Computational Visual Media
https://doi.org/10.1007/s41095-020-0163-y Vol. 6, No. 1, March 2020, 95–112

Research Article

Learning local shape descriptors for computing non-rigid dense correspondence

Jianwei Guo[1], Hanyu Wang[2], Zhanglin Cheng[3] (✉), Xiaopeng Zhang[1], and Dong-Ming Yan[1] (✉)

Abstract A discriminative local shape descriptor plays an important role in various applications. In this paper, we present a novel deep learning framework that derives discriminative local descriptors for deformable 3D shapes. We use local "geometry images" to encode the multi-scale local features of a point, via an intrinsic parameterization method based on geodesic polar coordinates. This new parameterization provides robust geometry images even for badly-shaped triangular meshes. Then a triplet network with shared architecture and parameters is used to perform deep metric learning; its aim is to distinguish between similar and dissimilar pairs of points. Additionally, a newly designed triplet loss function is minimized for improved, accurate training of the triplet network. To solve the dense correspondence problem, an efficient sampling approach is utilized to achieve a good compromise between training performance and descriptor quality. During testing, given a geometry image of a point of interest, our network outputs a discriminative local descriptor for it. Extensive testing of non-rigid dense shape matching on a variety of benchmarks demonstrates the superiority of the proposed descriptors over the state-of-the-art alternatives.

Keywords local feature descriptor; triplet CNN; dense correspondence; geometry image; non-rigid shape

1 National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: J. Guo, jianwei.guo@nlpr.ia.ac.cn; X. Zhang, xiaopeng.zhang@ia.ac.cn; D.-M. Yan, yandongming@gmail.com (✉).

2 University of Maryland-College Park, Maryland, USA. E-mail: hywang66@cs.umd.edu.

3 Shenzhen VisuCA Key Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. E-mail: zl.cheng@siat.ac.cn (✉).

Manuscript received: 2020-01-22; accepted: 2020-02-29

1 Introduction

With the rapid increase of available 3D models, 3D shape analysis has become an important research topic in the field of visual media computing. Designing local shape descriptors is one of the fundamental analysis tasks. Typically, a local descriptor refers to an informative representation stored in a multi-dimensional vector that describes the local geometry of the shape around a point. It plays a crucial role in a variety of vision tasks, such as shape matching [1], object recognition [2], shape retrieval [3], shape correspondence [4, 5], and surface registration [6], to name a few.

In recent decades, many local descriptors have been actively investigated by the research community. Despite this interest, however, designing discriminative and robust descriptors is still a non-trivial and challenging task. Early works focus on deriving shape descriptors based on hand-crafted features, including spin images [7], curvature features [8], heat kernel signatures [9], etc. Although these descriptors can represent local shape effectively, the performance of these methods is largely limited by the representative power of the hand-tuned parameters.

Recently, convolutional neural networks (CNNs) have achieved a significant performance breakthrough in many image analysis tasks. Inspired by the remarkable success of applying deep learning in many fields, recent approaches have been proposed to learn local descriptors for 3D shapes in either an extrinsic or intrinsic manner. The former usually takes multi-view images [10] or volumetric representations [11] as input, but is hindered by strong requirements on view selection or low voxel resolutions. The latter methods generalize the CNN paradigm to non-Euclidean manifolds [12],

Fig. 1 Our newly-learned descriptor can be easily used to establish dense correspondences between pairs of non-rigid shapes. The human body shapes (left) are from SCAPE [13] and dog shapes (right) are from TOSCA [14].

and are able to learn invariant shape signatures for non-rigid shape analysis. However, since these methods learn information relating to shape types and structures (e.g., mesh scale, topological structure, spatial resolution, etc.) that vary for different datasets, their generalizability is poor. As a result, these methods are not robust to domain changes.

In this paper, we propose a novel approach for learning local descriptors that can capture the local geometric essence of a 3D shape. We draw inspiration from the work of Ref. [15] which used geometry images for learning global surface features for shape classification. Unlike their work, we present an efficient intrinsic parameterization to construct a small set of geometry images from multi-scale local patches around each point on the surface. Then, the fundamental low-level geometric features can be encoded into the pixels of these regular geometry images; standard CNNs can then be applied directly to them. We leverage a triplet network [16] to perform deep metric learning with a pre-training phase and an improved triplet loss function. The objective is to learn a descriptor that minimizes the distances between corresponding points while maximizing the distances between non-corresponding points in descriptor space. In summary, our main contributions are the following:

- A new 3D local descriptor based on specially designed triplet networks dedicated to processing local geometry images encoding low-level geometric information. To generate geometry images, a robust intrinsic parameterization is constructed using geodesic polar coordinates.
- A novel triplet loss function that can control the dispersion of anchor-positive descriptor distance, effectively improving the performance of our descriptor. We also present a tractable and efficient feature point sampling approach—selecting a sufficient number of informative feature points leads to efficient and accurate training.
- Validation that the proposed concise framework is suitable for solving the dense correspondence problem for deformable shapes. Furthermore, it has better generalizability to different datasets than existing descriptors.

We note that a shorter conference version of this paper appeared in Ref. [17]; it did not address the dense correspondence problem. Specifically, this journal paper extends our earlier work as follows:

- Since the neural network in the conference paper was only trained using rigid keypoints, its performance on points defined in highly deformable regions was unsatisfactory. To address this issue, two modifications are adopted here. Firstly, rather than only using rigid keypoints for training, we generate local geometry images from landmark keypoints on both rigid parts and truly deformable regions. Secondly, we add a further intrinsic feature (HKS) to get a better local descriptor. This makes the proposed approach applicable to learning dense descriptor fields.
- We introduce a new and more robust parameterization method for local geometry image generation. Instead of the previous authalic parameterization, our new method is based on geodesic polar coordinates. This approach works well for badly-shaped triangular meshes, while the authalic parameterization may fail to parameterize some local patches due to imperfections in meshes. In this way, the process of preparing training data can be greatly accelerated.
- Extensive experiments and analysis using more standard quality measures have been conducted to verify the effectiveness of our approach. We also study its resistance to noise and partiality, and compare it to our earlier conference work to show the advantages of the new approach.

2 Related work

A large variety of 3D local feature descriptors has been proposed in the literature. These approaches can be roughly classified into two categories: traditional hand-crafted descriptors and learned local descriptors. The relevant work on matching shapes with non-rigid correspondences is also revisited.

2.1 Hand-crafted local descriptors

Early works focus on deriving shape descriptors based on hand-crafted features [18]. A detailed survey is beyond the scope of this paper, but we briefly review some representative techniques. For rigid shapes, some successful *extrinsic* descriptors have been proposed, for example, spin images (SI) [7], 3D shape contexts (3DSC) [19], MeshHOG descriptors [20], signatures of histogram of orientations (SHOT) [21], shape google [22], and rotational projection statistics (RoPS) [23]. Obviously, these approaches are invariant under rigid Euclidean transformations, but not under deformations. To deal with isometric deformations, some *intrinsic* descriptors have been based on geodesic distances [24] or spectral geometry. Such descriptors include heat kernel signatures (HKS) [9], wave kernel signatures (WKS) [25], intrinsic shape contexts (ISC) [26], and optimal spectral descriptors (OSD) [27]. In addition, several methods have been proposed to compute shape similarities and correspondences across a large shape database, for exploring large model repositories [28] or finding high quality point-to-point maps within a collection of related shapes [29]. However, both extrinsic and intrinsic descriptors rely on a limited predefined set of hand-tuned parameters, which are tailored for task-specific scenarios. Thus, these local descriptors are not discriminative enough to describe various 3D shape transformations.

2.2 Deep-learned local descriptors

Wei et al. [30] employ a CNN architecture to learn invariant descriptors for subjects in arbitrary complex poses and clothing; their system is trained with a large dataset of depth maps. Zeng et al. [11] present another data-driven 3D keypoint descriptor for robustly matching local RGB-D data. Since they use 3D volumetric CNNs, this voxel-based approach is limited to low resolution due to the high memory and computational cost. Charles et al. [31] propose a deep net framework, called PointNet, that can directly learn point features from unordered point sets to compute shape correspondences. Khoury et al. [32] present an approach to learn local *compact geometric features* (CGF) for unstructured point clouds by mapping high-dimensional histograms into low-dimensional Euclidean spaces. Huang et al. [10] recently introduce a new local descriptor by taking rendered views [33] at multiple scales and processing them through a classic 2D CNN. While this method has been successfully used in many applications, it still suffers from strong requirements on view selection, and as a result the projected 2D images are not necessarily geometrically informative. In addition, whether this approach can be used for non-rigid shape matching is somewhat elusive.

Another family of methods is based on the notion of *geometric deep learning* [34], which generalizes CNN to non-Euclidean manifolds. Various frameworks have been introduced to solve descriptor learning or correspondence learning problems, including localized spectral CNN (LSCNN) [35], geodesic CNN (GCNN) [36], anisotropic CNN (ACNN) [37], mixture model networks (MoNet) [12], deep functional maps (FMNet) [38], and so on. Unlike such methods, our work utilizes geometry images to locally flatten each non-Euclidean patch to the 2D domain so that standard convolutional networks can be used.

