Learning 3D Keypoint Descriptors for Non-Rigid Shape Matching

Hanyu Wang*, Jianwei Guo*[0000−0002−3376−1725], Dong-Ming Yan†[0000−0003−2209−2404], Weize Quan, and Xiaopeng Zhang

NLPR, Institute of Automation, Chinese Academy of Sciences
jianwei.guo@nlpr.ia.ac.cn, yandongming@gmail.com, xiaopeng.zhang@ia.ac.cn

Abstract. In this paper, we present a novel deep learning framework that derives discriminative local descriptors for 3D surface shapes. In contrast to previous convolutional neural networks (CNNs) that rely on rendering multi-view images or extracting intrinsic shape properties, we parameterize the multi-scale localized neighborhoods of a keypoint into regular 2D grids, which are termed as 'geometry images'. The benefits of such geometry images include retaining sufficient geometric information, as well as allowing the usage of standard CNNs. Specifically, we leverage a triplet network to perform deep metric learning, which takes a set of triplets as input, and a newly designed triplet loss function is minimized to distinguish between similar and dissimilar pairs of keypoints. At the testing stage, given a geometry image of a point of interest, our network outputs a discriminative local descriptor for it. Experimental results for non-rigid shape matching on several benchmarks demonstrate the superior performance of our learned descriptors over traditional descriptors and the state-of-the-art learning-based alternatives.

Keywords: Local feature descriptor · Triplet CNNs · Non-rigid Shapes

1 Introduction

Designing local descriptors for 3D surface points is within common interests in both computer vision and computer graphics communities. 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 keypoint. It plays a crucial role in a variety of vision tasks, such as shape correspondence [1, 2], object recognition [3], shape matching [4, 5], shape retrieval [6, 7], and surface registration [8], 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

[*] H. Wang and J. Guo are joint first authors with equal contribution. [†]D.-M. Yan is the corresponding author.

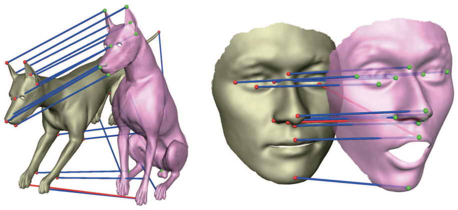

Fig. 1. Our non-rigid shape matching results using a set of landmark points (red and green spheres). The Dog shapes (21 correct matches from 22 keypoints) are from TOSCA [9] and Face shapes (13 correct matches from 15 keypoints) are from [10]. The incorrect correspondences are drawn using red lines.

features, including spin images [11], curvature features [12], heat kernel signatures [13], 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-view images [14] or volumetric representations [15] as input, but is 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 [16], 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 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 recent work of [17] which used geometry images for learning global surface features for shape classification. Different from their work, we construct a small set of geometry images from multi-scale local patches around each keypoint 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. More specifically, we train a well-known triplet network [18, 19] 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 keypoint descriptor based on specially designed triplet networks, which is dedicated to processing local geometry images encoding very low-level geometric information.

- 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 show that the proposed concise framework has better generalization capability across different datasets than existing descriptors.

2 Related Work

A large variety of 3D local feature descriptors have been proposed in the literature. These approaches can be roughly classified into two categories: traditional hand-crafted descriptors and learned local descriptors.

Hand-crafted local descriptors. Early works focus on deriving shape descriptors based on hand-crafted features[20, 21]. 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)[11], 3D shape context (3DSC)[22], MeshHOG descriptor[23], signature of histogram of orientations (SHOT)[24], 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[26] or spectral geometry. Such descriptors include heat kernel signature (HKS)[13], wave kernel signatures (WKS)[27], intrinsic shape context (ISC) [28] and optimal spectral descriptors (OSD)[29]. However, both extrinsic and intrinsic descriptors rely on a limited predefined set of hand-tuned parameters, which are tailored for task-specific scenarios.

Deep-learned local descriptors. Recently, deep learning based methods have attracted large attention because they tend to automatically learn features from raw input data, so as to avoid manually engineered features. Wei et al.[30] 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.[15] 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. [31] propose a deep net framework, named 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.[14] recently introduce a new local descriptor by taking multiple rendered views 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.

