Continuous Geodesic Convolutions for Learning on 3D Shapes

Zhangsihao Yang
Arizona State University*

Or Litany
NVIDIA*

Tolga Birdal
Stanford University

Srinath Sridhar
Brown University*

Leonidas Guibas
Stanford University

Abstract

The majority of descriptor-based methods for geometric processing of non-rigid shape rely on hand-crafted descriptors. Recently, learning-based techniques have been shown effective, achieving state-of-the-art results in a variety of tasks. Yet, even though these methods can in principle work directly on raw data, most methods still rely on hand-crafted descriptors at the input layer. In this work, we wish to challenge this practice and use a neural network to learn descriptors directly from the raw mesh. To this end, we introduce two modules into our neural architecture. The first is a local reference frame (LRF) used to explicitly make the features invariant to rigid transformations. The second is continuous convolution kernels that provide robustness to sampling. We show the efficacy of our proposed network in learning on raw meshes using two cornerstone tasks: shape matching, and human body parts segmentation. Our results show superior results over baseline methods that use hand-crafted descriptors.

1. Introduction

Shape descriptors are key to many applications in 3D computer vision and graphics. Examples include shape matching, segmentation, retrieval, and registration, to name a few. A good local descriptor should balance between two opposite forces. On the one hand, it needs to be discriminative enough to uniquely describe a surface local region. At the same time though, it needs to keep robustness to nuisance factors like noise or sampling. Depending on the task, other properties may be required. Common examples are invariance to rigid transformations [19, 17, 18] or isometric deformations [45, 7, 49]. To this end, many descriptors have been manually crafted with built-in invariance. However, these rely on one's ability to analytically model structure in the data which can often be too difficult. Alternatively, machine learning approaches and neural networks in particular, can recover complex patterns from training samples. Further, end-to-end learning is task aware and thus can tailor the learned descriptors to the specified task. Neural networks have proven very powerful in learning descriptors from raw data in various domains including images, text, audio and point clouds. Recently, an exciting research branch termed geometric deep learning has emerged, offering various techniques to process shape represented as meshes. Interestingly though, the vast majority of methods still rely on hand-crafted descriptors at the input to the network as these seem to perform better than working on raw data. In this work, we wish to challenge this practice and learn directly from the raw mesh. Our proposed method is data-driven in nature, however, we integrate the powerful local reference frame (LRF) module commonly used in hand-crafted descriptors into our network. We found through experimentation that structuring the learning through the LRF, is key to reach good performance. A second contribution is the use of continuous convolution kernels. This concept was recently shown to be quite powerful in point cloud networks [65]. Here, we show its usefulness in the context of deformable meshes. We show the efficacy of our proposed network in learning on raw meshes using two cornerstone tasks: shape matching, and human body parts segmentation. Our results show superior results over baseline methods that use hand-crafted descriptors.

Contributions Our contributions can be summarized as follows.

- We introduce a local reference frame (LRF) and continuous convolution kernel modules in the context of deformable shapes.
- Using these, we are able to work directly on raw mesh features and outperform previous methods that take hand-crafted descriptors as input.
- We achieve improved results on deformable shape matching, and human body part segmentation.

*Majority of work done while at Stanford

2. Related Work

2.1. Shape descriptors

Rigid Case Rotation invariant 3D local descriptors are of great interest in the realm of 3D computer vision. Most of the works consider the scenario where the 3D point sets undergo a rigid transformation. The first handcrafted family of works tried to achieve repeatability under those transformations by building certain invariances such as isometry invariance [62, 33, 24, 58, 57, 61]. Many of these works rely extensively on a local reference frame that is assumed to be repeatably constructed on the point sets [51, 47, 46]. With the advances in deep networks, these methods have been replaced by their learned counterparts [67, 35, 18, 17, 22, 13]. Be it data driven or not, a large portion of all these works owe their robustness to the local reference frames unless the input is made invariant to rotations [17, 68]. One of the aims of this paper is to extend the findings regarding locally rigid LRFs to the non-rigid.

Case of Deformable Shapes For the non-rigid shapes, pointwise descriptors are mostly designed to be intrinsic in order to handle isometric deformations [5, 56, 59] and scale [11]. However, designing a descriptor by hand is a cumbersome task. It requires manual balancing of the trade-off between robustness and discriminability, and often relies on heuristics to capture local patterns. Learning based methods are well suited for this task, given the availability of enough training samples. Pioneering works in the field include [9] which extended a "bag of features" approach to non-rigid 3D shape retrieval, and [43] which utilized a Mahalanobis metric learning to optimize a parametric spectral descriptor for shape matching. The success of deep learning in computer vision, has motivated a new active research area termed *Geometric Deep Learning* [10]. A main challenge in this field is how to construct basic operations such as convolution and pooling for geometric data structures. One line of work has opted for extracting geodesic patches [45, 7, 49]. This way, the convolution operator is defined intrinsically on the manifold guaranteeing invariance to isometric deformations. The challenge lies in constructing repeatable local patches with a canonical orientation. For nonrigid shapes these are often based on curvature, but other approaches exist such as [29] and the recently proposed GFrames [46]. On the other hand, spectral techniques [12, 28] generalize a convolutional network through the Graph Fourier transform, thus avoiding the need for a patch. A polynomial parameterization of the learned filters was proposed in [16] in order to spatially localize the kernel and reduce the learning complexity. Common to most learning based methods, is that they use precomputed descriptors as inputs and improve upon them through learned operations. Few works explored directly using raw mesh data. In [52] it was shown that using 3D coordinates underperform SHOT [62] as input; while in [63] the opposite conclusion was reached. MeshCNN [26] defined convolution on triangular meshes treating the *edge* as a first citizen, rather than the nodes. Using angles and edge ratios as input features they were able to work on the raw mesh while being rotation, translation and (uniform) scale invariant. More recently rotation equivariant features were introduced in [66].

