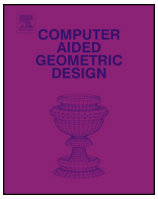
Computer Aided Geometric Design 62 (2018) 192–205

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd



Jointly learning shape descriptors and their correspondence via deep triplet CNNs

Mingjia Chen [a], Changbo Wang [a,*], Hong Qin [b]

[a] School of Computer Science and Software Engineering, East China Normal University, China
[b] Department of Computer Science, Stony Brook University, USA

ARTICLE INFO

Article history:
Available online 22 March 2018

Keywords:
Shape descriptor
Correspondence
Jointly learning
Deep triplet CNNs

ABSTRACT

Informative and discriminative feature descriptors play a crucial role in a large variety of shape analysis and processing applications, including shape matching, retrieval, segmentation, recognition, and synthesis. In this paper, in order to obtain more discriminative shape descriptors in an automated fashion, we develop a novel approach to jointly learning shape descriptors and their correspondence in deep triplet convolutional neural networks (CNNs). We first extract geometrically informative multi-scale features based on classical diffusion theories on manifold and kernel function on shapes. During the training process, the deep triplet CNNs is trained in an unsupervised manner by using a larger pool of extracted multi-scale features serving as the networks' inputs. One key component of our triplet CNNs is a delicate and unique design of the loss function favoring good matches while penalizing mis-matches simultaneously. Our newly-learned shape descriptors have better discriminative power than any existing kernel-based descriptors and could give rise to more accurate shape matching subject to various shape deformation constraints (e.g., isometric deformations and non-isometric deformations). To our best knowledge, this is the first attempt to learn more discriminative shape descriptors and more accurate shape correspondence jointly using the triplet CNNs. Extensive experiments have confirmed that our newly-learned shape descriptors have many attractive properties, including being concise, discriminative, and robust, and we demonstrate the superior performance of our approach over other state-of-the-art methods.

1. Introduction and motivation

In computer graphics and geometric modeling, shape understanding plays an important role to analyze and process 3D shape geometry. In order to understand shapes effectively and in a higher level, one key step is to find the correct matching points between a pair of shapes, serving as an enabling technique at a lower level for shape representation and modeling. This step, also called shape correspondence, could be applied to various shape analysis and processing applications, including shape retrieval (Bronstein et al., 2011), segmentation (Skraba et al., 2010), recognition (Bronstein and Kokkinos, 2010), and synthesis (Kalogerakis et al., 2012). As a result, shape matching is now becoming one of the most active yet challenging research areas. In shape matching, seeking the best possible feature descriptor is of fundamentally significance.

[*] Corresponding author.
E-mail addresses: mjchen611@gmail.com (M. Chen), cbwang@sei.ecnu.edu.cn (C. Wang).

https://doi.org/10.1016/j.cagd.2018.03.002

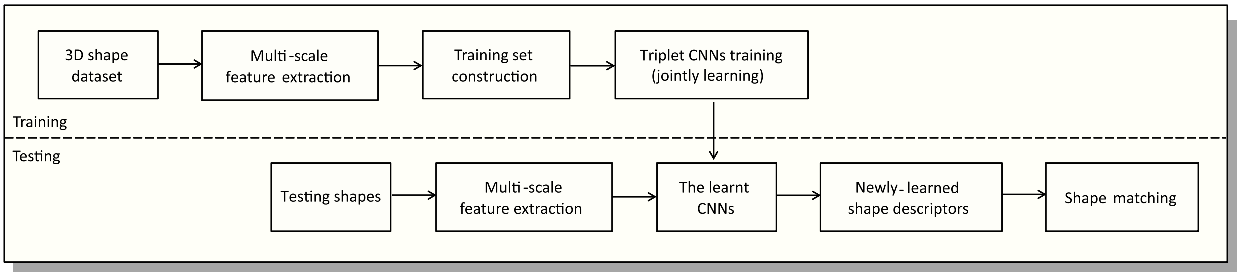

3D shape dataset
Multi-scale feature extraction
Training set construction
Triplet CNNs training (jointly learning)
Training
Testing
Testing shapes
Multi-scale feature extraction
The learnt CNNs
Newly-learned shape descriptors
Shape matching

Fig. 1. The entire process of our approach.

In technical essence, a local shape feature descriptor assigns to each point on the shape a vector in some single- or multi-dimensional feature space representing the local geometric properties of the shape in the vicinity of that point. A global feature descriptor describes the global geometric structure of the entire shape.

In the past few decades, many shape descriptors (Bronstein and Kokkinos, 2010; Aubry et al., 2011; Boscaini et al., 2015) have been proposed and well utilized in shape matching. However, the performance of these descriptors is still far from being satisfactory, especially in complicated matching tasks. The main issue results from the following aspects: First, these shape descriptors do not have enough discriminative power to describe various transformations of 3D shape; Second, by considering the complex topological structure and visibly variational geometry of 3D shapes, one type of feature descriptor can only extract limited information; Third, although some feature descriptors can collect enough information about the point's local region, they are not concise, which leads to the inefficient usage. Therefore, a key problem in shape matching is to construct discriminative feature descriptors that can provide accurate matching results for various deformations of 3D shapes with enhanced generalization capability.

According to the recent trends of image processing and analysis, it is suggested that a deep neural network structure could be employed to extract concise, discriminative, and robust feature representations from the input data (Masci and Meier, 2011; Yan et al., 2014). Inspired by this observation, we resort to the deep learning algorithm to extract powerful shape descriptors automatically. In particular, we develop a novel approach to jointly learning shape descriptors and their correspondence by using the deep triplet convolutional neural networks (CNNs). In our approach, a large pool of geometrically informative multi-scale features are first extracted to characterize each point on the shape. Given these features, we can thus train the deep triplet CNNs in an unsupervised manner. The key component of the deep triplet CNNs is a delicate and unique design of the loss function favoring good matches while penalizing mis-matches simultaneously. In such way, the deep triplet CNNs afford an automatic mechanism that expedite the joint learning of both shape descriptors and their correspondence. The entire process of our approach is shown in Fig. 1. Experimental results confirm that our approach is outperforming existing state-of-the-art approaches.

The main contributions of our article include:

- We articulate a novel approach to jointly learning shape descriptors and their correspondence via deep triplet CNNs.
- Our approach could learn more discriminative shape descriptors, which will give rise to more accurate shape matching subject to various shape deformation constraints (e.g., isometric deformations and non-isometric deformations).
- Our newly-learned shape descriptors have many attractive properties, including being concise, discriminative, and robust.

The rest of this paper is organized as follows: Section 2 reviews related works. An overview of the proposed approach appears in Section 3. Section 4 discusses how to jointly learn shape descriptors and their matching via deep triplet CNNs, and the learning process' major components are documented in details. Then we demonstrate impressive performance through extensive experiments in Section 5. Finally, we conclude the paper in Section 6 and point out possible future research directions.

2. Related works

As discussed previously, one of the key issues in shape correspondence is to find discriminative and robust shape feature descriptors. There have been several prior works relevant to shape feature descriptors, which mainly follow two methods: 1) constructing better shape feature descriptors in a handcrafted way, and 2) developing approaches to automatically learn feature descriptors (based on deep learning techniques). In this section, we shall briefly review the related works from these two aspects.

Shape feature descriptors. Early research works on feature descriptors mainly focus on invariance under global Euclidean transformations (e.g., rigid deformation), including shape context (Belongie et al., 2000), spin image (Johnson and Hebert, 2002), and integral volume descriptors (Manay et al., 2006). In the past decade, significant efforts have been invested in extending the invariance properties of shapes to non-rigid deformation. Some of the classical rigid shape feature descriptors are extended to the non-rigid case by replacing the Euclidean metric with its geodesic counterpart (Hamza and Krim, 2003; Elad and Kimmel, 2003).

Recently, the emerging field of diffusion geometry provides an effective method for the geometric analysis of non-rigid shapes. The research of diffusion geometry is based on the theoretical works by Bérard et al. (1994). Later on, Coifman and Lafon (2006) propose to use the eigenvalues and eigenvectors of the Laplace–Beltrami operator associated with the shape to construct diffusion distances. These distances as well as other diffusion geometry construct new shape descriptors that are significantly more robust than geodesic counterparts (Bronstein et al., 2009; Mémoli, 2009). Diffusion geometry provides an intuitive interpretation of many shape properties for spatial frequencies and allows us to use standard harmonic analysis tools.

Furthermore, with recent advancements in the discretization of the Laplace–Beltrami operator, there have been several efficient and robust numerical and computational tools. Lévy (2006) first explores these methods in the context of shape processing. There have also been several attempts to construct shape feature descriptors based on diffusion geometry properties of the shape. Rustamov (2007) proposes to construct the global point signature (GPS), which uses eigenvalues and eigenfunctions of the Laplace–Beltrami operator defined on 3D shape to characterize points. Later on, based on the fundamental solutions of the heat equation, Sun et al. (2009) introduce the heat kernel signature (HKS). And another physically-inspired descriptor, the wave kernel signature (WKS) (Aubry et al., 2011), is proposed as a solution to combat the excessive sensitivity of the HKS to low-frequency information. Both HKS and WKS have gained attention because of their multi-scale property and invariance to isometric deformations. As of now, these descriptors lie in the foundation of shape matching applications (Bronstein and Kokkinos, 2010; Bronstein et al., 2010).