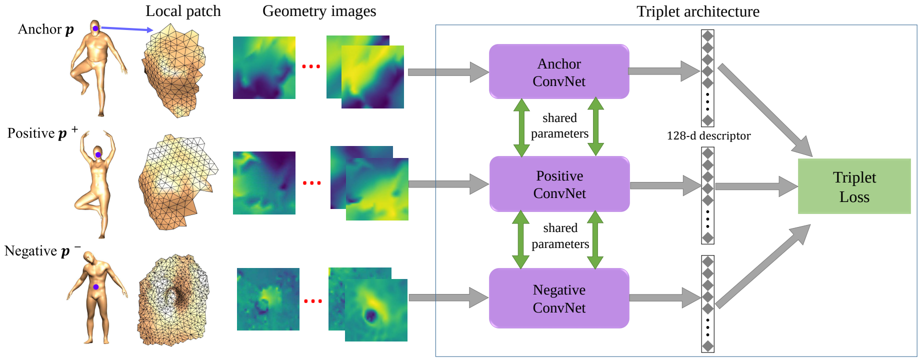

Fig. 2. Overview of our local descriptor training framework. We start with extracting local patches around the keypoints (shown in purple color), and generate geometry images for them. Then a triplet is formed and further processed through a triplet network, where we train this network using an objective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

Another family of methods are based on the notion of *geometric deep learning*[33], 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)[34], geodesic CNN (GCNN)[35], Anisotropic CNN (ACNN)[36], mixture model networks (MoNet)[16], deep functional maps (FMNet)[37], 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.

3 Methodology Overview

Given a keypoint (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 $f(\mathbf{p}) : \mathbb{R}^3 \to \mathbb{R}^d$ 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 keypoints is as small as possible. In this paper, we use the L_2 Euclidean norm as the similarity metric between descriptors: $D(X_{\mathbf{p}_i}, X_{\mathbf{p}_j}) = ||X_{\mathbf{p}_i} - X_{\mathbf{p}_j}||_2$.

Geometry image. Due to space limitations, here we just briefly review the concept of the *geometry image*, which is a new kind of mesh representation technique introduced by Gu et al. [38]. It represents an irregular mesh as a 2D image by parametrizing it onto a square domain. Using this parametrization, the geometric properties of the original mesh can be resampled and encoded into the pixels of an image. In order to parametrize arbitrary mesh onto a square, the mesh should be firstly cut into a topological disk.

Pipeline. The core part of our approach is a newly proposed 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 keypoints, 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, we could generate a 128-d local descriptor for a keypoint by applying the individual ConvNet on one input 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 [39], 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 our method.

Keypoints annotation. To detect the keypoints, we propose a semi-automatic approach. First, candidate keypoint locations can be determined by leveraging any 3D interest point detectors (e.g., 3D-Harris [40]). Then we manually adjust them by removing unsuitable candidates or adding some missing keypoints. Fortunately, since the ground-truth point-wise correspondence has already been defined in FAUST, the keypoint detection operation is only performed on one mesh, and each keypoint can easily be retrieved in all the other meshes. Thus it does not require too much manual effort. We finally annotate 48 keypoints on the FAUST dataset, as shown in Fig. 3.

Local geometry images generation. Partially motivated by [17], 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 keypoint.

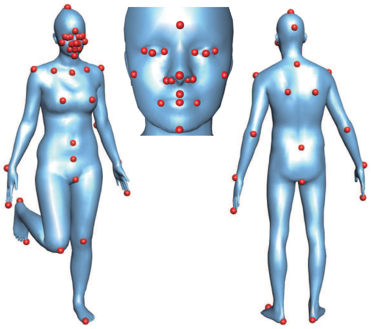

Fig. 3. Illustration of our annotated keypoints on two human models in dynamic poses in the FAUST dataset.

We now generate local geometry images for each keypoint. A local patch mesh is first built by extracting the neighbor triangles around the keypoint. Then we map the local patch to a 2D square grid. Sinha et al. [17] have demonstrated that geometry images using authalic parameterization encode more information of the shape as compared to conformal geometry images, especially when the resolution of the geometry images is limited. In our approach, we perform an authalic and intrinsic parameterization method [41] which minimizes the intrinsic distortion, then the local patch is resampled to generate one geometry image using this parameterization. Nevertheless, other appropriate parameterization methods, such as the geodesic polar coordinates used in [35], could also be used. The resolution of a geometry image depends on specific applications, here we set its size to be 32×32 for all our experiments. Additionally, to be invariant to rotation, we rotate the local patch $K = 12$ times at 30° intervals around the average normal direction of faces, and align it with respect to the principal curvature direction as in [42]. For each rotation, we generate a corresponding geometry image. Furthermore, in order to capture multi-scale contexts around this keypoint, we extract the local patch at $L = 3$ scales, with neighbor radius $6r$, $9r$ and $12r$, respectively. Here r is computed as the average edge length of the entire mesh.

