{b}^l$, respectively. The output of the layer is called the activation function. Usually, the activation function is non-linear, such as sigmoid function $\sigma(\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{x}}}$ or tanh function $\sigma(\boldsymbol{x}) = \frac{e^{\boldsymbol{x}}-e^{-\boldsymbol{x}}}{e^{\boldsymbol{x}}+e^{-\boldsymbol{x}}}$. Therefore, the output of the layer $l + 1$ is :

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

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

$$F(\boldsymbol{x}) = f_k(f_{k-1}(\cdots, f_2(\boldsymbol{x}))) \tag{6}$$

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

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

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

$$\begin{aligned} <\hat{\boldsymbol{W}}, \hat{\boldsymbol{b}}> = & argmin_{\boldsymbol{W},\boldsymbol{b}} \frac{1}{2} \sum_{i=1}^{N} \|\boldsymbol{x}_i - G(F(\boldsymbol{x}_i))\|_2^2 \\ & + \frac{1}{2}\lambda \|\boldsymbol{W}\|_2^2 \end{aligned} \tag{8}$$

where x_i represents the i^{th} training samples, N is 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 back-propagation 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 [23] 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 shape 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 $\boldsymbol{y}_{i,j}$ to index the j^{th} sample of the i^{th} shape class. For each shape $\boldsymbol{y}_{i,j}$, we extract HKS feature $\boldsymbol{S}_{i,j} \in \mathcal{R}^{N\times T}$, where $\boldsymbol{S}_{i,j} = [\boldsymbol{S}_{i,j}^1, \boldsymbol{S}_{i,j}^2, \cdots, \boldsymbol{S}_{i,j}^T]$, $\boldsymbol{S}_{i,j}^t$ denotes HKS of the shape $\boldsymbol{y}_{i,j}$ at the t^{th} scale, $t = 1, 2, \cdots, T$, N is the number of vertices of shape $\boldsymbol{y}_{i,j}$ and T is the number of scales. For the scale t, we calculate the histogram of $\boldsymbol{S}_{i,j}^t$ of N vertices of the shape $\boldsymbol{y}_{i,j}$ to form the shape distribution $\boldsymbol{h}_{i,j}^t$. 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,

$$\boldsymbol{h}_{i,j}^t = \frac{\boldsymbol{h}_{i,j}^t - \boldsymbol{h}^t}{\boldsymbol{v}^t} \tag{9}$$

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

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 $\boldsymbol{x}_i^t$ of the shape class i at the scale t, $\boldsymbol{x}_i^t = [\boldsymbol{h}_{i,1}^t, \boldsymbol{h}_{i,2}^t, \cdots, \boldsymbol{h}_{i,J}^t]$, we denote by $\boldsymbol{z}^t$ the features of the hidden layer k in the auto-encoder from all classes. We can write $\boldsymbol{z}^t$ as $\boldsymbol{z}^t = [\boldsymbol{z}_1^t, \boldsymbol{z}_2^t, \cdots, \boldsymbol{z}_C^t]$, where $\boldsymbol{z}_i^t = [\boldsymbol{z}_{i,1}^t, \boldsymbol{z}_{i,2}^t, \cdots, \boldsymbol{z}_{i,J}^t]$, $\boldsymbol{z}_{i,j}^t$ is the hidden layer feature of the j^{th} sample from the class i, $i = 1, 2, \cdots, C$, $j = 1, 2, \cdots, J$. Based on the Fisher discriminative criterion, the discrimination can be achieved by minimizing the within-class scatter of $\boldsymbol{z}^t$, denoted by $S_w(\boldsymbol{z}^t)$, and maximizing the between-class scatter of $\boldsymbol{z}^t$, denoted by $S_b(\boldsymbol{z}^t)$. $S_w(\boldsymbol{z}^t)$ and $S_b(\boldsymbol{z}^t)$ are defined as:

