

Learned Binary Spectral Shape Descriptor for 3D Shape Correspondence

Jin Xie, Meng Wang, and Yi Fang

NYU Multimedia and Visual Computing Lab, UAE

Department of Electrical and Computer Engineering, New York University Abu Dhabi
Department of Electrical and Computer Engineering, NYU Tandon School of Engineering

{jin.xie, mengw, yfang}@nyu.edu

Abstract

Dense 3D shape correspondence is an important problem in computer vision and computer graphics. Recently, the local shape descriptor based 3D shape correspondence approaches have been widely studied, where the local shape descriptor is a real-valued vector to characterize the geometrical structure of the shape. Different from these real-valued local shape descriptors, in this paper, we propose to learn a novel binary spectral shape descriptor with the deep neural network for 3D shape correspondence. The binary spectral shape descriptor can require less storage space and enable fast matching. First, based on the eigenvectors of the Laplace-Beltrami operator, we construct a neural network to form a nonlinear spectral representation to characterize the shape. Then, for the defined positive and negative points on the shapes, we train the constructed neural network by minimizing the errors between the outputs and their corresponding binary descriptors, minimizing the variations of the outputs of the positive points and maximizing the variations of the outputs of the negative points, simultaneously. Finally, we binarize the output of the neural network to form the binary spectral shape descriptor for shape correspondence. The proposed binary spectral shape descriptor is evaluated on the SCAPE and TOSCA 3D shape datasets for shape correspondence. The experimental results demonstrate the effectiveness of the proposed binary shape descriptor for the shape correspondence task.

1. Introduction

3D shape feature extraction and matching is an important topic in the community of computer vision and computer graphics. With the recent developments of the 3D scanning technology such as the Microsoft Kinect scanner, 3D shape correspondence has been receiving much more atten-

tion in many fields (e.g., molecular biology, mechanical engineering, and medical image analysis). Finding the intrinsic correspondences between 3D shapes can be applied to 3D scan alignment, texture mapping, shape morphing and animation, etc. Due to the large nonrigid deformations of the shapes, the shape correspondence task is usually very challenging.

Extensive research efforts have been dedicated to 3D shape correspondence in the past decades. Early shape correspondence approaches focused on the rigid shapes. The transformation between the rigid shapes can be parameterized by a few parameters such as the translation, rotation and scale factors. Thus, these parameters can be estimated by the optimization methods such as the iterative closest point (ICP) method [6] and graph matching methods [10, 12]. For example, the traditional ICP method first calculates the transformation between the rigid shapes with the least-square solution. Then, the points on one shape are transformed by the estimated rigid transformation. The correspondence-register cycle is iterated until the stopping criterion is satisfied. In addition, for the 3D point set matching problem, Jian *et al.* [13] proposed to represent the input point sets using the Gaussian mixture models. The correspondence problem is then converted into the problem of aligning two Gaussian mixture models such that the statistical discrepancy measure between the two models is minimized.

Compared to the rigid shape correspondence, establishing correspondence between two shapes with the nonrigid deformations is much more challenging. Nonrigid shape matching can be formulated as the graph matching problem [15, 18, 20, 25], where the point-to-point correspondence, i.e., the permutation matrix, can be obtained by minimizing the structural distortions such as the point-wise costs [3, 8, 22] and pairwise costs [3, 12]. By incorporating the point-wise costs and the pairwise costs, the graph match-

ing problem can be converted into the quadratic assignment problem. Since the quadratic assignment problem is NP-hard, many relaxation techniques have been proposed to solve this problem. In [14], the spectral graph matching algorithm was proposed by relaxing the constraint on the permutation matrix such that the Frobenius norm of the matrix is 1. In [23], by assuming the permutation matrix is an orthogonal matrix, the spectral embedding approach was proposed for shape correspondence.

Different from the aforementioned graph matching based shape correspondence methods, the local spectral shape descriptors have been proposed for 3D shape matching. Based on the fundamental solution of the heat diffusion equation, Sun *et al.* [21] developed heat kernel signature (HKS) as a point signature to describe the shape. In [4], based on the evolution of a quantum particle on the surface of the shape, the wave kernel signature (WKS) was proposed to characterize the shape. Due to the discriminative power of HKS and WKS, they can be used for shape correspondence. Based on the Laplace-Beltrami operator, Litman *et al.* [16] employed the classical metric learning method to learn the local spectral descriptor for shape correspondence. In [19], the authors converted the shape matching problem into the classification problem by employing the random forest classifier. The constructed random forest can vary the parameters of WKS to form a discriminative local shape descriptor for shape matching.

However, these local shape descriptors are real-valued. Recent advancement in the 3D shape acquisition technology has led to capturing large amounts of 3D shape data. It is desirable to develop the local binary shape descriptor for shape correspondence because the binary shape descriptor requires less storage and enables fast matching. In this paper, based on the Laplace-beltrami operator, we propose to learn the local binary spectral shape descriptor for shape correspondence. First, we construct a neural network to compute the responses of the eigenvectors of the Laplace-beltrami operator of each point to characterize the shape. For each point on the shape, we define the positive/negative points that are in/out of the neighborhoods of the point on the shape and the corresponding point on the deformed shape, respectively. We then train the constructed neural network such that the errors between the real-valued outputs of the network and their binary outputs are as small as possible. Moreover, we encourage that the variations of the outputs associated with the pairs of positive points are as small as possible and the variations of the outputs associated with the pairs of negative points are as large as possible. Finally, we binarize the outputs of the network to form a binary spectral shape descriptor for correspondence. Experimental results demonstrate that the learned binary spectral shape descriptor can yield good performance.