While geometry images can be encoded with any suitable feature of the surface mesh, we found that using only two fundamental low-level geometric features is sufficient in our approach: (1) vertex normal direction $\mathbf{n_v} = \{n_x, n_y, n_z\}$ 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.

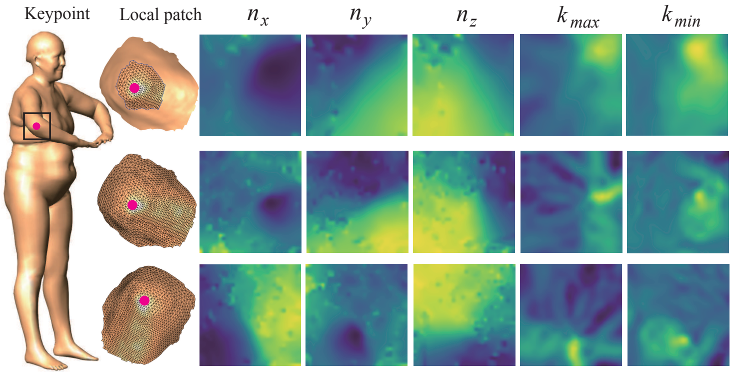

Fig. 4. Geometry images generated around a keypoint. From top to bottom are the geometry images of a smaller scale local patch, a larger scale local patch and a rotated larger scale local patch (rotation angle is 90° in clockwise). From left to right show the geometry images encoding normal $\{n_x, n_y,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

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 lead to poor training, because the noisy or poorly shaped local patches would cause great difficulties for defining good hard triplets. We use a stochastic gradient descent approach to generate the triplets within a mini-batch, similar to the approach used in [43] for 2D face recognition. Specifically, at each iteration of the training stage, we randomly select 16 keypoints out of 48 keypoints, then randomly select 8 geometry images out of $K \times M$ geometry images across the shapes for each keypoint, where $K = 12$ is the number of rotated geometry images of one keypoint 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 is a negative exemplar that is further away from the anchor than the positive, but still closer than other harder negatives. A rigorous definition of the hard and semi-hard negatives is given in the supplemental materials, or refer to [43] for more details.

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 keypoint 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 keypoint is closer to all other geometry images of the same type of keypoint and farther from geometry images of any other types of keypoint. To achieve this goal, we define the following classic triplet loss function [43]:

$$L = \sum_{i=1}^{N} \left[D_{pos}^{i} - D_{neg}^{i} + \alpha \right]_{+}, \tag{1}$$

$$D_{pos}^{i} = D\big(f(\mathbf{p}_i), f(\mathbf{p}_i^{+})\big),$$

$$D_{neg}^{i} = D\big(f(\mathbf{p}_i), f(\mathbf{p}_i^{-})\big),$$

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 keypoint 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:

$$L_{Min-CV} = \lambda \frac{\sigma(D_{pos})}{\mu(D_{pos})} + \sum_{i=1}^{N} \left[D_{pos}^{i} - D_{neg}^{i} + \alpha \right]_{+}, \tag{2}$$

where λ is a tunable non-negative parameter, $\sigma(\cdot)$ calculates the standard deviation among one batch, and $\mu(\cdot)$ calculates the empirical mean of one batch. Note that recent work [44, 45] also introduced the mean value and variance/standard

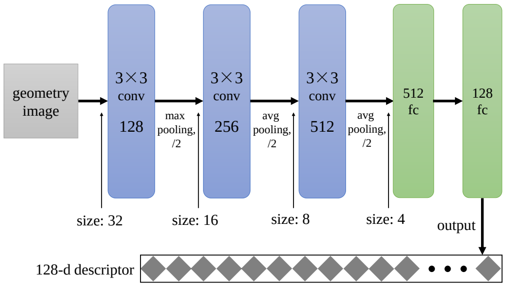

Fig. 5. Detailed network architecture of individual ConvNet shown in Fig. 2.