Feature learning. Over the past decades, due to the fact that conventional hand-crafted feature descriptors might not be discriminative enough to various transformations of 3D shapes, feature-learning-based approaches start to attract attention of many researchers. These learning-based methods provide an efficient way to construct more discriminative feature descriptors in an automated fashion.

In the research of Shape Google (Ovsjanikov et al., 2009; Bronstein et al., 2011), the bag-of-features (BoF) method is used to extract a frequency histogram of geometric words for shape analysis (Eitz et al., 2012, 2011; Hu and Collomosse, 2013). Despite the inception of BoF, the authors also introduce a similarity-sensitive hashing method to achieve more discriminative and compact representations. Castellani et al. (2008, 2011) propose a middle-level feature extraction scheme through learning hidden states from local basic descriptors for shape matching. In this approach, local patches are modeled as a stochastic process through a set of circular geodesic pathways and learned by using hidden Markov model.

With the development of diffusion geometry, the Laplacian-based descriptors achieve state-of-the-art performance. Nonetheless, they frequently focus on different properties of shape. In order to provide a generic feature descriptor for shapes, Litman and Bronstein (2014) develop a learning scheme to construct optimized spectral descriptors for deformable shape correspondence. Also, in order to collect rich information from the input data and select the most significant feature, Barra and Biasotti (2013) introduce a method using multiple kernel learning to find optimal linear combination of kernels in classification and retrieval tasks.

Recently, the studies of the deep learning algorithm show that deep neural networks have impressive capabilities in extracting effective and robust representations from the low-level features (Bengio, 2009; Hinton, 2010). As a special case, CNNs (Ranzato et al., 2007; Krizhevsky et al., 2012; Farabet et al., 2013) have been proved to be an effective way for extracting concise, discriminative, and robust features and could be applied to various applications. In this paper, we propose to jointly *learn* shape descriptors and their correspondence from a large pool of multi-scale features with deep triplet CNNs in order to continue to advance the research frontier of feature extraction and shape matching.

3. Method overview

Our goal is to obtain a discriminative and robust shape descriptor automatically, which could give rise to more accurate shape matching subject to various shape deformation constraints. The research is propelled by deep learning approaches. In this article, we make first attempt to jointly learn shape descriptors and their correspondence via deep triplet CNNs. The network supervised by triplet loss to map the input feature space into a newly-learned descriptor space, where the Euclidean distance of descriptors is directly related to the correspondence of points.

As shown in Fig. 2, the deep triplet CNNs comprises three identical copies of the same feed-forward CNNs which share the same parameters. The input of the network is a triplet of multi-scale features, including the feature of anchor point, corresponding point, and non-corresponding point. When fed with these three features, the deep triplet CNNs outputs two intermediate values, which are Euclidean distance between newly-learned descriptors of corresponding points and the one between descriptors of non-corresponding points. Then, we use these two Euclidean distances in the newly-learned descriptor space to define the triplet loss function.

During the training process, by effectively reducing the distance of good-matching descriptors while increasing the distance of mis-matching descriptors, the deep triplet CNNs guarantees that good-matches are closer than mis-matches in the newly-learned descriptor space. With the trained network, a concise, discriminative, and robust shape descriptor could be automatically extracted, which in turn results in more accurate shape matching subject to various shape deformation constraints.

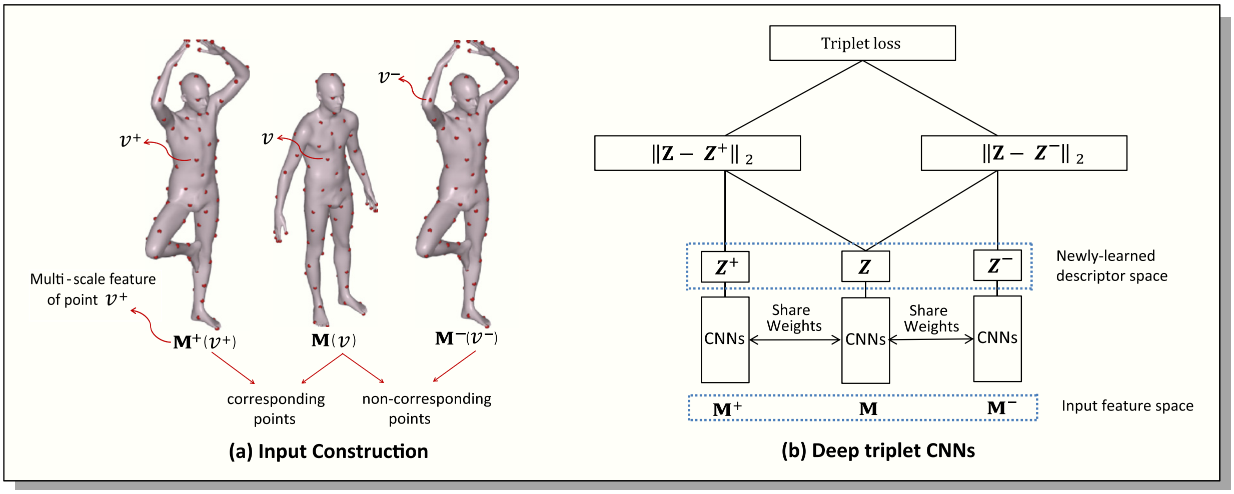


Fig. 2. Overview of the joint learning method based on the deep triplet CNNs (see Section 4 for details). (a) Input construction. Given a large pool of 3D shapes, we first extract multi-scale features for each point. A triplet of multi-scale features, including the feature of anchor point, corresponding point, and non-corresponding point, is fed into deep triplet CNNs for training. (b) Architecture of the deep triplet CNNs. The network is composed of three identical copies of the same feed-forward CNNs which shares the same parameterization. Supervised by the triplet loss, the deep triplet CNNs can transform the input feature space into a newly-learned descriptor space, where the Euclidean distance of newly-learned descriptors is directly related to the correspondence of points.

4. The new joint learning method based on triplet CNNs

In this section, we will present how to jointly learn shape descriptors and their correspondence by using deep triplet CNNs. First, we extract a large pool of geometrically informative multi-scale features to characterize each point on the shape. Second, we present the architecture of the deep triplet CNNs that is used to jointly learn shape descriptors and their correspondence from massive multi-scale features. Third, we show the details of joint learning method. Finally, we make a brief description of the training set construction.

4.1. Multi-scale feature extraction

For a 3D shape, the description value of the point does not provide sufficient discriminative information especially for the low-level descriptor. Usually, geometric properties of local region around the point provide much more information. Thus, we develop a geometrically informative multi-scale feature to characterize each point on a shape. The proposed multi-scale features capture geometric properties and hierarchical information of local region around the point.

Local region feature. Let v_i, $i = 1, \cdots, n$ be the point on shape X. The local region $R(i)$ is the bi-harmonic metric (Lipman et al., 2010) area of radius r centered at v_i. We first construct a local region feature to encode the geometric characteristics of local region $R(i)$. Let $\{H(v_j)\}_{j=1,\cdots,t}$ be the heat kernel signatures computed at every point inside the local region $R(i)$. We use matrix $\mathbf{M}_{his}(i)$ to give an estimation of probability distribution of HKS values at all vertices in the local region $R(i)$ and at all time samples. At each time sample, $\mathbf{M}_{his}(i)$ is defined based on the probability distribution of HKS at that time sample. In our article, we use histogram to give an estimation of probability distribution of HKS values. That is, given HKS has N_Q time samples in diffusion time and N_B is the number of bins used in the histogram, $\mathbf{M}_{his}(i)$ will be formed as a $N_Q \times N_B$ matrix, where $N_Q = N_B$ (we empirically use 32 samples in diffusion time and 32 bins in histogram). Meanwhile, we compute the covariance matrix $\mathbf{M}_{cov}(i)$ for $\{H(v_j)\}$, $j = 1, \cdots, t$:

$$\mathbf{M}_{cov}(i) = \frac{1}{t-1}\sum_{j=1}^{t}(H(v_j) - \mu)(H(v_j) - \mu)^T \tag{1}$$

where μ is the mean of heat kernel signatures computed in the local region $R(i)$. It has a fixed dimension $N_Q \times N_Q$, which independently of the size of the local region $R(i)$ (as described above, we empirically set $N_Q = 32$). It has been verified in Fang et al. (2015), Tabia and Laga (2015) that $\mathbf{M}_{his}(i)$ and $\mathbf{M}_{cov}(i)$ can efficiently capture geometric properties of local region $R(i)$. Till now, we have got the local region feature $\mathbf{M}_{loc}(i)$ if we define it as

$$\mathbf{M}_{loc}(i) = \lambda\mathbf{M}_{his}(i) + (1 - \lambda)\mathbf{M}_{cov}(i), \tag{2}$$

where λ is empirically set as 0.5.

