Computational Visual Media
DOI 10.1007/s41095-xxx-xxxx-x **Vol. x, No. x, mouth year, xx–xx**

Article Type (Research/Review)

Learning Local Shape Descriptors for Computing Non-Rigid Dense Correspondence

Jianwei Guo[1], **Hanyu Wang**[2], **Zhanglin Cheng**[3](✉), Xiaopeng Zhang[1], 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, where we introduce an intrinsic parameterization method based on geodesic polar coordinates. By this new parameterization, we could robustly generate the geometry images for even badly-shaped triangular meshes. Then a triplet network with shared architecture and parameters is proposed to perform deep metric learning, which aims to distinguish between similar and dissimilar pairs of points. Additionally, a newly designed triplet loss function is minimized for improved and accurate training of the triplet network. Besides, to solve the dense correspondence problem, an efficient sampling approach is utilized to achieve a good compromise between training performance and descriptor quality. At the testing stage, given a geometry image of a point of interest, our network outputs a discriminative local descriptor for it. An extensive comparison for 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 CNNs, Dense correspondence, Geometry images, Non-rigid shapes.

1 Introduction

With the rapid increase of available 3D models, 3D shape analysis becomes 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 [15], object recognition [23], shape retrieval [35], shape correspondence [57, 61], and surface registration [51], to name a few.

Over the last decades, a large number of local descriptors have been actively investigated by the research community. Despite the recent interests, 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 [29], curvature features [19], heat kernel signatures [54], etc. Although these descriptors can represent the local behavior of the shape effectively, the performance of these methods is still largely limited by the representation 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 an either extrinsic or intrinsic manner. The former usually takes multi-

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

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

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

Manuscript received: 2020-02-28;

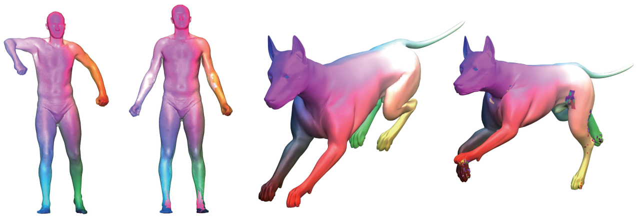

Fig. 1 Our newly-learned descriptor can be easily used to establish dense correspondences between pairs of non-rigid shapes. The Human Body shapes (left) are from SCAPE [2] and Dog shapes (right) are from TOSCA [11].

view images [26] or volumetric representations [66] as input, but it suffers from strong requirements on view selection and low voxel resolutions. While the latter kind of methods generalizes the CNN paradigm to non-Euclidean manifolds [42], they 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 from different datasets, their generalization ability is defective. As a result, these methods perform unstable on different domains.

In this paper, we propose another novel approach for local descriptors learning, that can capture the local geometric essence of a 3D shape. We draw inspiration from the work of [52] which used geometry images for learning global surface features for shape classification. Different from 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, on which standard CNNs can be applied directly. We leverage a triplet network [60] 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 corresponding points distance while maximizes the non-corresponding points distance in descriptor space. In summary, our main contributions are the following:

- We develop a new 3D local descriptor based on specially designed triplet networks, which is dedicated to processing local geometry images encoding low-level geometric information. To generate geometry images, a robust intrinsic parameterization is constructed by utilizing the geodesic polar coordinates.
- We design a novel triplet loss function that can control the dispersion of anchor-positive descriptor distance, thus improving the performance of our descriptor effectively. We also present a tractable and efficient feature points sampling approach, where selecting informative and sufficient number of feature points can lead to efficient and accurate training.
- We show that the proposed concise framework is discriminative for solving dense correspondence problem of deformable shapes. In addition, it has better generalization capability across different datasets than existing descriptors.

We note that a shorter conference version of this paper appeared in [59]. Our initial conference paper did not address the dense correspondence problem. Specifically, this journal paper extends our early conference work through the following aspects:

(1) Since the neural network in the previous conference paper is only trained using rigid keypoints, its performance on points defined in highly deformable regions is not satisfactory. To address this issue, two modifications are adopted in this paper. First, rather than only using rigid keypoints to train the neural network, we generate local geometry images from both landmark keypoints on the rigid parts and the points on truly deformable regions. Second, we add one more intrinsic feature (HKS) to generate a better local descriptor. In this sense, the applicability of the proposed approach to learn dense descriptor fields is achieved.

(2) We introduce a new and more robust parameterization method for local geometry images generation. Instead of the previously used authalic parametrization, we further introduce a more robust parameterization method based on geodesic polar coordinates. This approach works efficiently for badly-shaped triangular meshes, while the authalic parametrization may fail to parameterize some local patches due to imperfections on meshes. In doing so, the process of preparing training data can be greatly accelerated.

(3) Extensive experiments and analysis using more standard quality measures are conducted to verify the effectiveness of our approach. We also study the resistance to noise and partiality, and compare to our early conference work to show the advantages of the new approach.

2 Related Work

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

Hand-crafted local descriptors. Early works focus on deriving shape descriptors based on hand-crafted features[24]. A detailed survey is out of the scope of this paper, so we briefly review some representative techniques. For rigid shapes, some successful *extrinsic* descriptors have been proposed, for example, spin images (SI)[29], 3D shape context (3DSC)[18], MeshHOG descriptor[65], signature of histogram of orientations (SHOT)[56], shape google [10], rotational projection statistics (RoPS)[25]. Obviously, these approaches are invariant under rigid Euclidean transformations, but not under deformations. To deal with isometric deformations, there have been some *intrinsic* descriptors based on geodesic distances[17] or spectral geometry. Such descriptors include heat kernel signature (HKS)[54], wave kernel signatures (WKS)[3], intrinsic shape context (ISC) [32] and optimal spectral descriptors (OSD)[37]. In addition, several methods are proposed to compute shape similarities and correspondences across a large shape database, such as exploring large model repositories [20] or finding high quality point-to-point maps among a collection of related shapes [27]. 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.

Deep-learned local descriptors. Wei et al.[62] employe a CNN architecture to learn invariant descriptors in arbitrary complex poses and clothings, where their system is trained with a large dataset of depth maps. Zeng et al.[66] 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 resolutions due to the high memory and computational cost. Qi et al. [48] propose a deep net framework, called PointNet, that can directly learn point features from unordered point sets to compute shape correspondences. Khoury et al. [30] 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.[26] recently introduce a new local descriptor by taking multiple rendered views [4] in 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, as a result the 2D projection images are not geometrically informative. In addition, whether this approach can be used for non-rigid shape matching is somewhat elusive.

Another family of methods are based on the notion of *geometric deep learning*[12], where they generalize CNN to non-Euclidean manifolds. Various frameworks have been introduced to solve descriptor learning or correspondence learning problems, including localized spectral CNN (LSCNN)[7], geodesic CNN (GCNN)[39], Anisotropic CNN (ACNN)[8], mixture model networks (MoNet)[42], deep functional maps (FMNet)[36], and so on. Different from this kind of methods, our work utilizes geometry images to locally flatten the non-Euclidean patch to the 2D domain so that standard convolutional networks can be used.