2.2. Continuous convolution

Key to our approach is a continuous convolution operator. Realizing that the input points are merely samples from an underlying continuous surface, this makes a much more natural formulation than treating the points as an unstructured cloud. Several works have explored the use of learned continuous kernels. In [42, 20] self similarities in the mesh patches were used via dictionary learning. More recently [45, 7] studied the extraction of local geodesic patches for constructing the equivalent of a convolutional neural network for 2-dimensional manifolds. A generalized form of these was introduced in [49]. In [4] a continuous processing of pointcloud was proposed by defining an extension operator that maps pointclouds to continuous volumetric functions. Another line of works parameterizes the continuous ambient function by another network. This concept was first introduced in [32], where it was termed "dynamic filter" as it allows to modify the convolution filters according to the input, instead of using fixed ones. In [15] a deformation of the convolution kernel was used to dynamically react to the input image patch. A similar approach was taken for pointclouds in [60]. Instead of dynamically modifying the kernel, [39] proposed a $\chi-$transformation in order to to canonicalize the input points. Our continuous convolution resembles the most this line of work, however differently from pointcloud based dynamic filters we utilize the mesh structure to enrich our point features as described in Sec. 3.

2.3. Sampling on geodesic disk

SpiralNet [40] uses either an RNN or an MLP to consume an ordered representation of the points in the vicinity of a vertex. The network is made robust to vertex sampling by randomly sampling different points at train time. This augmentation-based robustness, is replaced in SpiralNet++ [23], by fixing the start point under the assumption of meshes having the same topology which limits the application of their method. Specifically, this approach cannot not be directly applied to the human body segmentation task where the connectivity changes between subjects. In our network, by introducing the local reference frame, we were able to solve the ambiguity mentioned in SpiralNet and successfully apply our network to meshes with different topology.

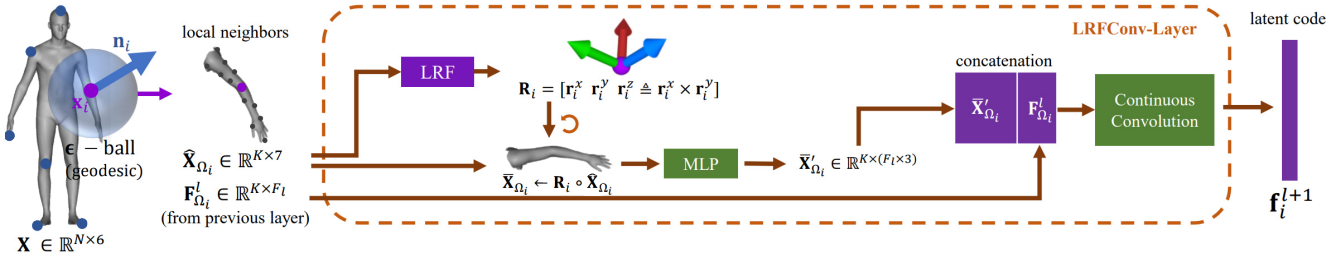

Figure 1. Our LRF-Conv layer. We treat a shape as a set of 3D coordinates and surface normals $\mathbf{X} \in \mathbb{R}^{N\times 6}$. Around each point $\mathbf{x}_i \in \mathbb{R}^6 \equiv [x, y, z, n_x, n_y, n_z]^\top$ we consider a 3D geodesic local neighborhood Ω_i: $\mathbf{X}^M_{\Omega_i} = \{\mathbf{x}_k \in \mathbf{X} : d(\mathbf{x}_i, \mathbf{x}_k) < \tau\}$, where τ is a threshold on the geodesic distance d. We then perform a farthest point sampling (FPS) on this set, $\mathbf{X}_{\Omega_i} = \text{FPS}(\mathbf{X}^M_{\Omega_i}, K)$, and retain K neighbors. We perform a de-mean operation, setting the coordinates of $\mathbf{x}_i$ as the local origin. We further augment each point with the geodesic distance to this new origin as well as the surface normal of the origin and let $\hat{\mathbf{x}}_{ij} \in \mathbb{R}^7 = [\hat{x}_{ij}, \hat{y}_{ij}, \hat{z}_{ij}, \hat{n}^x_{ij}, \hat{n}^y_{ij}, \hat{n}^z_{ij}, g_{ij} \triangleq d(\mathbf{x}_i, \mathbf{x}_{ij})]^\top$ represent the j^{th} point in the local frame of $\mathbf{x}_i$. We use $\hat{\mathbf{X}}_{\Omega_i} = \{\hat{\mathbf{x}}_{ij}\}_{j\in\Omega_i}$ to refer to this augmented, centered local set and this is precisely the input to our LRF-Conv layer. Our layer takes as input a local reference frame [62] $\mathbf{R}_i$ assigned to the patch i. Next, we re-orient the input by $\mathbf{R}_i$ to get $\{\bar{\mathbf{x}}_{ij}\}_j \triangleq \bar{\mathbf{X}}_{\Omega_i} = \mathbf{R}_i \circ \hat{\mathbf{X}}_{\Omega_i}$. Note that the operator $\circ$ does not act on the distances $\{d_{ij}\}_j$ while rotating the rest. We then use an MLP per entity (coordinates, normals, distances) to extend the input description to match the latent dimension and concatenate these with features extracted in the previous layer $\mathbf{F}^l_{\Omega_i}$, where $\mathbf{F}^0_{\Omega_i} \triangleq \mathbf{0}$. This concatenated feature is the input to our trainable continuous convolution yielding the latent features (output) of this $(l+1)^{\text{th}}$ layer. The learnable modules is depicted in green whereas the data containers in purple.