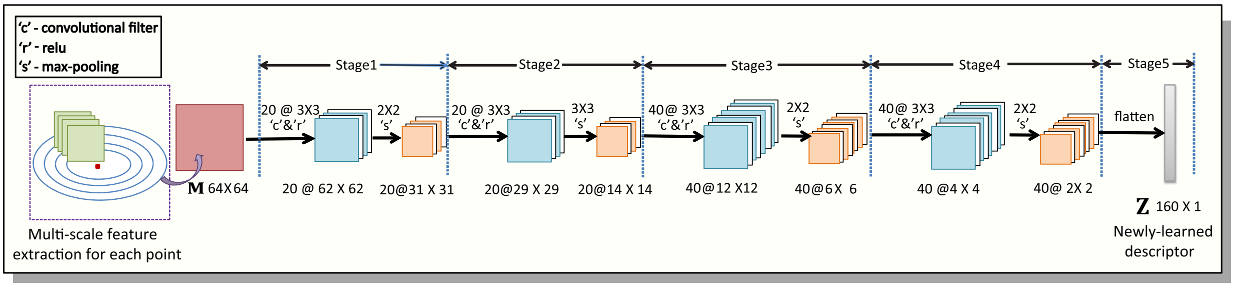


Fig. 3. Architecture of deep CNNs with five stages. Each of the first four stages contains a filter bank module, a nonlinear module with a relu activation function, and a spatial pooling module for sub-sampling. The last stage involves a flattening operation which is used to obtain the newly-learned descriptor of each point on the shape.

Multi-scale feature. In order to take into account the hierarchical information, we propose a multi-scale feature based on the local region feature. For each point v_i, we define several local regions $R_j(i)$, $j = 1, \cdots, l$ and $r_1 < \cdots < r_l$ (we empirically set $l = 4$). Thus, we get four local region features $\mathbf{M}_{loc}(i, j)$, $j = 1, \cdots, 4$ for point v_i. Given these features, we reorganize the four 32×32 features to form a 64×64 multi-scale feature $\mathbf{M}_i$. According to the convolutional characteristics of deep CNNs, the ordering of features would not influence the performance of the deep triplet CNNs.

4.2. Triplet CNNs structure

The multi-scale feature of each point can be used as the input of the deep triplet CNNs. The goal of the deep triplet CNNs is to map the input feature space into a newly-learned descriptor space, where the Euclidean distance of descriptors is directly related to the correspondence of points.

CNNs architecture. In our approach, CNNs can be seen as an end-to-end feature extractor. The multi-scale feature of each point can be used as the input of CNNs. With limited training shapes, the deep CNNs architecture should not be very complex, otherwise with the growth of layer number, the parameters would rapidly increase so as to produce overfit. Thus, the structure of CNNs consists of five major stages, as shown in Fig. 3.

In the first stage, the input multi-scale feature $\mathbf{M}$ passing through twenty 3×3 convolutional kernels $\{\mathbf{W}_i\}_{i=1}^{20}$ and bias values $\{b_i\}_{i=1}^{20}$:

$$\mathbf{Y}_i = \mathbf{W}_i * \mathbf{M} + b_i, i = 1, \cdots, 20, \tag{3}$$

where $*$ indicates the convolution operation and the bias b_i is same for all the convolutional kernels $\{\mathbf{W}_i\}_{i=1}^{20}$. After the convolution, twenty $62 * 62$ maps are generated by passing $\{\mathbf{Y}_i\}_{i=1}^{20}$ through nonlinearity module with a relu activation function. The nonlinearity is operated on each component of $\mathbf{Y}_i$. After the nonlinear module, we down-sample each dimensional of the feature maps by a factor of 2 to generate twenty 31×31 feature maps, denoted as $\left\{\dot{\mathbf{Y}}_i\right\}_{i=1}^{20}$.

In the second stage, we first generate 20 new feature maps $\left\{\mathbf{R}_j\right\}_{j=1}^{20}$ from all first-stage feature maps. In this process, each 29×29 map $\mathbf{R}_j$ is obtained by applying 20 different $3 * 3$ kernels $\left\{\dot{\mathbf{W}}_{ij}\right\}_{j=1}^{20}$ and biases $\dot{b}_j$ to the input feature maps $\left\{\dot{\mathbf{Y}}_i\right\}_{i=1}^{20}$:

$$\mathbf{R}_j = \dot{\mathbf{W}}_{ij} * \dot{\mathbf{Y}}_i + \dot{b}_j, \ j = 1, \cdots, 20. \tag{4}$$

Then, the same nonlinear module and max-pooling operation is conducted, which generate twenty 14×14 maps $\left\{\dot{\mathbf{R}}_j\right\}_{j=1}^{20}$. Max-pooling is performed over a 3×3 pixel window, with stride 2.

In the third stage, 20 input feature matrices $\left\{\dot{\mathbf{R}}_j\right\}_{j=1}^{20}$ passing through 40 different 3×3 convolutional filters $\left\{\ddot{\mathbf{W}}_{ijk}\right\}_{k=1}^{40}$ and bias values $\left\{\ddot{b}_k\right\}_{k=1}^{40}$:

$$\mathbf{G}_k = \ddot{\mathbf{W}}_{ijk} * \dot{\mathbf{R}}_j + \ddot{b}_k, k = 1, \cdots, 40. \tag{5}$$

Then, through the same nonlinear module and 2×2 max-pooling operation, we get 40 down-sampled 6×6 feature maps $\left\{\dot{\mathbf{G}}_k\right\}_{k=1}^{40}$. In the fourth stage, we generate 40 new feature maps $\{\mathbf{P}_t\}_{t=1}^{40}$ from the 40 input feature maps $\left\{\dot{\mathbf{G}}_k\right\}_{k=1}^{40}$:

$$\mathbf{P}_t = \dddot{\mathbf{W}}_{ijkt} * \dot{\mathbf{G}}_k + \dddot{b}_t, t = 1, \cdots, 40, \tag{6}$$

where $\left\{\dddot{\mathbf{W}}_{ijkt}\right\}_{t=1}^{40}$ are 40 different $3 * 3$ kernels and the same bias $\dddot{b}_t$ is added to all the components of $\dddot{\mathbf{W}}_{ijkt} * \dot{\mathbf{G}}_k$. After that, the same nonlinear module and 2×2 max-pooling operation is conducted, which eventually generate 40 down-sampled 2×2 feature maps.

In the last stage, we flatten these feature maps into a concise descriptor **Z**. In the process of extracting **Z**, we can see that each component of **Z** is actually generated by nonlinearly combining (i.e., relu function) and hierarchically compressing (i.e., max-pooling) a subset of the input features. Thus each newly-learned descriptor **Z** actually characterizes specific attributes of a point on the shape. In this process, any two elements from the input multi-scale feature can be nonlinearly combined, no matter how the local region features are organized in the multi-scale feature matrix. For the sake of simplification, we rewrite the descriptor **Z** as a function of **M**:

$$\mathbf{Z} = F_\theta(\mathbf{M}), \tag{7}$$

where F_θ denotes CNNs parameterized by θ (i.e., convolutional kernels $\mathbf{W}_i, \dot{\mathbf{W}}_{ij}, \ddot{\mathbf{W}}_{ijk}$ and $\dddot{\mathbf{W}}_{ijkt}$, and bias values $b_i, \dot{b}_j, \ddot{b}_k$ and $\dddot{b}_t$).

Triplet CNNs structure. The triplet CNNs consists of three identical copies of the same feed-forward CNNs which share the same parameters, as shown in Fig. 2. In this model, CNNs is used as an end-to-end feature extractor. The input of deep triplet CNNs is a triplet $(\mathbf{M}, \mathbf{M}^+, \mathbf{M}^-)$, which $\mathbf{M}, \mathbf{M}^+, \mathbf{M}^-$ are the multi-scale feature of anchor point v, corresponding point v^+ and non-corresponding point v^-, respectively. When fed with these three multi-scale features, the triplet CNNs outputs three newly-learned descriptors $\mathbf{Z}, \mathbf{Z}^+$ and $\mathbf{Z}^-$. Then, we compute Euclidean distances between each of $\mathbf{Z}^+$ and $\mathbf{Z}^-$ against **Z**. According to (7), these two Euclidean distances can be denoted as:

$$d_\theta^+ = \|(F_\theta(\mathbf{M}) - F_\theta(\mathbf{M}^+)\|_2, \tag{8}$$

and

$$d_\theta^- = \|(F_\theta(\mathbf{M}) - F_\theta(\mathbf{M}^-)\|_2. \tag{9}$$

That is, d_θ^+ and d_θ^- encode the Euclidean distances between each of $\mathbf{M}^+$ and $\mathbf{M}^-$ against **M** in the newly-learned descriptor space. Finally, we use the pair of distances d_θ^+ and d_θ^- to define the triplet loss. During the training process, by effectively reducing distances of good-matching features while increasing distances of mis-matching features, the deep triplet CNNs guarantees that good-matches are closer than mis-matches in the newly-learned descriptor space.