The main contribution of our work is that we construct

a neural network to form a parametric spectral representation to characterize the shape, and propose a learning based binary spectral shape descriptor for shape correspondence. It can be comparable to the real-valued local shape descriptors while it requires less storage and enables fast matching for correspondence. To the best of our knowledge, in the community of 3D shape analysis, this is the first work on developing the learning based binary 3D shape descriptor for correspondence.

The rest of the paper is organized as follows. Section 2 introduces the background of the local spectral shape descriptors. In Section 3, we present the proposed binary spectral shape descriptor for shape correspondence. Section 4 presents the experimental results and Section 5 concludes the paper.

2. Background

In this section, we briefly review the two local spectral shape descriptors. One is heat kernel signature (HKS) and the other is the learned spectral shape descriptor.

2.1. Heat kernel signature

Heat diffusion on the meshed surface X can be defined as :

$$\frac{\partial K_t}{\partial t} = -LK_t \quad (1)$$

where K_t denotes the heat kernel at diffusion time t , L is the Laplace-Beltrami operator. Given an initial Dirac delta distribution defined on X at time $t = 0$, based on the spectral decomposition theorem, the fundamental solution of Eq. (1) on vertices x and y , $K_t(x, y)$, can be obtained as:

$$K_t(x, y) = \sum_i e^{-v_i t} \phi_i(x) \phi_i(y) \quad (2)$$

where v_i and ϕ_i are the i^{th} eigenvalue and eigenvector of the Laplace-Beltrami operator L , respectively.

The fundamental solution $K_t(x, y)$ is also called the heat kernel. The heat kernel signature (HKS) [21] of vertex x at time t , $p_t(x)$, is defined as:

$$p_t(x) = \sum_i e^{-v_i t} \phi_i(x)^2. \quad (3)$$

Since HKS is highly related to the eigenvalue and eigenvector of the Laplace-Beltrami operator, it can capture the geometric structure of the neighborhood of point x on the shape.

2.2. Learned spectral descriptor

In HKS, from Eq. (3), one can see that the weight $e^{-v_i t}$ is obtained from the eigenvalue of the Laplace-Beltrami operator L . Litman *et al.* [16] proposed to learn the weight to

form a parametric spectral descriptor in a supervised learning way. For each point x_i on the shape, the learned parametric spectral descriptor $\mathbf{p}(x_i) \in \mathbb{R}^{n \times 1}$ can be defined as:

$$\mathbf{p}(x_i) = \mathbf{A}(\mathbf{b}(v_1), \mathbf{b}(v_2), \dots, \mathbf{b}(v_s))\phi(x_i) \quad (4)$$

where $\mathbf{b}(v_j) \in \mathbb{R}^{m \times 1}$ is the basis function such as the cubic B-spline function, v_j is the frequency component, $j = 1, 2, \dots, s$, $\phi(x_i) = [\phi_1^2(x_i), \phi_2^2(x_i), \dots, \phi_s^2(x_i)]^T$, \mathbf{A} is the $n \times m$ matrix of representation coefficients using the basis function. For each point x_i , by selecting the similar points on the pair of shapes as the positive samples and the dissimilar points on the pair of shapes as the negative samples, matrix \mathbf{A} can be learned by minimizing the Mahalanobis distances between the spectral descriptors of the set of the positive sample pairs and maximizing the Mahalanobis distances between the spectral descriptors of the set of the negative sample pairs, simultaneously.

From Eqs. (3) and (4), one can see that HKS can be viewed as a special case of the learned spectral descriptor. Nonetheless, different from HKS, matrix \mathbf{A} is learned from the training samples. Compared to HKS, the learned spectral descriptor is much more discriminative for the point-to-point correspondence.

3. Proposed Approach

In this section, we present our learned binary spectral shape descriptor for shape correspondence. Fig. 1 illustrates the shape matching framework with the proposed binary spectral shape descriptor. In subsection 3.1, based on the eigenvectors of the Laplace-Beltrami operator on the shape, we construct a metric network to compute the nonlinear representation of the eigenvectors to characterize the shape. In subsection 3.2, we present to learn the binary spectral descriptor with the deep metric learning method for shape correspondence.

3.1. Nonlinear parametric spectral representation

For each point x_i on the shape, $i = 1, 2, \dots, N$, we define the geometry vector $\mathbf{g}(x_i)$ as:

$$\mathbf{g}(x_i) = (\mathbf{b}(v_1), \mathbf{b}(v_2), \dots, \mathbf{b}(v_s))\phi(x_i) \quad (5)$$

where $\mathbf{b}(v_j) \in \mathbb{R}^{m \times 1}$ is the cubic B-spline basis function, v_j is the frequency component, $j = 1, 2, \dots, s$, $\phi(x_i) = [\phi_1^2(x_i), \phi_2^2(x_i), \dots, \phi_s^2(x_i)]^T$. The geometry vector $\mathbf{g}(x_i)$ can effectively capture the geometric structure of the neighborhood of point x_i . Based on the defined geometry vector $\mathbf{g}(x_i)$, we can construct a neural network to compute the representation of geometry vector $\mathbf{g}(x_i)$ by multiple layers of the nonlinear transformations. The constructed neural network architecture is shown in Fig. 2. The key advantage of employing the neural network [11, 5] is that we can map

the geometry vector $\mathbf{g}(x_i)$ to the nonlinear feature space to form a nonlinear spectral representation.