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

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

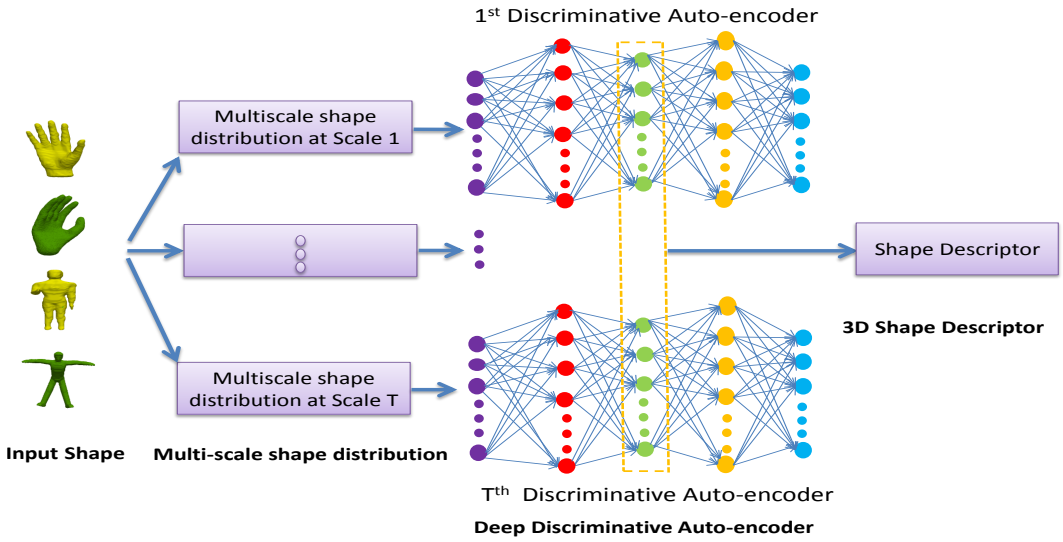


Figure 1. The framework of the proposed discriminative auto-encoder based shape descriptor.

function of the discriminative auto-encoder:

$$J(\boldsymbol{W}^t, \boldsymbol{b}^t) = argmin_{\boldsymbol{W}^t, \boldsymbol{b}^t} \sum_{i=1}^{C} \frac{1}{2}\|\boldsymbol{x}_i^t - G(F(\boldsymbol{x}_i^t))\|_2^2 + \frac{1}{2}\lambda\|\boldsymbol{W}^t\|_2^2 + \frac{1}{2}\gamma(tr(S_w(\boldsymbol{z}^t)) - tr(S_b(\boldsymbol{z}^t))). \tag{11}$$

For the sample $\boldsymbol{h}_{i,j}^t$, we define the following functions:

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

$$L_0(\boldsymbol{z}_{i,j}^t) = \frac{1}{2}tr((\boldsymbol{z}_{i,j}^t - \boldsymbol{m}_i^t)(\boldsymbol{z}_{i,j}^t - \boldsymbol{m}_i^t)^T) - \frac{1}{2}tr((\boldsymbol{m}_i^t - \boldsymbol{m}^t)(\boldsymbol{m}_i^t - \boldsymbol{m}^t)^T) \tag{13}$$

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, $b_p^{l,t}$ is the bias associated with the connection with the unit p in the layer l. The partial derivatives of the overall cost function $J(\boldsymbol{W}^t, \boldsymbol{b}^t)$ can be computed as:

$$\frac{\partial J(\boldsymbol{W}^t, \boldsymbol{b}^t)}{\partial \boldsymbol{W}^{l,t}} = \sum_{i=1}^{C}\sum_{\boldsymbol{h}_{i,j}^t \in i} \frac{\partial J_0(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{h}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}} + \lambda \boldsymbol{W}^{l,t} + \gamma \sum_{i=1}^{C}\sum_{\boldsymbol{z}_{i,j}^t \in i} \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}} \tag{14}$$

$$\frac{\partial J(\boldsymbol{W}^t, \boldsymbol{b}^t)}{\partial \boldsymbol{b}^{l,t}} = \sum_{i=1}^{C}\sum_{\boldsymbol{h}_{i,j}^t \in i} \frac{\partial J_0(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{h}_{i,j}^t)}{\partial \boldsymbol{b}^{l,t}} + \gamma \sum_{i=1}^{C}\sum_{\boldsymbol{z}_{i,j}^t \in i} \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{b}^{l,t}} \tag{15}$$

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

$$\boldsymbol{\delta}^{L,t} = -(\boldsymbol{h}_{i,j}^t - \boldsymbol{a}^{L,t}) \bullet \sigma'(\boldsymbol{u}^{L,t}) \tag{16}$$