2.3 Non-rigid shape correspondence

Plenty of algorithms have been proposed to compute correspondences between geometric shapes, with several recent surveys [4, 39] and tutorials [40] available for in-depth review of this area. Broadly speaking, these approaches can be classified into three major categories. First, *point-wise correspondence methods* establish the matching between (a subset of) the points on two or more shapes by minimizing metric distortion, using similarity of local descriptors [20, 27, 41], geodesic distances [42–44], or diffusion distances [45]. Second, *soft correspondence methods* aim to establish approximate correspondences between probability density functions. A family of such methods is based on functional maps [46], which model correspondences as linear operators between spaces of functions on manifolds [47–49]. Third, *learning-based methods* formulate correspondence computation as a learning problem [50], or design convolutional neural networks on Euclidean [30] and non-Euclidean [12, 37, 38] domains.

3 Methodology overview

3.1 Background

Given a feature point (or any point of interest) $\boldsymbol{p}$ on a surface shape $\mathcal{S} \subset \mathbb{R}^3$, our goal is to learn a non-linear feature embedding function $f(\boldsymbol{p}) : \mathbb{R}^3 \to \mathbb{R}^d$ which outputs a d-dimensional descriptor $X_{\boldsymbol{p}} \in \mathbb{R}^d$

for that point. The embedding function is carefully designed such that the distance between descriptors of geometrically and semantically similar points is as small as possible. In this paper, we use the L_2 Euclidean norm as the similarity metric between descriptors: $D(X_{\boldsymbol{p}_i}, X_{\boldsymbol{p}_j}) = ||X_{\boldsymbol{p}_i} - X_{\boldsymbol{p}_j}||_2$. Since our approach is built on the notion of a geometry image, we next briefly review the concept of geometry image, before introducing the pipeline of our framework.

The *geometry image* is a mesh representation technique introduced by Gu et al. [51]. It represents an irregular mesh of arbitrary topology using a completely regular grid of samples on a square domain. Given a 2-manifold surface mesh, the creation of a geometry image includes three steps: cutting, parameterization, and quantification. The first step converts the surface into a topological disk using a network of cuts, the second step parameterizes this disk onto a square domain, and the third step creates a regular grid over the square and resamples the surface via the parameterization. Using this representation, the geometric properties (e.g., positions, normals) as well as other attributes of the original mesh can be resampled and encoded into the pixels of an image. Geometry images have been demonstrated to be useful in various graphics applications, such as rendering, remeshing, and shape compression.

3.2 Pipeline

The core part of our approach is a full end-to-end learning framework, illustrated in Fig. 2. In an off-line training phase, we learn the descriptors by utilizing a triplet network composed of three identical convolutional networks (*ConvNet* for simplicity) sharing the same architecture and parameters. We feed a set of triplets into the ConvNet branches to characterize the descriptor similarity relationship. Here, a triplet $t = (I(\boldsymbol{p}), I(\boldsymbol{p}^+), I(\boldsymbol{p}^-))$ contains an anchor point $\boldsymbol{p}$, a positive point $\boldsymbol{p}^+$, and a negative point $\boldsymbol{p}^-$, where $I(\boldsymbol{p})$ represents a geometry image encoding the local geometric context around $\boldsymbol{p}$. By *positive* we mean that $\boldsymbol{p}$ and $\boldsymbol{p}^+$ are correspondingly similar surface points, and by *negative* we mean $\boldsymbol{p}^-$ is dissimilar to the anchor point $\boldsymbol{p}$. With the training data, we optimize the network parameters by using a minimized-deviation triplet loss function so that, in the final descriptor space, each positive point should be much closer to the anchor point than any negative point. Once trained, during the testing stage, we first generate a local geometry image for a point of interest on the surface, then we generate a 128-d local descriptor for this point by applying the individual ConvNet to the geometry image.

4 CNN architecture and training

In this section, we describe the details of our network architecture and how it can be trained automatically and efficiently to learn the embedding function.

4.1 Training data preparation

4.1.1 Requirements

A rich and representative training dataset is the key to the success of CNN-based methods. For non-

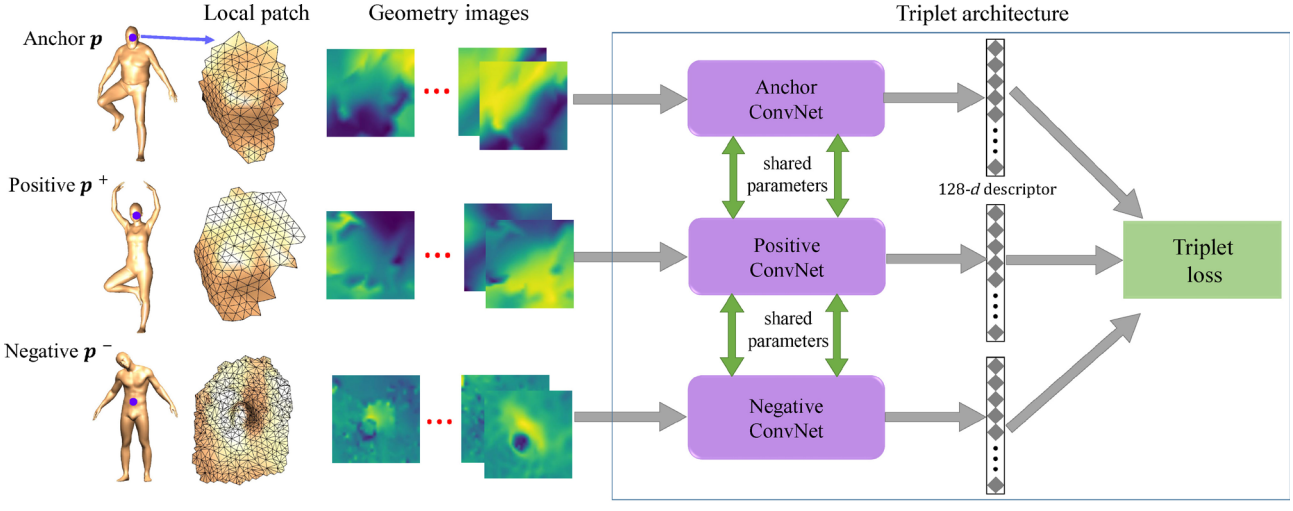

Fig. 2 Overview of our local descriptor training framework. We start by extracting local patches around the keypoints (purple), and generate geometry images for them. Then a triplet is formed and further processed through a triplet network, where we train this network using an objective function (triplet loss function).

rigid shape analysis, a good local descriptor should be invariant with respect to noise, transformation, and non-isometric deformation. To meet these requirements, we chose the most recent and particularly challenging FAUST dataset [52], which contains noisy, realistically deforming meshes of different people in a variety of poses. Furthermore, full-body ground-truth correspondences between the shapes are known for all points.

Note that our proposed approach is generalizable, i.e., after our network has been trained on one dataset, it can be applied to other datasets: see Section 5.

4.1.2 Feature point selection

Naively, we could use all points of the shape to generate geometry images for training. However, this approach does not work well in practice. The reasons are two-fold: first, it requires a huge amount of memory space to store the training data; second, the training process converges poorly due to the existence of many noisy and uninformative local regions. To overcome these issues, we use some representative feature points, of two kinds. First, 128 landmark keypoints are determined by leveraging existing 3D interest point detectors (e.g., 3D-Harris [53] is used in this paper). Then we randomly sample another 128 points by using farthest point sampling on surfaces [54], so that the sampled points are uniformly distributed to cover the entire shape. This finally gives 256 feature points, as shown for the FAUST dataset in Fig. 3. By this means, we not only consider keypoints on rigid parts of the shape, but also take into account points defined on deformable regions. In addition, since ground-truth point-wise correspondences are given in FAUST, feature point sampling operation need only be performed on one mesh, and then each feature point can be easily retrieved from all the other meshes.

4.2 Local geometry image generation

Partially motivated by Ref. [15], we use the geometry image representation to capture surface information: surface signals are stored in simple 2D arrays. Unlike previous work converting the entire 3D shape into a single geometry image for shape classification, we generate a set of local geometry images, one for each point of interest.

To generate a local geometry image for a surface point $\boldsymbol{p}$, a local patch mesh is first built by extracting the neighboring triangles around this point. Then we map the local patch to a 2D square grid. To speed up the training process and to make the descriptor more robust, we make two alignments of the local patch before parameterizing it. First, we align the average normal direction of the vertices inside the local patch with the z-axis, and then we rotate the local patch around z-axis so that the principal curvature direction is located in the x–z plane, following Ref. [55].

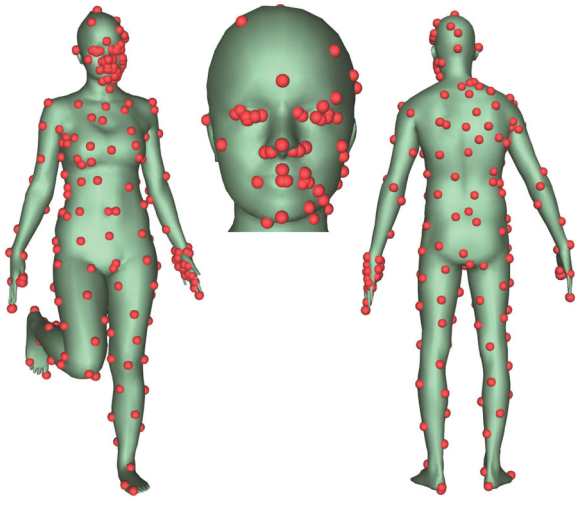

Fig. 3 The 256 sampled feature points on human models in dynamic poses from the FAUST dataset.