deviation into traditional triplet loss. Their loss functions (Kumar's [44] and Jan's [45]) are respectively defined as:

$$L_{Kumar's} = (\sigma^2(D_{pos}) + \sigma^2(D_{neg})) + \lambda max(0, \mu(D_{pos}) - \mu(D_{neg}) + \alpha), \quad (3)$$

$$L_{Jan's} = \sigma(D_{pos}) + \sigma(D_{neg}) + \mu(D_{pos}) + \lambda max(0, \alpha - \mu(D_{neg})), \quad (4)$$

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 8×8×256 tensor as input and operates 3×3×512 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) is adopted after each convolution or linear map of input but before non-linear activation.

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 [46] with $slope = 0.1$ for negative input as our activation function. Experimental results demonstrate the effectiveness of this strategy. In addition, to speed up training, we first train a classification network with same architecture and training data of our triplet CNNs except the fully connected layers. The classification labels are the indices of the vertices of the mesh. When it is closed to convergence, its parameters can be used to initialize the convolutional layers of our triplet CNN. Besides, Xavier initialization [47] is adopted to initialize all layers of the classification network and the fully connected layers of our triplet CNNs. In training procedure, Adam algorithm [48] 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 conduct a number of experiments on both real and synthetic datasets to demonstrate the efficacy of our learned local descriptors. 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 experiments. 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.

Datasets. In addition to FAUST, we further carry out experiments on four other public-domain datasets. The SCAPE dataset [49] contains 71 realistic registered meshes of a particular person in a variety of poses, while the TOSCA dataset [9] contains 80 synthetic models of animals and people with near-isometric deformations. The SPRING dataset [50] contains 3000 scanned body models which have also been placed in point to point correspondence. Finally, we test our method on the FACE models used in [10], where some facial expressions are provided.

Training settings. We separate the FAUST dataset into training models (75%), validation models (10%), and testing models (15%). 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 8.1×10^{11} different triplets that could be fed into our triplet CNNs for training (due to imperfections on meshes, local patches of some keypoints on certain models may not

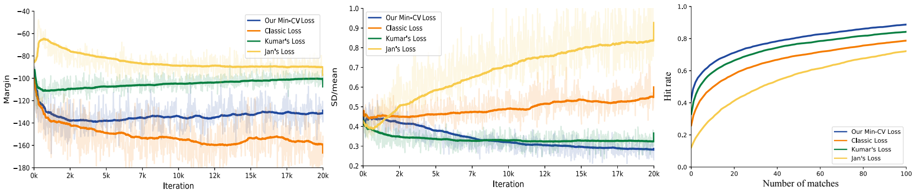

Fig. 6. Training behaviors using different triplet loss functions. Left: positive-negative margin curves. Middle: standard deviation mean ratio curves. Right: CMC curves.

Table 1. Numeric statistics of the CMC curve using different losses in the rightmost plots of Fig. 6.

Dataset	Method	$P_1\%$	$P_5\%$	$P_{10}\%$	$P_{20}\%$
FAUST	Ours	**40.42**	**55.94**	**64.76**	**71.29**
	Ours with Classic loss	25.93	42.37	49.66	57.82
	Ours with Kumar's loss	33.39	51.90	59.04	66.25
	Ours with Jan's loss	12.28	21.92	29.70	40.40

able to be parameterized correctly and thus are discarded), while the triplet validation set and testing set contains up to 1.7×10^9 and 6.1×10^9 triplets, respectively. Our method is implemented based on TensorFlow [51]. Using our hardware configuration shown above, one full training takes about 8 hours.

Next, we demonstrate the effectiveness of our proposed Min-CV triplet loss. In Fig. 6 we depict the training behaviors evaluated on validation dataset using classic triplet loss (Eq. 1), Kumar's loss [44] (Eq. 3), Jan's loss [45] (Eq. 4) and our Min-CV triplet loss (Eq. 2), 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. The positive-negative margin curve shows the average distance between anchor-positive and anchor-negative pairs in each batch, and it is calculated by $\sum_{i=1}^{N}\left[D_{pos}^{i} - D_{neg}^{i}\right]_{+}$. The standard deviation mean ratio curve shows the average ratio $\frac{\sigma(D_{pos})}{\mu(D_{pos})}$ along the iterations. From the left two figures in Fig. 6, 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 behaviors of our loss are better in both figures, thus it effectively improves the robustness and generalization ability of our learned descriptor. Taking advantage of this, our descriptor performs stably on various datasets. From the CMC curves (we will explain it below), our loss still outperforms Kumar's loss. A more thorough comparison is provided in Table 1.