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

$$\boldsymbol{\delta}^{l,t} = ((\boldsymbol{W}^{l,t})^T \boldsymbol{\delta}^{l+1,t}) \bullet \sigma'(\boldsymbol{u}^{L,t}) \tag{17}$$

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

$$\frac{\partial J_0(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{h}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}} = \boldsymbol{\delta}^{l+1,t}(\boldsymbol{a}^{l,t})^T \qquad \frac{\partial J_0(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{h}_{i,j}^t)}{\partial \boldsymbol{b}^{l,t}} = \boldsymbol{\delta}^{l+1,t} \tag{18}$$

Since $\boldsymbol{z}_{i,j}^t = \boldsymbol{W}^{k-1}\boldsymbol{a}^{k-1} + \boldsymbol{b}^{k-1}$, for $l \neq k-1$, $\frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}} = 0$ and $\frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{b}^{l,t}} = 0$. For $l = k-1$, $\frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial W_{p,q}^{l,t}}$

and $\frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial b_p^{l,t}}$ can be computed as follows:

$$\frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial W_{p,q}^{k-1,t}} = \frac{\partial z_{i,j,p}^t}{\partial W_{p,q}^{k-1,t}} \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial z_{i,j,p}^t} = a_q^{k-1,t} \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial z_{i,j,p}^t}$$
$$\frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial b_p^{k-1,t}} = \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial z_{i,j,p}^t} \tag{19}$$

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

$$\frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial z_{i,j,p}^t} = (1 - \frac{1}{n_i})(z_{i,j,p}^t - m_{i,p}^t) - (\frac{1}{n_i} - \frac{1}{\sum n_i})(m_{i,p}^t - m_p^t) \tag{20}$$

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

Therefore, based on Eqs. (18), (19) and (20), for $l \neq k-1$, $\frac{\partial J(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{h}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}} + \gamma \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}}$ and $\frac{\partial J(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{h}_{i,j}^t)}{\partial \boldsymbol{b}^{l,t}} + \gamma \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{b}^{l,t}}$ can be obtained by Eq. (18). For $l = k-1$, $\frac{\partial J(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{h}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}} + \gamma \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}}$ and $\frac{\partial J(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{x}_i^t)}{\partial \boldsymbol{b}^{l,t}} + \gamma \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{b}^{l,t}}$ can be computed:

$$\frac{\partial J_0(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{h}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}} + \gamma \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}} = (\boldsymbol{\delta}^{l+1,t} + \gamma(1 - \frac{1}{n_i})(\boldsymbol{z}_{i,j}^t - \boldsymbol{m}_i^t) - \gamma(\frac{1}{n_i} - \frac{1}{\sum n_i})(\boldsymbol{m}_i^t - \boldsymbol{m}^t))(\boldsymbol{a}^{l,t})^T \tag{21}$$

$$\frac{\partial J_0(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{h}_{i,j}^t)}{\partial \boldsymbol{b}^{l,t}} + \gamma \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{b}^{l,t}} = \boldsymbol{\delta}^{l+1,t} + \gamma(1 - \frac{1}{n_i})(\boldsymbol{z}_{i,j}^t - \boldsymbol{m}_i^t) - \gamma(\frac{1}{n_i} - \frac{1}{\sum n_i})(\boldsymbol{m}_i^t - \boldsymbol{m}^t) \tag{22}$$

Once the partial derivatives of the objective function of the discriminative auto-encoder with respect to $\boldsymbol{W}^t$ and $\boldsymbol{b}^t$ are computed, we can employ the conjugate gradient method to obtain $\boldsymbol{W}^t$ and $\boldsymbol{b}^t$. 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 $\boldsymbol{W}^t$ and $\boldsymbol{b}^t$ from a set of training shape distributions, i.e.,$\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \cdots, \boldsymbol{x}_C^t$, $t = 1, 2, \cdots, T$. 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.

Algorithm 1 Algorithm of discriminative auto-encoder.