Now we perform a local intrinsic parameterization with low metric distortion in the region of interest around $\boldsymbol{p}_i$, which is invariant under non-rigid shape transformations. In particular, an efficient method based on *discrete geodesic polar coordinates* (DGPC) [56] is utilized to map each neighborhood point $\boldsymbol{p}_i$ to a polar coordinate (ρ, θ) with respect to the base point $\boldsymbol{p}$, where ρ is the geodesic distance from $\boldsymbol{p}_i$ to $\boldsymbol{p}$, and θ is the polar angle. After the local geodesic polar map has been constructed, we convert the geodesic polar coordinates to Cartesian coordinates, allowing a 2D geometry image to be generated. This approach is very robust for badly-shaped triangular meshes. The resolution of geometry image used depends on specific application requirements; we set it to 32×32 in all our experiments. To further avoid rotational ambiguity, we rotate the 2D geometry image through $K = 12$ steps at 30° intervals, and generate a corresponding geometry image for each. Finally, in order to capture multi-scale context around this point, we extract the local patch at $L = 3$ scales, with neighborhood geodesic radius $2.0\rho_0$, $3.5\rho_0$, and $4.5\rho_0$, respectively, where ρ_0 is 1% of the geodesic diameter of the entire mesh.

While geometry images can be encoded with any suitable feature of the surface mesh, choice depends on specific applications. For solving the sparse correspondence problem, we found that just using two fundamental low-level geometric features suffices in our approach: (1) vertex normal direction $\vec{\boldsymbol{n}}_{\boldsymbol{v}} = \{n_x, n_y, n_z\}$ at each vertex $\boldsymbol{v}$, calculated by weighted averaging face normals of the vertex's incident triangles; (2) the principal curvatures $\kappa_{\min}$ and $\kappa_{\max}$, that measure minimum and maximum bending in orthogonal directions at a surface point, respectively. Therefore, each geometry image is encoded with 15 feature channels: $\{n_x^i, n_y^i, n_z^i, \kappa_{\min}^i, \kappa_{\max}^i\}_{i=1}^{L=3}$, where i represents each scale. Figure 4 shows some example geometry images for different scales and orientations. To learn a dense descriptor, we only need to add one more intrinsic feature: HKS [9]. We select HKS because of its invariance under isometric deformation and its multi-scale property of capturing the point's local and global geometric information. Specifically, it represents increasingly global properties of the shape with increasing time. We will show its effects in the results section.

4.3 Triplet sampling

For fast training convergence, it is important to select meaningful and discriminative triplets as input to the triplet network. The purpose of training is to learn a discriminative descriptor from positive or negative points that are hard to differentiate from the anchor point. Thus, given an anchor point $\boldsymbol{p}$, we want to select a positive point $\boldsymbol{p}^+$ (*hard positive*) with $\text{argmax}||f(\boldsymbol{p}_i) - f(\boldsymbol{p}_i^+)||_2$, and similarly, a negative point $\boldsymbol{p}^-$ (*hard negative*) with $\text{argmax}||f(\boldsymbol{p}) - f(\boldsymbol{p}^-)||_2$. The question is, given an anchor point $\boldsymbol{p}$, how to select these hard positive and negative points? The most straightforward way is to pick samples by considering all possible triplets across the whole training set. However, this global approach is time-consuming and may provide misleading information that undermines the convergence of the triplet network, due to noisy or poorly shaped local patches.

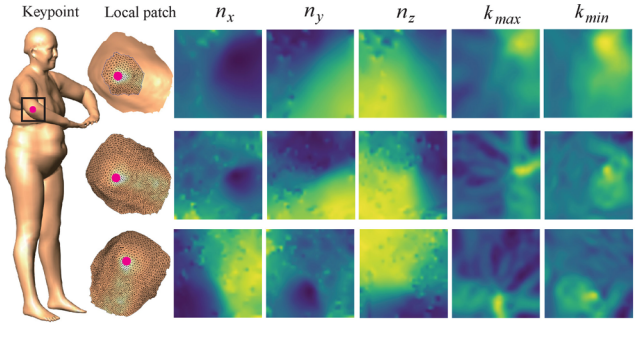

Fig. 4 Geometry images generated around a keypoint. Top to bottom: geometry images of a small local patch, a large local patch and a rotated (90° clockwise) large local patch. Left to right: geometry images encoding normal $\{n_x, n_y, n_z\}$ and curvature $\{\kappa_{\max}, \kappa_{\min}\}$ features.

Instead, we use a stochastic gradient descent approach to generate the triplets within a mini-batch, following the approach used in Ref. [57] for 2D face recognition. Specifically, at each iteration of the training stage, we randomly select 16 points out of 256 feature points, and then randomly select 8 geometry images out of $K \times M$ geometry images across the shapes for each point, where $K = 12$ is the number of rotated geometry images of one feature point on one shape, and M is the number of shape models in the training set. This give a batch size of 128. Then for all anchor-positive pairs within the batch, we select semi-hard negatives instead of the hardest ones, as the hardest negatives can in practice lead to bad local minima early in the training process. Here a semi-hard negative $\boldsymbol{p}_{\text{semi}}^-$ is defined as

$$||f(\boldsymbol{p}_i) - f(\boldsymbol{p}_i^+)||_2 < ||f(\boldsymbol{p}_i) - f(\boldsymbol{p}_{\text{semi}}^-)||_2 \qquad (1)$$

While the semi-hard negative is a negative exemplar that is further away from the anchor than the positive, it is still closer than other harder negatives.

4.4 Min-CV triplet loss

The main concern in real tasks such as shape matching and shape alignment is the discrimination ability of a local shape descriptor. Since we employ CNNs to embed geometry images of keypoints into a d-dimensional Euclidean space, an effective loss function must be designed. It causes the CNN to regard a geometry image of a specific type of surface point as being closer to all other geometry images of the same type of surface point, and farther from geometry images of any other type of surface point. To achieve this goal, we define the following classic triplet loss function [57]:

$$\begin{aligned} L &= \sum_{i=1}^{N} \left[D_{\text{pos}}^i - D_{\text{neg}}^i + \alpha \right]_+ \\ D_{\text{pos}}^i &= D\big(f(\boldsymbol{p}_i), f(\boldsymbol{p}_i^+)\big) \\ D_{\text{neg}}^i &= D\big(f(\boldsymbol{p}_i), f(\boldsymbol{p}_i^-)\big) \end{aligned} \qquad (2)$$

where N is the batch size, and α is the margin distance

that we expect between anchor-positive and anchor-negative pairs.

Combined with hard mining, such triplet loss functions are widely used in various metric learning tasks and perform well or at least acceptably. However, they suffer from problems when used with our model and evaluation dataset. In particular, in training with this loss function, the average loss continually decreased, but the single-triplet loss oscillated violently. Furthermore, for many triplets, the distance between the anchor and the positive geometry images in descriptor space were still large compared to the distance between anchor and negative. Only a few triplets with almost zero loss led to the decrease in average loss. This phenomenon indicated that our CNNs were failing to learn intrinsic local features and were trapped in a local optimum.

To solve this problem, we proposed a new triplet loss function, which minimizes the ratio of standard deviation to mean value, *coefficient of variation* (CV), of anchor-positive distance within one batch. This modification was inspired by the intuition that distance in descriptor space for one geometry image pair at a point should be similar (at least the same order of magnitude) to the distances for other geometry image pairs for the same keypoint. Adding this requirement to the classic triplet loss gives our minimized-CV (Min-CV) triplet loss:

$$L_{\text{Min-CV}} = \lambda \frac{\sigma(D_{\text{pos}})}{\mu(D_{\text{pos}})} + \sum_{i=1}^{N} \left[D^i_{\text{pos}} - D^i_{\text{neg}} + \alpha \right]_+ \quad (3)$$

where λ is a tunable non-negative parameter, $\sigma(\cdot)$ denotes standard deviation of one batch, and $\mu(\cdot)$ denotes arithmetic mean. Note that other recent work [58, 59] has also introduced the mean value and variance/standard deviation into traditional triplet loss. Their loss functions, L_{Kumar} [58] and L_{Jan} [59], are respectively defined as

$$\begin{aligned} L_{\text{Kumar}} = {} & (\sigma^2(D_{\text{pos}}) + \sigma^2(D_{\text{neg}})) \\ & + \lambda \max(0, \mu(D_{\text{pos}}) - \mu(D_{\text{neg}}) + \alpha) \end{aligned} \quad (4)$$

$$\begin{aligned} L_{\text{Jan}} = {} & \sigma(D_{\text{pos}}) + \sigma(D_{\text{neg}}) + \mu(D_{\text{pos}}) \\ & + \lambda \max(0, \alpha - \mu(D_{\text{neg}})) \end{aligned} \quad (5)$$

where $\sigma^2(\cdot)$ denotes variance. Unlike these two approaches, we minimize CV directly instead of the variance: unlike variance, CV measures dispersion of D_{pos} without being influenced by the numerical scale of the descriptor distance (or the magnitude of the data). Thus, scaling the descriptor distance decreases the variance but CV is unaffected. Thus, CV better reflects the degree of data deviation. We compare our approach with these two loss functions in Section 5. Furthermore, extensive experiments show that our Min-CV triplet loss can help CNNs to learn significant features from one dataset while generalizing well to other datasets.