4.3. Joint learning method

Given the structure of the deep triplet CNNs, we have to optimize its parameters (i.e., θ) to jointly learn shape descriptor and their correspondence. The deep triplet CNNs should guarantee that descriptors of corresponding points are closer than ones of non-corresponding points in the newly-learned descriptor space. Formally, given T triplets $(\mathbf{M}_t, \mathbf{M}_t^+, \mathbf{M}_t^-)$, $t = 1, \cdots, T$, we want to enforce $d_\theta^-(t) > d_\theta^+(t)$. Therefore, the triplet loss of our deep triplet CNNs is defined as follows:

$$L_\theta = \sum_{t=1}^{T} max(0, d_\theta^+(t) - d_\theta^-(t) + \alpha), \tag{10}$$

where α represents the gap parameters between two distances. We set $\alpha = 0.5$ in the experiment. The objective of training process is $L_\theta \to 0$, that is $d_\theta^+(t) \to 0$ and $d_\theta^-(t) \to \alpha$. In our implementation, the minimization of (10) mainly consists of iterative feed-forward and back-propagation processes. We initialize all convolution kernels with small random values.

Feed-forward. In the feed-forward process, we first pass the input triplet $(\mathbf{M}_t, \mathbf{M}_t^+, \mathbf{M}_t^-)$ through the deep triplet CNNs to get the pair of Euclidean distances $d_\theta^+(t)$ and $d_\theta^-(t)$. Then, we use these two distances to update the triplet loss (10).

Back-propagation. In the back-propagation process, the objective is to propagate backward the triplet loss and reduce it by tuning the parameters layer by layer. According to the sharing parameters mechanism in deep triplet CNNs, we allow the back-propagation operation to update the three same CNNs simultaneously. Here we use the stochastic gradient descent with momentum algorithm to update parameters layer by layer. In this process, we set a learning rate of 0.01 and a weight decay of 0.0005. The feed-forward and back-propagation operations are iteratively conducted until the triplet loss converge or a predefined number of iterations is reached. Fig. 4 shows two examples to illuminate the convergency of our method. In general, our approach can achieve convergency through 150 iterations.

4.4. Training set construction

So far, we have introduced the details of joint learning method. One important part of our approach is the training set construction. These triplets in the training set are contribute to loss reduction, affecting the convergency speed of the training process. So the construction of the training triplets is crucial for the jointly learning process. According to Litman and Bronstein (2014), the construction of the training set follows two aspects:

1) Given the point-to-point correspondence between shapes, we first randomly sample two shapes. Then, for each point (anchor point) on the first shape, we denote the corresponding point on the second sampled shape as good matching point, and a random point on the second shape as mis-matching point.

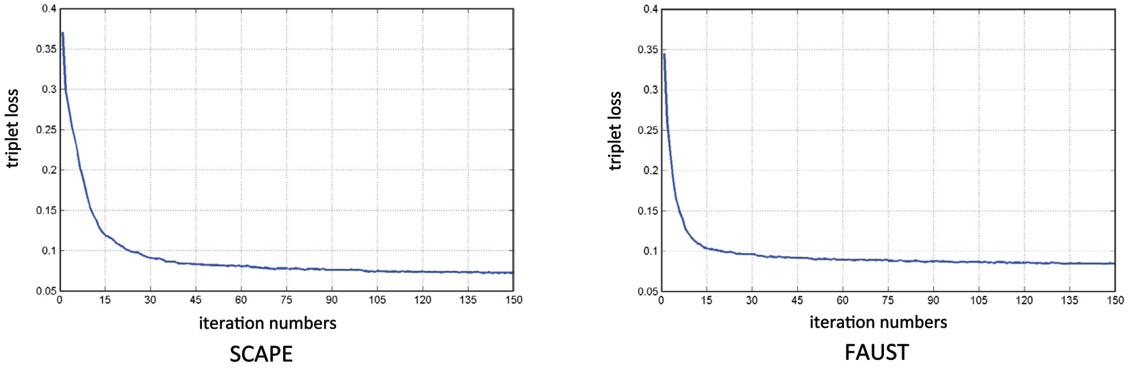


Fig. 4. Triplet loss convergence curves.

2) Let v be a point (anchor point) on a shape, and $r < R$ are the geodesic radii centered at v. We deem all points v^+ lie in the geodesic ball of radius r as good matching points, while deeming mis-matching points all v^- outside the geodesic ball of radius R.

The training triplets are generated by sampling many anchor points, good matching points and mis-matching points on a collection of training shapes. The two sampling strategies above guarantee the discriminative power and the robustness of the newly-learned descriptor.

5. Experimental results and evaluation

5.1. Experimental setup and data

Datasets. To evaluate the performance of our approach, three correspondence benchmarks are used in our experiments, including SCAPE (Anguelov et al., 2005), FAUST (Bogo et al., 2014) and TOSCA (Bronstein et al., 2008) datasets. The SCAPE dataset contains a scanned human figure in 71 different poses, the FAUST dataset consists of 100 scanned human shapes (10 subjects, 10 shapes per subject), and 71 synthetic models of humans and animals are in the TOSCA dataset. To better compare with the other benchmarks, we limit the evaluation only to human shapes (Michael) for the latter dataset. In order to reduce the computational and storage complexity, the shapes in SCAPE and TOSCA are downsampled to 10K points. FAUST shapes contain 6.8K points. All shapes are scaled to unit geodesic diameter. We use the cotangent weight scheme (Meyer et al., 2002) to compute the first 200 eigenvalues and eigenvectors of the Laplace–Beltrami operator on each shape. For the HKS, the time scale is set to be $[1, 10]$ with an interval of 0.1, and the log time base is set to 2. And we use the first 32 samples. For each point, we set the four local region radii to $0.125, 0.25, 0.375$ and 0.5 of the max bi-harmonic distance, respectively.

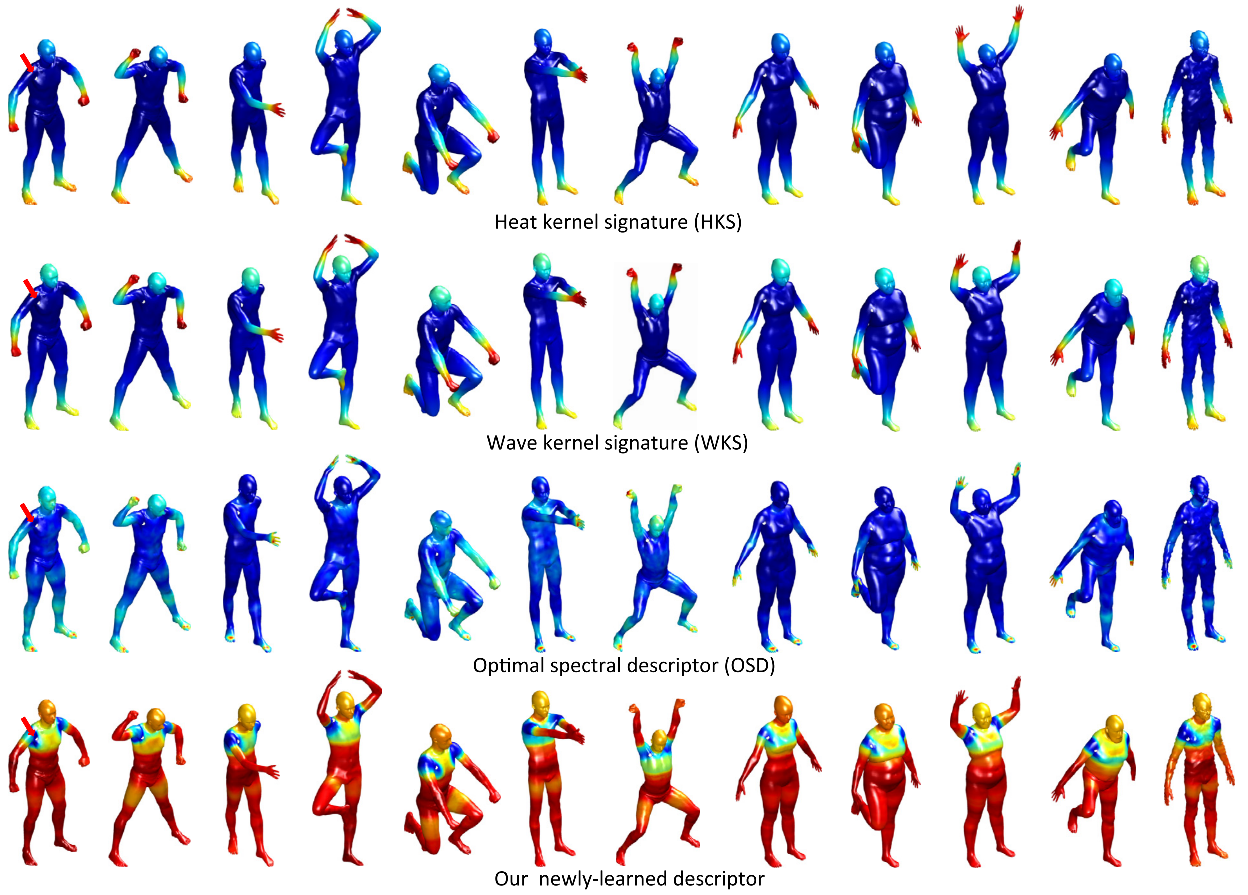
For comparison, we also evaluate the HKS, WKS and OSD descriptors. The HKS time scales are set as described above. The energy levels and variance σ^2 of WKS are set as introduced in Aubry et al. (2011). And the OSD related parameters are set according to Litman and Bronstein (2014). To guarantee the fairness of comparison, Euclidean distance is used for all descriptors.