Non-rigid shape correspondence. Plenty of algorithms are proposed to compute correspondence between geometric shapes, and several recent surveys [5, 57] and tutorials [45] are available for an 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, which can include similarity of local descriptors [37, 46, 65], geodesic [13, 41, 58] or diffusion distances [14]. Second, *soft correspondence methods* aim to establish approximate correspondences between probability density functions. A family of such methods are based on functional maps [44], which model correspondences as linear operators between spaces of functions on manifolds [33, 43, 47]. Third, *learning-based methods* formulate the correspondence computation as a learning problem [49] or design convolutional neural networks on Euclidean [62] and non-Euclidean [8, 36, 42] domains.

3 Methodology Overview

Given a feature point (or any point of interest) $\mathbf{p}$ on a surface shape $\mathcal{S} \subset \mathbb{R}^3$, our goal is to learn a non-linear feature embedding function $f(\mathbf{p}) : \mathbb{R}^3 \rightarrow \mathbb{R}^d$ which outputs a $d-$dimensional descriptor $X_\mathbf{p} \in \mathbb{R}^d$ 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: $D(X_{\mathbf{p}_i}, X_{\mathbf{p}_j}) = ||X_{\mathbf{p}_i} - X_{\mathbf{p}_j}||_2$. Since our approach is built on the notion of geometry image, in this section, we first briefly review the concept of geometry image,

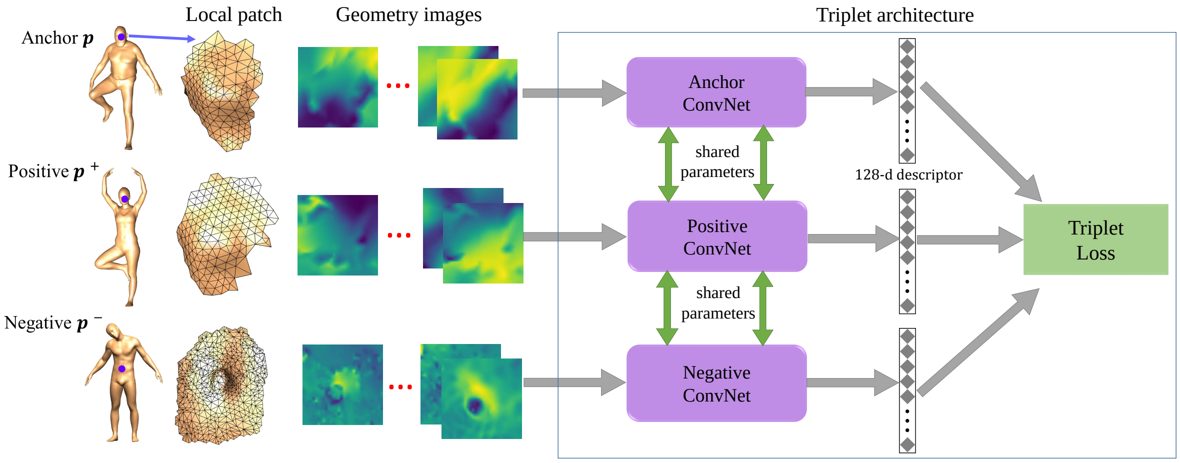

Fig. 2 Overview of our local descriptor training framework. We start with extracting local patches around the keypoints (shown in purple color), and generate geometry images for them. Then a triplet is formed and further processed through a triplet network, where we train this network using an objective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

then the pipeline of our framework is introduced.

Geometry image. The *geometry image* is a new kind of mesh representation technique introduced by Gu et al. [22]. 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, parametrization and quantification. The first step converts the surface into a topological disk using a network of cuts, the second step parametrizes this disk onto a square domain, and the third step creates a regular grid over the square and resamples the surface via the parametrization. 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 has been demonstrated to be useful in various graphics applications, such as rendering, remeshing and shape compression.

Pipeline. The core part of our approach is a full end-to-end learning framework as illustrated in Fig. 2. At off-line training phase, we propose to learn the descriptors by utilizing a triplet network, which are 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 $t = (I(\mathbf{p}), I(\mathbf{p}^{+}), I(\mathbf{p}^{-}))$ contains an anchor point $\mathbf{p}$, a positive point $\mathbf{p}^{+}$, and a negative point $\mathbf{p}^{-}$, where $I(\mathbf{p})$ represents a geometry image encoding the local geometric context around $\mathbf{p}$. By "positive" we mean that $\mathbf{p}$ and $\mathbf{p}^{+}$ are correspondingly similar surface points, and by "negative" we mean $\mathbf{p}^{-}$ is dissimilar to the anchor point $\mathbf{p}$. Based on the training data, we optimize the network parameters by using a minimized-deviation triplet loss function to enforce that, in the final descriptor space, the positive point should be much closer to the anchor point than any other negative points. Once trained, at 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 on 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

A rich and representative training dataset is the key to the success of CNN-based methods. For our non-rigid shape analysis purpose, a good local descriptor should be invariant with respect to noise, transformations, and non-isometric deformations. To meet above requirements, we choose the most recent and particularly challenging FAUST dataset [6], 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.

However, note that our proposed approach is generalizable, that is to say, our network is trained on one dataset, but can be applied to other datasets. In Sec. 5, we will demonstrate the generalization ability of

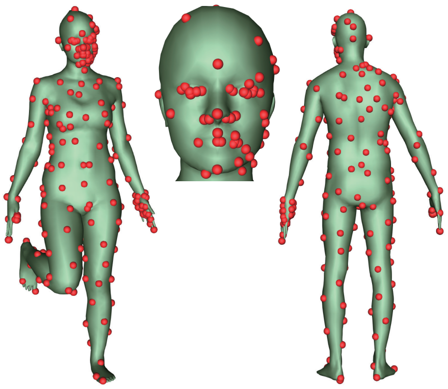

Fig. 3 Illustration of our sampled 256 feature points on two human models in dynamic poses from the FAUST dataset.

our method.

Feature points selection. Intuitively, we could use all the points of the shape to generate geometry images for training. However, this approach does not work well in practice. The reason is two-fold: first, it requires a huge amount of memory space that stores the training data; second, the training process is very hard to converge due to the existence of so many noisy and uninformative local regions. To stay within the memory budget, as well as reduce the training complexity, we propose to use some representative feature points, which consist of two kinds of points. First, 128 landmark keypoints are determined by leveraging any existing 3D interest point detectors (e.g., 3D-Harris [53] used in this paper). Then we randomly sample another 128 points by using the farthest point sampling method on surfaces [63], where the sampling points are uniformly distributed to cover the entire shape. We finally select 256 feature points on the FAUST dataset, as shown in Fig. 3. By this means, we not only consider the keypoints on the rigid parts of the shape, but also take into account the points defined on truly deformable regions. In addition, since the ground-truth point-wise correspondence has already been defined in FAUST, the feature point sampling operation is only performed on one mesh, and each point can be easily retrieved in all the other meshes.