3. LRFConv Layers

At the core of our contribution lies a continuous convolution layer that operates on a locally rectified point set and its geodesics. We call this the *LRFConv* Layer and illustrate it in Figure 1. LRFConv receives a local patch $\hat{\mathbf{X}}_{\Omega_i} = \{\hat{\mathbf{x}}_{ij}\}_j = \{\hat{\mathbf{x}}_{i1}, \hat{\mathbf{x}}_{i2}, \ldots, \hat{\mathbf{x}}_{iK}\}$ that is already centered on a given reference point $\mathbf{x}_i$ as input. This local patch is composed of a collection of those K points lying in the region Ω_i being subsampled using a farthest point sampling algorithm [54, 48]. Along with its 3D coordinates, each j^{th} point in this patch also carries two additional pieces of geometric information: the surface normal $\hat{\mathbf{n}}_{ij} \in \mathbb{S}^3$ and the geodesic distance to the reference (center) point $d(\mathbf{x}_i, \mathbf{x}_{ij})$, a quantity that is preserved under isometries. For a vertex j centered around the i^{th} vertex, this yields a 7-dimensional point representation $\hat{\mathbf{x}}_{ij} \in \mathbb{R}^7$:

$$\hat{\mathbf{x}}_{ij} = \begin{bmatrix} \hat{x}_{ij} & \hat{y}_{ij} & \hat{z}_{ij} & \hat{n}^x_{ij} & \hat{n}^y_{ij} & \hat{n}^z_{ij} & g_{ij} \end{bmatrix}^\top \quad (1)$$

where $g_{ij} \triangleq d(\mathbf{x}_i, \mathbf{x}_{ij})$ and the subscript ij refers to the index of the j^{th} point in the neighborhood of i^{th} vertex: $j \in \Omega_i$. In order to build resilience among six degrees of freedom (DoF) rotations we re-orient this patch using a local reference frame. To this end, we first compute an LRF for all points in the vertex set $\mathbf{X}$. Then each patch $\hat{\mathbf{X}}_{\Omega_i}$ is assigned an LRF in accordance with its index. The axes of this LRF are assembled into a rotation matrix $\mathbf{R}_i \in SO(3)$ which can be used to transform the patch to a canonical alignment: $\bar{\mathbf{X}}_{\Omega_i} = \mathbf{R}_i \circ \hat{\mathbf{X}}_{\Omega_i}$. Here the $\circ$ operator only acts on coordinates and normals separately. We then consider the aligned local patch $\bar{\mathbf{X}}_{\Omega_i} \triangleq \{\bar{\mathbf{x}}_{ij} \in \mathbb{R}^7\}_j$. Note that after such transformations $\bar{\mathbf{x}}_i \equiv \mathbf{o}$, a constant vector. We compute an intermediate feature representation for the whole patch using the entirety of the information collected up to this point and extend it with the help of three multi layer perceptrons (MLPs). We use one MLP per each of the coordinates, normals and geodesic distances in order to match the dimension of the latent features propagated from the previous layer, denoted as $\mathbf{F}^l_{\Omega_i} \in \mathbb{R}^{K\times F_l}$ where F_l is the dimension of the features. This generates $\bar{\mathbf{X}}'_{\Omega_i} \in \mathbb{R}^{K\times(F_l\times 3)}$. Note that an essential quantity in feeding forward the information generated in the previous layers to the later layers of our network is $\mathbf{F}^l_{\Omega_i}$. Thus, we concatenate the output of the said MLPs with $\mathbf{F}^l_{\Omega_i}$ and feed the resulting matrix into a continuous convolution operation producing the output features of this layer $\mathbf{f}^{l+1}_i \in \mathbb{R}^{F_{l+1}}$. Note that in the beginning we initialize the features to zeros: $\mathbf{F}^0_{\Omega_i} \triangleq \mathbf{0}$.

In the following, we will present the details of our LRF computation and dig deeper into the continuous convolutions.

Local Reference Frame (LRF) In order to introduce invariance to translations and rotations as well as building robustness to noise, many of the handcrafted descriptors rely on the estimation of a local coordinate system that varies equivariantly with the global transformation of the object. We use a similar idea to endow our deep features with invariances. A frame of reference (LRF) can be parameterized as a rotation matrix $\mathbf{R}_i = [\mathbf{r}^x_i, \mathbf{r}^y_i, \mathbf{r}^z_i] \in SO(3)$ where each column corresponds to an axis of the local coordinate frame. In our work, we switch between two LRFs depending on whether the data is real or synthetic. For scanned point sets, we use SHOT's LRF [62] thanks to its uniqueness and robustness to noise. The second kind that is suited to less noisy, synthetic meshes is inspired by Texturenet [30]: The first axis is aligned with the surface normal at $\mathbf{x}_i$: $\mathbf{n}_i$. The second axis is determined by the direction of maximum curvature projected on the tangent

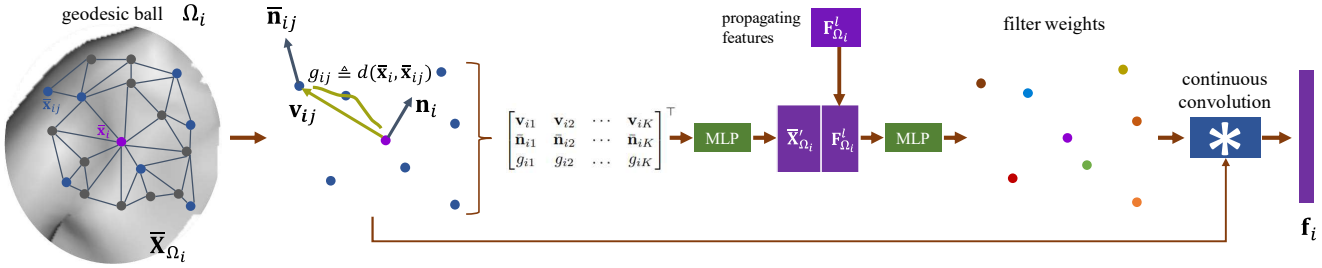

Figure 2. Details of our continuous geodesic convolution. The figure shows in more detail the computation of the input to the MLP and subsequently to the convolution in the unstructured domain. Our MLP that regresses the convolution kernel learns a mapping from the high dimensional point/patch representation to a matrix of weights.

plane defined by the surface normal (first axis). The third and the final axis is simply the cross product of the two: $\mathbf{r}_i^z = \mathbf{r}_i^x \times \mathbf{r}_i^y$. Such LRF construction reduces the degree of ambiguity from four as in [30] to two.