The constructed neural network can map the geometry vector $\mathbf{g}(x_i) \in \mathbb{R}^{m \times 1}$ to the output $\mathbf{h}_i^K \in \mathbb{R}^{r \times 1}$, where m and r are the dimensions of the geometry vector and the output, respectively, K is the number of the layers. And each neuron in layer k is connected to all neurons in layer $k + 1$. The output of layer $k + 1$ can be represented as:

$$\mathbf{h}_i^{k+1} = f_{k+1}(\mathbf{h}_i^k) = \varphi(\mathbf{W}^k \mathbf{h}_i^k + \mathbf{b}^k) \quad (6)$$

where $f_{k+1}(\mathbf{h}_i^k)$ is the nonlinear mapping function in layer $k + 1$, \mathbf{h}_i^k is the neuron in layer k , $\varphi(\mathbf{x})$ is the nonlinear activation function. The nonlinear representation of the geometry vector $\mathbf{g}(x_i)$ across K layers, $F_K(\mathbf{g}(x_i))$, is:

$$F_K(\mathbf{g}(x_i)) = f_K(f_{K-1}(\dots, f_2(\mathbf{g}(x_i)))) \quad (7)$$

The matrices \mathbf{W} and \mathbf{b} are the weights and biases of all layers in the neural network, $\mathbf{W} = [\mathbf{W}^1, \mathbf{W}^2, \dots, \mathbf{W}^{K-1}]$ and $\mathbf{b} = [\mathbf{b}^1, \mathbf{b}^2, \dots, \mathbf{b}^{K-1}]$, where \mathbf{W}^k and \mathbf{b}^k are the weight and bias associated with the connection between layer k and layer $k + 1$, respectively.

In [16], for each point x_i on the shape, matrix \mathbf{A} linearly maps the geometry vector $\mathbf{g}(x_i)$ to the linear feature space. Nonetheless, since there are usually large deformations with the shape, the linear mapping function cannot discriminatively characterize the shape well. From Eq. (6), one can see that in our work we employ the neural network to nonlinearly map the geometry vector to the nonlinear feature space. Compared to the linear mapping function in [16], the nonlinear mapping function $F_K(\mathbf{g}(x_i))$ can better characterize the manifold that the geometry vector $\mathbf{g}(x_i)$ of the shape lies on.

3.2. Learned binary spectral shape descriptor

Before learning the binary spectral shape descriptor, we first construct the training samples to train the constructed neural network. Let $B_r(x_i)$ be the ball of radius r centered at point x_i , where r is set to 2% of the average intrinsic shape diameter. For point x_i , the point in the geodesic metric ball $B_r(x_i)$ is defined as the positive sample x_{i+} . Also, the point in $B_r(\eta(x_i))$ is defined as the positive sample of point x_i , where η is the transform between a pair of matched shapes. The point out of the ball is defined as the negative sample x_{i-} . We use the farthest point sampling (FPS) method [9] with the geodesic distance to select some reference points on the shape. For each selected point, we randomly chose 50 positive and negative points to form the pairs of positive and negative points, respectively. Then we can use the pairs of positive geometry vectors $(\mathbf{g}(x_i), \mathbf{g}(x_{i+}))$ and the pairs of negative geometry vectors $(\mathbf{g}(x_i), \mathbf{g}(x_{i-}))$ as the inputs to the constructed neural network.

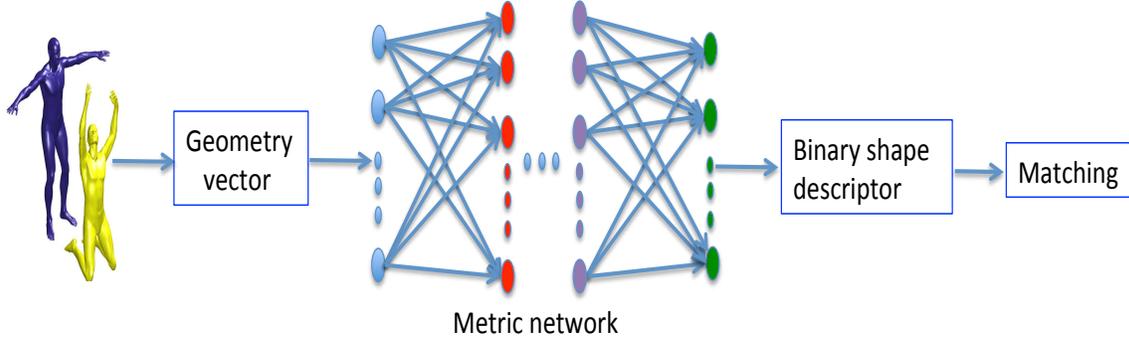


Figure 1. The shape matching framework with the proposed binary spectral shape descriptor. The geometry vectors of the points on a pair of shapes are used as the inputs to the metric network to form a nonlinear spectral representation. In the constructed metric network, the outputs of the pairs of positive points are required to be as similar as possible, the outputs of the pairs of negative points are required to be as dissimilar as possible, and the errors between the real-valued outputs of the network and their binary outputs are encouraged to be as small as possible.

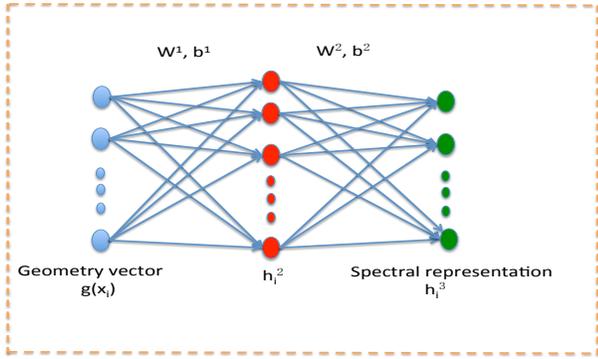


Figure 2. The network used in our method. The input to the network is the geometry vector, the hidden layer and the output are \mathbf{h}_i^2 and \mathbf{h}_i^3 , respectively. Here \mathbf{W}^1 , \mathbf{W}^2 , \mathbf{b}^1 and \mathbf{b}^2 are the parameters to be learned in our constructed network.

