Geodesic Convolutional Neural Networks on Riemannian Manifolds

Jonathan Masci[†][*]   Davide Boscaini[†][*]   Michael M. Bronstein[†]   Pierre Vandergheynst[‡]

[†]USI, Lugano, Switzerland   [‡]EPFL, Lausanne, Switzerland

Abstract

Feature descriptors play a crucial role in a wide range of geometry analysis and processing applications, including shape correspondence, retrieval, and segmentation. In this paper, we introduce Geodesic Convolutional Neural Networks (GCNN), a generalization of the convolutional networks (CNN) paradigm to non-Euclidean manifolds. Our construction is based on a local geodesic system of polar coordinates to extract "patches", which are then passed through a cascade of filters and linear and non-linear operators. The coefficients of the filters and linear combination weights are optimization variables that are learned to minimize a task-specific cost function. We use GCNN to learn invariant shape features, allowing to achieve state-of-the-art performance in problems such as shape description, retrieval, and correspondence.

1. Introduction

Feature descriptors are ubiquitous tools in shape analysis. Broadly speaking, a *local* feature descriptor assigns to each point on the shape a vector in some multi-dimensional descriptor space representing the local structure of the shape around that point. A *global* descriptor describes the whole shape. Local feature descriptors are used in higher-level tasks such as establishing correspondence between shapes [35], shape retrieval [8], or segmentation [43]. Global descriptors are often produced by aggregating local descriptors e.g. using the bag-of-features paradigm. Descriptor construction is largely application dependent, and one typically tries to make the descriptor discriminative (capture the structures that are important for a particular application, e.g. telling apart two classes of shapes), robust (invariant to some class of transformations or noise), compact (low dimensional), and computationally-efficient.

Previous work   Early works on shape descriptors such as spin images [19], shape distributions [34], and integral volume descriptors [32] were based on *extrinsic* structures that are invariant under Euclidean transformations. The fol-

lowing generation of shape descriptors used *intrinsic* structures such as geodesic distances [15] that are preserved by isometric deformations. The success of image descriptors such as SIFT [31], HOG [13], MSER [33], and shape context [2] has led to several generalizations thereof to non-Euclidean domains (see e.g. [49, 14, 24], respectively). The works [11, 28] on diffusion and spectral geometry have led to the emergence of intrinsic spectral shape descriptors that are *dense* and isometry-invariant by construction. Notable examples in this family include heat kernel signatures (HKS) [45] and wave kernel signatures (WKS) [1].

Arguing that in many cases it is hard to model invariance but rather easy to create examples of similar and dissimilar shapes, Litman and Bronstein [29] showed that HKS and WKS can be considered as particular parametric families of transfer functions applied to the Laplace-Beltrami operator eigenvalues and proposed to learn an optimal transfer function. Their work follows the recent trends in the image analysis domain, where hand-crafted descriptors are abandoned in favor of learning approaches. The past decade in computer vision research has witnessed the re-emergence of "deep learning" and in particular, convolutional neural network (CNN) techniques [17, 27], allowing to learn task-specific features from examples. CNNs achieve a breakthrough in performance in a wide range of applications such as image classification [26], segmentation [10], detection and localization [38, 42] and annotation [16, 21].

Learning methods have only recently started penetrating into the 3D shape analysis community in problems such as shape correspondence [39, 37], similarity [20], description [29, 47, 12], and retrieval [30]. CNNs have been applied to 3D data in the very recent works [48, 44] using standard (Euclidean) CNN architectures applied to volumetric 2D views shape representations, making them unsuitable for deformable shapes. Intrinsic versions of CNNs that would allows dealing with shape deformations are difficult to formulate due to the lack of shift invariance on Riemannian manifolds; we are aware of two recent works in that direction [9, 5].

Contribution   In this paper, we propose Geodesic CNN (GCNN), an extension of the CNN paradigm to non-Euclidean manifolds based on local geodesic system of coor-

---

[*]equal contribution

dinates that are analogous to 'patches' in images. Compared to previous works on non-Euclidean CNNs [9, 5], our model is generalizable (i.e., it can be trained on one set of shapes and then applied to another one), local, and allows to capture anisotropic structures. We show that HKS [45], WKS [1], optimal spectral descriptors [29], and intrinsic shape context [24] can be obtained as particular configurations of GCNN; therefore, our approach is a generalization of previous popular descriptors. Our experimental results show that our model can be applied to achieve state-of-the-art performance in a wide range of problems, including the construction of shape descriptors, retrieval, and correspondence.

2. Background

We model a 3D shape as a connected smooth compact two-dimensional manifold (surface) X, possibly with a boundary ∂X. Locally around each point x the manifold is homeomorphic to a two-dimensional Euclidean space referred to as the *tangent plane* and denoted by $T_x X$. A *Riemannian metric* is an inner product $\langle \cdot, \cdot \rangle_{T_x X} : T_x X \times T_x X \to \mathbb{R}$ on the tangent space depending smoothly on x.

Laplace-Beltrami operator (LBO) is a positive semidefinite operator $\Delta_X f = -\mathrm{div}(\nabla f)$, generalizing the classical Laplacian to non-Euclidean spaces. The LBO is *intrinsic*, i.e., expressible entirely in terms of the Riemannian metric. As a result, it is invariant to isometric (metric-preserving) deformations of the manifold. On a compact manifold, the LBO admits an eigendecomposition $\Delta_X \phi_k = \lambda_k \phi_k$ with real eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots$. The corresponding eigenfunctions $\phi_1, \phi_2, \ldots$ form an orthonormal basis on $L^2(X)$, which is a generalization of the Fourier basis to non-Euclidean domains.