Local geometry images generation. Partially motivated by [52], we use the geometry image representation to capture surface information, where 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 for each point of interest.

We now generate local geometry images for a surface point $\mathbf{p}$. A local patch mesh is first built by extracting the neighbor triangles around the this point. Then we map the local patch to a 2D square grid. To speed up the training process and 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 to the Z axis, and then we rotate the local patch around Z axis to make the principal curvature direction located in the X-Z plane (similar to [9]).

Now we perform a local intrinsic parameterization with low metric distortion in the region of interest around $\mathbf{p}_i$, which is invariant to non-rigid shape transformations. In particular, an efficient method of *Discrete Geodesic Polar Coordinates* (DGPC) [40] is utilized to map each neighbor point $\mathbf{p}_i$ to a polar coordinate (ρ, θ) with respect to the base point $\mathbf{p}$, where ρ is the geodesic distance from $\mathbf{p}_i$ to $\mathbf{p}$, and θ is the polar angle. After the local geodesic polar map is constructed, we convert the geodesic polar coordinates to Cartesian coordinates, hence one 2D geometry image can be generated. This approach is very robust for badly-shaped triangular meshes. The resolution of a geometry image depends on specific applications, here we set its size to be 32×32 for all our experiments. To further solve the rotation ambiguity, we rotate the 2D geometry image $K = 12$ times at 30° intervals. For each rotation, we generate a corresponding geometry image. Finally, in order to capture multi-scale contexts around this point, we extract the local patch at $L = 3$ scales, with neighbor geodesic radius $2.0\rho_0$, $3.5\rho_0$ and $4.5\rho_0$, respectively. Here ρ_0 is computed as 1% of the geodesic diameter of the entire mesh.

While geometry images can be encoded with any suitable feature of the surface mesh, it also depends on specific applications. For solving the sparse correspondence problem, we found that using only two fundamental low-level geometric features is sufficient in our approach: (1) vertex normal direction $\vec{\mathbf{n}}_{\mathbf{v}} = \{n_x, n_y, n_z\}$ at each vertex $\mathbf{v}$, which are calculated by weighted averaging face normals of its incident triangles; (2) two principal curvatures κ_{min} and κ_{max}, that measure the minimum and maximum bending in orthogonal directions of 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. Fig. 4 shows some geometry image examples with different scales and rotations. To learn a dense descriptor, we only need to add one more intrinsic feature: HKS [54]. We select HKS because of its invariance to isometric deformations and

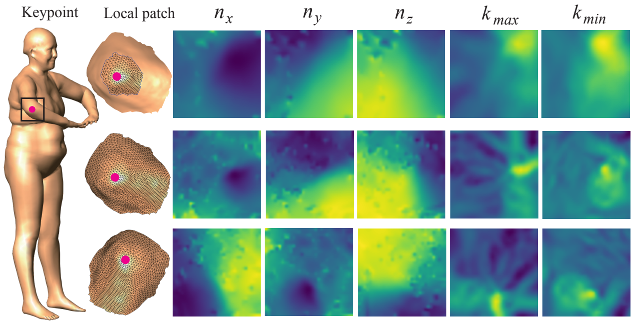

Fig. 4 Geometry images generated around a keypoint. From top to bottom are the geometry images of a smaller scale local patch, a larger scale local patch and a rotated larger scale local patch (rotation angle is 90° in clockwise). From left to right show the geometry images encoding normal $\{n_x, n_y, n_z\}$ and curvature $\{\kappa_{max}, \kappa_{min}\}$ features.

multi-scale property that captures the point's local and global geometric information. Specifically, it could represent increasingly global properties of the shape with increasing time. We will show its effect in the results section.

4.2 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 with the positive or negative points that are hard to be identified from the anchor point. That is to say, given an anchor point $\mathbf{p}$, we want to select a positive point $\mathbf{p}^+$ (*hard positive*) such that $argmax||f(\mathbf{p}_i)-f(\mathbf{p}_i^+)||_2$, and similarly, a negative point $\mathbf{p}^-$ (*hard negative*) such that $argmin||f(\mathbf{p}) - f(\mathbf{p}^-)||_2$. Then, the question becomes: given an anchor point $\mathbf{p}$, how to select the hard positive and negative points? The most straightforward way is to pick samples by hard mining from all of the possible triplets across the whole training set. However, this global manner is time-consuming and may provide misleading information that undermine the convergence of the triplet network, because the noisy or poorly shaped local patches would cause great difficulties for defining good hard triplets.

In our approach, we use a stochastic gradient descent approach to generate the triplets within a mini-batch, similar to the approach used in [50] for 2D face recognition. Specifically, at each iteration of the training stage, we randomly select 16 points out of 256 feature points, 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, M is the number of shape models in training set. Totally, the batch size equals to 128. Then for all anchor-positive pairs within the batch, we select the semi-hard negatives instead of the hardest ones, because the hardest negatives can in practice lead to bad local minima early in training process. Here a semi-hard negative $\mathbf{p}_{semi}^-$ is defined as:

$$||f(\mathbf{p}_i) - f(\mathbf{p}_i^+)||_2 < ||f(\mathbf{p}_i) - f(\mathbf{p}_{semi}^-)||_2. \quad (1)$$

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

4.3 Min-CV Triplet Loss

According to the requirements in real tasks such as shape matching and shape aligning, the pivotal property of an appropriate local descriptor is its discriminability. Since we employ CNNs to embed geometry images of keypoints into a $d-$dimensional Euclidian space, an effective loss function must be designed. It encourages the CNNs to regard that a geometry image of a specific type of surface point is closer to all other geometry images of the same type of surface point and farther from geometry images of any other types of surface point. To achieve this goal, we define the following classic triplet loss function [50]:

$$L = \sum_{i=1}^{N} \left[D_{pos}^i - D_{neg}^i + \alpha\right]_+, \quad (2)$$
$$D_{pos}^i = D\big(f(\mathbf{p}_i), f(\mathbf{p}_i^+)\big),$$
$$D_{neg}^i = D\big(f(\mathbf{p}_i), f(\mathbf{p}_i^-)\big),$$

where N is the batch size, α is the margin distance parameter that we expect between anchor-positive and anchor-negative pairs.