Input:training set $\boldsymbol{x}_i^t$;the layer size of the auto-encoder;λ; γ.
Output: $\boldsymbol{W}^t$ and $\boldsymbol{b}^t$.
Initialize $\Delta \boldsymbol{W}^{l,t}$ and $\Delta \boldsymbol{b}^{l,t}$ with the restricted Boltzman machine for all l.
For all $\boldsymbol{h}_{i,j}^t$:

1. Compute $\frac{\partial J_0(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{h}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}} + \gamma \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}}$ and $\frac{\partial J_0(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{h}_{i,j}^t)}{\partial \boldsymbol{b}^{l,t}} + \gamma \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{b}^{l,t}}$: $l \neq k-1$, compute them with Eq. (18); $l = k-1$, compute them with Eqs. (21) and (22).
2. Set $\Delta \boldsymbol{W}^{l,t}$ to $\Delta \boldsymbol{W}^{l,t} + \frac{\partial J_0(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{h}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}} + \gamma \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{W}^{l,t}}$.
3. Set $\Delta \boldsymbol{b}^{l,t}$ to $\Delta \boldsymbol{b}^{l,t} + \frac{\partial J_0(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{x}_i^t)}{\partial \boldsymbol{b}^{l,t}} + \gamma \frac{\partial L_0(\boldsymbol{z}_{i,j}^t)}{\partial \boldsymbol{b}^{l,t}}$.

Update $\boldsymbol{W}^{l,t}$ and $\boldsymbol{b}^{l,t}$: $\boldsymbol{W}^{l,t} = \boldsymbol{W}^{l,t} - \alpha(\Delta \boldsymbol{W}^{l,t} + \lambda \boldsymbol{W}^{l,t})$ $\boldsymbol{b}^{l,t} = \boldsymbol{b}^{l,t} - \alpha \Delta \boldsymbol{b}^{l,t}$.
Output $\boldsymbol{W}^{l,t}$ and $\boldsymbol{b}^{l,t}$ until the values of $J(\boldsymbol{W}^t, \boldsymbol{b}^t, \boldsymbol{x}_i^t)$ in adjacent iterations are close enough or the maximum number of iterations is reached.

Denote the t^{th} encoder of the multiple discriminative auto-encoders by F^t, which corresponds to the input of the multicale 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 :

$$\boldsymbol{\alpha}_{i,j} = [F^1(\boldsymbol{h}_{i,j}^1), F^2(\boldsymbol{h}_{i,j}^2), \cdots, F^T(\boldsymbol{h}_{i,j}^T)]. \tag{23}$$

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 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), λ and γ 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 [29]. 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, $Noise \sim N_3(\mu, NR * \Sigma)$, where $\mu = [E[X_1], E[X_2], \cdots, E[X_k]]$ is the 3-dimensional mean vector of the coordinates of all vertices, $\Sigma = [Cov[X_i, X_j]]$ is the 3×3 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 [29], SHREC'10 ShapeGoogle [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.

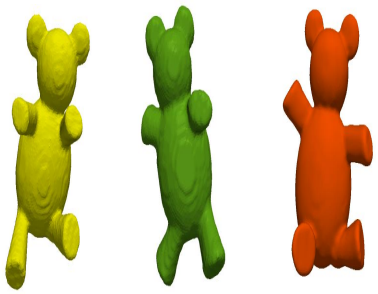
We compare our proposed method to the state-of-the-art methods: the Hybrid BOW [25], the PCA based VLAT method [32], the graph-based method [1], the hybrid 2D/3D approach [20] and covariance descriptor [31]. We denote our proposed discriminative auto-encoder based shape descriptor by DASD. In our proposed DASD method, 10 shapes per class are randomly chosen as the training samples to train the discriminative auto-encoder. The proposed method is evaluated with different performance measures, namely, Nearest Neighbor (NN), the First Tier (1-Tier), the Second Tier (2-Tier) and the Discounted Cumulative Gain (DGC). The retrieval performance of these methods is illustrated in Table 1. From this table, compared to the state-of-the-art methods [25, 32, 1, 20, 31], 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 auto-encoder, as shown in Fig. 2, DASD is still robust to non-rigid deformations. Therefore, our proposed DASD can all obtain better performance with the four different retrieval criteria.

Table 1. Retrieval results on the McGill dataset.