The binary spectral shape descriptor can be obtained by binarizing the outputs of the neural network as follows:

$$\mathbf{b}_i = \text{sgn}(\mathbf{h}_i^K) \quad (8)$$

where \mathbf{b}_i is the binary vector associated with point x_i on the shape, $\text{sgn}(v)$ is 1 if $v > 0$ and 0 otherwise. We formulate the following objective function to learn the parameters of the neural network:

$$\begin{aligned} J(\mathbf{W}, \mathbf{b}) = & \underset{\mathbf{W}, \mathbf{b}}{\text{argmin}} \frac{\alpha}{M} \sum_{i=1}^N \sum_{j \in x_{i+}} \frac{1}{2} \|\mathbf{h}_i^K - \mathbf{h}_j^K\|_2^2 - \\ & \frac{1-\alpha}{M} \sum_{i=1}^N \sum_{j \in x_{i-}} \frac{1}{2} \|\mathbf{h}_i^K - \mathbf{h}_j^K\|_2^2 + \frac{\lambda}{N} \sum_{i=1}^N \frac{1}{2} \|\mathbf{b}_i - \mathbf{h}_i^K\|_2^2 \\ & + \frac{1}{2} \gamma \|\mathbf{W}\|_F^2 \end{aligned} \quad (9)$$

where M is the number of all positive/negative training pairs, \mathbf{b}_i and \mathbf{h}_i^K are the binary and real-valued outputs, respectively, $0 \leq \alpha \leq 1$ controls the tradeoff between the

distances from the positive samples and the negative samples, parameters λ and γ are the positive scalars.

In the proposed learning model in Eq. (9), the first two terms minimize the distances between the outputs associated with the positive samples and simultaneously maximize the distances between the outputs associated with the negative samples so that the spectral shape descriptor of x_i is as close as possible to the descriptor of x_{i+} and is as far as possible to the descriptor of x_{i-} . In order to learn the binary descriptor, we furthermore enforce the the binary outputs of the network to be as close as possible to the real-valued outputs of the network such that the quantization loss is minimized.

It is noted that for each pair of positive/negative samples x_i and x_j , the distance between the outputs of the network, \mathbf{h}_i^K and \mathbf{h}_j^K , can be re-written as:

$$\|\mathbf{h}_i^K - \mathbf{h}_j^K\|_2 = \|F_K(\mathbf{g}(x_i)) - F_K(\mathbf{g}(x_j))\|_2. \quad (10)$$

From Eq. (10), one can see that the nonlinear mapping function F_K in our constructed neural network can transfer the Euclidean distance between the geometry vectors $\mathbf{g}(x_i)$ and $\mathbf{g}(x_j)$ to the Euclidean distance between outputs of the network. In [16], with the linear mapping matrix \mathbf{A} , the Euclidean distance between the spectral descriptors is equal to the Mahalanobis distance between the corresponding geometry vectors. Thus, learning the linear mapping matrix can be converted into the linear Mahalanobis metric learning problem [24]. Compared to the linear Mahalanobis metric learning method in [16] that linearly maps the geometry vectors to the linear feature space, our constructed neural network can seek a nonlinear mapping function to characterize the high dimensional geometry vector space better.

To solve the optimization problem in Eq. (9), we employ the back-propagation method to learn parameters \mathbf{W} and \mathbf{b} in the constructed neural network. Since each term in Eq. (9) can be optimized separately, we first define the following

functions:

$$\begin{aligned} J_1(\mathbf{h}_i^K, \mathbf{h}_j^K) &= \frac{1}{2} \|\mathbf{h}_i^K - \mathbf{h}_j^K\|_2^2 \\ J_2(\mathbf{b}_i, \mathbf{h}_i^K) &= \frac{1}{2} \|\mathbf{b}_i - \mathbf{h}_i^K\|_2^2. \end{aligned} \quad (11)$$

The partial derivatives of the objective function $J(\mathbf{W}, \mathbf{b})$ with respect to \mathbf{W}^k and \mathbf{b}^k can be computed as:

$$\begin{aligned} \frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial \mathbf{W}^k} &= \frac{\alpha}{M} \sum_{i=1}^N \sum_{j \in x_{i+}} \frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{W}^k} - \\ &\frac{1-\alpha}{M} \sum_{i=1}^N \sum_{j \in x_{i-}} \frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{W}^k} + \frac{\lambda}{N} \sum_{i=1}^N \frac{\partial J_2(\mathbf{b}_i, \mathbf{h}_i^K)}{\partial \mathbf{W}^k} \\ &+ \gamma \mathbf{W}^k \end{aligned} \quad (12)$$

$$\begin{aligned} \frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial \mathbf{b}^k} &= \frac{\alpha}{M} \sum_{i=1}^N \sum_{j \in x_{i+}} \frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{b}^k} - \\ &\frac{1-\alpha}{M} \sum_{i=1}^N \sum_{j \in x_{i-}} \frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{b}^k} + \frac{\lambda}{N} \sum_{i=1}^N \frac{\partial J_2(\mathbf{b}_i, \mathbf{h}_i^K)}{\partial \mathbf{b}^k}. \end{aligned} \quad (13)$$