Combined with hard mining, such kinds of triplet loss functions are widely used in various metric learning tasks and perform well or at least acceptable. However, it suffers from some problems in our evaluation dataset. In particular, when training our model with this loss function, the average loss was continually decreasing, however, the single-triplet loss was oscillating violently. Besides, we noticed that for a large number of triplets, the distance between the anchor and the positive geometry images in descriptor space are still considerably large compared with the distance of anchor and negative. Only a few triplets resulted in almost zero loss that led to the decrease in average loss. This phenomenon indicated that our CNNs were failed to learn intrinsic local features but trapped into a local optimum.

To solve this problem, we propose a new triplet loss function, which minimizes the ratio of standard deviation to mean value (also called coefficient of variation-CV) of anchor-positive distance among one batch. This modification is inspired by the intuition that measured by distance in our descriptor space, one geometry image pair of a point should be as similar (at least same order of magnitude) as other geometry image pairs of the same keypoint. By adding this part to the classic triplet loss, we get our minimized-CV (referred to as 'Min-CV') triplet loss:

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

where λ is a tunable non-negative parameter, $\sigma(\cdot)$ calculates the standard deviation among one batch, and $\mu(\cdot)$ calculates the arithmetic mean of one batch. Note that recent work [34, 55] also introduced the mean value and variance/standard deviation into traditional triplet loss. Their loss functions, L_{Kumar} [34] and L_{Jan} [55],are respectively defined as:

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

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

where $\sigma^2(\cdot)$ calculates the variance among one batch. Different from these two approaches, we minimize the CV instead of the variance directly. The reason is that compared to the variance, the CV could measure the dispersion of D_{pos} without being influenced by the numerical scale of the descriptor distance (or the magnitude of the data), e.g., scaling down the descriptor distance will decrease the variance but not affect the CV. Thus, the CV better reflects the degree of data deviation. We make a comparison with these two loss functions in Sec. 5. Furthermore, extensive experiments show that our Min-CV triplet loss is able to help CNNs to learn significant features from one dataset and generalize well to other datasets.

4.4 CNN Architecture and Configuration

Considering the particularity and complexity of our task, we design a special CNN architecture dedicated to processing geometry images in our triplet structure, which is presented below.

Network architecture. Fig. 5 illustrates the architecture of our CNN model. In this figure, we have a compact stack of three convolutional layers ("conv", colored in blue), three pooling layers and two fully connected layers ("fc", colored in green). In particular, each convolutional layer is equipped with the size of convolution kernel shown above and the number of output feature maps shown below. For each fully connected layer, we show the number of units above. The "size" represents the length and the width of the tensor which is fed into next layer, e.g., from left to right, the third layer is a convolutional layer that takes an $8 \times 8 \times 256$ tensor as input and operates $3 \times 3 \times 512$ convolution on it, resulting in an $8 \times 8 \times 512$ tensor flowed to pooling operation. Next, we apply max pooling with a stride of 2 on the output of the first convolutional layer and average pooling with the same stride on the outputs of the other two convolutional layers. Batch normalization (BN) [28] is adopted after each convolution or linear map of input but before non-linear activation. Note that the function of BN layer is different from that of our Min-CV loss: BN layer normalizes the mean and variance of a batch data in deep neural networks. It solves the vanishing gradient problem and exploding gradient problem during back-propagation stage of training. By contrast, our new loss directly act on the output of our neural network and influence the whole network by guiding the training. It is more similar to a kind of prior knowledge based on the intuition that the degree of difference among descriptors of corresponding points should be as small as possible.

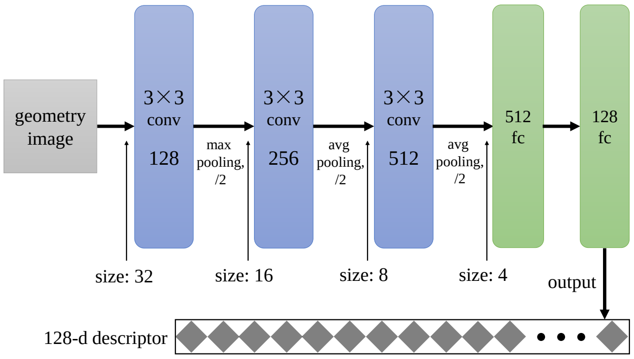

Fig. 5 Detailed network architecture of individual ConvNet shown in Fig. 2.

CNN configuration. The detailed configuration of our triplet CNN is set to adapt our architecture and gain the best performance. Because triplet loss is not as stable as other frequently-used loss functions, our old-version CNN with traditional ReLU activation often suffers from dying ReLU problem that may reduce the effective capacity of our CNN model and then lead to failure in generating meaningful descriptors. To avoid this defect, we employ leaky ReLU [38] 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 by a main part that is identical to the anchor net in our triplet CNNs and a softmax layer using FAUST dataset. The classification labels are the indices of the vertices of the mesh. After it converges, we use the parameters in the main part of the classification network to initialize the convolutional layers of our triplet CNN. Besides, Xavier initialization [21] is adopted to initialize all layers of the classification network and the fully connected layers of our triplet CNNs. In training procedure, Adam algorithm [31] is employed to optimize the loss function. In all of our experiments, the learning rate starts with 0.01 and decreases by a factor of 10 every time when the validation loss begins to oscillate periodically. 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 with qualitative and quantitative evaluations for computing dense correspondence. We also compare to our early conference work [59] to show the advantages of the new approach. The shown results are obtained on an Intel Core i7-3770 Processor with 3.4 GHz and 16GB RAM. Offline training runs on an NVIDIA GeForce TITAN X Pascal (12GB memory) GPU.

5.1 Experimental Setup

Datasets. In addition to FAUST, we further carry out experiments on two other public-domain datasets. The SCAPE [2] contains 71 realistic registered meshes of a particular person in a variety of poses, while the SPRING [64] contains 3000 scanned body models. In these datasets, groundtruth point-wise correspondence between the shapes are known for all points.

Training settings. We separate 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). Any geometry image triplet is generated from one of above subsets depending on the stage it is used for, resulting in the triplet training set, validation set, and testing set, respectively. The training set contains, counted by combination, up to 2.35×10^{13} different triplets that could be fed into our triplet CNNs for training, while the triplet validation set and testing set contains up to 5.08×10^{10} and 1.78×10^{11} triplets, respectively. Our method is implemented based on TensorFlow [1]. Using our hardware configuration shown above, one full training takes about 10 hours.

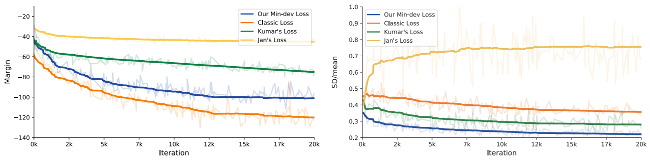

Fig. 6 Training behaviors using different triplet loss functions. Left: positive-negative margin curves. Right: standard deviation mean ratio curves.