Continuous Graph Convolution An important portion of the success of the CNNs is attributed to the 2D convolutions that are well suited to the structured grid underlying the pixels. Unfortunately, for unstructured 3D data, defining such a grid is not trivial and hence the primary tools for point set processing such as PointNet [53] prefer to ignore the domain and apply point-wise convolutions. Yet, taking into account the neighborhood structure is shown to be advantageous [54]. Thanks to the availability of the mesh structure, we could define an unstructured convolution analogous to the 2D that considers the mesh surface. To this end, we use continuous graph convolutions., whose details we show in Figure 2 and explain below.

As discussed in the introduction of this section, we first extract a patch $\mathbf{X}^M_{\Omega_i} = \{\mathbf{x}_k \in \mathbf{X} : d(\mathbf{x}_i, \mathbf{x}_k) < \tau\}$ according to the geodesic distance τ of the reference point $\mathbf{x}_i$. Then we use FPS to get K points within $\mathbf{X}^M_{\Omega_i}$ and center them given $\mathbf{x}_i$ to get $\hat{\mathbf{X}}_{\Omega_i}$. After that, we rotate $\hat{\mathbf{X}}_{\Omega_i}$ using the LRF to obtain $\bar{\mathbf{X}}_{\Omega_i}$. $\bar{\mathbf{X}}_{\Omega_i}$ carries four feature components per each point in the neighborhood of $\mathbf{x}_i$: the coordinate $\mathbf{v}_{ij}$, the aligned normal $\bar{\mathbf{n}}_{ij}$, the geodesic distance g_{ij}, and the feature from previous layer $\mathbf{f}^l_{ij}$. They could be expressed as $\mathbf{v}_{ij} = \bar{x}_{ij} - \bar{x}_i$, $\bar{\mathbf{n}}_{ij} = \mathbf{R}_i \hat{\mathbf{n}}_{ij}$, and $g_{ij} = d(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_{ij})$. Note that $\mathbf{f}^l_{ij}$ is a rotation invariant feature. We will justify choosing these four feature components in Section 5.5. In our implementation, the first layer omits this feature. Though, signals such as color could be whenever available. For the following layers, $\mathbf{f}^l_{ij}(l > 0)$ is the computed rotation invariant feature. If $\mathbf{f}^l_{ij}$ exists, then we first use three MLPs (MLP_{vb}, MLP_{nb}, and MLP_g) to expand the dimension of $\mathbf{v}_{ij}$, $\bar{\mathbf{n}}_{ij}$, and g_{ij} to match the dimension of $\mathbf{f}^l_{ij}$. If $\mathbf{f}^l_{ij}$ does not exist, we expand the dimension of $\mathbf{v}_{ij}$ to 9. This can help making the network treat each feature's components equally in the next step. After concatenating the expanded input descriptions $\bar{\mathbf{x}}'_{ij}$ and $\mathbf{f}^l_{ij}$, we use another MLP_w to regress a weight matrix whose size is is $F_l \times F_{l+1}$. Subsequently, we apply a discrete convolution operation between this convolution weights matrix and the input features:

$$\mathbf{f}_i^{l+1} = \sum_{j=1}^{K} MLP_w([\bar{\mathbf{x}}'_{ij}, \mathbf{f}^l_{ij}])[\bar{\mathbf{x}}'_{ij}, \mathbf{f}^l_{ij}] \tag{2}$$

With that we update the feature at the reference points $\mathbf{x}_i$. The whole process is depicted in Figure 2.

4. Network Architecture

Our network architecture as shown in Figure 3 consumes each local part separately and involves stacking of the LRFConv layers with skip connections. The upper skip links marked in brown denote that the coordinates, normals, and geodesic distances are also fed forward. In addition, by speaking of LRFConvBN in Figure 3, it means a LRFConv layer followed by a batch normalization layer [31] ,and a non-linear layer, in our case ReLU [50]. The part of network before branching into task specific modules is what we call the *learned shape descriptor (LSD)*. The architecture of LSD is motivated by ResNet [27]. As the layers deepen, we gradually increase the perception field of the network and the length of the feature while having skip connections to increase the depth of the network. This avoids the vanishing gradient problem. Once the LSD is extracted for each vertex (local patch) on the shape, we use it for two tasks, human body segmentation and shape correspondence.

Part Segmentation The fully connected network for part segmentation is composed of 7 residual blocks. For each block the dimension of the output feature is reduced by half with the first dimension being 512. In the last residual block, the dimension equals to the number of classes M which is 8. And a softmax layer is followed. We get an output label $\mathbf{s}_i$ for each patch anchored at $\mathbf{x}_i$. For the segmentation task, we minimize the cross-entropy loss between the output predictions $\{s_i\}$ and the ground truth segmentation