Based on the chain rule of the partial derivative, $\frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{W}^k}$ can be re-written as:

$$\begin{aligned} \frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{W}^k} &= \frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{s}_i^{k+1}} \frac{\partial \mathbf{s}_i^{k+1}}{\partial \mathbf{W}^k} \\ &+ \frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{s}_j^{k+1}} \frac{\partial \mathbf{s}_j^{k+1}}{\partial \mathbf{W}^k} \end{aligned} \quad (14)$$

where $\mathbf{s}_i^{k+1} = \mathbf{W}^k \mathbf{h}_i^K + \mathbf{b}_i^k$, $k = 1, 2, \dots, K-1$. Similarly, the partial derivative of $J_2(\mathbf{b}_i, \mathbf{h}_i^K)$ with respect to \mathbf{W}^k can be represented as:

$$\frac{\partial J_2(\mathbf{b}_i, \mathbf{h}_i^K)}{\partial \mathbf{W}^k} = \frac{\partial J_2(\mathbf{b}_i, \mathbf{h}_i^K)}{\partial \mathbf{s}_i^{k+1}} \frac{\partial \mathbf{s}_i^{k+1}}{\partial \mathbf{W}^k}. \quad (15)$$

We denote $\frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{s}_i^{k+1}}$, $\frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{s}_j^{k+1}}$ and $\frac{\partial J_2(\mathbf{b}_i, \mathbf{h}_i^K)}{\partial \mathbf{s}_i^{k+1}}$ by the errors $\delta_{k+1,i}^{1,K}$, $\delta_{k+1,j}^{1,K}$ and $\delta_{k+1,i}^{2,K}$, respectively. For $k = K-1$, the errors $\delta_{k+1,i}^{1,K}$, $\delta_{k+1,j}^{1,K}$ and $\delta_{k+1,i}^{2,K}$ can be represented as:

$$\begin{aligned} \delta_{K,i}^{1,K} &= (\mathbf{h}_i^K - \mathbf{h}_j^K) \bullet \varphi'(\mathbf{s}_i^K) \\ \delta_{K,j}^{1,K} &= (-\mathbf{h}_i^K + \mathbf{h}_j^K) \bullet \varphi'(\mathbf{s}_j^K) \\ \delta_{K,i}^{2,K} &= (-\mathbf{b}_i + \mathbf{h}_i^K) \bullet \varphi'(\mathbf{s}_i^K) \end{aligned} \quad (16)$$

where $\varphi'(\mathbf{s}_i^K)$ is the derivative of the activation function in the output layer and \bullet denotes the element-wise multiplication. For layer $k = K-2, K-3, \dots, 1$, with the back-propagation method, the errors $\delta_{k+1,i}^{1,K}$, $\delta_{k+1,j}^{1,K}$ and $\delta_{k+1,i}^{2,K}$

can be represented as:

$$\begin{aligned} \delta_{k+1,i}^{1,K} &= ((\mathbf{W}^{k+1})^T \delta_{k+2,i}^{1,K}) \bullet \varphi'(\mathbf{s}_i^{k+1}) \\ \delta_{k+1,j}^{1,K} &= ((\mathbf{W}^{k+1})^T \delta_{k+2,j}^{1,K}) \bullet \varphi'(\mathbf{s}_j^{k+1}) \\ \delta_{k+1,i}^{2,K} &= ((\mathbf{W}^{k+1})^T \delta_{k+2,i}^{2,K}) \bullet \varphi'(\mathbf{s}_i^{k+1}). \end{aligned} \quad (17)$$

Thus, $\frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{W}^k}$ and $\frac{\partial J_2(\mathbf{b}_i, \mathbf{h}_i^K)}{\partial \mathbf{W}^k}$ can be calculated as:

$$\begin{aligned} \frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{W}^k} &= \delta_{k+1,i}^{1,K} (\mathbf{h}_i^k)^T + \delta_{k+1,j}^{1,K} (\mathbf{h}_j^k)^T \\ \frac{\partial J_2(\mathbf{b}_i, \mathbf{h}_i^K)}{\partial \mathbf{W}^k} &= \delta_{k+1,i}^{2,K} (\mathbf{h}_i^k)^T. \end{aligned} \quad (18)$$

Similarly, $\frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{b}^k}$ and $\frac{\partial J_2(\mathbf{b}_i, \mathbf{h}_i^K)}{\partial \mathbf{b}^k}$ can be represented as:

$$\begin{aligned} \frac{\partial J_1(\mathbf{h}_i^K, \mathbf{h}_j^K)}{\partial \mathbf{b}^k} &= \delta_{k+1,i}^{1,K} + \delta_{k+1,j}^{1,K} \\ \frac{\partial J_2(\mathbf{b}_i, \mathbf{h}_i^K)}{\partial \mathbf{b}^k} &= \delta_{k+1,i}^{2,K}. \end{aligned} \quad (19)$$

By substituting Eqs. (18) and (19) into Eqs. (12) and (13), we can calculate $\frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial \mathbf{W}^k}$ and $\frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial \mathbf{b}^k}$. Then \mathbf{W}^k and \mathbf{b}^k can be updated with the gradient descent algorithm. The optimization algorithm of the objective function (9) is summarized in Algorithm 1.

Once the weight \mathbf{W} and the bias \mathbf{b} are learned, we can use Eqs. (7) and (8) to obtain the binary spectral shape descriptor \mathbf{b}_i . Then the Hamming distance between the learned binary spectral shape descriptors can be computed for shape matching. Since the calculation of the Hamming distance between the binary vectors can be implemented by the bitwise XOR operator, the matching process is very fast.