Evaluation metrics. Next, we thoroughly compare our method with several local descriptors of different types, including extrinsic hand-crafted features spin images (SI) [11], SHOT [24], and RoPS [25], intrinsic hand-crafted features HKS [13] and WKS [27], learning-based descriptor OSD [29], and the state-of-the-art deep-learned descriptors LSCNN [34], MoNet [16], FMNet [37]. All the

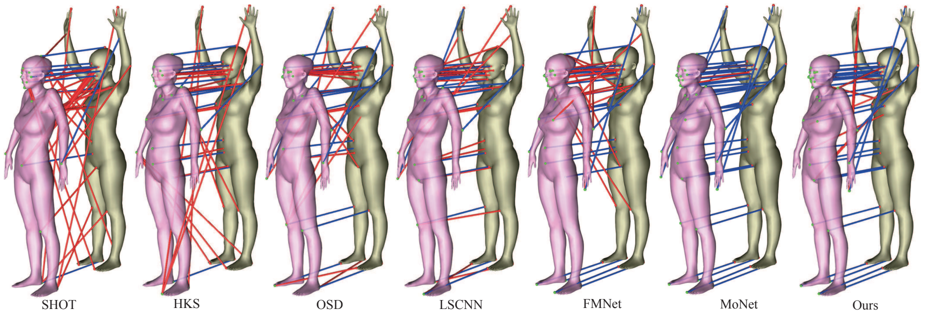

Fig. 7. Selected comparison result of non-rigid shape matching on FAUST, where the incorrect matches are shown in red lines. The total number of used landmark points is 48. From left to right are SHOT (11 matches), HKS (16 matches), OSD (20 matches), LSCNN (19 matches), FMNet (21 matches), MoNet (41 matches) and our descriptor (33 matches).

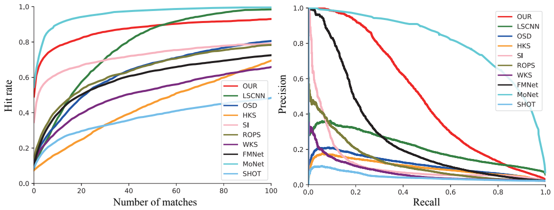

Fig. 8. Performance of different descriptors on FAUST dataset, measured using the CMC (left) and PR (right) plots.

learning-based methods are trained on our above FAUST train-test split. For fair comparison with others, FMNet is not post-processed with the correspondence refinement technique as used in their paper. We believe it makes sense because we focus on the performance of different descriptors, rather than the correspondence. The comparison contains two evaluation metrics that are commonly used in the literature. The first measure is the *cumulative match characteristic* (CMC) curve, which evaluates the probability of finding a correct correspondence among the $k-$nearest neighbors in the descriptor space. Another popular measure is the *precision-recall* (PR) curve with the average precision (i.e., area under PR curve, denoted by AP), that is based on two basic evaluation measures: recall and precision.

Comparison on FAUST dataset. Fig. 8 shows the CMC and PR plots for all the descriptors on the FAUST dataset. The numerical statistics about the curves are presented in Table 2. For a fair and unbiased comparison, we randomly select 200 pairs of shapes from the dataset. And for each pair of shapes, we generate 1000 feature points on them by using 3D-Harris detector [40]. Then the plots are drawn by averaging the calculation results of 200 pairs of shapes. From the curves, we observe that MoNet performs best. However, in fact MoNet does

Table 2. Numeric statistics of the CMC and PR curves for all the methods on different datasets. The best result of each measurement is marked in **bold** font. Here $P_k\%$ is the fraction of correct correspondences within the first k ranks in CMC curve; AP is the average precision, i.e., the area under the PR curve.