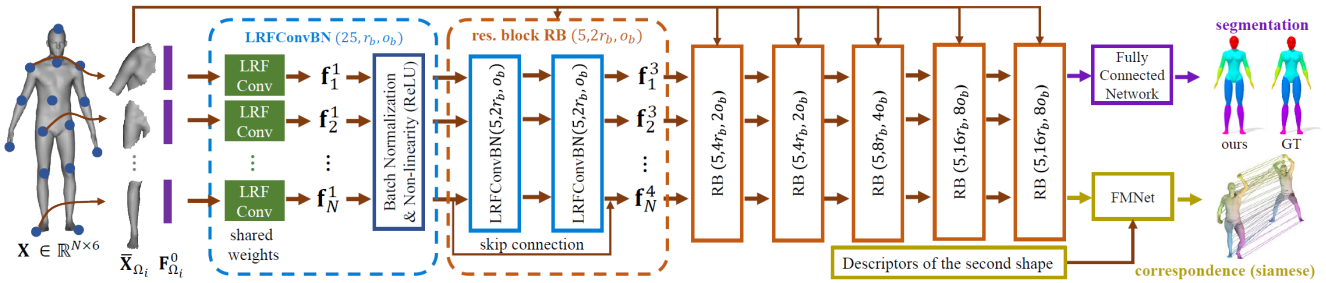

Figure 3. Entirety of our architecture. We extract a local patch centered around each point by querying a geodesic neighborhood. These patches are sent into a sequence of LRFConv layers followed by a batch normalization (BN) and ReLU non-linearity. We also add skip connections to be able to increase the depth and avoid the vanishing gradients. After 13 layers of LRFConv we arrive at our latent features which can be used to address common tasks such as human body segmentation or correspondence estimation. When LRFConv is followed by a BN and ReLU, we call this an *LRFConvBN* layer and parametrize it by three respective arguments: K the number of points in the patch, $r_b = 0.003$ the base radius that determines the size of the neighborhood, $o_b = 32$ where $o_l = \lambda o_b$ sets the dimension of the output features. The residual blocks (RB) that are composed by two LRFConBNs are similarly parametrized. Note that for correspondence estimation we have a weight sharing siamese architecture where the latent features of a paired shape are fed into a deep functional map network [41] along with the features of the base (current) mesh.

labels $\{y_i\}$.

$$\ell = -\frac{1}{M}\sum_{i=1}^{M} y_i \log s_i \tag{3}$$

Correspondence Estimation The fully connected network for estimating correspondence consists of 7 residual blocks. In order to have a fair comparison with FMNet [41], the dimension of the output / feature of each block is set to 352. This is identical to the length of the latent feature used in FMNet. We use the shared weights of LSD followed by the fully connected network to extract the feature from the target shape $\mathcal{Y}$ and the source shape $\mathcal{X}$. Then we follow the loss function proposed in FMNet:

$$\ell = \frac{1}{|\mathcal{X}|}||\mathbf{P} \circ (\mathbf{D}_{\mathcal{Y}}\Pi^*)||_F^2 \tag{4}$$

Note that our correspondence estimation approach resembles FMNet's. However, in addition to our continuous convolutions, we avoid using the handcrafted SHOT descriptors and replace them with LSDs.

5. Experimental Evaluation

5.1. Datasets

We demonstrate the efficacy of our learned descriptor on two cornerstone tasks in shape analysis: dense shape correspondence and part segmentation. To this end, we utilize two datasets.

Part Segmentation. For the segmentation task, we use the human segmentation benchmark introduced in [44]. This dataset consists of 370 models fused from multiple human shape collections: SCAPE [2], FAUST [6], MIT animation datasets [64] and Adobe Fuse [1]. All models were manually segmented into eight parts [34]. The test set is the 18 human models from the SHREC dataset [21]. The variety of data sources makes the problem especially challenging as each collection has a different sampling and appearance. Moreover, the SHREC dataset was used solely for testing which calls for a high generalization ability.

Shape Correspondence. To showcase our descriptor learning module in the task of shape matching we use the FAUST dataset [6]. The data contains 100 human high resolution scans belonging to 10 different individuals at 10 different poses each. The scans were all registered to a parametric model with 6890 vertices and consistent triangulation. We call this set "Synthetic FAUST". We also test our method on the more challenging set of the original scans.

5.2. Part Segmentation

Given an input mesh we use our network to predict for each vertex the part segment it belongs to. At train time, we select 2000 random points from each mesh as input to the network, and use the corresponding segmentation label as the supervision signal. We train our network for 200 epochs. In all our experiments we use the Adam optimizer [37] with a fixed learning rate of 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We compare our results with the two variations of MDGCNN [52] as proposed by the authors, using either raw 3D coordinates or precomputed SHOT [62] descriptors. To better understand the influence of the continuous convolution module (CC) we also compare with a simplified version of our pipeline, where the continuous convolution is replaced by a standard PointNet (PN) [53]. Our results are summarized in Table 1. It can be seen that our method in both of its forms outperforms MDGCNN, while using the continuous convolution further boosts the performance.

In MDGCNN [52] table 2, the authors demonstrated improved performance when using SHOT as input instead of

Method	Input feature	Accuracy
MDGCNN [52]	3D coords	88.61
MDGCNN [52]	$SHOT_{12}$	89.47
Ours PN	3D coords	89.69
Ours CC	3D coords	**89.88**

Table 1. We compare our results with the two variations of MDGCNN [52], using either raw 3D coordinates or precomputed SHOT [62] descriptors. To better understand the influence of the continuous convolution module (CC) we also compare with a simplified version of our pipeline, where the continuous convolution is replaced by a standard PointNet (PN) [53].

raw data. Perhaps surprisingly, in FeaStNet [63], the finding was different: the raw data achieved better performance than when using SHOT. This highlights the importance of the architecture in extracting information from raw data, and can explain why our proposed modification contributed to the improved results.

Importantly, we achieve this by using raw 3D coordinates as input and by which bridges the gap reported in MDGCNN allowing to remove the dependence on manually designed features. This desired behaviour is expected since our network imitates the design philosophy of SHOT. We further present a qualitative evaluation of the part segmentation in Figure 4.

5.3. Rotation invariance

As described in the introduction, depending on the dataset and task at hand, a descriptor should be invariant to different transformations. In the case of human body models and part segmentation it is natural to ask for invariance to rigid transformations and articulations. While the latter is achieved through learning from examples, the former is taken care of by construction using our proposed LRF. Baking invariance into the descriptor by construction is not only natural to the problem setting but also more sample efficient. To see this, we take two baseline methods which are not rotation invariant, and train them with and without rotation augmentation. We also create an augmented test set, where each shape has been rotated in 128 different angles. We summarize the results of these methods and ours in Table 2. As can be seen, when the non rotation invariant networks are tested on rotated examples their performance drops significantly. Adding rotation at train time helps shrinking the performance gap however, both still under perform compared to our rotation invariant solution.