Evaluation metrics. To evaluate our learned local descriptor and compare with others, we adopt various measures that are 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 the descriptor space.
- *Princeton protocol*, which measures correspondence quality by plotting the percentage of nearest-neighbor matches that are at most r-geodesically distant from the ground-truth correspondence.
- *similarity map*, which qualitatively 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 as well as its transformations.
- *point-wise map*, which visualizes the correspondence as a vertex-to-vertex map (corresponding points w.r.t a ground-truth reference are shown in the same color).

Description of competing algorithms. We thoroughly compare our method against multiple local descriptors of different types:

- *extrinsic descriptors* including hand-crafted like features spin images (SI) [29], SHOT [56], RoPS [25], and learning-based CGF-32 [30].
- *intrinsic descriptors* including hand-crafted like features HKS [54] and WKS [3], learning-based descriptor OSD [37], and the state-of-the-art deep-learned descriptors LSCNN [7], MoNet [42], FMNet [36].

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 depict the training behaviors evaluated on the validation dataset using classic triplet loss (Eq. 2),

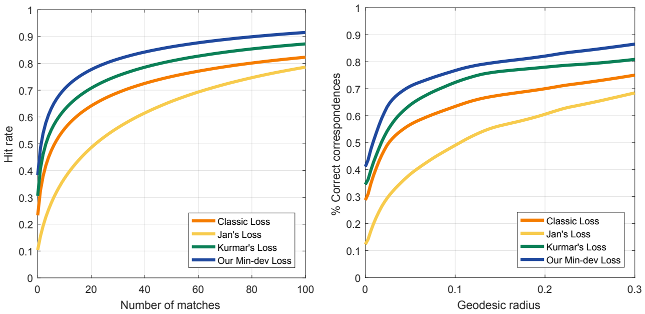

Fig. 7 Performance of different losses on FAUST testing models, measured using the CMC (left) and Princeton protocol (right) plots.

Kumar's loss [34] (Eq. 4), Jan's loss [55] (Eq. 5) and our Min-CV triplet loss (Eq. 3), where the margin distance parameter α is empirically set to a large number (e.g., 100 in this paper) and λ is set to 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, and it is calculated by $\sum_{i=1}^{N}\left[D^i_{pos}-D^i_{neg}\right]_+$. The standard deviation mean ratio curve shows the average ratio $\frac{\sigma(D_{pos})}{\mu(D_{pos})}$ along the iterations. From the curves, we see 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. Compared with Kumar's loss, the training behavior of our loss is better in both comparisons, thus it effectively improves the robustness and generalization ability of our learned descriptor. Taking advantage of this, our descriptor performs stably on various datasets. Further, we 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 has a better performance.

5.3 Dense Correspondence Task

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

Comparison. We measure the performance of all shape descriptors using the CMC and Princeton protocol plots. For all comparisons, the learned methods (OSD, LSCNN, MoNet, FMNet and ours) are trained on FAUST dataset, then applied to other datasets. Fig. 8 reports the comparison results.

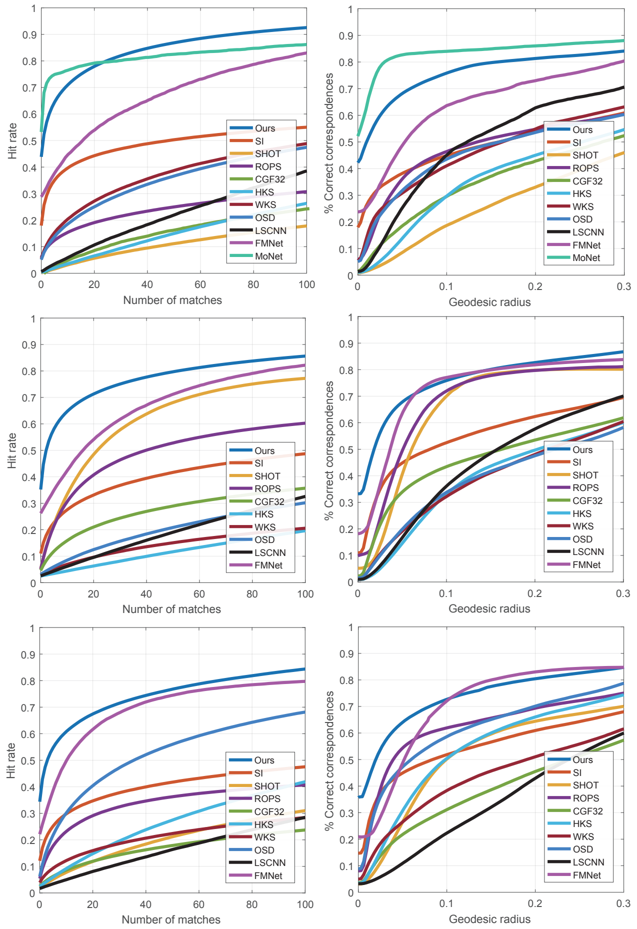

Fig. 8 Performance of different descriptors for solving dense correspondences task on FAUST (top), SPRING (middle)and SCAPE (bottom) datasets, measured using the CMC (left) and Princeton protocol (right) plots.

we observe that MoNet has the best performance on FAUST, however, 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 once it is trained on FAUST, because the labelling spaces can be quite different. Compared to other methods, our performance is higher on FAUST. In addition, we show that our approach has better generalization capability than others on other datasets.