4.5 CNN architecture and configuration

Considering the nature and complexity of our task, we have designed a specific CNN architecture dedicated to processing geometry images in our triplet structure, which we now presented.

4.5.1 Network architecture

Figure 5 illustrates the architecture of our CNN model. We have a compact stack of three convolutional layers (*conv*, blue), three pooling layers, and two fully connected layers (*fc*, green). Each convolutional layer has the size of convolution kernel given above and the number of output feature maps given below. Each fully connected layer has the number of units given above. The *size* is length and width of the tensor which is fed into next layer, so e.g., the third layer from the left is a convolutional layer that takes an $8 \times 8 \times 256$ tensor as input and performs a $3 \times 3 \times 512$ convolution on it, resulting in an $8 \times 8 \times 512$ tensor flowing into a pooling operation. Next, we apply max pooling with a stride of 2 to the output of the first convolutional layer and average pooling with the same stride to the outputs of the other two convolutional layers. Batch normalization (BN) [60] is adopted after each convolution or linear map of input but before non-linear activation. Note that the function in the BN layer differs from our Min-CV loss: the BN layer normalizes the mean and variance of batch data in deep neural networks. It

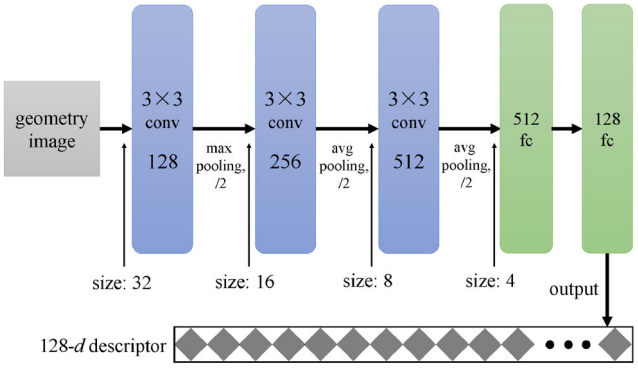

Fig. 5 Detailed network architecture of individual ConvNet shown in Fig. 2.

solves the vanishing and exploding gradient problems during the back-propagation stage of training. In contrast, our new loss directly acts on the output of our neural network and influences the whole network by guiding the training. It imposes the intuition that the degree of difference between descriptors of corresponding points should be as small as possible.

4.5.2 CNN configuration

The detailed configuration of our triplet CNN is adjusted to adapt our architecture to give the best performance. Because triplet loss is not as stable as other frequently-used loss functions, our previous CNN with traditional ReLU activation often suffered from the dying ReLU problem, reducing the effective capacity of our CNN model, leading to a failure to generate meaningful descriptors. To avoid this defect, we employ leaky ReLU [61] with slope = 0.1 for negative input as our activation function. Experimental results demonstrate the effectiveness of this strategy.

In addition, we use a pre-training strategy to speed up training. Firstly we train a classification network constructed with a main part identical to the anchor net in our triplet CNNs and a softmax layer, using the FAUST dataset. The classification labels are the indices of the vertices of the mesh. After convergence, we use the parameters from the main part of the classification network to initialize the convolutional layers of our triplet CNN. Furthermore, Xavier initialization [62] is adopted to initialize all layers of the classification network and the fully connected layers of our triplet CNNs. In training, the Adam algorithm [63] is employed to optimize the loss function. In all of our experiments, the learning rate starts at 0.01 and decreases by a factor of 10 every time the validation loss begins to oscillate. To avoid overfitting, L_2 regularization is also used with coefficient 0.005.

5 Experimental results

In this section, we first give training details and evaluate the performance of our Min-CV triplet loss. Then we provide a complete comparison with state-of-the-art approaches for computing dense correspondence, with qualitative and quantitative evaluations. We also compare the current approach to our earlier conference work [17] to show the advantages of the new approach. These results were obtained on a 3.4 GHz Intel Core i7-3770 processor with 16 GB RAM. Offline training runs used a NVIDIA GeForce TITAN X Pascal GPU with 12 GB RAM.

5.1 Experimental setup

5.1.1 Datasets

In addition to FAUST, we carried out experiments on two other public-domain datasets. SCAPE [13] contains 71 realistic registered meshes of a particular person in a variety of poses, while SPRING [64] contains 3000 scanned body models. In these datasets, ground truth point-wise correspondence between the shapes are known for all points.

5.1.2 Training settings

We separated the FAUST dataset into disjoint training models (70%, subjects 1–7 with 10 poses per subject), validation models (10%, subject 8), and testing models (20%, subjects 9–10). Each geometry image triplet is generated from one of these subsets at the appropriate stage, resulting in a triplet training set, validation set, and testing set, respectively. The training set contains about 2.35×10^{13} different triplets that could selected for feeding into our triplet CNNs for training, while the validation and testing sets contain 5.08×10^{10} and 1.78×10^{11} potential triplets, respectively. Our method implementation is based on TensorFlow [65]. Using our hardware configuration given above, full training takes about 10 hours.

5.1.3 Evaluation metrics

To evaluate our learned local descriptor and to compare with other algorithms, we adopt various measures commonly used in the literature:

- *Cumulative match characteristic* (CMC) curve, which evaluates the probability of finding a correct correspondence among the k-nearest neighbors in descriptor space.
- *Princeton protocol*, which measures correspondence quality by plotting the proportion of nearest-neighbor matches that are at most r-geodesically distant from the ground-truth correspondence.
- *Similarity map*, which qualitatively depicts Euclidean distance in descriptor space between the descriptor at a reference point and the remaining points on the same shape, as well as its transformations.

- *Point-wise map*, which visualizes the correspondence as a vertex-to-vertex map (corresponding points with respect to a ground-truth reference are shown in the same color).

5.1.4 Competing algorithms

We have compared our method against many local descriptors of different types:

- *extrinsic descriptors* including hand-crafted features: spin images (SI) [7], SHOT [21], RoPS [23], and a learning-based method: CGF-32 [32];
- *intrinsic descriptors* including hand-crafted features: HKS [9] and WKS [25], a learning-based descriptor: OSD [27], and state-of-the-art deep-learned descriptors: LSCNN [35], MoNet [12], FMNet [38].

5.2 Ablation study of loss functions

First, we demonstrate the effectiveness of our proposed Min-CV triplet loss by an ablation study. In Fig. 6, we show the training behaviors evaluated on the validation dataset using classic triplet loss (Eq. (2)), Kumar's loss [58] (Eq. (4)), Jan's loss [59] (Eq. (5)), and our Min-CV triplet loss (Eq. (3)), where the margin distance parameter α is empirically set to a large number (100 used here) and $\lambda = 1.0$. To be fair, we use the same network architecture and parameters proposed in this paper for different losses. In Fig. 6, the positive–negative margin curve shows the average distance between anchor-positive and anchor-negative pairs in each batch, calculated as $\sum_{i=1}^{N} \left[D^i_{\text{pos}} - D^i_{\text{neg}}\right]_+$. The standard deviation mean ratio curve shows variation of average ratio $\frac{\sigma(D_{\text{pos}})}{\mu(D_{\text{pos}})}$ with number of iterations. These results show that Jan's loss performs worst in our task, and classic loss cannot control the degree of deviation of anchor-positive distance, while both Kumar's loss and our Min-CV loss significantly reduce it. The training behavior of our loss is better than Kumar's in both comparisons, so it provides improved robustness and generalizability for our learned descriptor; our descriptor performs stably on various datasets. We further compare the performance of different losses on the testing models. As shown in the CMC and Princeton protocol curves (see Fig. 7), our loss still performs better.

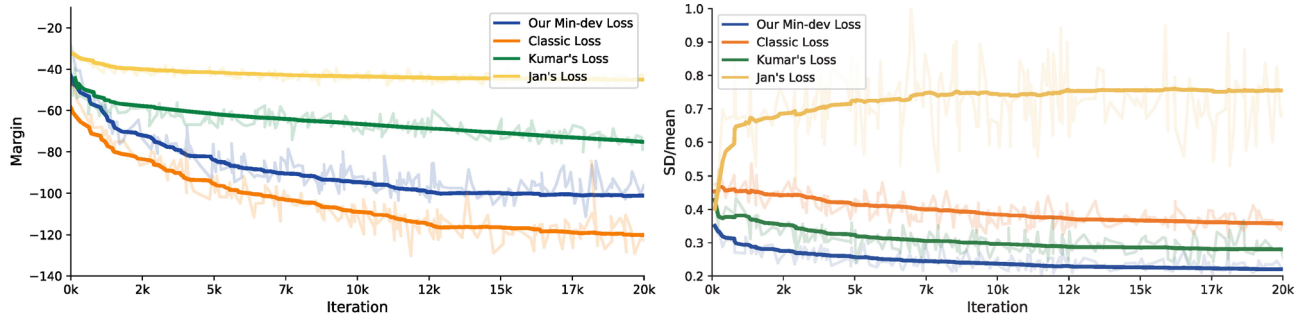

Fig. 6 Training behavior with different triplet loss functions. Left: positive–negative margin curves. Right: standard deviation mean ratio curves.