Heat diffusion on manifolds is governed by the *diffusion equation*,

$$\left(\Delta_X + \frac{\partial}{\partial t} \right) u(x, t) = 0; \quad u(x, 0) = u_0(x), \quad (1)$$

where $u(x, t)$ denotes the amount of heat at point x at time t, $u_0(x)$ is the initial heat distribution; if the manifold has a boundary, appropriate boundary conditions must be added. The solution of (1) is expressed in the spectral domain as

$$u(x, t) = \int_X u_0(x') \underbrace{\sum_{k \geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(x')}_{h_t(x, x')} \, dx', \quad (2)$$

where $h_t(x, x')$ is the *heat kernel*. Interpreting the LBO eigenvalues as 'frequencies', the coefficients $e^{-t\lambda}$ play the role of a transfer function corresponding to a low-pass filter sampled at $\{\lambda_k\}_{k \geq 1}$.

Discretization In the discrete setting, the surface X is sampled at N points $x_1, \ldots, x_N$. On these points, we construct a triangular mesh (V, E, F) with vertices $V = \{1, \ldots, N\}$, in which each interior edge $ij \in E$ is shared by exactly two triangular faces ikj and $jhi \in F$, and boundary edges belong to exactly one triangular face. The set of vertices $\{j \in V : ij \in E\}$ directly connected to i is called the *1-ring* of i. A real-valued function $f : X \to \mathbb{R}$ on the surface is sampled on the vertices of the mesh and can be identified with an N-dimensional vector $\mathbf{f} = (f(x_1), \ldots, f(x_N))^\top$. The discrete version of the LBO is given as an $N \times N$ matrix $\mathbf{L} = \mathbf{A}^{-1} \mathbf{W}$, where

$$w_{ij} = \begin{cases} (\cot \alpha_{ij} + \cot \beta_{ij})/2 & ij \in E; \\ -\sum_{k \neq i} w_{ik} & i = j; \\ 0 & \text{else}; \end{cases} \quad (3)$$

α_{ij}, β_{ij} denote the angles $\angle ikj, \angle jhi$ of the triangles sharing the edge ij, and $\mathbf{A} = \mathrm{diag}(a_1, \ldots, a_N)$ with $a_i = \frac{1}{3} \sum_{jk:ijk \in F} A_{ijk}$ being the local area element at vertex i and A_{ijk} denoting the area of triangle ijk [36].

The first $K \leq N$ eigenfunctions and eigenvalues of the LBO are computed by performing the generalized eigendecomposition $\mathbf{W\Phi} = \mathbf{A\Phi\Lambda}$, where $\mathbf{\Phi} = (\phi_1, \ldots, \phi_K)$ is an $N \times K$ matrix containing as columns the discretized eigenfunctions and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_K)$ is the diagonal matrix of the corresponding eigenvalues.

3. Spectral descriptors

Many popular spectral shape descriptors are constructed taking the diagonal values of heat-like operators. A generic descriptor of this kind has the form

$$\mathbf{f}(x) = \sum_{k \geq 1} \boldsymbol{\tau}(\lambda_k) \phi_k^2(x) \approx \sum_{k=1}^{K} \boldsymbol{\tau}(\lambda_k) \phi_k^2(x) \quad (4)$$

where $\boldsymbol{\tau}(\lambda) = (\tau_1(\lambda), \ldots, \tau_Q(\lambda))^\top$ is a bank of transfer functions acting on LBO eigenvalues, and Q is the descriptor dimensionality. Such descriptors are dense (computed at every point x), intrinsic by construction, and typically can be efficiently computed using a small number K of LBO eigenfunctions and eigenvalues.

Heat kernel signature (HKS) [45] is a particular setting of (4) using parametric low-pass filters of the form $\tau_t(\lambda) = e^{-t\lambda}$, which allows to interpret them as diagonal values of the heat kernel taken at some times $t_1, \ldots, t_Q$. The physical interpretation of the HKS is *autodiffusivity*, i.e., the amount of heat remaining at point x after time t, which is equal (up to constant) to the Gaussian curvature for small t. A notable drawback of HKS stemming from the use of low-pass filters is its poor spatial localization.

Wave kernel signature (HKS) [1] arises from the model of a quantum particle on the manifold possessing some initial energy distribution, and boils down to a particular setting of (4) with band-pass filters of the form $\tau_\nu(\lambda) = \exp\left(\frac{\log \nu - \log \lambda}{2\sigma^2}\right)$, where ν is the initial mean energy of the particle. WKS have better localization, but at the same time tend to produce noisier matches.

Optimal spectral descriptors (OSD) [29] use parametric transfer functions expressed as

$$\tau_q(\lambda) = \sum_{m=1}^{M} a_{qm}\beta_m(\lambda) \qquad (5)$$

in the B-spline basis $\beta_1(\lambda), \ldots, \beta_M(\lambda)$, where a_{qm} ($q = 1, \ldots, Q, m = 1, \ldots, M$) are the parametrization coefficients. Plugging (5) into (4) one can express the qth component of the spectral descriptor as

$$f_q(x) = \sum_{k \geq 1} \tau_q(\lambda_k)\phi_k^2(x) = \sum_{m=1}^{M} a_{qm} \underbrace{\sum_{k \geq 1} \beta_m(\lambda_k)\phi_k^2(x)}_{g_m(x)}, \quad (6)$$

where $\mathbf{g}(x) = (g_1(x), \ldots, g_M(x))^\top$ is a vector-valued function referred to as *geometry vector*, dependent only on the intrinsic geometry of the shape. Thus, (4) is parametrized by the $Q \times M$ matrix $\mathbf{A} = (a_{lm})$ and can be written in matrix form as $\mathbf{f}(x) = \mathbf{A}\mathbf{g}(x)$. The main idea of [29] is to *learn* the optimal parameters $\mathbf{A}$ by minimizing a task-specific loss which reduces to a Mahalanobis-type metric learning.