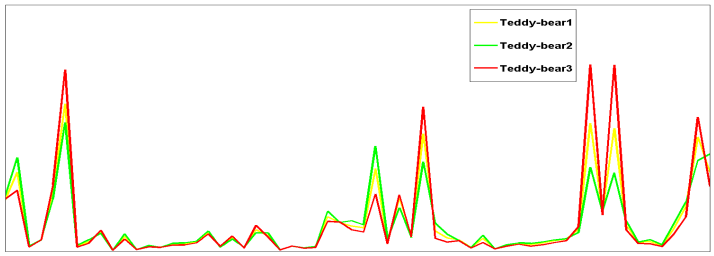
Methods	NN	1-Tier	2-Tier	DCG
Covariance method [31]	0.977	0.732	0.818	0.937
Graph-based method [1]	0.976	0.741	0.911	0.933
PCA-based VLAT [32]	0.969	0.658	0.781	0.894
Hybrid BOW [25]	0.957	0.635	0.790	0.886
Hybrid 2D/3D [20]	0.925	0.557	0.698	0.850
DASD	**0.988**	**0.812**	**0.934**	**0.955**

4.2.2 SHREC'10 ShapeGoogle Dataset

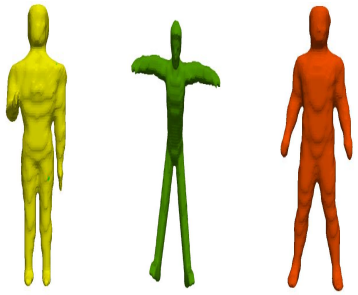
SHREC'10 ShapeGoogle dataset [5] contains 1184 synthetic shapes. In this dataset, there are 715 shapes from 13 classes are generated with the five simulated transformations, i.e., isometry, topology, isometry+topology, partiality and triangulation, and there are 456 unrelated distractor shapes. Following the setting in [21], 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



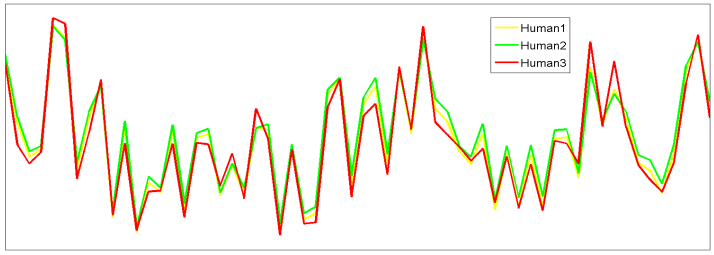
(a) Teddy-bear models: Teddy-bear1, Teddy-bear2, Teddy-bear3.



(b) Descriptors of the Teddy-bear models



(c) Human models: Human1, Human2, Human3.



(d) Descriptors of the Human models

Figure 2. Descriptors of the Teddy-bear model and the Human model. In (b), the descriptors of the shapes are plotted by the yellow, green and red curves, which correspond to Teddy-bear 1, Teddy-bear 2, Teddy-bear 3 while in (d) these curves correspond to Human 1, Human 2 and Human 3, respectively.

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 Shape-Google dataset.

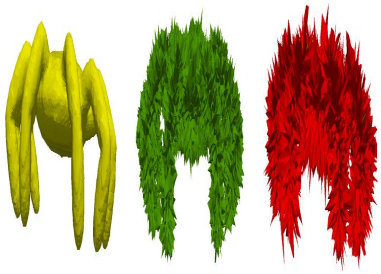
We compared the proposed DASD to the bag of feature (BOF) descriptor based on standard vector quantization (VQ) [5], sparse coding with unsupervised dictionary learning (DL) [21] and sparse coding with supervised DL [21]. We used the mean average precision criterion to evaluate our proposed method. Evaluation results are summarized in Table 2. From this table, one can see that our proposed DASD is superior to the BOF descriptors based on standard VQ [5], sparse coding with unsupervised DL [21] and sparse coding with supervised DL [21] in the case of different transformations. Compared to the dictionary learning based shape descriptors, since the deep auto-encoder has the good ability to model nonlinearity, DASD can characterize the low-dimensional manifold embedded in the high-dimensional shape space better. Therefore, the proposed DASD can obtain better performance. For example, in the cases of isometry+topology and partiality, the supervised dictionary learning based shape descriptor can obtain accuracies of 0.956 and 0.951 while our proposed DASD can achieve accuracies of 0.982 and 0.973, respectively.