Training set. For each dataset, the training, validation and testing sets are disjoint. On the SCAPE dataset, we use shapes 20–39 and 50–70 for training, five random shapes from the remaining ones for validation, and the rest for testing. On FAUST, we use subjects 1–7 for training, subject 8 for validation and subjects 9–10 for testing. On the TOSCA set, we use the human shapes (Michael) for testing. For each shape, we use the farthest point sampling (FPS) strategy (Eldar et al., 1997) to sample 2K feature points.

The construction of training set from two aspects (Litman and Bronstein, 2014): 1) We first sample two shapes. Then, for each feature point on the first shape, we deem the corresponding point on the second sampled shape as good matching point, and a random point on the second shape as mis-matching point; 2) For each feature point on a shape, we pair 50 good matching points and 50 mis-matching points on the same shape. These good matches are sampled from the ball of radius r centered at that point. Half of the mis-matches are picked from the ring lying between the radii R and $4R$ around that point; another half are filled with points farther than $4R$. We empirically set the radii r and R to 2% and 4% of the max geodesic distance, respectively. Thus, we generate 143K training triplets on SCAPE dataset and 181K training triplets on FAUST dataset.

5.2. Experiments and analysis

In this subsection, we conduct extensive experiments to show the performance of our newly-learned shape descriptors in shape matching. Qualitative and quantitative descriptor evaluation are done using three criteria, including similarity map,


Heat kernel signature (HKS)

Wave kernel signature (WKS)

Optimal spectral descriptor (OSD)

Our newly-learned descriptor

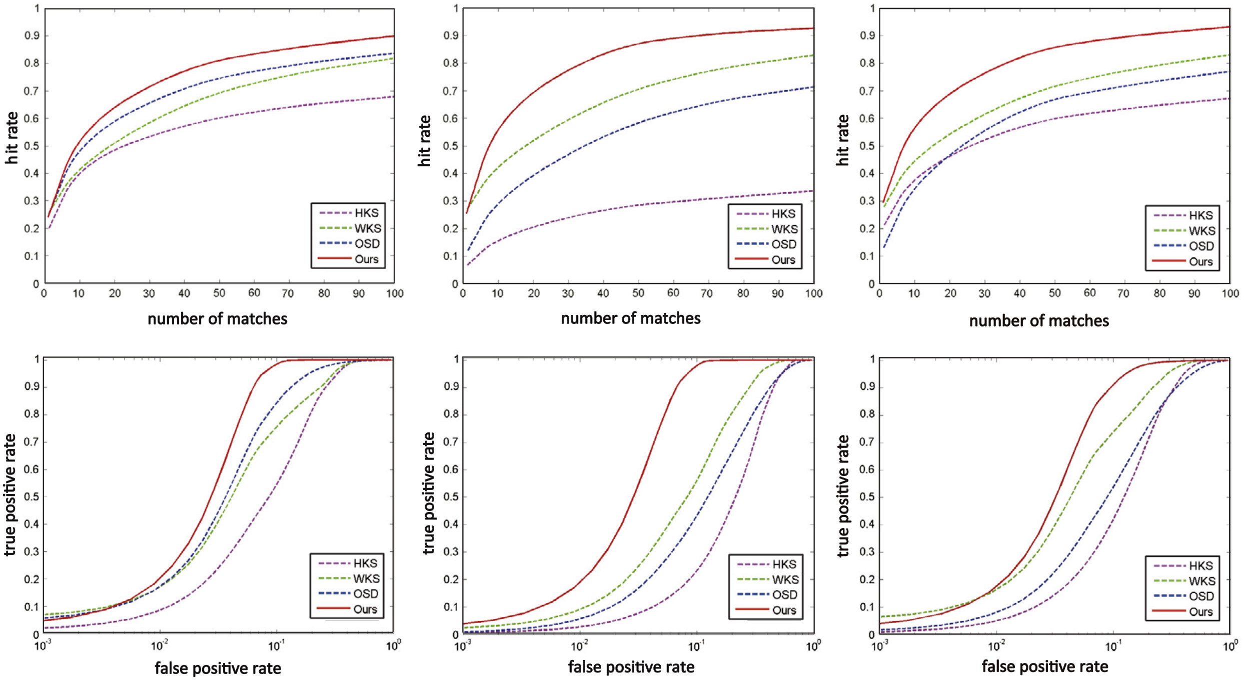
Fig. 5. Normalized Euclidean distance between the descriptor at the reference point on the right shoulder (white sphere pointed by red arrow) and the descriptors computed at rest of the points for different transformations (shown left-to-right: original, 6 near isometric deformations, 4 non-isometric deformations, 1 noise). Dark blue and red colors represent small and large distances, respectively; colors are saturated at the median distance. Ideal descriptors would produce a distance map with a sharp minimum at the corresponding point and no false local minima at other locations. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

cumulative match characteristic (CMC) and receiver operating characteristic (ROC). Similarity map depicts the Euclidean distance in the descriptor space between the descriptor at a reference point and the rest of the points on the same shape. The CMC evaluates the probability of finding the correct matching among the k nearest neighbors in the descriptor space. The ROC measures the percentage of good matches and mis-matches falling below various thresholds of their distance in the descriptor space (true positive and negative rates, respectively).

Similarity map. We first show the qualitative evaluation of our newly-learned descriptors using similarity map, as shown in Fig. 5. It depicts the Euclidean distance in the descriptor space between the descriptor at a reference point (on the right shoulder) and the rest of the points on the same shape as well as its transformations. We observe that the newly-learned shape descriptor manifest better localization and better discriminative power than other kernel-based descriptors. Moreover, our newly-learned shape descriptors are robust to various transformations, including isometric deformations, non-isometric deformations, and noise. In contrast, our method can well represent these shapes.

Descriptor evaluation. To illustrate the discriminative power and robustness of our newly-learned shape descriptors, we measure the performance of newly-learned shape descriptors using the CMC and ROC. As shown in Fig. 6(a), we compare the newly-learned descriptor to HKS, WKS and OSD on the SCAPE dataset, we can see that the newly-learned descriptor exhibits excellent performance (over 80% hit rate at $k = 50$), while other kernel-based descriptors perform significantly worse (e.g., lower than 58% hit rate via using HKS). Additionally, to show the robustness to non-isometric deformations of the newly-learned shape descriptor, we make a comparison with kernel-based descriptors on the FAUST dataset. As shown in Fig. 6(b), we observe that the hit rate can reach 88% at $k = 50$ with the newly-learned descriptor, while only 26% via using HKS. The main reason is that the descriptors learned in CNNs can better represent a shape. Through multilayer convolutions and nonlinear mapping, the descriptor space becomes much easier to be divided.

Generalization capability. In order to demonstrate the generalization capability of our newly-learned descriptors, we conduct an additional experiment to test the performance of the newly-learned descriptor. We applied the net trained on FAUST to the TOSCA test set. As shown in Fig. 6(c), we see that the trained net transfers well to a new dataset and that the



(a) SCAPE (b) FAUST (c) TOSCA

Fig. 6. Performance of different descriptors (HKS, WKS, OSD and Ours) measured using the CMC (first row) and ROC (second row); higher curves represent better performance. (a) First column shows results for the network trained and tested on disjoint sets of the SCAPE dataset. (b) Second column shows results for the network trained and tested on disjoint sets of the FAUST dataset. (c) Third column shows results for a transfer learning experiment where the network has been trained on FAUST and tested to TOSCA.