4. Convolutional neural networks on manifolds

4.1. Geodesic convolution

We introduce a notion of convolution on non-Euclidean domains that follows the 'correlation with template' idea by employing a local system of geodesic polar coordinates constructed at point x, shown in Figure 1, to extract patches on the manifold. The radial coordinate is constructed as ρ-level sets $\{x' : d_X(x, x') = \rho\}$ of the geodesic (shortest path) distance function for $\rho \in [0, \rho_0]$; we call ρ_0 the radius of the geodesic disc. [1] Empirically, we see that choosing a sufficiently small $\rho_0 \approx 1\%$ of the geodesic diameter of the shape produces valid topological discs. The angular coordinate is constructed as a set of geodesics $\Gamma(x, \theta)$ emanating from x in direction θ; such rays are perpendicular to the geodesic distance level sets. Note that the choice of the origin of the angular coordinate is arbitrary. For boundary points, the procedure is very similar, with the only difference that instead of mapping into a disc we map into a half-disc.

---

[1] If the radius ρ_0 of the geodesic ball $B_{\rho_0}(x) = \{x' : d_X(x, x') \leq \rho_0\}$ is sufficiently small w.r.t the local convexity radius of the manifold, then the resulting ball is guaranteed to be a topological disc.
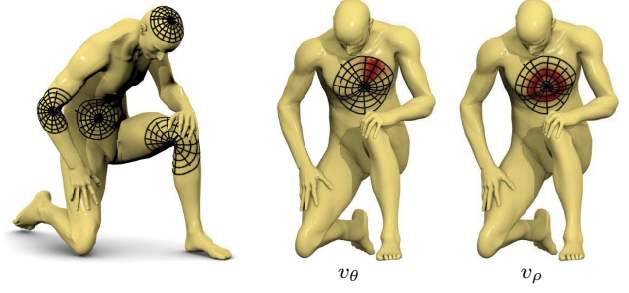

Figure 1: Construction of local geodesic polar coordinates on a manifold. Left: examples of local geodesic patches, center and right: example of angular and radial weights v_θ, v_ρ, respectively (red denotes larger weights).

Let $\Omega(x): B_{\rho_0}(x) \to [0, \rho_0] \times [0, 2\pi)$ denote the bijective map from the manifold into the local geodesic polar coordinates (ρ, θ) around x, and let $(D(x)f)(\rho, \theta) = (f \circ \Omega^{-1}(x))(\rho, \theta)$ be the *patch operator* interpolating f in the local coordinates. We can regard $D(x)f$ as a 'patch' on the manifold and use it to define what we term the *geodesic convolution* (GC),

$$(f \star a)(x) = \sum_{\theta, r} a(\theta + \Delta\theta, r)(D(x)f)(r, \theta), \quad (7)$$

where $a(\theta, r)$ is a filter applied on the patch. Due to angular coordinate ambiguity, the filter can be rotated by arbitrary angle $\Delta\theta$.

Patch operator Kokkinos et al. [24] construct the patch operator as

$$(D(x)f)(\rho, \theta) = \int_X v_{\rho, \theta}(x, x')f(x')dx', \quad (8)$$

$$v_{\rho, \theta}(x, x') = \frac{v_\rho(x, x')v_\theta(x, x')}{\int_X v_\rho(x, x')v_\theta(x, x')dx'}. \quad (9)$$

The radial interpolation weight is a Gaussian $v_\rho(x, x') \propto e^{-(d_X(x,x')-\rho)^2/\sigma_\rho^2}$ of the geodesic distance from x, centered around ρ (see Figure 1, right). The angular weight is a Gaussian $v_\theta(x, x') \propto e^{-d_X^2(\Gamma(x,\theta),x')/\sigma_\theta^2}$ of the point-to-set distance $d_X(\Gamma(x, \theta), x') = \min_{x'' \in \Gamma(x,\theta)} d_X(x'', x')$ to the geodesic $\Gamma(x, \theta)$ (see Figure 1, center).

Discrete patch operator On triangular meshes, a discrete local system of geodesic polar coordinates has N_θ angular and N_ρ radial bins. Starting with a vertex i, we first partition the 1-ring of i by N_θ rays into equi-angular bins, aligning the first ray with one of the edges (Figure 2). Next, we propagate the rays into adjacent triangles using an unfolding procedure resembling one used in [23], producing poly-lines that form the angular bins (see Figure 2). Radial bins are created as level sets of the geodesic distance function computed using fast marching [23].
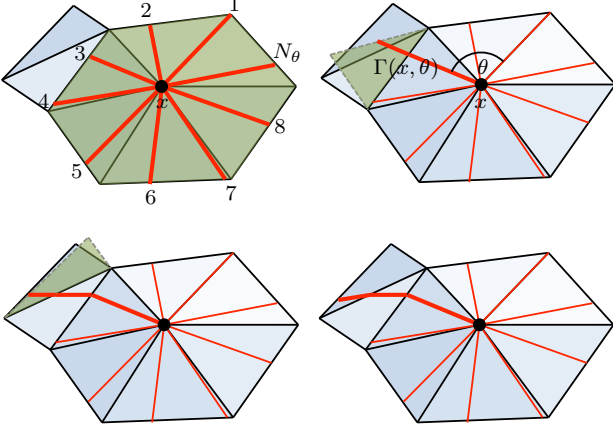
Figure 2: Construction of local geodesic polar coordinates on a triangular mesh. Shown clock-wise: division of 1-ring of vertex x_i into N_θ equi-angular bins; propagation of a ray (bold line) by unfolding the respective triangles (marked in green).