4. Experimental Results

4.1. Experimental settings

We test our proposed method on the SCAPE [2] and TOSCA [7] datasets. The SCAPE dataset only consists of 3D human shapes, each shape with the 70 pose changes. Following the setting in [16], the shapes are also re-scaled to have about 10000 vertices and the vertex-wise correspondences are kept. In the TOSCA dataset, 3D shapes with deformations are from 7 classes: centaur, david, dog, horse, michael, cat and victoria. For each class of shapes there are different nearly iso-metric deformations. In order to reduce computational complexity and storage space, all shapes are downsampled to have 10000 vertices with the compatible triangulations and the same vertex-wise correspondences.

We compute the first 300 eigenvalues and the corresponding eigenvectors of the Laplace-Beltrame operator on each shape for a fair comparison to the learned spectral

Algorithm 1 Training algorithm of the proposed binary spectral shape descriptor learning model .

Input: training sample \mathbf{x}_i ; the set of positive samples \mathbf{x}_{i+} ; the set of negative samples \mathbf{x}_{i-} ; layer K of the neural network; weight α ; regularization parameters λ and γ ; learning rate β .

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

For $z = 1, 2, \dots, Z$:

1. Compute the outputs of the neural network with forward propagation for all input geometry vectors $\mathbf{g}(x_i)$;
2. For $k = K - 1, K - 2, \dots, 1$
 Compute $\frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial \mathbf{W}^k}$ with Eqs. (18) and (12);
 Compute $\frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial \mathbf{b}^k}$ with Eqs. (19) and (13);
3. Update \mathbf{W}^k and \mathbf{b}^k for $k = 1, 2, \dots, K - 1$:

$$\mathbf{W}^k = \mathbf{W}^k - \beta \frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial \mathbf{W}^k};$$

$$\mathbf{b}^k = \mathbf{b}^k - \beta \frac{\partial J(\mathbf{W}, \mathbf{b})}{\partial \mathbf{b}^k}.$$

Output \mathbf{W}^k and \mathbf{b}^k until the values of $J(\mathbf{W}, \mathbf{b})$ in the adjacent iterations are smaller than the setting threshold.

shape descriptor in [16]. The discretization of the Laplace-Beltrami operator is implemented by employing the cotangent scheme in [17]. And 150 segments are used in the cubic B-spline basis function to compute the geometry vector, i.e., $m = 150$. In the proposed method, the neural networks are empirically set as 150-100-80-16, 150-100-80-32 and 150-100-80-64 to form the corresponding 16, 32 and 64-bit binary shape descriptors. Moreover, in Eq. (9), parameters α , λ and γ are set to 0.25, 0.06 and 0.001, respectively.

4.2. Comparison evaluation

In this subsection, we evaluate our proposed learned binary spectral shape descriptor in terms of matching performance and computational time. The shape correspondence experiments are conducted on two benchmark datasets, i.e., SCAPE dataset [2] and TOSCA dataset [7].

4.2.1 Matching performance evaluation

We denote our proposed learned binary spectral shape descriptor by LBSSD. For HKS [21], WKS [4] and the learned optimal spectral shape descriptor [16], we employ the local sensitive hashing method [1] to form the binary shape descriptors for shape correspondence. We denote the binary HKS, WKS and optimal spectral shape descriptor with the local sensitive hashing method by HKS-LSH [21, 1], WKS-LSH [4, 1] and OSSD-LSH [16, 1], respectively. Following the evaluation criteria in [16], the cumulative match characteristic (CMC) is used to evaluate the performance of these binary local shape descriptors for the shape correspondence

task. The CMC is the probability of the correct matches that occurs in the top c matches. Given the top c matches, the hit rate calculates the percentage of the positive points associated with the ground truth in the top c matching points. The hit rate is a monotonically increasing function of the top match number c .

For the SCAPE dataset [2], we sample the points as described in Section 3.2 to form 99550 positive/negative pairs to train the neural network. The remaining shapes are used for testing. With the trained neural network, we form the 16, 32 and 64-bit binary shape descriptors. The proposed learned binary spectral shape descriptors with 16, 32, and 64 bits are compared to the three binary shape descriptors for shape correspondence. The CMC curves in the cases of 16, 32 and 64 bits are plotted in Fig. 3. It is noted that in Fig. 3 the X-axis represents the percentage of the top matches in the whole matches. From this figure, one can see that the proposed binary shape descriptor LBSSD is superior to the three binary shape descriptors. Since the dimension of the 64-bit binary shape descriptor is discriminative enough to describe the local geometric structure of the shape, one can see that the hit rate of our proposed binary shape descriptor is slightly higher than that of OSSD-LSH. Nonetheless, the hit rate at the first matching of our proposed LBSSD is much higher than that of OSSD-LSH. Moreover, the hit rate of our proposed binary shape descriptor LBSSD is higher than those of HKS-LSH and WKS-LSH.

For the TOSCA dataset [7], 98750 pairs of the positive/negative geometry vectors are used to train the neural network. With the learned parameters \mathbf{W} and \mathbf{b} , we can obtain the 16, 32 and 64-bit binary outputs of the neural network, i.e., LBSSD, for shape correspondence. The CMC curves for the binary 3D shape descriptors are plotted in Fig. 4. From this figure, one can see that our proposed LBSSD is also superior to the three 3D shape descriptors in terms of the hit rate. We also demonstrate the correctly matching points within 10% of the shape diameter among the sampled 100 points on a pair of human shapes with the four binary local shape descriptors in Fig. 5. Due to the discriminative power of our proposed LBSSD, the number of the correctly matching points with the LBSSD is greater than those with the HKS-LSH, WKS-LSH and OSSD-LSH methods. The number of the correctly matching points with our proposed LBSSD are 28 while the numbers of the correctly matching points with HKS-LSH, WKS-LSH and OSSD-LSH are 8, 5 and 11, respectively.