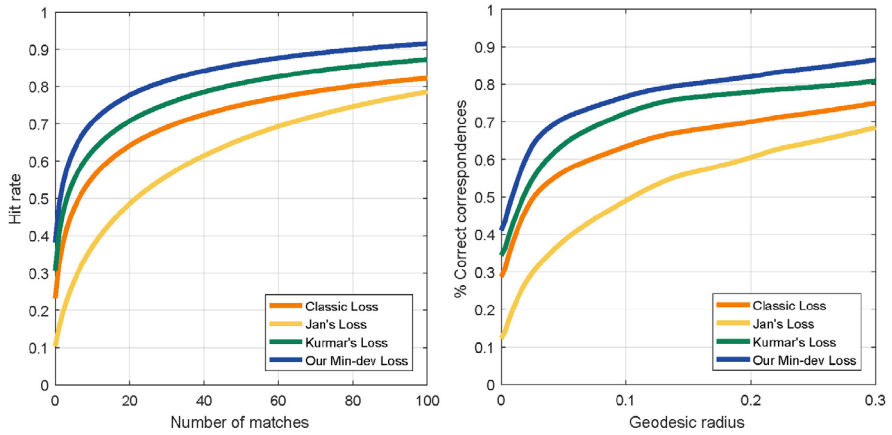

Fig. 7 Performance of different losses on FAUST testing models, measured using CMC (left) and the Princeton protocol (right).

5.3 Dense correspondence task

We next demonstrate the advantages of our local descriptor in solving the dense correspondence problem. We retrained our CNN network by adding geometry images with the HKS feature, and tested it on FAUST, SPRING, and SCAPE datasets.

5.3.1 Comparison

We measured the performance of all shape descriptors using CMC and Princeton protocol metrics. For all comparisons, the learning methods (OSD, LSCNN, MoNet, FMNet, and ours) were trained on the FAUST dataset, and then applied to other datasets. Figure 8 reports the results. We observe that MoNet performs best on FAUST, but MoNet does not learn a real descriptor, and it casts shape correspondence as a labelling problem. Thus, it cannot be directly generalized to other datasets having been trained on FAUST, because the labelling spaces can be quite different. Compared to the remaining methods, our

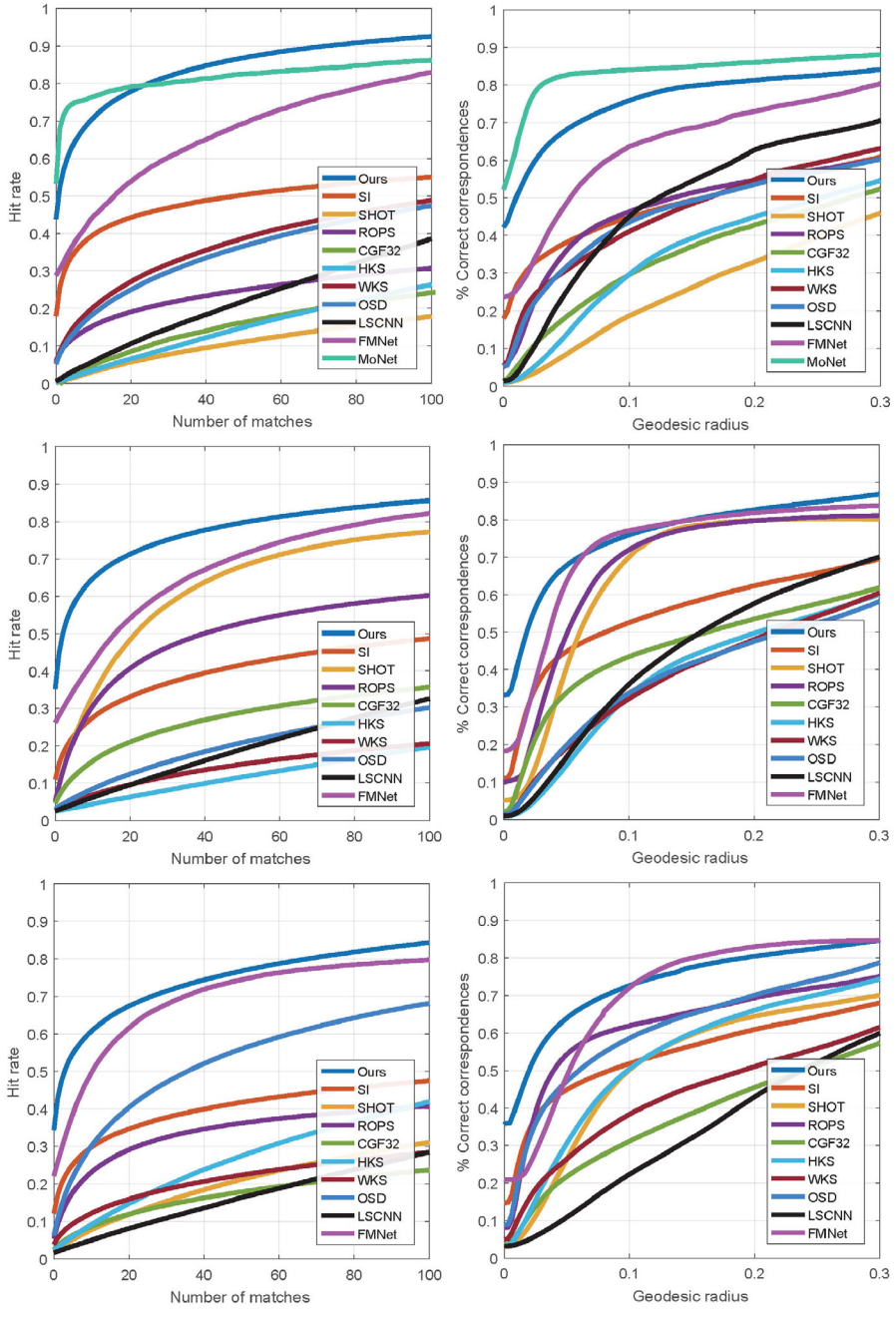

Fig. 8 Performance of different descriptors for solving dense correspondences task on FAUST (top), SPRING (center), and SCAPE (bottom) datasets, using CMC (left) and the Princeton protocol (right).

performance is higher on FAUST. In addition, we show that our approach has better generalizability than other methods to other datasets.

We provide further results of shape matching using a similarity map and a point-wise map in Figs. 9 and 10 for FAUST. Note that for the point-wise map, we show the matching results for the top $k = 20$ ranks in the CMC curves. The similarity map shows that our proposed approach is more discriminative and robust to various transformations. The point-wise map also demonstrates that our newly-learned descriptor has superior performance.

5.3.2 Noise and partial shapes

To demonstrate our approach is robust, we first train our descriptor only on the clean data in FAUST. Then we test it on noisy data, obtained by adding three levels of Gaussian noise. As shown in Fig. 12, our performance slightly reduces as the level of noise increases, but we still perform well on noisy data.

Matching deformable 3D shapes when only parts of them are present is a challenging problem. Since our approach only exploits local geometry images, it does not necessarily require the objects to be complete shapes. To demonstrate that our descriptor has a