Looking closely at the results of PointNet++ [54] one notices that the results improve on the non-rotated test set by adding the augmentation at train time. This is explained by the fact that the models in both train and test sets are either upright or lying down but with very different distribution between the two poses. What is less intuitive is why this improvement is not achieved for DGCNN. From our experiments we conclude that the method could not benefit from the augmentation and instead reduced to solving the average case, perhaps due to limited capacity. In Figure 5 we compare the results of our network with DGCNN on each of the 18 test shapes. To better visualize the effect of the rotation angle we sort the test samples (x-axis) according to the performance of DGCNN. This clearly shows a gap between samples where the input shape was lying down (first 6 examples) and the ones which were upright.

Train w/ rot	Test w/ rot	PointNet++	DGCNN	Ours(PN)	Ours(CC)
Yes	Yes	85.35	43.88	89.69	**89.88**
Yes	No	85.85	36.99		
No	Yes	56.95	36.65		
No	No	75.81	66.35		

Table 2. The accuracy on human body segmentation of different learning-based approaches under training and test with and without rotation augmentation

5.4. Shape matching

Finding dense shape correspondences between a pair of shapes is one of the most important and most explored problems in shape analysis. As described earlier, state of the art methods utilize hand crafted descriptors as input. Here, instead, we propose to learn the descriptor directly from raw 3D coordinates by utilizing our proposed LRFConv. Our descriptor can in principle be combined with any matching pipeline. In this work we make use of FMNet [41], one of the best performing methods in the task of shape matching. Specifically, it accepts a pair of shapes as input together with their computed Laplacian eigenfunctions and per-point features. Then, both shape features are passed through a (siamese) feed forward network to get refined descriptors. These, in turn, are used to compute a functional map aligning the shape eignefunctions. Finally, these are used to predict a point-to-point soft-correspondences which are converted to match by taking the maximal probability per point. In the original work of [41] and its unsupervised follow up [25] SHOT descriptors were used. Here, instead, we replace it with our learned descriptors and train the network in an end-to-end fashion.

Synthetic FAUST We first demonstrate our performance on the synthetic FAUST dataset described in 5.1. We follow the evaluation protocol as prescribed by [49] were the 100 models are split into 80 train and 20 test shapes, and the matching is performed with respect to a single fixed null shape. We train the network for 200 epochs using the same optimization hyper parameters as described in 5.2. The results are summarized in Figure 7. As can be seen, by using our proposed descriptor we were able to improve upon the results of FMNet with SHOT. We include the performance of other methods for the sake of completeness.

In Figure 6 we show a qualitative evaluation of our matching results. When evaluating the temporal complex-

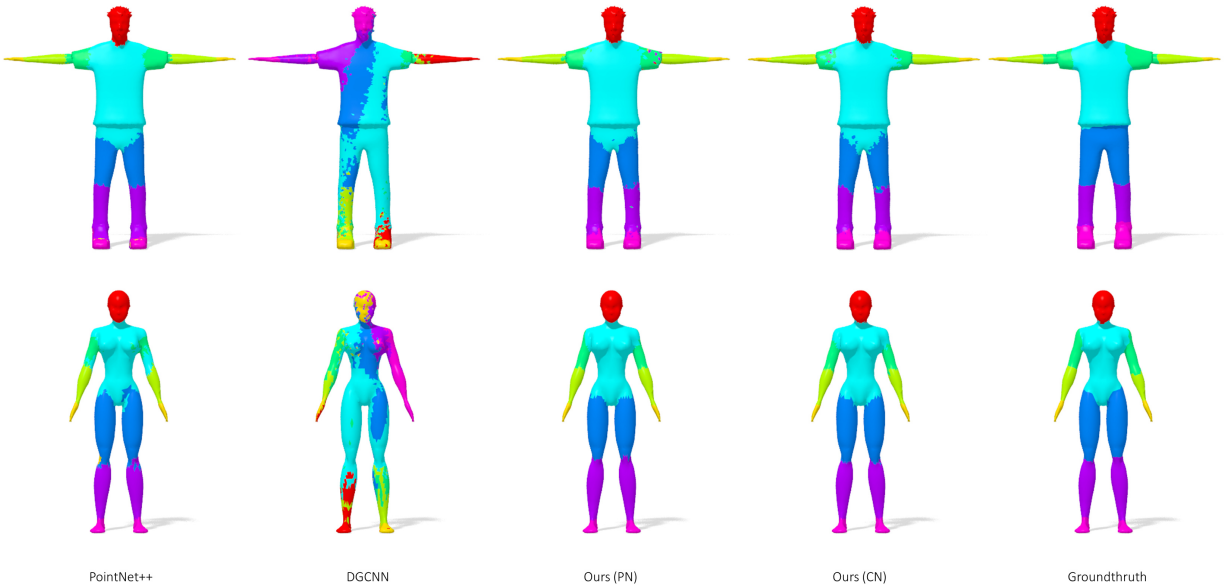

Figure 4. A qualitative evaluation of our method on human body segmentation comparing to other learning-based approaches.

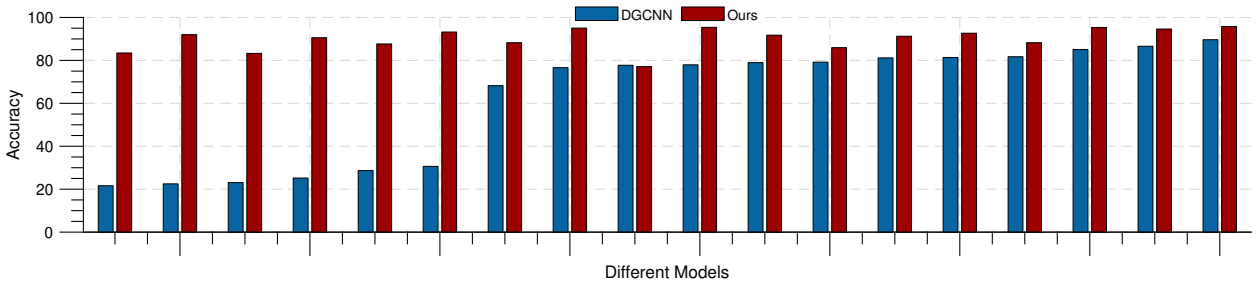

Figure 5. We compare the results of our network with DGCNN on each of the 18 test shapes. To better visualize the effect of the rotation angle we sort the test samples (x-axis) according to the performance of DGCNN. This clearly shows a gap between samples where the input shape was lying down (first 6 examples) and the ones which were upright.

ity, we found that while our network inference time is pretty fast (less than 1.5 seconds), the computation of the geodesic distance is costly. Specifically, for a single mesh in FAUST registration dataset (with 6890 vertices and 13776 faces), the time to compute its geodesic distance is 30.43. In contrast, the time to compute SHOT on such a mesh is only 0.69 seconds. A for the inference time, compared to FMNet, our network average prediction run time for a pair of FAUST models is 1.37 instead of 0.25 seconds. A possible remedy to speed up the pre-processing is to use approximated geodesics, like the one proposed by [14].