We represent the discrete patch operator as an $N_\theta N_\rho N \times N$ matrix applied to a function defined on the mesh vertices and producing the patches at each vertex. The matrix is very sparse since the values of the function at a few nearby vertices only contribute to each local geodesic polar bin.

4.2. Geodesic Convolutional Neural Networks

Using the notion of geodesic convolution, we are now ready to extend CNNs to manifolds. GCNN consists of several layers that are applied subsequently, i.e. the output of the previous layer is used as the input into the subsequent one (see Figure 3). We distinguish between the following types of layers:

Linear (LIN) layer typically follows the input layer and precedes the output layer to adjust the input and output dimensions by means of a linear combination,

$$f_q^{\text{out}}(x) = \xi\left(\sum_{p=1}^{P} w_{qp} f_p^{\text{in}}(x)\right); \quad q = 1, \ldots, Q, \quad (10)$$

optionally followed by a non-linear function such as the ReLU, $\xi(t) = \max\{0, t\}$.

Geodesic convolution (GC) layer replaces the convolutional layer used in classical Euclidean CNNs. Due to the angular coordinate ambiguity, we compute the geodesic convolution result for *all N_θ rotations* of the filters,

$$f_{\Delta\theta,q}^{\text{out}}(x) = \sum_{p=1}^{P} (f_p \star a_{\Delta\theta,qp})(x), \quad q = 1, \ldots, Q, \quad (11)$$

where $a_{\Delta\theta,qp}(\theta, r) = a_{qp}(\theta + \Delta\theta, r)$ are the coefficients of the pth filter in the qth filter bank rotated by $\Delta\theta = 0, \frac{2\pi}{N_\theta}, \ldots, \frac{2\pi(N_\theta-1)}{N_\theta}$, and the convolution is understood in the sense of (7).

Angular max-pooling (AMP) is a fixed layer used in conjunction with the GC layer, that computes the maximum over the filter rotations,

$$f_p^{\text{out}}(x) = \max_{\Delta\theta} f_{\Delta\theta,p}^{\text{in}}(x), \quad p = 1, \ldots, P = Q, \quad (12)$$

where $f_{\Delta\theta,p}^{\text{in}}$ is the output of the GC layer (11).

Fourier transform magnitude (FTM) layer is another fixed layer that applies the patch operator to each input dimension, followed by Fourier transform w.r.t. the angular coordinate and absolute value,

$$f_p^{\text{out}}(\rho, \omega) = \left| \sum_{\theta} e^{-i\omega\theta} (D(x) f_p^{\text{in}}(x))(\rho, \theta) \right|, \quad (13)$$

$p = 1, \ldots, P = Q$. The Fourier transform translates rotational ambiguity into complex phase ambiguity, which is removed by taking the absolute value [25, 24].

Covariance (COV) layer is used in applications such as retrieval where one needs to aggregate the point-wise descriptors and produce a global shape descriptor [46],

$$\mathbf{f}^{\text{out}} = \int_X (\mathbf{f}^{\text{in}}(x) - \boldsymbol{\mu})(\mathbf{f}^{\text{in}}(x) - \boldsymbol{\mu})^\top dx, \quad (14)$$

where $\mathbf{f}^{\text{in}}(x) = (f_1^{\text{in}}(x), \ldots, f_P^{\text{in}}(x))^\top$ is a P-dimensional input vector, $\boldsymbol{\mu} = \int_X \mathbf{f}^{\text{in}}(x)dx$, and $\mathbf{f}^{\text{out}}$ is a $P \times P$ matrix column-stacked into a P^2-dimensional vector.

5. Comparison to previous approaches

Our approach is perhaps the most natural way of generalizing CNNs to manifolds, where convolutions are performed by sliding a window over the manifold, and local geodesic coordinates are used in place of image 'patches'. Such patches allow capturing local anisotropic structures. Our method is generalizable and unlike spectral approaches does not rely on the approximate invariance of Laplacian eigenfunctions across the shapes.

Spectral descriptors can be obtained as particular configurations of GCNN applied on geometry vectors input. HKS [45] and WKS [1] descriptors are obtained by using a fixed LIN layer configured to produce low- or band-pass filters, respectively. OSD [29] is obtained by using a learnable LIN layer. Intrinsic shape context [24] is obtained by using a fixed LIN layer configured to produce HKS or WKS descriptors, followed by a fixed FTM layer.
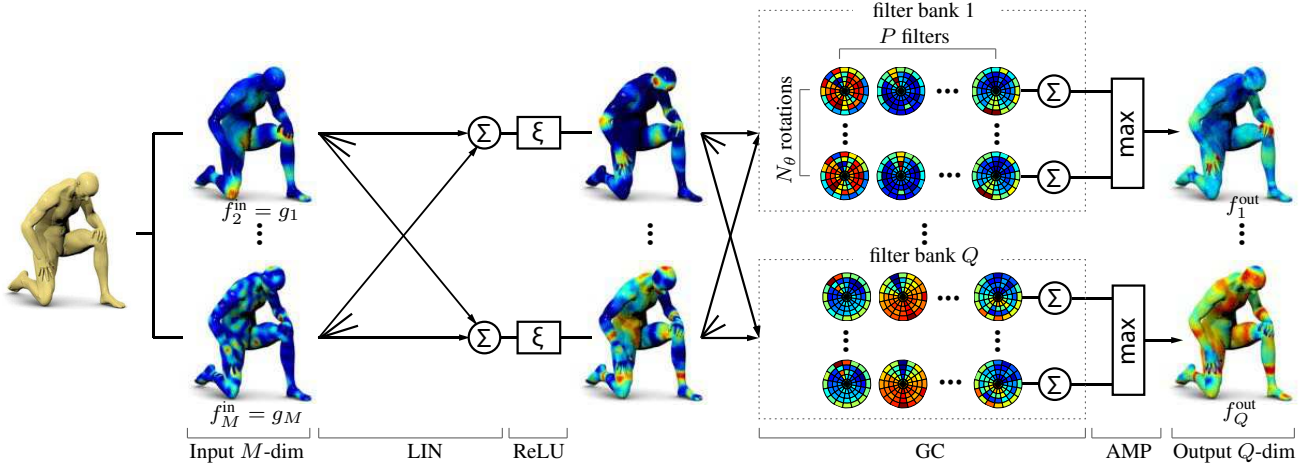
Figure 3: The simple GCNN1 architecture containing one convolutional layer applied to $M = 150$-dimensional geometry vectors (input layer) of a human shape, to produce a $Q = 16$-dimensional feature descriptor (output layer).