Dataset	Method	$P_1\%$	$P_5\%$	$P_{10}\%$	$P_{20}\%$	AP
FAUST	SI	34.66	56.00	61.19	64.14	0.116
	RoPS	14.04	29.95	40.64	51.85	0.128
	SHOT	8.77	17.94	23.36	29.07	0.045
	HKS	7.47	11.71	17.78	24.14	0.098
	WKS	11.26	21.24	28.55	38.98	0.071
	OSD	13.19	23.85	33.45	47.45	0.113
	LSCNN	11.97	22.02	38.12	58.61	0.210
	FMNet	12.43	27.12	38.10	49.59	0.508
	MoNet	**56.93**	**84.62**	**90.82**	**96.93**	**0.677**
	Ours	49.14	70.93	76.63	81.70	0.500
SPRING	SI	43.03	60.33	64.57	69.88	0.445
	RoPS	22.13	40.68	46.00	50.30	0.558
	SHOT	23.10	56.68	69.60	77.14	0.244
	HKS	8.58	14.83	19.75	28.73	0.348
	WKS	13.80	31.07	40.42	49.55	0.299
	OSD	10.52	26.60	37.95	50.58	0.327
	LSCNN	8.80	17.17	24.43	38.53	0.359
	FMNet	13.40	47.48	63.53	78.07	0.528
	Our	**63.30**	**77.71**	**81.70**	**85.99**	**0.631**
FACE	SI	19.33	34.03	40.13	47.23	0.304
	RoPS	26.93	47.27	55.17	60.77	0.629
	SHOT	16.50	35.53	45.93	55.20	0.479
	HKS	14.57	21.97	30.77	38.47	0.273
	WKS	12.67	19.57	24.63	25.43	0.193
	OSD	17.46	24.20	33.93	42.17	0.367
	LSCNN	15.53	18.47	20.83	23.70	0.140
	FMNet	12.00	36.89	48.30	56.67	0.558
	Ours	**35.22**	**63.22**	**71.76**	**80.94**	**0.619**

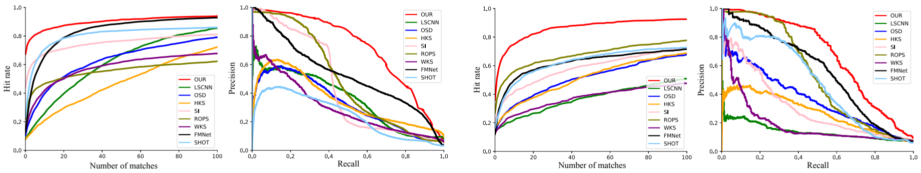

Fig. 9. Performance of different descriptors on SPRING and FACE dataset. The left two figures are the CMC and PR plots on SPRING respectively, while the right two figures are the CMC and PR plots on FACE.

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. Our learned descriptor performs better than all of the extrinsic and intrinsic hand-crafted features. Although our CMC curve converges a little slower than LSCNN, we have higher rank k CMC-percentage, i.e., more corresponding keypoints can be correctly matched in the top k ranks (see Table 2 for details). In addition, we show that our approach has better generalization capability than others in later experiments.

Next, as an application, we test the performance of different local descriptors for non-rigid shape matching, which is performed by computing the landmark correspondences. From the comparison in Fig. 7, we see that our learned local descriptor produces outstanding matching result.

Comparison on other datasets. In order to test our generalization ability, we perform a series of experiments on several other datasets. Here we only show the experimental results on the SPRING and FACE datasets. More exhaustive analysis and comparisons are provided in the supplemental materials. For all comparisons, the learned methods (OSD, LSCNN, FMNet and ours) are trained on FAUST dataset, then applied to other datasets. The evaluation curves are depicted in Fig. 9, and the numeric statistics are shown in Table 2. Note that for the 3D FACE dataset, we manually annotate 15 keypoints by considering the 2D facial point annotations [52] (see Fig.1).

As it can be observed, hand-crafted features behave differently on different datasets, so their robustness is not strong. Another interesting phenomenon is that LSCNN performs similarly with OSD on the SPRING dataset, but it is the worst on the FACE dataset. The reason is that LSCNN uses a domain-dependent spectral basis (the human body shapes in this case) for learning, thus it does not generalize well on different domains. Our approach performs even better on the SPRING than on the FAUST dataset, while a negligible drop in the CMC is observed on the FACE data. Moreover, among all the descriptors, FMNet shows good generalization, but we still achieve the best performance on both datasets. It demonstrates that our approach has the best generalization ability.

6 Conclusion and Future Work

In this paper, we have proposed a new 3D keypoint descriptor based on end-to-end deep learning techniques. A triplet network is designed and efficiently trained, where we introduce a new triplet-based loss function to characterize the relative ordering of the corresponding and non-corresponding keypoint pairs. The significant advantage of our framework is that we can learn the descriptors using local geometry images, that encodes more surface information than rendered views or 3D voxels. Although many local descriptors exist, we have demonstrated better discriminability, robustness and generalization capability of our approach through a variety of experiments.