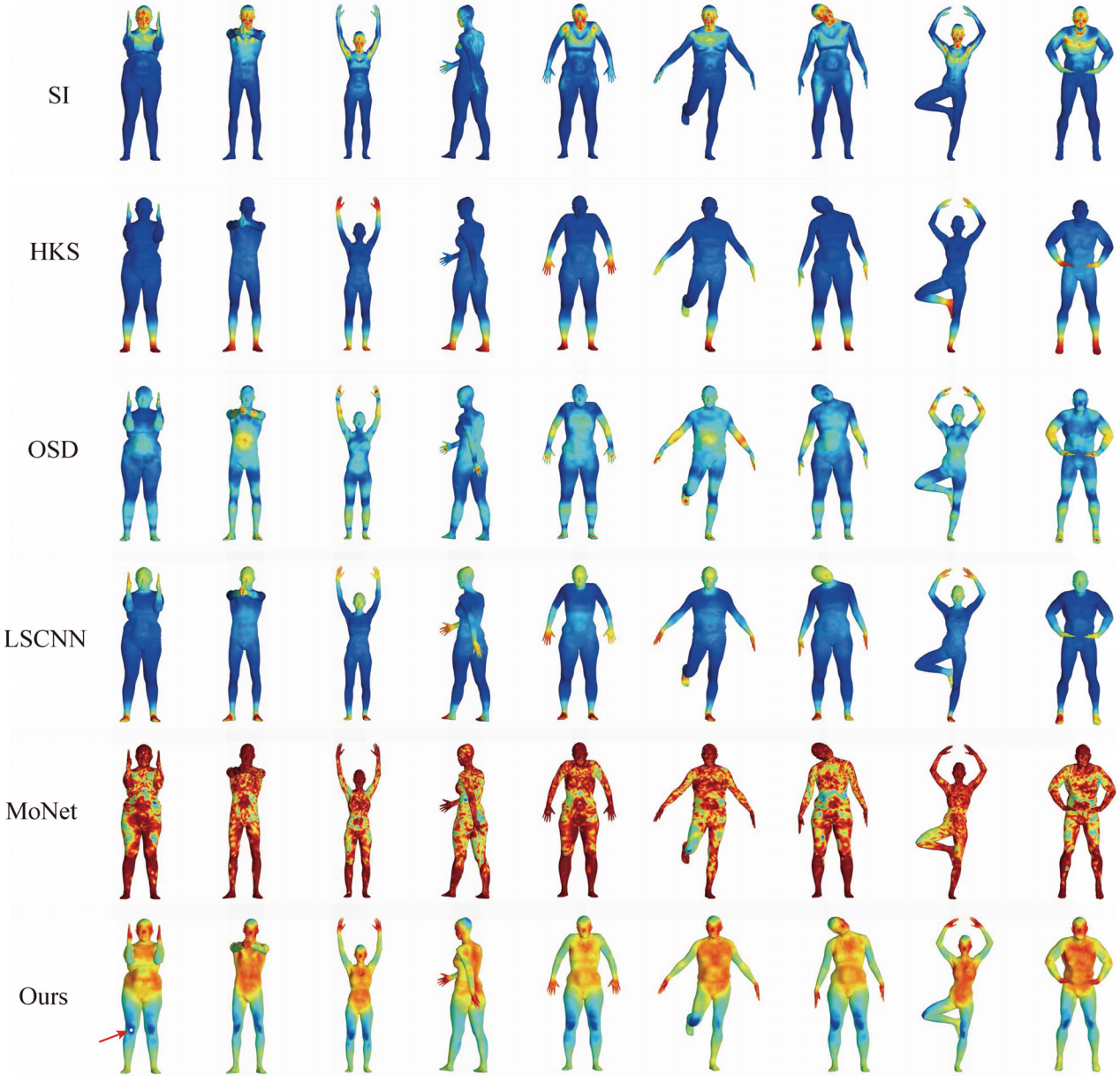

Fig. 9 Similarity map in descriptor space. We compute the distance between the descriptor at a point of the knee on the reference shape (leftmost) and the descriptors at all other points on the same and on other shapes. Colder colors indicate smaller distances in descriptor space.

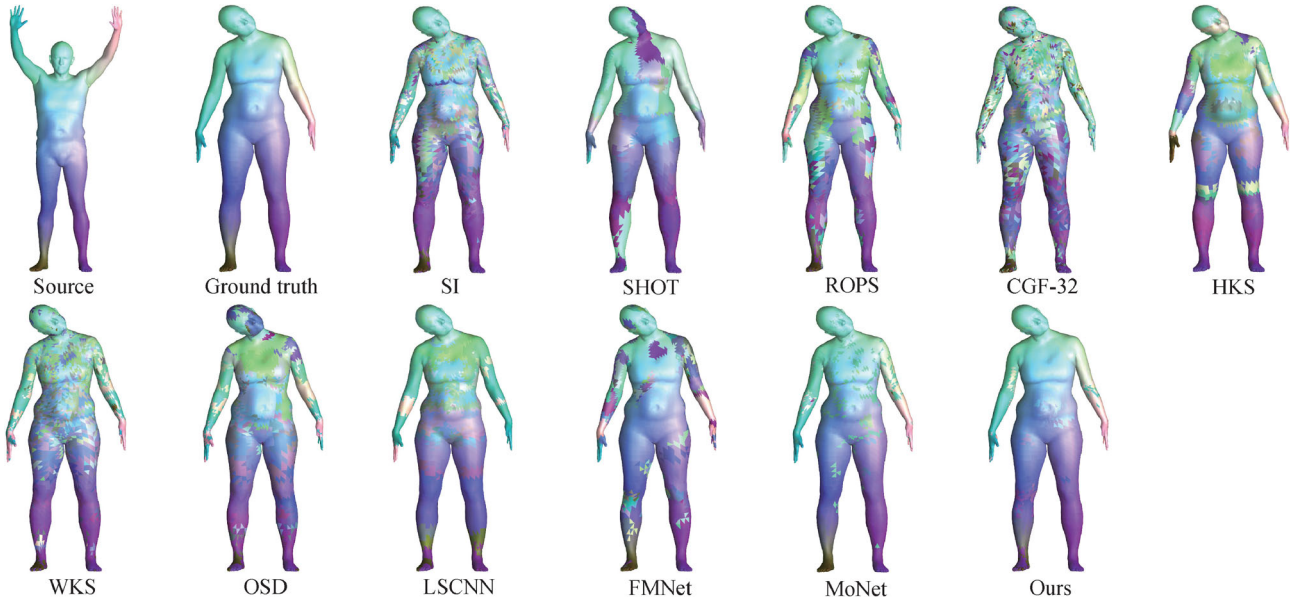

Fig. 10 Visualization of dense correspondence on FAUST dataset as vertex-to-vertex map (corresponding points are shown in the same color). Full reference shape is shown on the top left.

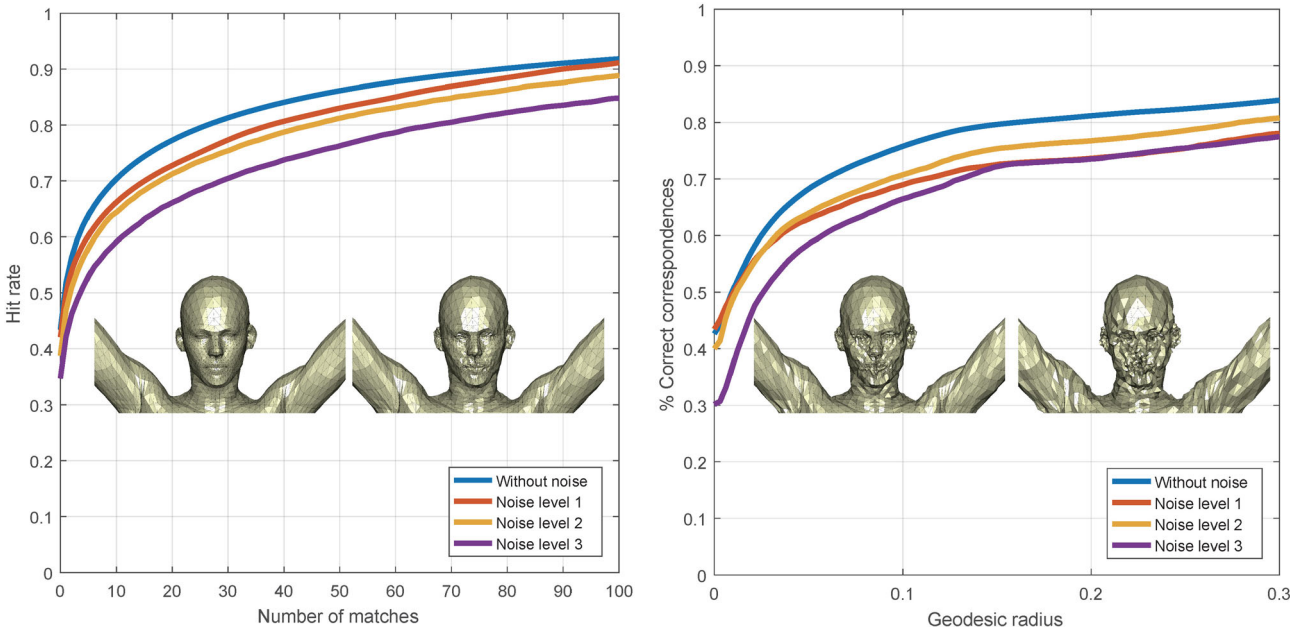

Fig. 11 Performance of our approach with data at different noise levels.

Fig. 12 Partial matching on the SHREC'16 benchmark using point-wise map ($k = 50$ ranks). Each partial shape is matched to the full shape on the left.

certain robustness for this challenging task, we ran our method on the recent public SHREC'16 Partial Correspondence dataset [66]. The shapes in the benchmark are based on the TOSCA high-resolution dataset and span different classes, exemplifying different kinds of incompleteness. The qualitative results in Fig. 12 show that our approach works well with incomplete shapes.

5.4 Comparison to our earlier work

Since the neural network in our previous conference paper [17] is only trained using rigid keypoints, its performance on points defined in highly deformable regions is unsatisfactory—see Fig. 13. By learning from examples also randomly sampled on deformable regions, we now achieve better performance. Furthermore, when using authalic parameterization [17], nearly 3% to 7% of local patches cannot be parameterized correctly. Figure 14 shows failures of the parameterization algorithm used in Ref. [17], caused by degenerate and ill-shaped triangles in a low quality input mesh. In contrast, the DGPC method maps each surface point to a polar