Spectral nets [9] are a spectral formulation of CNNs using the notion of generalized (non shift-invariant) convolution that relies on the analogy between the classical Fourier transform and the Laplace-Beltrami eigenbasis, and the fact that the convolution operator is diagonalized by the Fourier transform. The main drawback of this approach is that while it allows to extend CNNs to a non-Euclidean domain (in particular, the authors considered a graph), it does not generalize across *different* domains; the convolution coefficients are expressed in a domain-dependent basis. Another drawback of spectral nets is that they do not use locality.

Localized spectral nets [5] are an extension of [9] using the Windowed Fourier transform (WFT) [40] on manifolds. Due to localization, this method has better generalization abilities, however, it might have problems in the case of strongly non-isometric deformations due to the variability of the Laplacian eigenfunctions. Furthermore, while WFT allows capturing local structures, it is *isotropic*, i.e., insensitive to orientations.

6. Applications

GCNN model can be thought of as a non-linear hierarchical parametric function $\psi_{\boldsymbol{\Theta}}(\mathbf{F})$, where $\mathbf{F} = (\mathbf{f}(x_1), \ldots, \mathbf{f}(x_N))$ is a $P \times N$ matrix of input features (such as HKS, WKS, geometry vectors, or anything else) at all the points of the mesh, and $\boldsymbol{\Theta}$ denotes the parameters of all the layers. Depending on the application in hand, these parameters are learned by minimizing some loss function. We describe three examples of such task-specific losses.

Invariant descriptors Applying the GCNN point-wise on some input feature vector $\mathbf{f}(x)$, the output $\psi_{\boldsymbol{\Theta}}(\mathbf{f}(x))$ can be regarded as a dense local descriptor at point x. Our goal is to make the output of the network as similar as possible at corresponding points (*positives*) across a collection of shapes, and as dissimilar as possible at non-corresponding points (*negatives*). For this purpose, we use a *siamese network* configuration [6, 18, 41], composed of two identical copies of the same GCNN model sharing the same parameterization and fed by pairs of knowingly similar or dissimilar samples, and minimize the *siamese loss*

$$
\begin{aligned}
\ell(\boldsymbol{\Theta}) = & (1 - \gamma) \sum_{i=1}^{|\mathcal{T}_+|} \|\psi_{\boldsymbol{\Theta}}(\mathbf{f}_i) - \psi_{\boldsymbol{\Theta}}(\mathbf{f}_i^+)\|^2 \quad (15) \\
& + \gamma \sum_{i=1}^{|\mathcal{T}_-|} (\mu - \|\psi_{\boldsymbol{\Theta}}(\mathbf{f}_i) - \psi_{\boldsymbol{\Theta}}(\mathbf{f}_i^-)\|)_+^2,
\end{aligned}
$$

where $\gamma \in [0, 1]$ is a parameter trading off between the positive and negative losses, μ is a margin, $(t)_+ = \max\{0, t\}$, and $\mathcal{T}_\pm = \{(\mathbf{f}_i, \mathbf{f}_i^\pm)\}$ denotes the sets of positive and negative pairs, respectively.

Shape correspondence Finding the correspondence in a collection of shapes can be posed as a labelling problem, where one tries to label each vertex of a given *query* shape X with the index of a corresponding point on some *reference* shape Y [37]. Let $y_1, \ldots, y_{N'}$ be the vertices of Y, and let y_{j_i} denote the vertex corresponding to x_i for $i = 1, \ldots, N$. GCNN applied point-wise on X is used to produce an N'-dimensional vector encoding the probability distribution on Y, which acts as a 'soft correspondence'. The *multinomial regression loss*

$$
\ell(\boldsymbol{\Theta}) = -\sum_{i=1}^{|\mathcal{T}|} \mathbf{e}_{j_i} \log \psi_{\boldsymbol{\Theta}}(\mathbf{f}_i) \quad (16)
$$

is minimized on a training set of known correspondence $\mathcal{T} = \{\mathbf{f}(x_i), j_i\}$ to achieve the optimal correspondence (here e_i is a unit vector with a one at index i).

Shape retrieval In the shape retrieval application, we are interested in producing a global shape descriptor that discriminates between shape classes (note that in a sense this is the converse of invariant descriptors for correspondence, which we wanted to be oblivious to different classes). In order to aggregate the local features we use the COV layer in GCNN and regard $\psi_\Theta(\mathbf{F})$ as a global shape descriptor. Training is done by minimizing the siamese loss, where positives and negatives are shapes from same and different classes, respectively.