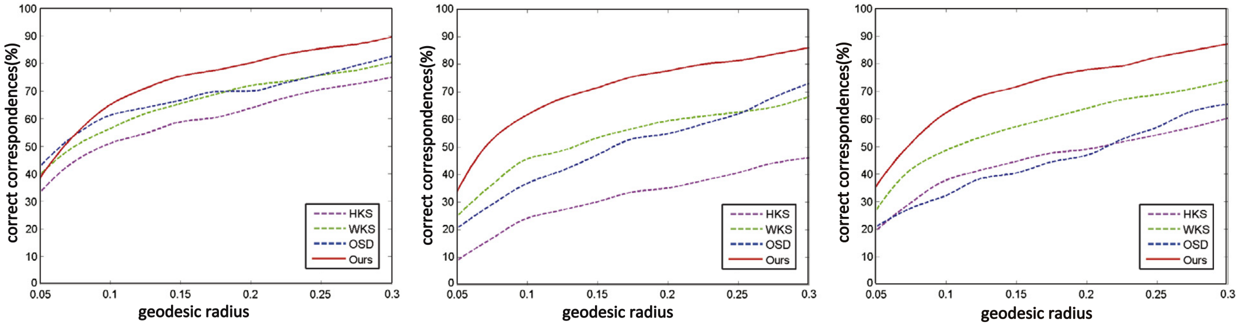
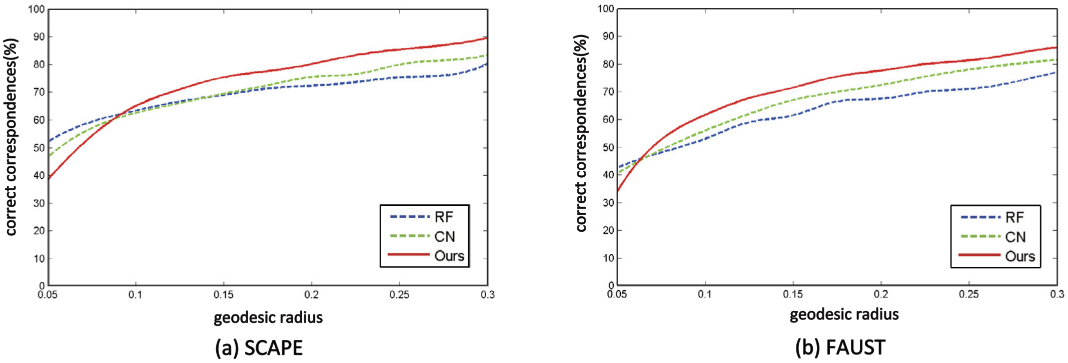


Fig. 7. Performance of different descriptors (HKS, WKS, OSD and Ours) in the shape correspondence task on the SCAPE (left), FAUST (middle) and TOSCA (right) datasets. Higher curves represent better performance.

newly-learned shape descriptor outperforms other kernel-based descriptors. We note that at $k = 50$ the hit rate can reach 86% with the learned descriptor, while only 58% via using HKS. These results prove that the newly-learned descriptor has better generalization capability.

Shape correspondence. We also measure the performance of newly-learned shape descriptors in the shape correspondence task. In this experiment, we define the geodesic distance between groundtruth corresponding point and matched point on the target shape as geodesic radius and use spectral matching algorithm (Leordeanu and Hebert, 2005) to compute the point-to-point correspondence between the shapes. For each shape, 100 feature points are sampled via FPS (Eldar et al., 1997). Fig. 7 shows the performance of different descriptors on the SCAPE, FAUST and TOSCA datasets. We can see that the newly-learned descriptor performs significantly better than other kernel-based descriptors. We also show good correspondences (geodesic radius below 0.15) obtained based on different descriptors, as shown in Fig. 9. We observe that the newly-learned descriptor generates the largest number of good correspondences. To illustrate the effectiveness of our method, we also compare the proposed approach to RF (Rodolà et al., 2014) and CN (Wei et al., 2016). As shown in Fig. 8, our method can get better results than that of these two methods (10% improvement than RF (Rodolà et al., 2014) with geodesic radius below 0.3 on both datasets), while the shapes can be matched at a finer level in our approach.



(a) SCAPE

(b) FAUST

Fig. 8. Performance of different methods in the shape correspondence task on the (a) SCAPE and (b) FAUST datasets. Higher curve represents better performance.

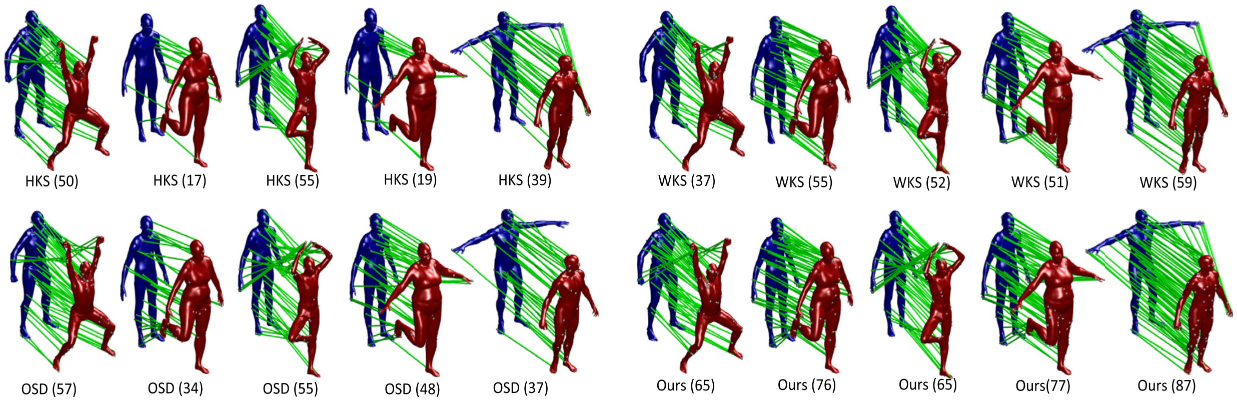


Fig. 9. Shape matching based on HKS, WKS, OSD and the newly-learned descriptor. Shown in green lines are good correspondence with geodesic radius below 0.15 (the number of good correspondence appears in parenthesis).

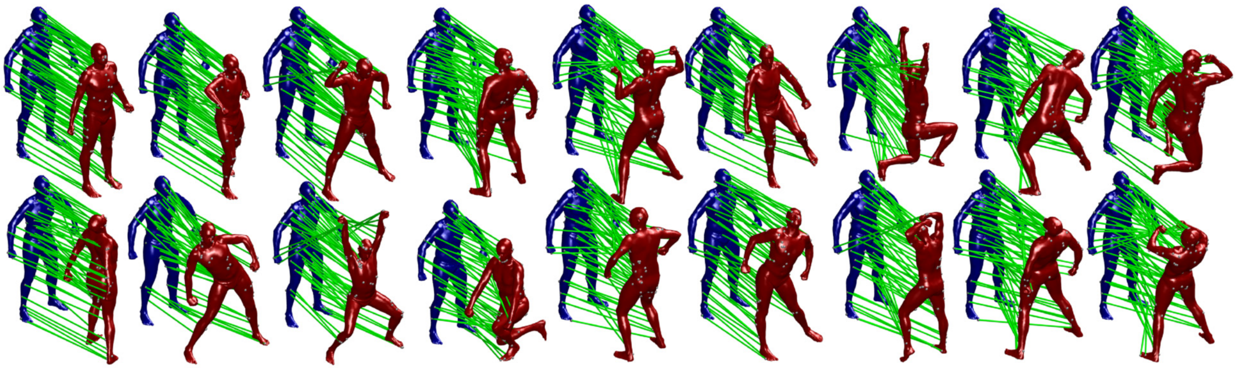


Fig. 10. More shape matching results of our approach on the SCAPE dataset.

More results. We also give more results of shape matching and similarity map. More shape matching results produced by our method are shown in Fig. 10, Fig. 11 and Fig. 12. Especially in Fig. 12, we see that our approach is applicable to large deformations. These results demonstrate that our approach is effective and robust for various shape transformation constraints. More similarity map results produced by our approach are shown in Fig. 13. As shown in Fig. 14 and Table 1, we also conduct an experiment on other categories, including cat and centaur. From the above experiments, we can see that the proposed approach is superior to state-of-the-art methods.

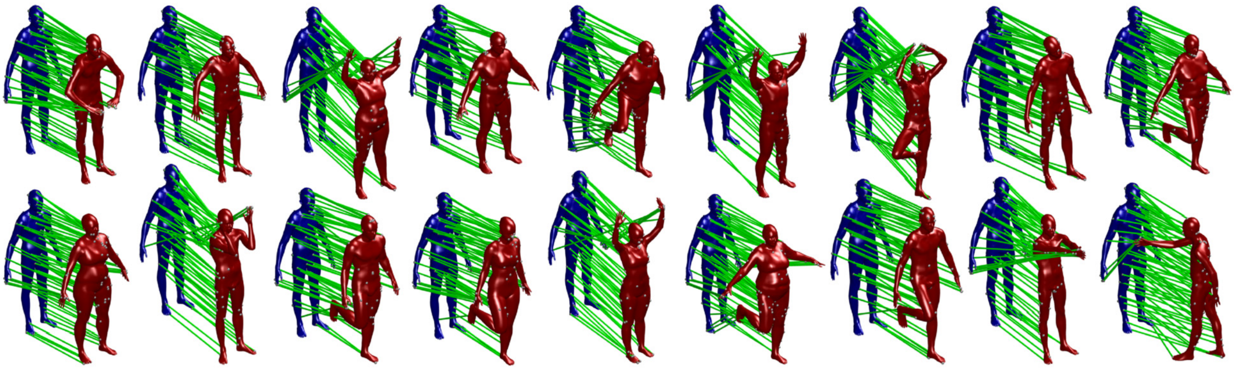


Fig. 11. More shape matching results of our approach on the FAUST dataset.