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

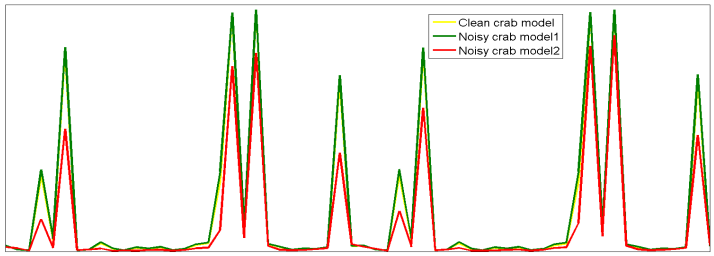
Transformation	VQ [5]	UDL [21]	SDL [21]	DASD
Isometry	0.988	0.977	0.994	**0.998**
Topology	**1.000**	**1.000**	**1.000**	0.996
Isometry+Topology	0.933	0.934	0.956	**0.982**
Partiality	0.947	0.948	0.951	**0.973**
Triangulation	0.954	0.950	**0.955**	**0.955**

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.

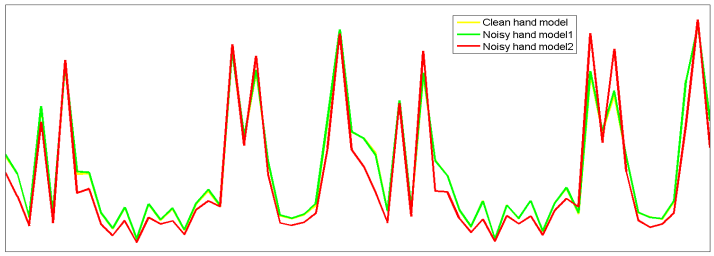


(a) Clean and noisy models of the shape Crab.



(b) Descriptors of the clean and noisy crab models.

(c) Clean and noisy models of the shape Human.



(d) Descriptors of the clean and noisy human models.

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.

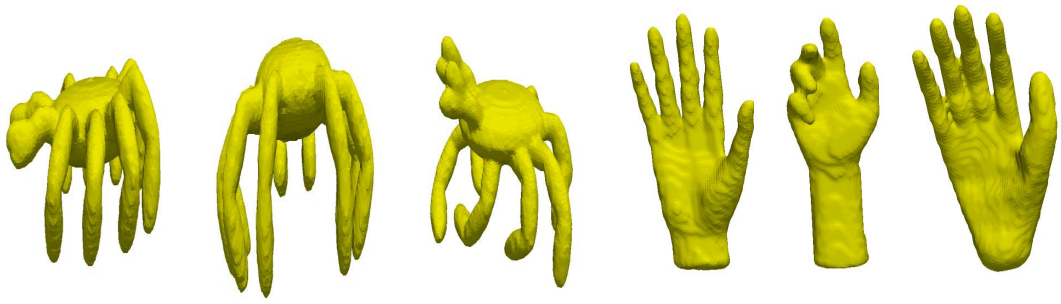


Figure 4. Example shapes in the McGill dataset. The left three columns show the shapes of Crab while the right three columns show the shapes of Hand with nonrigid transformations.

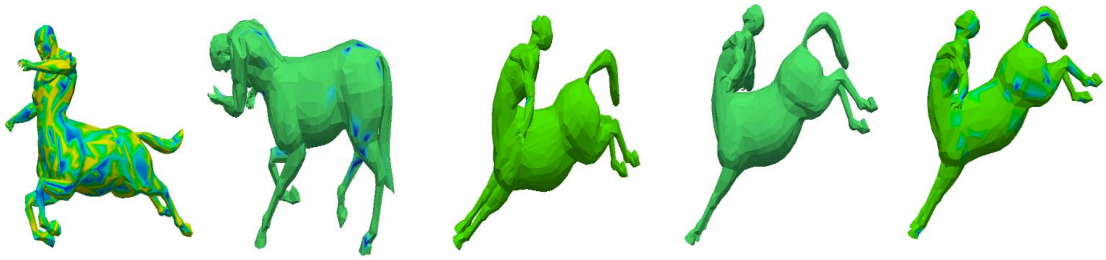


Figure 5. Example shapes with different transformations in the SHREC'10 ShapeGoogle dataset. From left to right, the Centaur shapes with the isometry, isometry+topology, topology, partiality and triangulation transformations are shown, respectively.