7. Results

We used the FAUST [4] dataset containing scanned human shapes in different poses and the TOSCA [7] dataset containing synthetic models of humans in a variety of near-isometric deformations. The meshes in TOSCA were resampled to 10K vertices; FAUST shapes contained 6.8K points. All shapes were scaled to unit geodesic diameter. GCNN was implemented in Theano [3]. Geodesic patches were generated using the code and settings of [24] with $\rho_0 = 1\%$ geodesic diameter. Training was performed using the Adadelta stochastic optimization algorithm [50] for a maximum of 2.5K updates. Typical training times on FAUST shapes were approximately 30 and 50 minutes for one- and two-layer models (GCNN1 and GCNN2, respectively). The application of a trained GCNN model to compute feature descriptors was very efficient: 75K and 45K vertices/sec for the GCNN1 and GCNN2 models, respectively. Training and testing was done on disjoint sets. As the input to GCNN, we used $M = 150$-dimensional geometry vectors computed according to (5)–(6) using B-spline bases. Laplace-Beltrami operators were discretized using the cotangent formula (3); $K = 300$ eigenfunctions were computed using MATLAB `eigs` function.

7.1. Intrinsic shape descriptors

We first used GCNN to produce dense intrinsic pose- and subject-invariant descriptors for human shapes, following nearly-verbatim the experimental setup of [29]. For reference, we compared GCNN to HKS [45], WKS [1], and OSD [29] using the code and settings provided by the respective authors. All the descriptors were $Q = 16$-dimensional as in [29]. We used two configurations: GCNN1 (150-dim input, LIN16+ReLU, GC16+AMP shown in Figure 3), and GCNN2 (same as GCNN1 with additional ReLU, FTM, LIN16 layers); Training of GCNN was done using the loss (15) with positive and negative sets of vertex pairs generated on the fly. On the FAUST dataset, we used subjects 1–7 for training, subject 8 for validation, and subject 9–10 for testing. On TOSCA, we test on all the deformations of the Victoria shape.

Figure 4 depicts the Euclidean distance in the descriptor space between the descriptor at a selected point and the rest of the points on the same shape as well as its transformations. GCNN descriptors manifest both good localization (better than HKS) and are more discriminative (less spurious minima than WKS and OSD), as well as robustness to different kinds of noise, including isometric and non-isometric deformations, geometric and topological noise, different sampling, and missing parts.

Quantitative descriptor evaluation was done using three criteria: *cumulative match characteristic* (CMC), *receiver operator characteristic* (ROC), and the *Princeton protocol* [22]. The CMC evaluates the probability of a correct correspondence among the k nearest neighbors in the descriptor space. The ROC measures the percentage of positives and negatives pairs falling below various thresholds of their distance in the descriptor space (*true positive* and *negative rates*, respectively). The Princeton protocol counting the percentage of nearest-neighbor matches that are at most r-geodesically distant from the groundtruth correspondence. Figure 5 (first row) shows the performance of different descriptors on the FAUST dataset. We observe that GCNN descriptors significantly outperform other descriptors, and that the more complex model (GCNN2) further boosts performance. In order to test the generalization capability of the learned descriptors, we applied OSD and GCNN learned on the FAUST dataset to TOSCA shapes (Figure 5, second row). We see that the learned model transfers well to a new dataset.

7.2. Shape correspondence

To show the application of GCNN for computing intrinsic correspondence, we reproduced the experiment of Rodolà et al. [37] on the FAUST dataset, replacing their random forest with a GCNN architecture GCNN3 containing three convolutional layers (input: 150-dimensional geometry vectors, LIN16+ReLU, GC32+AMP+ReLU, GC64+AMP+ReLU, GC128+AMP+ReLU, LIN256, LIN6890). Zeroth FAUST shape containing $N' = 6890$ vertices was used as reference; for each point on the query shape, the output of GCNN representing the soft correspondence as an 6890-dimensional vector was converted into a point correspondence by taking the maximum. Training was done by minimizing the loss (16); training and test sets were as in the previous experiment. Figure 6 shows the performance of our method evaluated using the Princeton benchmark, and Figure 7 shows correspondence examples where colors are transferred using raw point-wise correspondence in input to the functional maps algorithm. GCNN shows significantly better performance than previous methods [22, 35, 37].

7.3. Shape retrieval

In our final experiment, we performed pose-invariant shape retrieval on the FAUST dataset. This is a hard fine-grained classification problem since some of the human

Heat kernel signature (HKS)

Wave kernel signature (WKS)