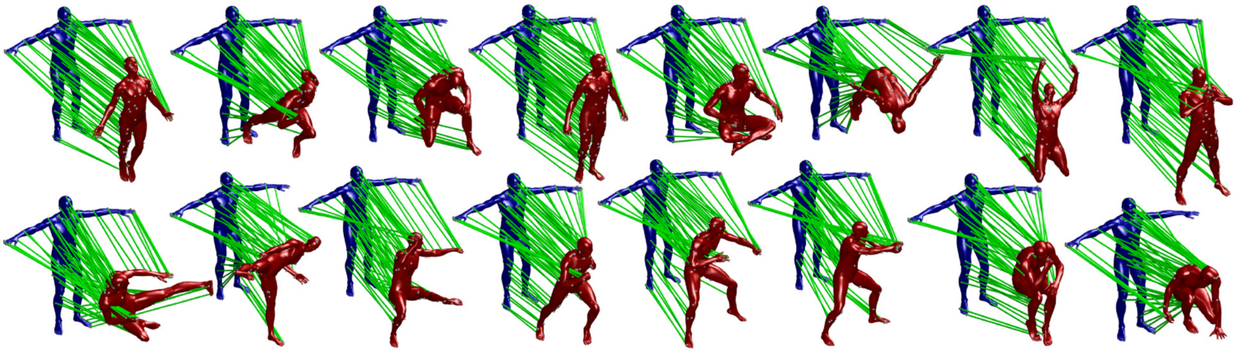


Fig. 12. More shape matching results of our approach on the TOSCA dataset.

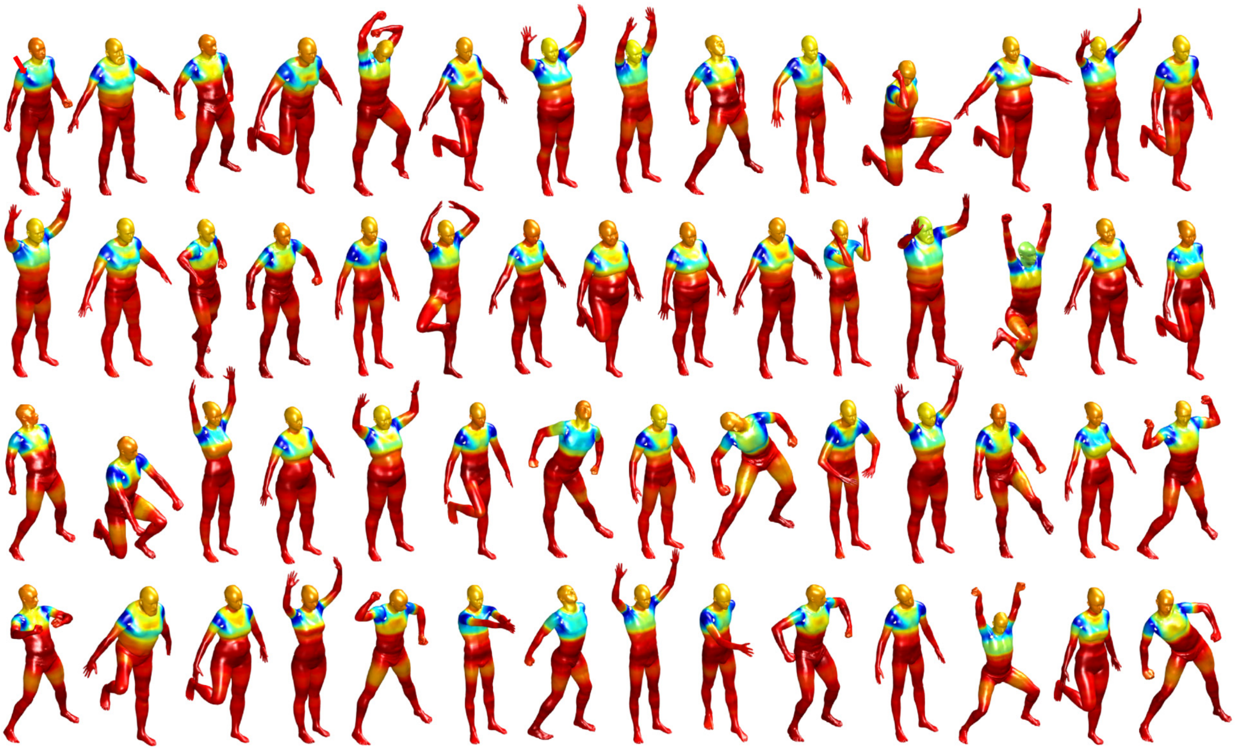


Fig. 13. More similarity map results of our method.

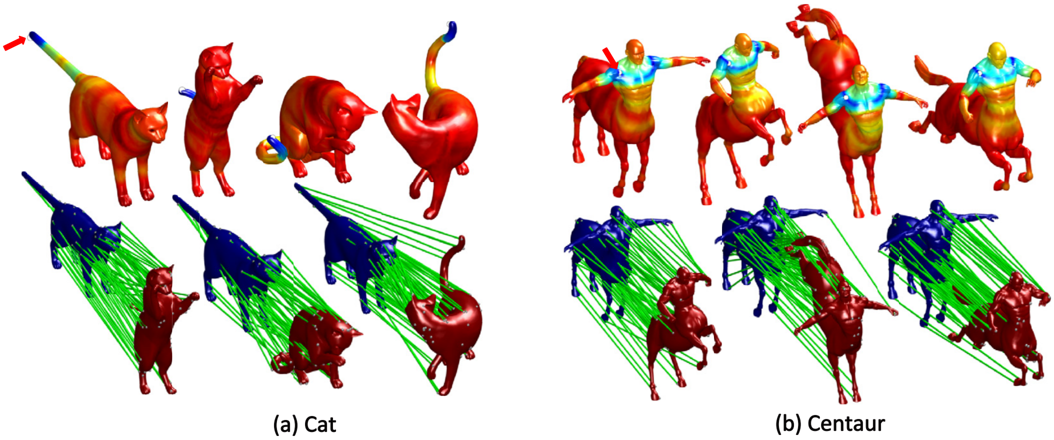


Fig. 14. More results of other categories on the TOSCA dataset.

Table 1
The performance statistics of our method on different datasets. The second column is the number of training triplets. The third and fourth columns are the timings of training and testing process (in minutes), respectively. TOSCA (human) just used for testing. And the last column is the averaged correspondence accuracy with geodesic radius below 0.3.

Dataset	Training triplets	Training	Testing (per mesh)	Accuracy
SCAPE (human)	143K	648 min	0.74 min	89.60%
FAUST (human)	181K	887 min	0.52 min	87.20%
TOSCA (human)	–	–	0.78 min	88.25%
TOSCA (cat)	111K	523 min	0.76 min	86.64%
TOSCA (centaur)	108K	514 min	0.78 min	88.17%

Performance. Our implementation runs on a desktop machine with an Intel Core I7-4790K CPU (3.6 GHz, 16GB memory) and a GeForce 970 GPU (4GB memory, CUDA 8.0). The GPU implementation of triplet CNNs uses the PyTorch framework.[1] It takes about 11 hours to train the model with 143K training triplets on SCAPE dataset, and 15 hours with 181K training triplets on FAUST dataset. Once the training process is finished, we can efficiently compute shape descriptors using the trained model. It takes about 40 seconds for a shape with 6.8K points. More timing details are shown in Table 1.

Limitations. In our approach, the learned descriptors may have difficulties in dealing with shape defects, such as the existence of big holes or missing large organic parts. Another limitation is that we use the fixed size and number of local regions to generate the multi-scale feature, however, for different points of 3D shapes the selection of the size and number of local regions should be considered with its structure. These will be the topics for our future research.

6. Conclusion, discussion, and future works

In this paper, we have presented a novel approach to jointly learning shape descriptors and their correspondence based on the deep triplet CNNs. Extensive experimental results have confirmed the discriminative capability of the newly-learned shape descriptor in shape matching. Geometrically informative multi-scale features are extracted first. We feed our designed deep triplet convolutional neural networks with these features to learn the parameters of the networks, yielding newly-learned descriptors which are more concise, discriminative, and robust than existing kernel-based descriptors. For a pair of new 3D shapes, we then extract the newly-learned descriptors with our learnt CNNs. Finally, a well known spectral matching algorithm is applied to obtain a point-to-point correspondences between the shapes. It may be noted that, our approach is an unsupervised learning algorithm which is applied to a deep triplet CNNs. Key to the success of such network is the loss function that tends to reduce the distance of good-matches while increasing of the distance of mis-matches. Consequently, this network is devised to accommodate descriptor space specially tailored for the shape matching task. With the trained deep convolutional neural networks, a concise, discriminative, and robust shape descriptor could be automatically extracted, which in turn results in more accurate shape matching subject to various shape transformation constraints.