In the OSSD-LSH method, the learned linear mapping function maps the geometry vectors to the linear feature space, which cannot discriminatively characterize the geometric structures of the neighborhoods of the points on the shape. Due to the quantization loss with the LSH method, the formed binary descriptor OSSD-LSH furthermore weakens the discriminative power of the descriptor.

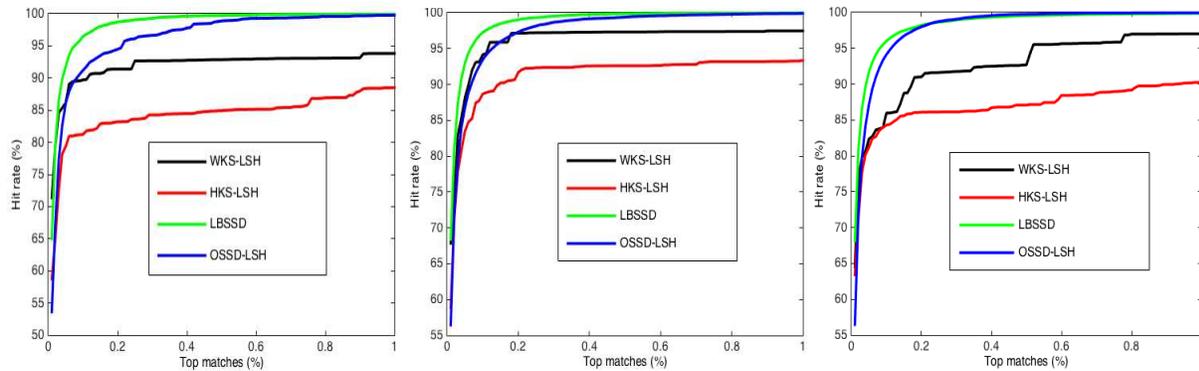


Figure 3. The CMC curves for HKS-LSH, WKS-LSH, OSSD-LSH and the proposed LBSSD on the SCAPE shape dataset: from left to right, 16 bits, 32 bits and 64 bits.

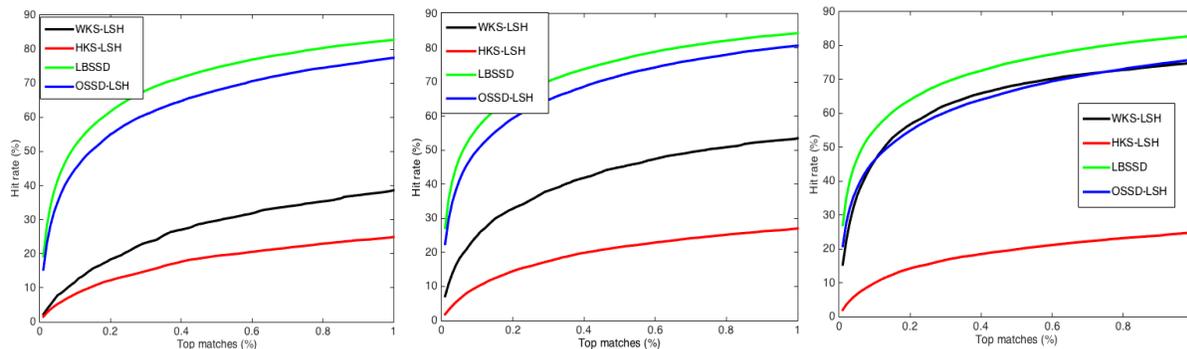


Figure 4. The CMC curves for HKS-LSH, WKS-LSH, OSSD-LSH and the proposed LBSSD on the TOSCA shape dataset: from left to right, 16 bits, 32 bits and 64 bits.

In our proposed method, we employ the neural network to form the spectral representation of the shape and learn the binary descriptor with the nonlinear metric learning method simultaneously. The trained neural network can learn a nonlinear mapping function that can not only represent the shape well but also reduce the quantization loss between the binary outputs and real-valued outputs of the network. Thus, the learned binary shape descriptor is discriminative to describe the local geometric structure for the point-wise correspondence. As shown in Figs. 3 and 4, compared to the OSSD-LSH method, the proposed LBSSD can obtain better performance.

4.2.2 Computational time evaluation

The proposed method was implemented in Matlab and tested on a Dell mobile workstation with an Intel Core i7 and 8GB memory. We evaluate computational time of our proposed LBSSD on the SCAPE dataset. For choosing 99550 positive/negative pairs to train the constructed neural network, the learning process on this dataset takes about 4.5 min. Average time of forming the LBSSD is about 8.75s. For HKS-LSH, WKS-LSH and OSSD-LSH, each costs about 8.13s, 8.13s and 9.75s, respectively. In terms of the matching accuracy and computational time, our pro-

posed LBSSD is superior to OSSD-LSH. Although our proposed LBSSD is slightly slower than HKS-LSH and WKS-LSH, our proposed LBSSD can obtain better matching performance.

5. Conclusions

In this paper, we proposed a binary 3D shape descriptor for shape correspondence. We first constructed a neural network to form a parametric spectral representation. We then developed a nonlinear metric learning technique to train the constructed neural network to form a binary spectral shape descriptor. Finally, the Hamming distance between the proposed binary spectral shape descriptors is used for shape correspondence. The proposed binary spectral shape descriptor requires less storage space and enables matching fast. We conducted the shape correspondence experiments on the two benchmark SCAPE and TOSCA shape datasets to demonstrate its correspondence performance.