FAUST scans. We also evaluate the performance on FAUST scans intra-challenge. To this end, we downsampled the scanned meshes to 15K vertices, and fixed them using MeshFix [3] followed by another downsampling to 7K vertices. We then used the registered models to create ground truth to train our network. Here, we also used a batch size of 8 using 8 GPUs to train the network. Different from the synthetic experiment, here we randomly chose both input shape, only restricting them to come from the same subject. At inference time, we retrieve the high-resolution correspondences for each pairs by upscaling the correspondence using functional maps as done in [41]. Our re-implementation of FMNet achieved an average error of $10.8cm$ while using our learned descriptor to replace SHOT we were able to reduce that error down to $6.7cm$[1].

5.5. Ablation Study

In this section we provide analysis of the design choices made when constructing our descriptor learning network. To this end, we utilize the synthetic FAUST dataset and the task of part segmentation. Since we are working in a simplified setting with only one dataset, we make two modi-

[1]We note that the errors reported in the original FMNet manuscript were lower, however we could not reproduce the results perhaps to a difference in the upscaling procedure

Figure 6. A qualitative evaluation of our matching results on synthetic FAUST. On the left of the vertical border is a textured target shape. To the right are four source shapes from the test set with their texture pulled back from the target according to the recovered correspondences.

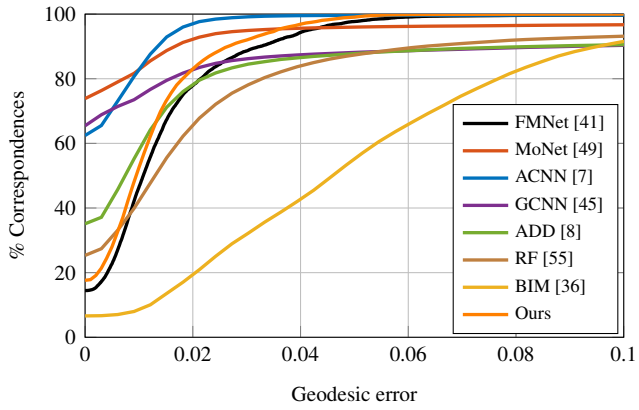

Figure 7. Comparison with learning-based approaches on the FAUST humans dataset. By using our proposed descriptor, we were able to improve upon the results of FMNet with SHOT.

Net.	FUll	CC→ PN	No g_{ij}	No $\mathbf{n}_{ij}$	No LRF	No $\mathbf{f}_{ij}^{l}$	No $\mathbf{v}_{ij}$
Acc.	98.30	98.29	98.24	97.98	95.24	94.40	75.21

Table 3. Ablation study on the design choices of our network ingredients. By removing different components and retraining we evaluate their importance.

fications to the network. First, we reduce the base feature dimensionality (see o_b in Figure 4) from 32 to 4. Second, since the number of vertices is kept fixed we use all vertices instead of subsampling a 2K subset. To evaluate the importance of each ingredient, we retrain and evaluate the network performance with and without it. Results are summarized in Table 3. It can be seen that the accuracy achieved by both CC and PN, is very high. This minor difference may be a result of the relatively small dataset size.

The importance of the rest of the components in descending order is: the coordinates $\mathbf{v}_{ij}$, removing feature propagation from previous layers $\mathbf{f}_{ij}^{l}$, LRF, normals $\mathbf{n}_{ij}$, and geodesic distance g_{ij}. As expected, the coordinate $\mathbf{v}_{ij}$ is playing the biggest role as it holds the most information of the local patch geometry. Another motivation to include normals comes from the results reported in [38], where it was shown that all state-of-the-art networks struggle with accurately estimating the normal of a patch. The results justify the inclusion of each of the components.

Consider the component of PointNet++ that has been used in the segmentation task, it uses an FPS-based geometric distance, a PointNet-based feature extraction, and tri-linear interpolation during the upsampling stage. While our network has the following components: FPS-based on geodesic distance, continuous convolution based feature extraction, and local reference frame. To appreciate the difference in performance between our method and PointNet++ when using the raw data, one should observe the results reported in Table 3 of our method when removing the LRF. The drop in performance is by a non-negligible 3 points.

6. Conclusion

In this work we have studied the problem of learning shape descriptors. A main motivation for this work was the fact that sate-of-the-art learning based techniques were still relying on hand crafted descriptors. Here, we showed that this is mainly due to the usage of the LRF. By baking the computation of an LRF into the design of the network we were able to bridge the gap and outperform manual descriptor- based methods with using raw mesh features: coordinates, normals, and geodesic distances. In addition, we introduced a continuous convolution kernel which allows the filters to dynamically react to the input features. We demonstrated the performance of our proposed method on two important tasks: shape matching and part segmentation. Albeit the usage of a continuous convolution, current method including ours, still rely heavily on the set of sampling points and the sampling method (FPS in our case). This is of course an unwanted behaviour as the result should depend on the underlying surface. In future work we plan to explore this direction.

References

[1] Adobe. Adobe fuse 3d characters., 2016.

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM transactions on graphics (TOG)*, volume 24, pages 408–416. ACM, 2005.

[3] Marco Attene. A lightweight approach to repairing digitized polygon meshes. *The visual computer*, 26(11):1393–1406, 2010.