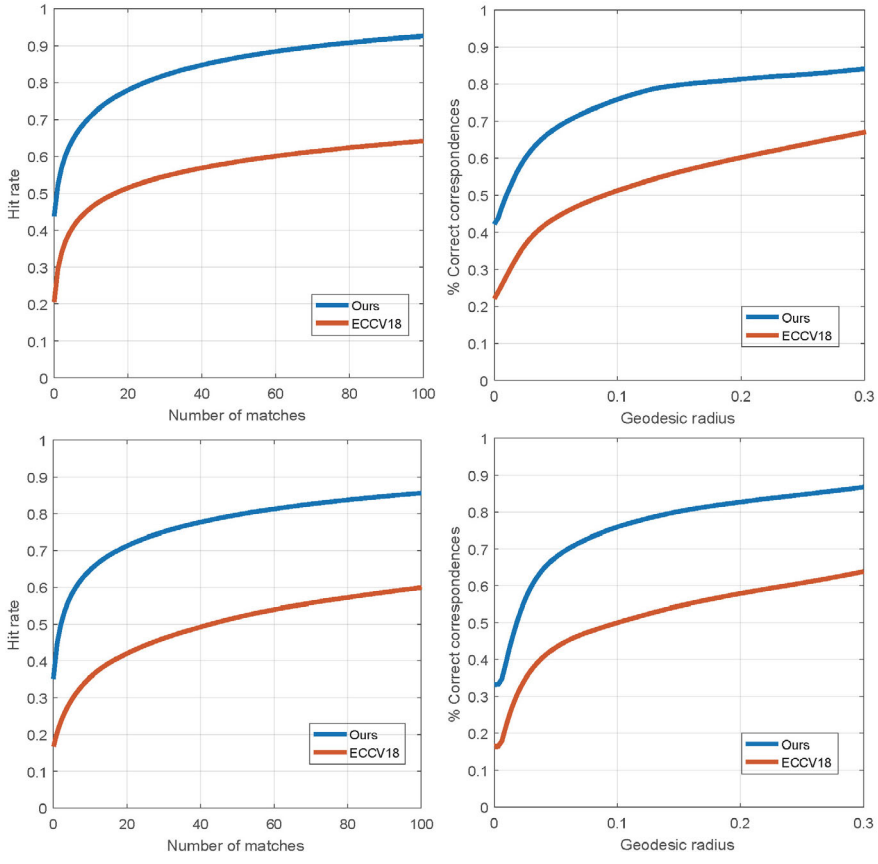

Fig. 13 Comparison to our earlier conference work, for the task of finding dense correspondences on FAUST (top) and SPRING (bottom), assessed using CMC (left) and the Princeton protocol (right).

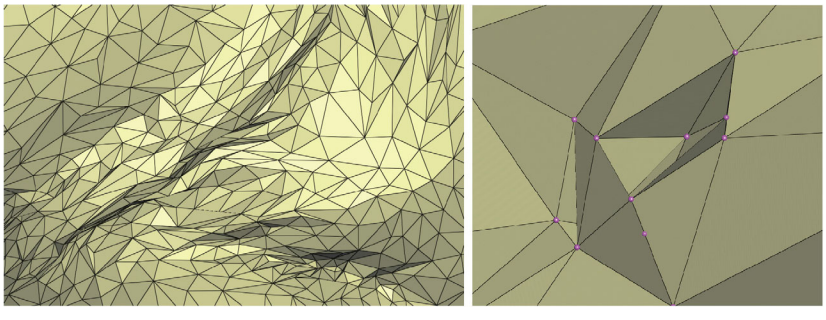

Fig. 14 The ill-shaped (left) and degenerate triangles (right) which cause problems for the parameterization method used in our early conference work [17].

coordinate based on geodesic distance. It works by propagating distances in an ordered fashion, from vertices close to the base point to those farther away. Thus it is robust in the presence of ill-shaped triangles, and for the dataset we used it can handle all points correctly. Furthermore, during testing, the time needed to process one FAUST shape with 6890 points was reduced from ∼8 to ∼2 minutes.

5.5 Limitations

We have successfully used deep neural networks to learn local descriptors for 3D shapes. Nevertheless, since our approach is based on parameterized geometry images, we require that the surface shapes should be locally a manifold triangular mesh. Thus, we currently cannot handle non-manifold local patches or other shape representations, such as point clouds and triangle soups. However, thanks to the many existing remeshing and mesh repair algorithms, a manifold triangle mesh can be easily produced nowadays.

Secondly, since we use low-level features (normal vectors and curvature) for the construction of the geometry images, and these are sensitive to mesh resolution and sampling, our descriptors, as well as other methods, are not very robust to resolution. We would like to address this issue in the future.

6 Conclusions and future work

We have presented a new approach for discriminative descriptor learning for non-rigid 3D shapes. First, we robustly parameterize multi-scale localized neighborhoods of a surface point into geometry images, which encode more geometric information in a local region than rendered views or 3D voxels. Invariance to deformation is then obtained via an efficiently trained triplet network, where we introduce a new metric learning loss function to characterize the relative ordering of the corresponding and non-corresponding point pairs. An efficient feature point sampling approach is also introduced to solve the dense correspondence problem. We have experimentally demonstrated better discrimination ability, robustness, and generalizability of our approach on a variety of datasets.

In future work, we would like to investigate more advanced training strategies or networks (e.g., graph CNNs) to further improve performance. We also wish to extend our flexible approach to other data-driven shape analysis, such as shape segmentation, 3D saliency detection, and point cloud recognition.

Acknowledgements

This work was partially funded by the National Key R&D Program of China (2018YFB2100602), the National Natural Science Foundation of China (61802406, 61772523, 61702488), Beijing Natural Science Foundation (L182059), the CCF–Tencent Open Research Fund, Shenzhen Basic Research Program (JCYJ20180507182222355), and the Open Project Program of the State Key Lab of CAD&CG (A2004) Zhejiang University.

References

[1] Corman, É.; Ovsjanikov, M.; Chambolle, A. Supervised descriptor learning for non-rigid shape matching. In: *Computer Vision – ECCV 2014 Workshops. Lecture Notes in Computer Science, Vol. 8928*. Agapito, L.; Bronstein, M.; Rother, C. Eds. Springer Cham, 283–298, 2015.

[2] Guo, Y. L.; Bennamoun, M.; Sohel, F.; Lu, M.; Wan, J. W. 3D object recognition in cluttered scenes with local surface features: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 36, No. 11, 2270–2287, 2014.

[3] Lian, Z. H.; Godil, A.; Bustos, B.; Daoudi, M.; Hermans, J.; Kawamura, S.; Kurita, Y.; Lavoué, G.; Van Nguyen, H.; Ohbuchi, R.; et al. A comparison of methods for non-rigid 3D shape retrieval. *Pattern Recognition* Vol. 46, No. 1, 449–461, 2013.

[4] Van Kaick, O.; Zhang, H.; Hamarneh, G.; Cohen-Or, D. A survey on shape correspondence. *Computer Graphics Forum* Vol. 30, No. 6, 1681–1707, 2011.

[5] Wang, Y. Q.; Guo, J. W.; Yan, D. M.; Wang, K.; Zhang, X. P. A robust local spectral descriptor for matching non-rigid shapes with incompatible shape structures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6231–6240, 2019.

[6] Shah, S. A. A.; Bennamoun, M.; Boussaid, F. A novel 3D vorticity based approach for automatic registration of low resolution range images. *Pattern Recognition* Vol. 48, No. 9, 2859–2871, 2015.

[7] Johnson, A. E.; Hebert, M. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 21, No. 5, 433–449, 1999.

[8] Gal, R.; Cohen-Or, D. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics* Vol. 25, No. 1, 130–150, 2006.

[9] Sun, J.; Ovsjanikov, M.; Guibas, L. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum* Vol. 28, No. 5, 1383–1392, 2009.

[10] Huang, H. B.; Kalogerakis, E.; Chaudhuri, S.; Ceylan, D.; Kim, V. G.; Yumer, E. Learning local shape descriptors from part correspondences with multiview convolutional networks. *ACM Transactions on Graphics* Vol. 37, No. 1, Article No. 6, 2018.

[11] Zeng, A.; Song, S. R.; NieBner, M.; Fisher, M.; Xiao, J. X.; Funkhouser, T. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 199–208, 2017.

[12] Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; Bronstein, M. M. Geometric deep learning on graphs and manifolds using mixture model CNNs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5425–5434, 2017.

[13] Anguelov, D.; Srinivasan, P.; Koller, D.; Thrun, S.; Rodgers, J.; Davis, J. SCAPE: Shape completion and animation of people. In: Proceedings of the SIGGRAPH '05: ACM SIGGRAPH 2005 Papers, 408–416, 2005.

[14] Bronstein, A.; Bronstein, M.; Kimmel, R. In the rigid kingdom. In: *Numerical Geometry of Non-Rigid Shapes*. Springer New York, 119–135, 2008.

[15] Sinha, A.; Bai, J.; Ramani, K. Deep learning 3D shape surfaces using geometry images. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9910*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 223–240, 2016.

[16] Wang, J.; Song, Y.; Leung, T.; Rosenberg, C.; Wang, J. B.; Philbin, J.; Chen, B; Wu, Y. Learning fine-grained image similarity with deep ranking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1386–1393, 2014.

[17] Wang, H. Y.; Guo, J. W.; Yan, D. M.; Quan, W. Z.; Zhang, X. P. Learning 3D keypoint descriptors for non-rigid shape matching. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11212*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 3–20, 2018.

[18] Guo, Y. L.; Bennamoun, M.; Sohel, F.; Lu, M.; Wan, J. W.; Kwok, N. M. A comprehensive performance evaluation of 3D local feature descriptors. *International Journal of Computer Vision* Vol. 116, No. 1, 66–89, 2016.

[19] Frome, A.; Huber, D.; Kolluri, R.; Bülow, T.; Malik, J. Recognizing objects in range data using regional point descriptors. In: *Computer Vision – ECCV 2004. Lecture Notes in Computer Science, Vol. 3023*. Pajdla, T.; Matas, J. Eds. Springer Berlin Heidelberg, 224–237, 2004.

[20] Zaharescu, A.; Boyer, E.; Varanasi, K.; Horaud, R. Surface feature detection and description with applications to mesh matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 373–380, 2009.

[21] Tombari, F.; Salti, S.; di Stefano, L. Unique signatures of histograms for local surface description. In: *Computer Vision – ECCV 2010. Lecture Notes in Computer Science, Vol. 6313*. Daniilidis K.; Maragos P.; Paragios N. Eds. Springer Berlin Heidelberg, 356–369, 2010.