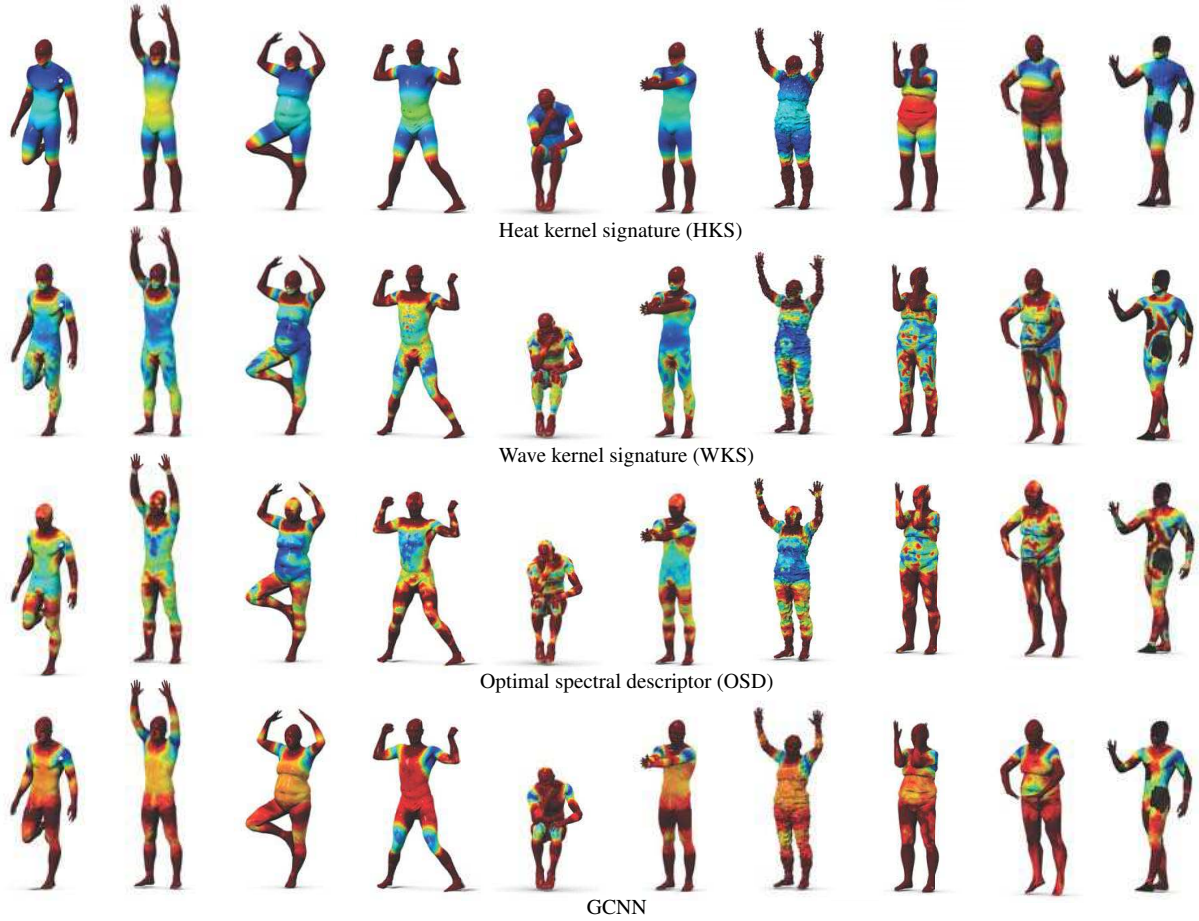Optimal spectral descriptor (OSD)

GCNN

Figure 4: Normalized Euclidean distance between the descriptor at a reference point on the shoulder (white sphere) and the descriptors computed at the rest of the points for different transformations (shown left-to-right: near isometric deformations, non-isometric deformations, topological noise, geometric noise, uniform/non-uniform subsampling, missing parts). Cold and hot colors represent small and large distances, respectively; distances are saturated at the median value. Ideal descriptors would produce a distance map with a sharp minimum at the corresponding point and no spurious local minima at other locations.

subjects look nearly identical. We used a GCNN architecture with one convolutional layer (input: 16-dimensional HKS descriptors, LIN8, GC8+AMP, COV), producing a 64-dimensional output used as the global shape descriptor. Training set consisted of five poses per subject (a total of 50 shapes); testing was performed on the 50 remaining shapes in a leave-one-out fashion. Evaluation was done in terms of *precision* (percentage of retrieved shapes matching the query class) and *recall* (percentage of shapes from the query class that is retrieved). Figure 8 shows the PR curve. For comparison, we show the performance of other descriptors (HKS, WKS, and OSD) aggregated into a global covariance shape descriptor. GCNN outperforms significantly all other methods.

8. Conclusions

We presented GCNN, a generalization of CNNs allowing to learn hierarchical task-specific features on non-Euclidean manifolds for applications such as shape correspondence or retrieval. Our model is very generic and flexible, and can be made arbitrarily complex by stacking multiple layers. Applying GCNN on other shape representations such as point clouds could be achieved by modifying the local geodesic charting procedure. Though in this paper we used intrinsic spectral properties of the shape as the the input to the network, GCNN can be applied on any function defined on the manifold, and it would be particularly natural to use it to construct descriptors of textured surfaces.

References

[1] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. ICCV*, 2011. 1, 2, 3, 4, 6