References

- [1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008.

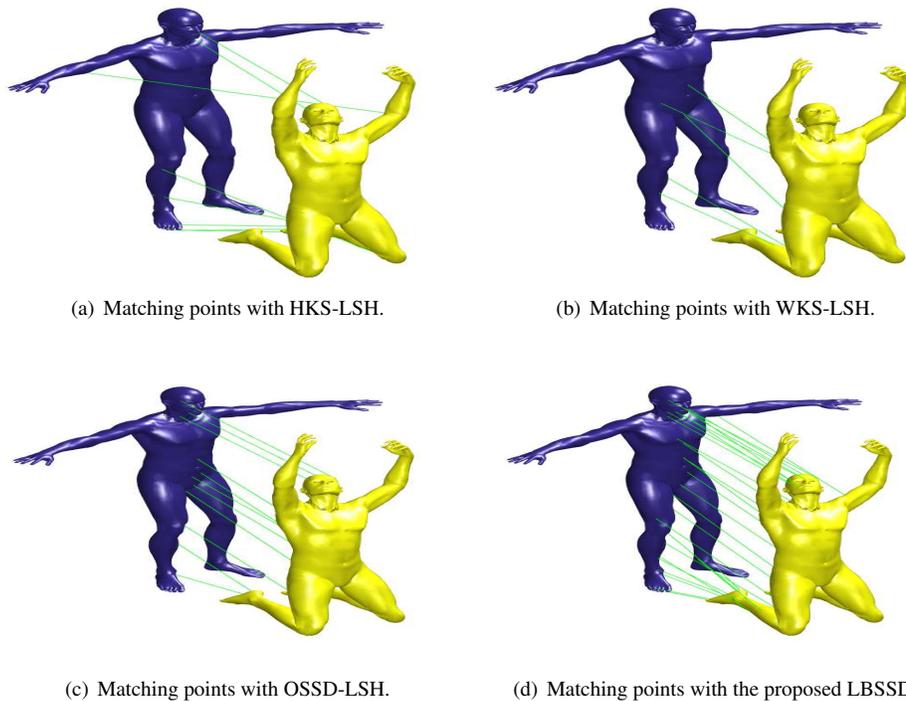


Figure 5. Matching points with the different 32-bit binary shape descriptors are shown with the geodesic distance distortion below 10% of the shape diameter among the sampled 100 points. (a): HKS-LSH, 8 matches; (b): WKS-LSH, 5 matches; (c): OSSD-LSH, 11 matches; (d): LBSSD, 28 matches.

- [2] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. SCAPE: shape completion and animation of people. *ACM Trans. Graphics*, 24(3):408–416, 2005.
- [3] D. Anguelov, P. Srinivasan, H. Pang, D. Koller, S. Thrun, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Advances in Neural Information Processing Systems, Vancouver, British Columbia, Canada*, pages 33–40, 2004.
- [4] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *IEEE International Conference on Computer Vision Workshops, Barcelona, Spain*, pages 1626–1633, 2011.
- [5] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.
- [6] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [7] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Monographs in Computer Science. Springer, 2009.
- [8] W. Chang and M. Zwicker. Automatic registration for articulated shapes. *Computer Graphics Forum*, 27(5):1459–1468, 2008.
- [9] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi. The farthest point strategy for progressive image sampling. *IEEE Trans. Image Processing*, 6(9):1305–1315, 1997.
- [10] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Third Eurographics Symposium on Geometry Processing, Vienna, Austria*, pages 197–206, 2005.
- [11] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.
- [12] Q. Huang, B. Adams, M. Wicke, and L. J. Guibas. Non-rigid registration under isometric deformations. *Computer Graphics Forum*, 27(5):1449–1457, 2008.
- [13] B. Jian and B. C. Vemuri. Robust point set registration using gaussian mixture models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, 2011.
- [14] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *IEEE International Conference on Computer Vision, Beijing, China*, pages 1482–1489, 2005.
- [15] Y. Lipman and T. A. Funkhouser. Möbius voting for surface correspondence. *ACM Trans. Graphics*, 28(3), 2009.
- [16] R. Litman and A. M. Bronstein. Learning spectral descriptors for deformable shape correspondence. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 36(1):171–180, 2014.
- [17] M. Meyer, M. Desbrun, P. Schrder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, pages 35–57. Springer-Verlag, 2002.

- [18] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. J. Guibas. One point isometric matching with the heat kernel. *Computer Graphics Forum*, 29(5):1555–1564, 2010.
- [19] E. Rodolà, S. R. Bulò, T. Windheuser, M. Vestner, and D. Cremers. Dense non-rigid shape correspondence using random forests. In *IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA*, pages 4177–4184, 2014.
- [20] Y. Sahillioglu and Y. Yemez. 3D shape correspondence by isometry-driven greedy optimization. In *IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA*, pages 453–458, 2010.
- [21] J. Sun, M. Ovsjanikov, and L. J. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009.
- [22] A. Tevs, M. Bokeloh, M. Wand, A. Schilling, and H. Seidel. Isometric registration of ambiguous and partial data. In *IEEE Conference on Computer Vision and Pattern Recognition, Miami, Florida, USA*, pages 1185–1192, 2009.
- [23] S. Umeyama. An eigen decomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(5):695–703, 1988.
- [24] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems, Vancouver, British Columbia, Canada*, pages 505–512, 2002.
- [25] F. Zhou and F. D. la Torre. Deformable graph matching. In *IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA*, pages 2922–2929, 2013.