[22] Bronstein, A. M.; Bronstein, M. M.; Guibas, L. J.; Ovsjanikov, M. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics* Vol. 3, No. 1, Article No. 1, 2011.

[23] Guo, Y. L.; Sohel, F.; Bennamoun, M.; Lu, M.; Wan, J. W. Rotational projection statistics for 3D local surface description and object recognition. *International Journal of Computer Vision* Vol. 105, No. 1, 63–86, 2013.

[24] Elad, A.; Kimmel, R. On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 25, No. 10, 1285–1295, 2003.

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

[26] Kokkinos, I.; Bronstein, M. M.; Litman, R.; Bronstein, A. M. Intrinsic shape context descriptors for deformable shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 159–166, 2012.

[27] Litman, R.; Bronstein, A. M. Learning spectral descriptors for deformable shape correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 36, No. 1, 171–180, 2014.

[28] Gao, L.; Cao, Y. P.; Lai, Y. K.; Huang, H. Z.; Kobbelt, L.; Hu, S. M. Active exploration of large 3D model repositories. *IEEE Transactions on Visualization and Computer Graphics* Vol. 21, No. 12, 1390–1402, 2015.

[29] Huang, Q. X.; Zhang, G. X.; Gao, L.; Hu, S. M.; Butscher, A.; Guibas, L. An optimization approach for extracting and encoding consistent maps in a shape collection. *ACM Transactions on Graphics* Vol. 31, No. 6, Article No. 167, 2012.

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

[31] Charles, R. Q.; Hao, S.; Mo, K. C.; Guibas, L. J. PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 77–85, 2017.

[32] Khoury, M.; Zhou, Q.Y.; Koltun, V. Learning compact geometric features. In: Proceedings of the IEEE International Conference on Computer Vision, 153–161, 2017.

[33] Bai, S.; Bai, X.; Zhou, Z. C.; Zhang, Z. X.; Tian, Q.; Latecki, L. J. GIFT: Towards scalable 3D shape retrieval. *IEEE Transactions on Multimedia* Vol. 19, No. 6, 1257–1271, 2017.

[34] Bronstein, M. M.; Bruna, J.; LeCun, Y.; Szlam, A.; Vandergheynst, P. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine* Vol. 34, No. 4, 18–42, 2017.

[35] Boscaini, D.; Masci, J.; Melzi, S.; Bronstein, M. M.; Castellani, U.; Vandergheynst, P. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Computer Graphics Forum* Vol. 34, No. 5, 13–23, 2015.

[36] Masci, J.; Boscaini, D.; Bronstein, M. M.; Vandergheynst, P. Geodesic convolutional neural networks on Riemannian manifolds. In: Proceedings of the IEEE International Conference on Computer Vision Workshop, 37–45, 2015.

[37] Boscaini, D.; Masci, J.; Rodolà, E.; Bronstein, M. Learning shape correspondence with anisotropic convolutional neural networks. In: Proceedings of the Advances in Neural Information Processing Systems, 3189–3197, 2016.

[38] Litany, O.; Remez, T.; Rodola, E.; Bronstein, A.; Bronstein, M. Deep functional maps: Structured prediction for dense shape correspondence. In: Proceedings of the IEEE International Conference on Computer Vision, 5660–5668, 2017.

[39] Biasotti, S.; Cerri, A.; Bronstein, A.; Bronstein, M. Recent trends, applications, and perspectives in 3D shape similarity assessment. *Computer Graphics Forum* Vol. 35, No. 6, 87–119, 2016.

[40] Ovsjanikov, M.; Corman, E.; Bronstein, M.; Rodolà, E.; Ben-Chen, M.; Guibas, L.; Chazal, F.; Bronstein, A. Computing and processing correspondences with functional maps. In: Proceedings of the SIGGRAPH ASIA 2016 Courses, Article No. 9, 2016.

[41] Ovsjanikov, M.; Mérigot, Q.; Mémoli, F.; Guibas, L. One point isometric matching with the heat kernel. *Computer Graphics Forum* Vol. 29, No. 5, 1555–1564, 2010.

[42] Mémoli, F.; Sapiro, G. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics* Vol. 5, No. 3, 313–347, 2005.

[43] Chen, Q. F.; Koltun, V. Robust nonrigid registration by convex optimization. In: Proceedings of the IEEE International Conference on Computer Vision, 2039–2047, 2015.

[44] Vestner, M.; Litman, R.; Rodola, E.; Bronstein, A.; Cremers, D. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6681–6690, 2017.

[45] Coifman, R. R.; Lafon, S.; Lee, A. B.; Maggioni, M.; Nadler, B.; Warner, F.; Zucker, S. W. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America* Vol. 102, No. 21, 7426–7431, 2005.

[46] Ovsjanikov, M.; Ben-Chen, M.; Solomon, J.; Butscher, A.; Guibas, L. Functional maps: A exible representation of maps between shapes. *ACM Transactions on Graphics* Vol. 31, No. 4, Article No. 30, 2012.

[47] Pokrass, J.; Bronstein, A. M.; Bronstein, M. M.; Sprechmann, P.; Sapiro, G. Sparse modeling of intrinsic correspondences. *Computer Graphics Forum* Vol. 32, No. 2pt4, 459–468, 2013.

[48] Kovnatsky, A.; Bronstein, M. M.; Bresson, X.; Vandergheynst, P. Functional correspondence by matrix completion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 905–914, 2015.

[49] Nogneng, D.; Ovsjanikov, M. Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum* Vol. 36, No. 2, 259–267, 2017.

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

[51] Gu, X. F.; Gortler, S. J.; Hoppe, H. Geometry images. *ACM Transactions on Graphics* Vol. 21, No. 3, 355–361, 2002.

[52] Bogo, F.; Romero, J.; Loper, M.; Black, M. J. FAUST: Dataset and evaluation for 3D mesh registration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3794–3801, 2014.

[53] Sipiran, I.; Bustos, B. Harris 3D: A robust extension of the Harris operator for interest point detection on 3D meshes. *The Visual Computer* Vol. 27, No. 11, 963–976, 2011.

[54] Yan, D. M.; Guo, J. W.; Jia, X. H.; Zhang, X. P.; Wonka, P. Blue-noise remeshing with farthest point optimization. *Computer Graphics Forum* Vol. 33, No. 5, 167–176, 2014.

[55] Boscaini, D.; Masci, J.; Rodolà, E.; Bronstein, M. M.; Cremers, D. Anisotropic diffusion descriptors. *Computer Graphics Forum* Vol. 35, No. 2, 431–441, 2016.

[56] Melvaer, E. L.; Reimers, M. Geodesic polar coordinates on polygonal meshes. *Computer Graphics Forum* Vol. 31, No. 8, 2423–2435, 2012.

[57] Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 815–823, 2015.

[58] Vijay Kumar, B, G.; Carneiro, G.; Reid, I. Learning local image descriptors with deep Siamese and triplet convolutional networks by minimizing global loss functions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5385–5394, 2016.

[59] Svoboda, J.; Masci, J.; Bronstein, M. M. Palmprint recognition via discriminative index learning. In: Proceedings of the 23rd International Conference on Pattern Recognition, 4232–4237, 2016.

[60] Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning, 448–456, 2015.

[61] Maas, A. L.; Hannun, A. Y.; Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the 30th International Conference on Machine Learning, 2013.

[62] Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, 249–256, 2010.

[63] Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint* arXiv:1412.6980, 2014.

[64] Yang, Y. P.; Yu, Y.; Zhou, Y.; Du, S. D.; Davis, J.; Yang, R. G. Semantic parametric reshaping of human body models. In: Proceedings of the 2nd International Conference on 3D Vision, 41–48, 2014.

[65] Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint* arXiv:1603.04467, 2016.

[66] Cosmo, L.; Rodolà, E.; Bronstein, M. M.; Torsello, A.; Cremers, D.; Sahillioglu, Y. SHREC'16: Partial matching of deformable shapes. In: Proceedings of the Eurographics Workshop on 3D Object Retrieval, 2016.

Jianwei Guo is an associate professor in the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA). He received his Ph.D. degree in computer science from CASIA in 2016, and bachelor degree from Shandong University in 2011. His research interests include computer graphics and geometry processing.

Hanyu Wang is working toward M.S. and Ph.D. degrees in computer science at the University of Maryland, College Park. In 2017–2018, he was an intern in CASIA. He obtained his bachelor degree from Xi'an Jiaotong University in 2018. His research interests include 3D computer vision and generative models.

Zhanglin Cheng received his Ph.D. degree from CASIA in 2008. He is currently an associate professor with the Shenzhen VisuCA Key Lab, Shenzhen Institutes of Advanced Technology (SIAT), CAS. His research interests include computer graphics and visualization.

Xiaopeng Zhang is a professor in NLPR at CASIA. He received his Ph.D. degree in computer science from the Institute of Software, CAS, in 1999. He received the National Scientific and Technological Progress Prize (second class) in 2004. His main research interests include computer graphics and image processing.

Dong-Ming Yan is a professor in NLPR at CASIA. He received his Ph.D. degree in computer science from Hong Kong University in 2010, and his master and bachelor degrees in computer science and technology from Tsinghua University in 2005 and 2002, respectively. His research interests include computer graphics, geometric processing, and visualization.