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

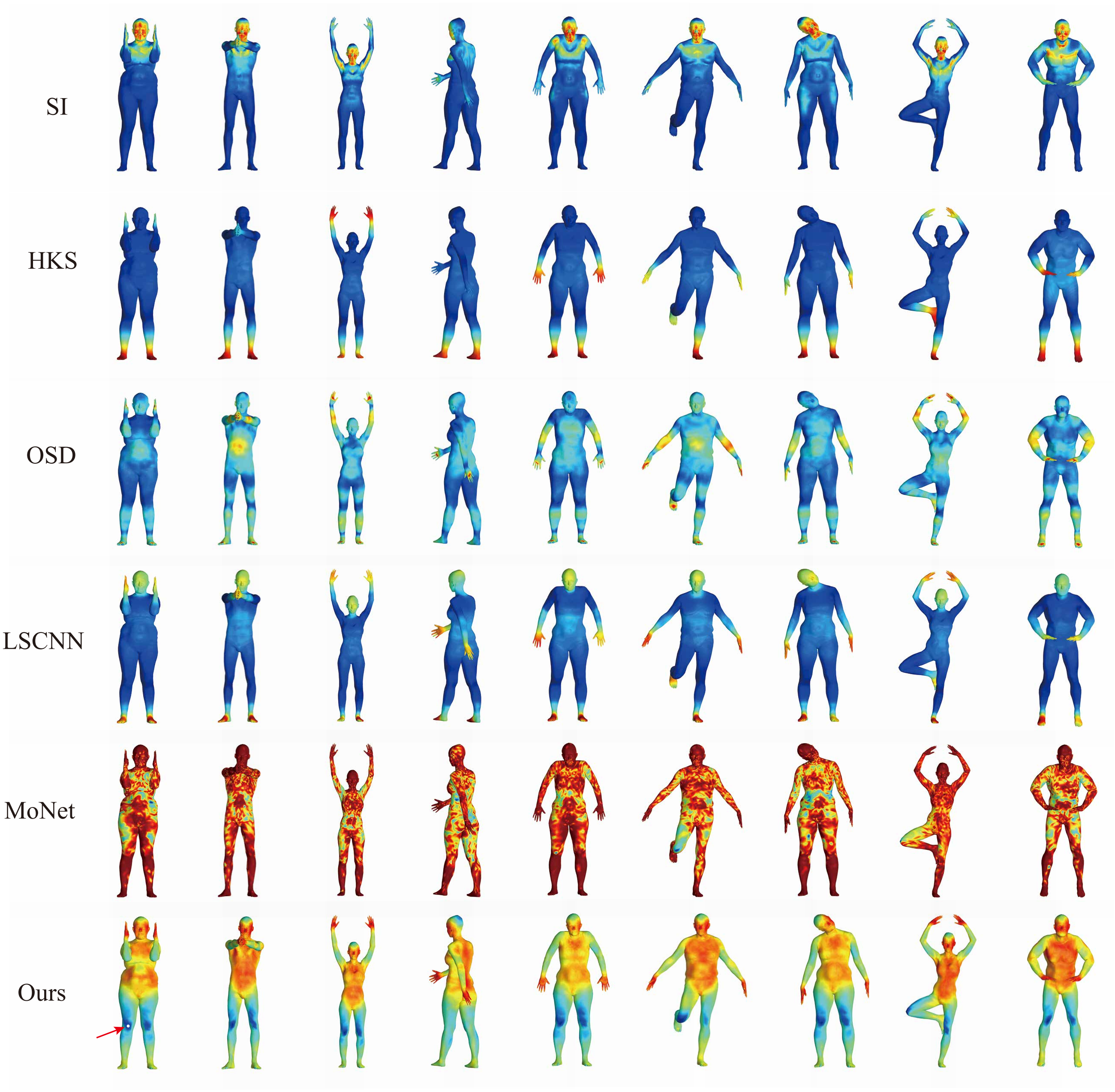

Fig. 9 Similarity map in the descriptor space. We compute the distance between the descriptor at a point of the knee on the reference shape (leftmost) and the descriptors at all other points on the same and on other shapes. Cold color indicates small distance in the descriptor space.

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

Partial matching. Matching deformable 3D shapes under partiality transformations 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 our descriptor has certain robustness for this challenging task, we run our method on the recent public SHREC'16 Partial Correspondence dataset [16]. The shapes in the benchmark are based on the TOSCA high-resolution dataset and span different classes, exemplifying different kinds of partiality. In Fig. 12, the qualitative result shows our approach works well in the case of partiality.

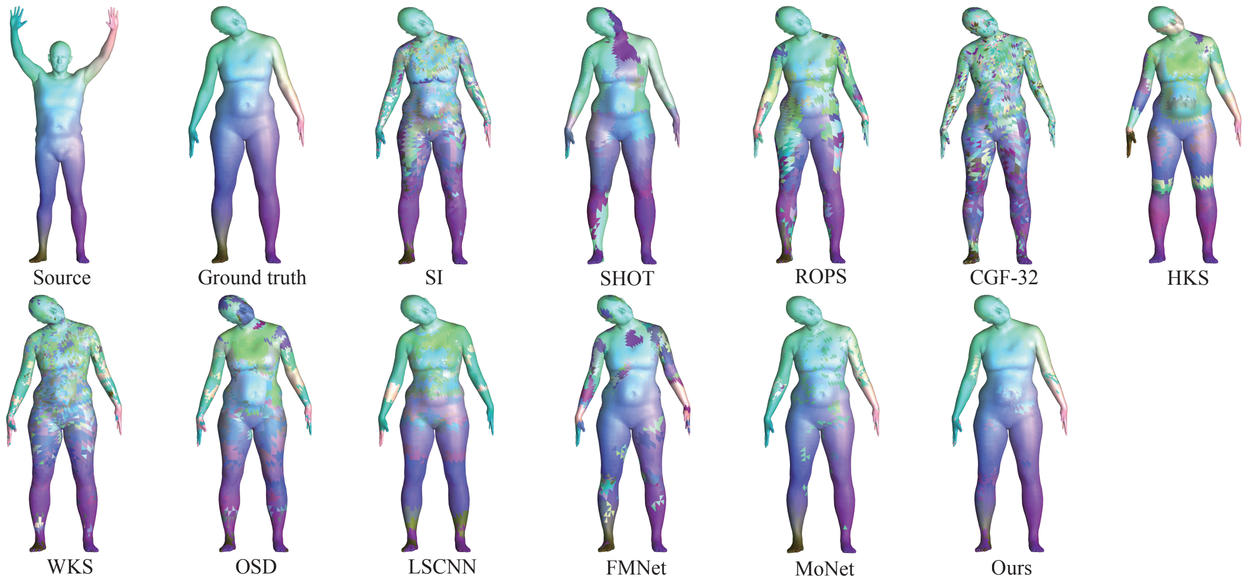

Fig. 10 Visualization of dense correspondence on FAUST dataset as vertex-to-vertex map (corresponding points are shown in the same color). Full reference shape is shown on the top left.

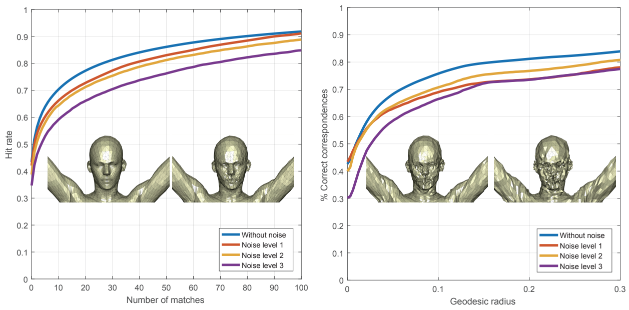

Fig. 11 Performance of our approach reacting with data at different noise levels.

5.4 Comparison to our Conference Work [59]

Since the neural network in the previous conference paper [59] is only trained using rigid keypoints, its performance on points defined in highly deformable regions is not satisfactory, as shown in Fig. 13. By learning from examples that are randomly sampled on truly deformable regions, we now can achieve better performance. Besides, using the authalic parametrization [59], nearly 3% to 7% of local patches cannot be parameterized correctly. Fig. 14 shows the failure cases for the parametrization algorithm used in [59], where the method easily interrupts when handling degenerate and ill-shaped triangles in which some elements of the input mesh have a very low mesh quality. By contrast, our used DGPC method maps each surface point to a polar coordinate based on the 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 for ill-shaped triangles, and for our used dataset it can handle all the points correctly. Meanwhile, at the testing stage, the time processing one FAUST shape with 6890 points is reduced from $\sim$ 8 minutes to $\sim$ 2 minutes.