There are many other deep learning structures, such as deep belief networks (Lee et al., 2009), which have exhibited a good capability for shape analysis. It would be an interesting research to apply other models to 3D shape matching in the near future. We design the deep triplet CNNs for several reasons. The shared weights mechanism of the deep triplet CNNs enables us to accelerate the training time considerably and reduce the complexity of the network. And our training samples

[1] https://github.com/gmp2018code/DeepTripletCNNs.

are points, which afford more than tens of thousands samples in the training sets, thus the over-fitting problem could be excluded. The deep CNNs structure we design is not very complex, this is due to the fact that, with the increase of the layer number, the parameters would rapidly growth so as to produce overfit, especially with the limited training shapes. Thus, it would be an interesting attempt for more complex networks in the future research. Moreover, with the deep triplet CNNs, we naturally couple the learning of shape descriptors with the learning of their correspondence together. Although our approach achieves a good performance, there still remain some challenging problems that call for future exploration. Our approach can achieve better performance based on a larger training set. However, the accuracy of our method in shape matching is in fact remaining less satisfactory if the size of the training dataset is small, which shall be improved in our future work.

Acknowledgements

We would like to thank the anonymous reviewers and chairs for their constructive comments. This work is partially supported by Natural Science Foundation of China (61532002 and 61672237), National Science Foundation of USA (IIS-0949467, IIS-1047715, IIS-1049448, and IIS-1715985), and National High-tech R&D Program of China (863 Program, 2015AA016404).

References

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J., 2005. Scape: shape completion and animation of people. ACM Trans. Graph. 24 (3), 408–416.

Aubry, M., Schlickewei, U., Cremers, D., 2011. The wave kernel signature: a quantum mechanical approach to shape analysis. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 1626–1633.

Barra, V., Biasotti, S., 2013. Learning kernels on extended Reeb graphs for 3D shape classification and retrieval. In: Proceedings of the Eurographics Workshop on 3D Object Retrieval, pp. 25–32.

Belongie, S., Malik, J., Puzicha, J., 2000. Shape context: a new descriptor for shape matching and object recognition. In: Proceedings of the Neural Information Processing Systems, pp. 831–837.

Bengio, Y., 2009. Learning deep architectures for AI. Found. Trends Mach. Learn. 2 (1), 1–127.

Bérard, P., Besson, G., Gallot, S., 1994. Embedding Riemannian manifolds by their heat kernel. Geom. Funct. Anal. 4 (4), 373–398.

Bogo, F., Romero, J., Loper, M., Black, M.J., 2014. Faust: dataset and evaluation for 3D mesh registration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3794–3801.

Boscaini, D., Masci, J., Melzi, S., Bronstein, M.M., Castellani, U., Vandergheynst, P., 2015. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. Comput. Graph. Forum 34 (5), 13–23.

Bronstein, A.M., Bronstein, M.M., Castellani, U., Dubrovina, A., Guibas, L.J., Horaud, R.P., Kimmel, R., Knossow, D., Lavante, E.V., Mateus, D., 2010. Shrec 2010: robust correspondence benchmark. In: Proceedings of the Eurographics Workshop on 3D Object Retrieval, pp. 87–91.

Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M., 2011. Shape Google: geometric words and expressions for invariant shape retrieval. ACM Trans. Graph. 30 (1), 1.

Bronstein, A.M., Bronstein, M.M., Kimmel, R., 2008. Numerical Geometry of Non-rigid Shapes. Springer Science & Business Media.

Bronstein, A.M., Bronstein, M.M., Kimmel, R., Mahmoudi, M., Sapiro, G., 2009. A Gromov–Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 256–263.

Bronstein, M.M., Kokkinos, I., 2010. Scale-invariant heat kernel signatures for non-rigid shape recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1704–1711.

Castellani, U., Cristani, M., Fantoni, S., Murino, V., 2008. Sparse points matching by combining 3D mesh saliency with statistical descriptors. Comput. Graph. Forum 33 (2), 643–652.

Castellani, U., Cristani, M., Murino, V., 2011. Statistical 3D shape analysis by local generative descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 33 (12), 2555–2560.

Coifman, R.R., Lafon, S., 2006. Diffusion maps. Appl. Comput. Harmon. Anal. 21 (1), 5–30.

Eitz, M., Hays, J., Alexa, M., 2012. How do humans sketch objects? ACM Trans. Graph. 31 (4), 1–10.

Eitz, M., Hildebrand, K., Boubekeur, T., Alexa, M., 2011. Sketch-based image retrieval: benchmark and bag-of-features descriptors. IEEE Trans. Vis. Comput. Graph. 17 (11), 1624–1636.

Elad, A., Kimmel, R., 2003. On bending invariant signatures for surfaces. IEEE Trans. Pattern Anal. Mach. Intell. 25 (10), 1285–1311.

Eldar, Y., Lindenbaum, M., Porat, M., Zeevi, Y.Y., 1997. The farthest point strategy for progressive image sampling. IEEE Trans. Image Process. 6 (9), 1305–1315.

Fang, Y., Xie, J., Dai, G., Wang, M., Zhu, F., Xu, T., Wong, E., 2015. 3D deep shape descriptor. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2319–2328.

Farabet, C., Couprie, C., Najman, L., Lecun, Y., 2013. Learning hierarchical features for scene labeling. IEEE Trans. Pattern Anal. Mach. Intell. 35 (8), 1915–1929.

Hamza, A.B., Krim, H., 2003. Geodesic Object Representation and Recognition. Springer Berlin Heidelberg.

Hinton, G., 2010. A practical guide to training restricted Boltzmann machines. Momentum 9 (1), 926.

Hu, R., Collomosse, J., 2013. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. Comput. Vis. Image Underst. 117 (7), 790–806.

Johnson, A.E., Hebert, M., 2002. Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. Pattern Anal. Mach. Intell. 21 (5), 433–449.

Kalogerakis, E., Chaudhuri, S., Koller, D., Koltun, V., 2012. A probabilistic model for component-based shape synthesis. ACM Trans. Graph. (TOG) 31 (4), 1–11.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Proceedings of the International Conference on Neural Information Processing Systems, pp. 1106–1114.

Lee, H., Grosse, R., Ranganath, R., Ng, A.Y., 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the International Conference on Machine Learning, pp. 609–616.

Leordeanu, M., Hebert, M., 2005. A spectral technique for correspondence problems using pairwise constraints. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1482–1489.

Lévy, B., 2006. Laplace–Beltrami eigenfunctions towards an algorithm that understands geometry. In: IEEE International Conference on Shape Modeling and Applications, p. 13.

Lipman, Y., Rustamov, R.M., Funkhouser, T.A., 2010. Biharmonic distance. ACM Trans. Graph. 29 (3), 27.

Litman, R., Bronstein, A.M., 2014. Learning spectral descriptors for deformable shape correspondence. IEEE Trans. Pattern Anal. Mach. Intell. 36 (1), 171–180.

Manay, S., Cremers, D., Hong, B.W., Yezzi, A.J., Soatto, S., 2006. Integral invariants for shape matching. IEEE Trans. Pattern Anal. Mach. Intell. 28 (10), 1602–1618.

Masci, J., Meier, U., 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. In: Proceedings of the International Conference on Artificial Neural Networks, pp. 52–59.

Mémoli, F., 2009. Spectral Gromov–Wasserstein distances for shape matching. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 256–263.

Meyer, M., Desbrun, M., Schroder, P., Barr, A.H., 2002. Discrete differential-geometry operators for triangulated 2-manifolds. Math. Vis. 6 (8–9), 35–57.

Ovsjanikov, M., Bronstein, A.M., Bronstein, M.M., Guibas, L.J., 2009. Shape Google: a computer vision approach to isometry invariant shape retrieval. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 320–327.

Ranzato, M., Huang, F.J., Boureau, Y.L., Lecun, Y., 2007. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Rodolà, E., Rota Bulo, S., Windheuser, T., Vestner, M., Cremers, D., 2014. Dense non-rigid shape correspondence using random forests. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4177–4184.

Rustamov, R.M., 2007. Laplace–Beltrami eigenfunctions for deformation invariant shape representation. In: Proceedings of the Eurographics Symposium on Geometry Processing, pp. 225–233.

Skraba, P., Ovsjanikov, M., Chazal, F., Guibas, L., 2010. Persistence-based segmentation of deformable shapes. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 45–52.

Sun, J., Ovsjanikov, M., Guibas, L., 2009. A concise and provably informative multi-scale signature based on heat diffusion. Comput. Graph. Forum 28 (5), 1383–1392.

Tabia, H., Laga, H., 2015. Covariance-based descriptors for efficient 3D shape matching, retrieval, and classification. IEEE Trans. Multimed. 17 (9), 1591–1603.

Wei, L., Huang, Q., Ceylan, D., Vouga, E., Li, H., 2016. Dense human body correspondences using convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1544–1553.

Yan, X., Chang, H., Shan, S., Chen, X., 2014. Modeling video dynamics with deep dynencoder. In: Proceedings of the European Conference on Computer Vision, pp. 215–230.