References

[1] 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. Proceedings*, pages 29–36, 2009. 6

[2] J. Assfalg, M. Bertini, A. D. Bimbo, and P. Pala. Content-based retrieval of 3-d objects using spin image signatures. *IEEE Transactions on Multimedia*, 9(3):589–599, 2007. 1

[3] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 831–837, 2000. 1

[4] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009. 2

[5] 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.*, 30(1):1, 2011. 6, 7

[6] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Sci. Comput.*, 28:1812–1836, September 2006. 1

[7] 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. Vision*, 89:266–286, 2010. 1

[8] D.-Y. Chen, X.-P. Tian, Y. te Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. *Computer Graphics Forum*, 22:223–232, 2003. 1

[9] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 2009. 1

[10] F. De Goes, S. Goldenstein, and L. Velho. A hierarchical segmentation of articulated bodies. *Computer Graphics Forum*, 27:1349–1356, 2008. 1

[11] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Ed)*. Wiley, 2001. 3

[12] M. Elad, A. Tal, and S. Ar. Content based retrieval of vrml objects - an iterative and interactive approach. *Proc. Sixth Eurographics Workshop Multimedia*, pages 97–108, 2001. 1

[13] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006. 2, 3, 4

[14] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, 37(5):509 – 530, 2005. Geometric Modeling and Processing 2004. 1

[15] V. Jain and H. Zhang. A spectral approach to shape-based retrieval of articulated 3d models. *Computer-Aided Design*, 39(5):398 – 407, 2007. Geometric Modeling and Processing 2006. 1

[16] V. Jain and H. Zhang. A spectral approach to shape-based retrieval of articulated 3d models. *Computer-Aided Design*, 39(5):398–407, 2007. 1

[17] M. Kan, S. Shan, H. Chang, and X. Chen. Stacked progressive auto-encoders (SPAE) for face recognition across poses. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 1883–1890, 2014. 2

[18] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21:649–658, 2005. 1

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012. 2

[20] G. Lavoué. Combination of bag-of-words descriptors for robust partial shape retrieval. *The Visual Computer*, 28(9):931–942, 2012. 1, 6

[21] 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*, 33(5):127–136, 2014. 2, 6, 7

[22] R. Osada, T. Funkhouser, B. Chazelle, and D. Dokin. Shape distributions. *ACM Transactions on Grphics*, 33:133–154, 2002. 1

[23] R. Osada, T. A. Funkhouser, B. Chazelle, and D. P. Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, 2002. 3

[24] M. Ovsjanikov, A. Bronstein, and M. Bronstein. Shape google: a computer vision approach to invariant shape retrieval. *Proc. NORDIA*, 2009. 1

[25] 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, 3DOR 2008, Crete, Greece, 2008. Proceedings*, pages 9–16, 2008. 6

[26] R. M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 225–233, 2007. 1

[27] D. Saupe and D. V. Vranic. 3d model retrieval with spherical harmonics and moments. *DAGM*, pages 392–397, 2001. 1

[28] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. *In Shape Modeling International*, pages 167–178, 2004. 1

[29] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. J. Dickinson. Retrieving articulated 3-d models using medial surfaces. *Mach. Vis. Appl.*, 19(4):261–275, 2008. 6

[30] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *SGP '09: Proceedings of the Symposium on Geometry Processing*, pages 1383–1392, 2009. 1, 2

[31] H. Tabia, H. Laga, D. Picard, and P. H. Gosselin. Covariance descriptors for 3d shape matching and retrieval. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, June 23-28, 2014*, pages 4185–4192, 2014. 6

[32] 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. Proceedings*, pages 17–24, 2013. 6

[33] J. W. H. Tangelder and R. C. Veltkamp. A survey of content based 3d shape retrieval methods. *In Shape Modeling International*, pages 145–156, 2004. 1

[34] D. V. Vranic, D. Saupe, and J. Richter. Tools for 3d-object retrieval: Karhunen-loeve transform and spherical harmonics. *IEEE MMSP 2001*, pages 293–298, 2001. 1

[35] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems 25, Lake Tahoe, Nevada, United States.*, pages 350–358, 2012. 2

[36] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 373–380, 2009. 1