5.5 Limitations

We successfully used deep neural networks to learn local descriptors for 3D shapes. Nevertheless, since our approach is based on the parameterized geometry images, we require that the surface shapes should be locally 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 so many existing meshing/remeshing and mesh repairing algorithms, manifold triangle mesh can be easily achieved nowadays.

Second, since we use low-level features (normal vectors and curvature) for the construction of the geometry images, our resulting descriptors is sensitive to mesh resolution and sampling because such low-level features are sensitive precisely to these nuisance factors. We find that our approach, as well as other methods, are not very robust to resolution. We would like to address this issue in the future.

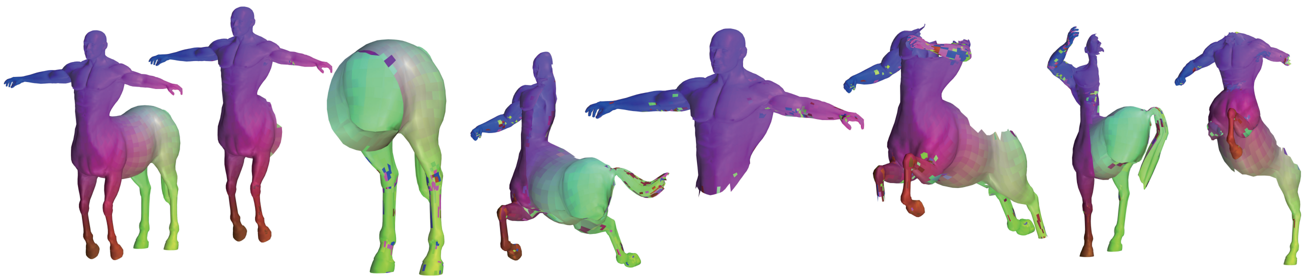

Fig. 12 Results for partial matching on the SHREC'16 benchmark using point-wise map (at top $k = 50$ ranks). Each partial shape is matched to the full shape on the left.

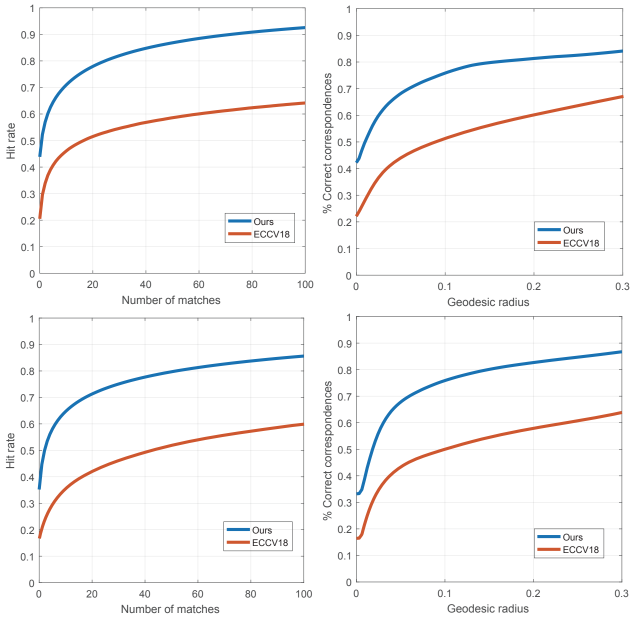

Fig. 13 Comparison to our early conference work (ECCV18) [59] for solving dense correspondences task on FAUST (top) and SPRING (bottom), measured using the CMC (left) and Princeton protocol (right) plots.

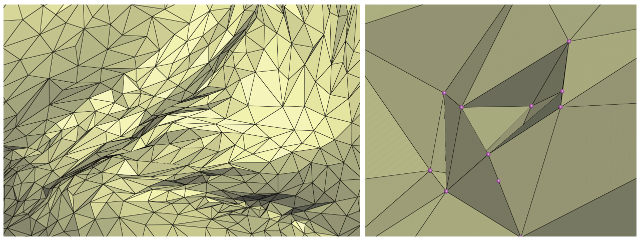

Fig. 14 The ill-shaped (left) and degenerate triangles (right) which cause problems for the parametrization method used in our early conference work [59].

6 Conclusion and Future Work

We presented a new approach for discriminative descriptor learning for non-rigid 3D shapes. First, we robustly parameterize the multi-scale localized neighborhoods of a surface point into the so-called geometry images, which encode more geometric information in the local region than rendered views or 3D voxels. Then the invariance to deformation is 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 points sampling approach is also introduced to solve the dense correspondence problem. We have experimentally demonstrated better discriminability, robustness and generalization capability 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 the performance. We also wish to extend our flexible approach to other data-driven shape analysis, such as shape segmentation, 3D saliency detection, point cloud recognition.

Acknowledgements

This work is partially funded by the National Key R&D Program of China (2018YFB2100602), National Natural Science Foundation of China (61802406, 61772523, 61702488), Beijing Natural Science Foundation (L182059), 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] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2015. https://www.tensorflow.org/.

[2] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 24(3):408–416, 2005.

[3] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1626–1633, 2011.

[4] S. Bai, X. Bai, Z. Zhou, Z. Zhang, Q. Tian, and L. J. Latecki. Gift: Towards scalable 3d shape retrieval. *IEEE Transactions on Multimedia*, 19(6):1257–1271, 2017.

[5] S. Biasotti, A. Cerri, A. Bronstein, and M. Bronstein. Recent trends, applications, and perspectives in 3d shape similarity assessment. In *Computer Graphics Forum*, volume 35, pages 87–119. Wiley Online Library, 2016.

[6] F. Bogo, J. Romero, M. Loper, and M. J. Black. Faust: Dataset and evaluation for 3d mesh registration. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3794–3801, 2014.

[7] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Computer Graphics Forum (Proc. SGP)*, 34(5):13–23, 2015.

[8] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing (NIPS)*, pages 3189–3197, 2016.

[9] D. Boscaini, J. Masci, E. Rodolà, M. M. Bronstein, and D. Cremers. Anisotropic diffusion descriptors. *Computer Graphics Forum (Proc. EUROGRAPHICS)*, 35(2):431–441, 2016.

[10] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Trans. on Graphics*, 30(1):1, 2011.

[11] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes.* Springer Science & Business Media, 2008.

[12] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[13] Q. Chen and V. Koltun. Robust nonrigid registration by convex optimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2039–2047, 2015.

[14] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. 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*, 102(21):7426–7431, 2005.

[15] É. Corman, M. Ovsjanikov, and A. Chambolle. Supervised descriptor learning for non-rigid shape matching. In *European Conference on Computer Vision*, pages 283–298. Springer, 2014.