Though we only use low-level geometric information in this paper, any other extrinsic or intrinsic surface properties can also be encoded into the geometry images. In future work, we would like to extend our flexible approach to other data-driven 3D vision applications, e.g., shape segmentation, 3D saliency detection, etc.

Acknowledgments. We thank anonymous reviewer for their valuable comments and suggestions. This work is partially funded by the National Natural Science Foundation of China (No. 61620106003), Beijing Natural Science Foundation (4184102), National Natural Science Foundation of China (No. 61772523, 61761003, 61571439, 61702488).

References

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

2. Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.: Functional maps: a flexible representation of maps between shapes. ACM Trans. on Graphics (Proc. SIGGRAPH) **31**(4) (2012) 30
3. Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J.: 3D object recognition in cluttered scenes with local surface features: a survey. IEEE Trans. on Pattern Analysis and Machine Intelligence **36**(11) (2014) 2270–2287
4. Corman, É., Ovsjanikov, M., Chambolle, A.: Supervised descriptor learning for non-rigid shape matching. In: European Conference on Computer Vision (ECCV), Springer (2014) 283–298
5. Cosmo, L., Rodola, E., Masci, J., Torsello, A., Bronstein, M.M.: Matching deformable objects in clutter. In: 3D Vision (3DV), 2016 Fourth International Conference on, IEEE (2016) 1–10
6. Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M.: Shape google: Geometric words and expressions for invariant shape retrieval. ACM Trans. on Graphics **30**(1) (2011) 1
7. Lian, Z., 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 **46**(1) (2013) 449–461
8. Shah, S.A.A., Bennamoun, M., Boussaid, F.: A novel 3D vorticity based approach for automatic registration of low resolution range images. Pattern Recognition **48**(9) (2015) 2859–2871
9. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical geometry of non-rigid shapes. Springer Science & Business Media (2008)
10. Sumner, R.W., Popović, J.: Deformation transfer for triangle meshes. ACM Trans. on Graphics (Proc. SIGGRAPH) **23**(3) (2004) 399–405
11. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. on Pattern Analysis and Machine Intelligence **21**(5) (1999) 433–449
12. Gal, R., Cohen-Or, D.: Salient geometric features for partial shape matching and similarity. ACM Trans. on Graphics **25**(1) (2006) 130–150
13. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. Computer Graphics Forum (Proc. SGP) **28**(5) (2009) 1383–1392
14. Huang, H., Kalogerakis, E., Chaudhuri, S., Ceylan, D., Kim, V., Yumer, E.: Learning local shape descriptors from part correspondences with multi-view convolutional networks. ACM Trans. on Graphics **37**(1) (2018) 6:1–6:14
15. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3DMatch: Learning local geometric descriptors from rgb-d reconstructions. In: IEEE Computer Vision and Pattern Recognition (CVPR). (2017) 199–208
16. Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model cnns. In: IEEE Computer Vision and Pattern Recognition (CVPR). (2017) 5425–5434
17. Sinha, A., Bai, J., Ramani, K.: Deep learning 3D shape surfaces using geometry images. In: European Conference on Computer Vision (ECCV). (2016) 223–240
18. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a" siamese" time delay neural network. In: Advances in Neural Information Processing Systems. (1994) 737–744
19. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: IEEE Computer Vision and Pattern Recognition (CVPR). (2014) 1386–1393

20. Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., Kwok, N.M.: A comprehensive performance evaluation of 3D local feature descriptors. Int. Journal of Computer Vision **116**(1) (2016) 66–89
21. Yang, J., Zhang, Q., Cao, Z.: The effect of spatial information characterization on 3D local feature descriptors: A quantitative evaluation. Pattern Recognition **66** (2017) 375–391
22. Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: European Conference on Computer Vision (ECCV). (2004) 224–237
23. Zaharescu, A., Boyer, E., Varanasi, K., Horaud, R.: Surface feature detection and description with applications to mesh matching. In: IEEE Computer Vision and Pattern Recognition (CVPR). (2009) 373–380
24. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: European Conference on Computer Vision (ECCV). (2010) 356–369
25. Guo, Y., Sohel, F., Bennamoun, M., Lu, M., Wan, J.: Rotational projection statistics for 3D local surface description and object recognition. Int. Journal of Computer Vision **105**(1) (2013) 63–86
26. Elad, A., Kimmel, R.: On bending invariant signatures for surfaces. IEEE Trans. on Pattern Analysis and Machine Intelligence **25**(10) (2003) 1285–1295
27. Aubry, M., Schlickewei, U., Cremers, D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In: IEEE International Conference on Computer Vision Workshops (ICCV Workshops). (2011) 1626–1633
28. Kokkinos, I., Bronstein, M.M., Litman, R., Bronstein, A.M.: Intrinsic shape context descriptors for deformable shapes. In: IEEE Computer Vision and Pattern Recognition (CVPR), IEEE (2012) 159–166
29. Litman, R., Bronstein, A.M.: Learning spectral descriptors for deformable shape correspondence. IEEE Trans. on Pattern Analysis and Machine Intelligence **36**(1) (2014) 171–180
30. Wei, L., Huang, Q., Ceylan, D., Vouga, E., Li, H.: Dense human body correspondences using convolutional networks. In: IEEE Computer Vision and Pattern Recognition (CVPR). (2016) 1544–1553
31. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3D classification and segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). (2017) 77–85
32. Khoury, M., Zhou, Q.Y., Koltun, V.: Learning compact geometric features. In: IEEE International Conference on Computer Vision (ICCV). (2017) 153–161
33. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine **34**(4) (2017) 18–42
34. 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 (Proc. SGP) **34**(5) (2015) 13–23
35. Masci, J., Boscaini, D., Bronstein, M., Vandergheynst, P.: Geodesic convolutional neural networks on riemannian manifolds. In: IEEE International Conference on Computer Vision Workshops (ICCV Workshops). (2015) 37–45
36. Boscaini, D., Masci, J., Rodolà, E., Bronstein, M.: Learning shape correspondence with anisotropic convolutional neural networks. In: Advances in Neural Information Processing (NIPS). (2016) 3189–3197

37. Litany, O., Remez, T., Rodola, E., Bronstein, A.M., Bronstein, M.M.: Deep functional maps: Structured prediction for dense shape correspondence. In: IEEE International Conference on Computer Vision (ICCV). Volume 2. (2017) 8
38. Gu, X., Gortler, S.J., Hoppe, H.: Geometry images. ACM Trans. on Graphics (Proc. SIGGRAPH) **21**(3) (2002) 355–361
39. Bogo, F., Romero, J., Loper, M., Black, M.J.: Faust: Dataset and evaluation for 3D mesh registration. In: IEEE Computer Vision and Pattern Recognition (CVPR). (2014) 3794–3801
40. Sipiran, I., Bustos, B.: Harris 3D: a robust extension of the harris operator for interest point detection on 3D meshes. The Visual Computer **27**(11) (2011) 963–976
41. Desbrun, M., Meyer, M., Alliez, P.: Intrinsic parameterizations of surface meshes. In: Computer Graphics Forum. Volume 21., Wiley Online Library (2002) 209–218
42. Boscaini, D., Masci, J., Rodolà, E., Bronstein, M.M., Cremers, D.: Anisotropic diffusion descriptors. Computer Graphics Forum (Proc. EUROGRAPHICS) **35**(2) (2016) 431–441
43. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: IEEE Computer Vision and Pattern Recognition (CVPR). (2015) 815–823
44. Kumar, B., Carneiro, G., Reid, I., 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). (2016) 5385–5394
45. Svoboda, J., Masci, J., Bronstein, M.M.: Palmprint recognition via discriminative index learning. In: Pattern Recognition (ICPR), 2016 23rd International Conference on, IEEE (2016) 4232–4237
46. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. ICML. Volume 30. (2013)
47. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. (2010) 249–256
48. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
49. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. ACM Trans. on Graphics (Proc. SIGGRAPH) **24**(3) (2005) 408–416
50. Yang, Y., Yu, Y., Zhou, Y., Du, S., Davis, J., Yang, R.: Semantic parametric reshaping of human body models. In: 2nd International Conference on 3D Vision (3DV). Volume 2., IEEE (2014) 41–48
51. 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 (2015) https://www.tensorflow.org/.
52. Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: Database and results. Image and Vision Computing **47** (2016) 3–18