[4] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*, 2018.

[5] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 1626–1633. IEEE, 2011.

[6] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014.

[7] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 3189–3197, 2016.

[8] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Michael M Bronstein, and Daniel Cremers. Anisotropic diffusion descriptors. In *Computer Graphics Forum*, volume 35, pages 431–441. Wiley Online Library, 2016.

[9] Alexander M Bronstein, Michael M Bronstein, Leonidas J Guibas, and Maks Ovsjanikov. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics (TOG)*, 30(1):1, 2011.

[10] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[11] Michael M Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1704–1711. IEEE, 2010.

[12] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[13] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 8958–8966, 2019.

[14] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Transactions on Graphics (TOG)*, 32(5):1–11, 2013.

[15] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[16] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[17] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *European conference on computer vision (ECCV)*, 2018.

[18] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 195–205, 2018.

[19] Haowen Deng, Tolga Birdal, and Slobodan Ilic. 3d local features for direct pairwise registration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[20] Julie Digne, Raphaëlle Chaine, and Sébastien Valette. Self-similarity for accurate compression of point sampled surfaces. In *Computer Graphics Forum*, volume 33, pages 155–164. Wiley Online Library, 2014.

[21] Daniela Giorgi, Silvia Biasotti, and Laura Paraboschi. Shape retrieval contest 2007: Watertight models track. 2007.

[22] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5545–5554, 2019.

[23] Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. Spiralnet++: A fast and highly efficient mesh convolution operator, 2019.

[24] Yulan Guo, Ferdous A Sohel, Mohammed Bennamoun, Jianwei Wan, and Min Lu. Rops: A local feature descriptor for 3d rigid objects based on rotational projection statistics. In *Communications, Signal Processing, and their Applications (ICCSPA), 2013 1st International Conference on*, pages 1–6. IEEE, 2013.

[25] Oshri Halimi, Or Litany, Emanuele Rodol'a, Alex Bronstein, and Ron Kimmel. Self-supervised learning of dense shape correspondence. *arXiv preprint arXiv:1812.02415*, 2018.

[26] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: A network with an edge. *ACM Transactions on Graphics (TOG)*, 38(4):90, 2019.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[29] Chun-Hao Huang, Federico Tombari, and Nassir Navab. Repeatable local coordinate frames for 3d human motion tracking: From rigid to non-rigid. In *2015 International Conference on 3D Vision*, pages 371–379. IEEE, 2015.

[30] Jingwei Huang, Haotian Zhang, Li Yi, Thomas Funkhouser, Matthias Nießner, and Leonidas J Guibas. Texturenet: Consistent local parametrizations for learning from high-resolution signals on meshes. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 4440–4449, 2019.

[31] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[32] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems*, pages 667–675, 2016.

[33] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999.

[34] Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. Learning 3d mesh segmentation and labeling. In *ACM Transactions on Graphics (TOG)*, volume 29, page 102. ACM, 2010.

[35] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning compact geometric features. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[36] Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)*, volume 30, page 79. ACM, 2011.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[38] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9601–9611, 2019.

[39] Yangyan Li, Rui Bu, Mingchao Sun, and Baoquan Chen. Pointcnn. *arXiv preprint arXiv:1801.07791*, 2018.

[40] Isaak Lim, Alexander Dielen, Marcel Campen, and Leif Kobbelt. A simple approach to intrinsic correspondence learning on unstructured 3d meshes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[41] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5659–5667, 2017.

[42] Or Litany, Tal Remez, Tal, and Alex Bronstein. Cloud dictionary: Sparse coding and modeling for point clouds. *SPARS*, 2017.

[43] Roee Litman and Alexander M Bronstein. Learning spectral descriptors for deformable shape correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):171–180, 2013.

[44] H. Maron, M. Galun, N. Aigerman, M. Trope, N. dym, , E. Yumer, V. Kim, and Y. Lipman. Convolutional neural networks on surfaces via seamless toric covers. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017)*, 2017.

[45] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.

[46] Simone Melzi, Riccardo Spezialetti, Federico Tombari, Michael M. Bronstein, Luigi Di Stefano, and Emanuele Rodola. Gframes: Gradient-based local reference frame for 3d shape matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[47] Ajmal Mian, Mohammed Bennamoun, and Robyn Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2-3):348–361, 2010.

[48] Carsten Moenning and Neil A Dodgson. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory, 2003.

[49] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[50] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[51] Alioscia Petrelli and Luigi Di Stefano. On the repeatability of the local reference frame for partial shape matching. In *2011 International Conference on Computer Vision*, pages 2244–2251. IEEE, 2011.

[52] Adrien Poulenard and Maks Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution. *ACM Transactions on Graphics (TOG)*, 37(6):236, 2019.

[53] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[54] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[55] Emanuele Rodolà, Samuel Rota Bulo, Thomas Windheuser, Matthias Vestner, and Daniel 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.

[56] Raif M Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 225–233. Eurographics Association, 2007.

[57] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.

[58] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz. Persistent Point Feature Histograms for 3D Point Clouds. In *Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS-10), Baden-Baden, Germany*, 2008.

[59] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat

diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.

[60] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. *arXiv preprint arXiv:1904.08889*, 2019.

[61] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique shape context for 3d data description. In *Proceedings of the ACM workshop on 3D object retrieval*, pages 57–62. ACM, 2010.

[62] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010.

[63] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2598–2606, 2018.

[64] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. In *ACM Transactions on Graphics (TOG)*, volume 27, page 97. ACM, 2008.

[65] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2589–2597, 2018.

[66] Ruben Wiersma, Elmar Eisemann, and Klaus Hildebrandt. Cnns on surfaces using rotation-equivariant features. *arXiv preprint arXiv:2006.01570*, 2020.

[67] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[68] Yongheng Zhao, Tolga Birdal, Jan Eric Lenssen, Emanuele Menegatti, Leonidas Guibas, and Federico Tombari. Quaternion equivariant capsule networks for 3d point clouds. *arXiv preprint arXiv:1912.12098*, 2019.