[16] L. Cosmo, E. Rodolà, M. Bronstein, A. Torsello, D. Cremers, and Y. Sahillioglu. Shrec'16: Partial matching of deformable shapes. *Proc. 3DOR*, 2, 2016.

[17] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):1285–1295, 2003.

[18] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *European Conference on Computer Vision (ECCV)*, pages 224–237, 2004.

[19] R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Trans. on Graphics*, 25(1):130–150, 2006.

[20] L. Gao, Y. Cao, Y. Lai, H. Huang, L. Kobbelt, and S. Hu. Active exploration of large 3d model repositories. *IEEE Transactions*

on Visualization and Computer Graphics, 21(12):1390–1402, Dec 2015.

[21] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[22] X. Gu, S. J. Gortler, and H. Hoppe. Geometry images. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 21(3):355–361, 2002.

[23] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan. 3d object recognition in cluttered scenes with local surface features: a survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(11):2270–2287, 2014.

[24] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok. A comprehensive performance evaluation of 3d local feature descriptors. *Int. Journal of Computer Vision*, 116(1):66–89, 2016.

[25] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan. Rotational projection statistics for 3d local surface description and object recognition. *Int. Journal of Computer Vision*, 105(1):63–86, 2013.

[26] H. Huang, E. Kalogerakis, S. Chaudhuri, D. Ceylan, V. Kim, and E. Yumer. Learning local shape descriptors from part correspondences with multi-view convolutional networks. *ACM Trans. on Graphics*, 30(1):1, 2017.

[27] Q.-X. Huang, G.-X. Zhang, L. Gao, S.-M. Hu, A. Butscher, and L. Guibas. An optimization approach for extracting and encoding consistent maps in a shape collection. *ACM Trans. on Graphics*, 31(6):1–11, 2012.

[28] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[29] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[30] M. Khoury, Q.-Y. Zhou, and V. Koltun. Learning compact geometric features. In *International Conference on Computer Vision (ICCV)*, 2017.

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

[32] I. Kokkinos, M. M. Bronstein, R. Litman, and A. M. Bronstein. Intrinsic shape context descriptors for deformable shapes. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 159–166. IEEE, 2012.

[33] A. Kovnatsky, M. M. Bronstein, X. Bresson, and P. Vandergheynst. Functional correspondence by matrix completion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 905–914, 2015.

[34] B. Kumar, G. Carneiro, I. Reid, et al. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 5385–5394, 2016.

[35] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H. Van Nguyen, R. Ohbuchi, et al. A comparison of methods for non-rigid 3d shape retrieval. *Pattern Recognition*, 46(1):449–461, 2013.

[36] O. Litany, T. Remez, E. Rodola, A. M. Bronstein, and M. M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, page 8, 2017.

[37] R. Litman and A. M. Bronstein. Learning spectral descriptors for deformable shape correspondence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(1):171–180, 2014.

[38] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, 2013.

[39] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 37–45, 2015.

[40] E. L. Melvær and M. Reimers. Geodesic polar coordinates on polygonal meshes. In *Computer*

Graphics Forum, volume 31, pages 2423–2435. Wiley Online Library, 2012.

[41] F. Mémoli and G. Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics*, 5(3):313–347, 2005.

[42] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

[43] D. Nogneng and M. Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. In *Computer Graphics Forum*, volume 36, pages 259–267. Wiley Online Library, 2017.

[44] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 31(4):30, 2012.

[45] M. Ovsjanikov, E. Corman, M. Bronstein, E. Rodolà, M. Ben-Chen, L. Guibas, F. Chazal, and A. Bronstein. Computing and processing correspondences with functional maps. In *SIGGRAPH ASIA 2016 Courses*, page 9. ACM, 2016.

[46] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas. One point isometric matching with the heat kernel. In *Computer Graphics Forum*, volume 29, pages 1555–1564. Wiley Online Library, 2010.

[47] J. Pokrass, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, and G. Sapiro. Sparse modeling of intrinsic correspondences. In *Computer Graphics Forum*, volume 32, pages 459–468. Wiley Online Library, 2013.

[48] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.

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

[50] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[51] S. A. A. Shah, M. Bennamoun, and F. Boussaid. A novel 3d vorticity based approach for automatic registration of low resolution range images. *Pattern Recognition*, 48(9):2859–2871, 2015.

[52] A. Sinha, J. Bai, and K. Ramani. Deep learning 3d shape surfaces using geometry images. In *European Conference on Computer Vision (ECCV)*, pages 223–240, 2016.

[53] I. Sipiran and B. Bustos. Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27(11):963–976, 2011.

[54] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer Graphics Forum (Proc. SGP)*, volume 28, pages 1383–1392. Wiley Online Library, 2009.

[55] J. Svoboda, J. Masci, and M. M. Bronstein. Palmprint recognition via discriminative index learning. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 4232–4237. IEEE, 2016.

[56] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *European Conference on Computer Vision (ECCV)*, pages 356–369, 2010.

[57] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011.

[58] M. Vestner, R. Litman, E. Rodolà, A. Bronstein, and D. Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *Proceedings of CVPR*, 2017.

[59] H. Wang, J. Guo, D.-M. Yan, W. Quan, and X. Zhang. Learning 3d keypoint descriptors for non-rigid shape matching. In *European Conference on Computer Vision (ECCV)*, pages 3–20. Springer, 2018.

[60] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu.

Learning fine-grained image similarity with deep ranking. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1386–1393, 2014.

[61] Y. Wang, J. Guo, D.-M. Yan, K. Wang, and X. Zhang. A robust local spectral descriptor for matching non-rigid shapes with incompatible shape structures. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 6231–6240, 2019.

[62] L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li. Dense human body correspondences using convolutional networks. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1544–1553, 2016.

[63] D.-M. Yan, J. Guo, X. Jia, X. Zhang, and P. Wonka. Blue-noise remeshing with farthest point optimization. *Computer Graphics Forum (Proc. SGP)*, 33(5):167–176, 2014.

[64] Y. Yang, Y. Yu, Y. Zhou, S. Du, J. Davis, and R. Yang. Semantic parametric reshaping of human body models. In *2nd International Conference on 3D Vision (3DV)*, volume 2, pages 41–48. IEEE, 2014.

[65] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 373–380, 2009.

[66] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

Jianwei Guo is an associate professor in 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 the M.S. and Ph.D. degree in computer science at the University of Maryland - College Park. In 2017-2018, he was an intern in Institute of Automation, Chinese Academy of Science. He obtained his bachelor degree from Xi'an Jiaotong University in 2018. His research interests include 3D computer vision and generative model.

Zhanglin Cheng received the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences (CAS) 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 National Laboratory of Pattern Recognition at Institute of Automation, Chinese Academic of Sciences (CAS). He received his Ph.D. degree in Computer Science from 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 National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences(CAS). 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.