[2] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Proc. NIPS*, 2000. 1

[3] J. Bergstra et al. Theano: a CPU and GPU math expression compiler. In *Proc. SciPy*, June 2010. 6
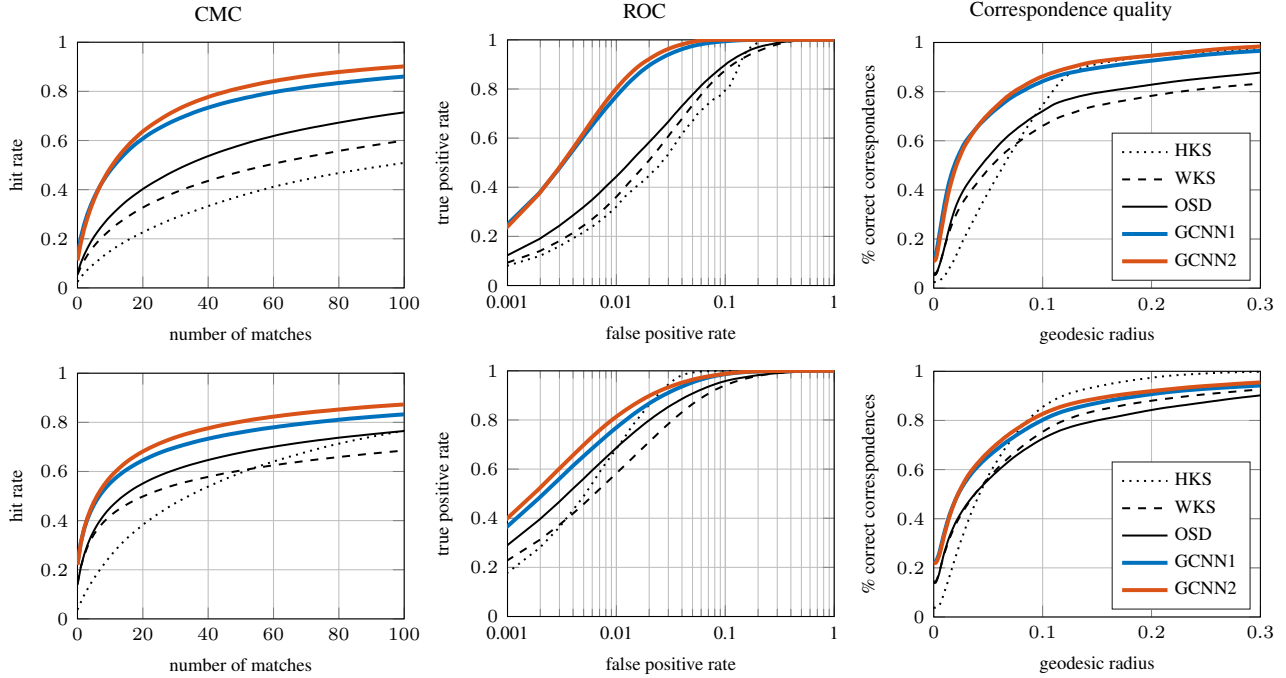
Figure 5: Performance of different descriptors measured using the CMC (left), ROC (center) and Princeton protocol for nearest-neighbor correspondence (right); higher curves correspond to better performance. First row show results for GCNN trained and tested on disjoint sets of the FAUST dataset. Second row shows results for a transfer learning experiment where the net has been trained on FAUST and applied to the TOSCA test set. GCNN (red and blue curves) significantly outperforms other standard descriptors.
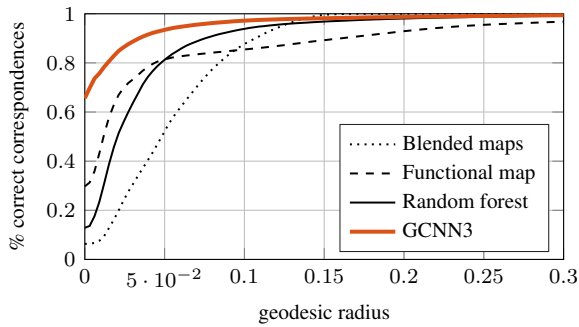
Figure 6: Performance of shape correspondence on the FAUST dataset evaluated using the Princeton benchmark. Higher curve corresponds to better performance.
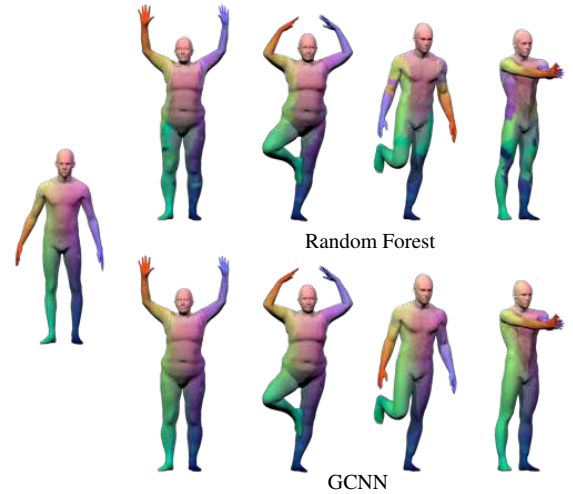

Figure 7: Example of correspondence obtained with GCNN (bottom) and random forest (top). Similar colors encode corresponding points.

[4] F. Bogo et al. FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR*, 2014. 6

[5] D. Boscaini et al. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *CGF*, 34(5):13–23, 2015. 1, 2, 5

[6] J. Bromley et al. Signature verification using a "Siamese" time delay neural network. In *Proc. NIPS*. 1994. 5

[7] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer, 2008. 6

[8] A. M. Bronstein et al. Shape Google: Geometric words and expressions for invariant shape retrieval. *TOG*, 30(1):1–20, 2011. 1

[9] J. Bruna et al. Spectral networks and locally connected networks on graphs. In *Proc. ICLR*, 2014. 1, 2, 5

[10] D. C. Ciresan et al. Deep neural networks segment neuronal membranes in electron microscopy images. In *Proc. NIPS*, 2012. 1
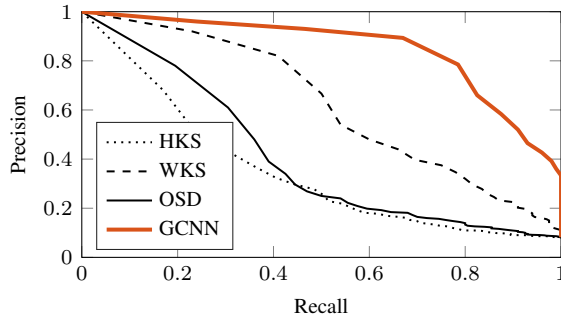
Figure 8: Performance (in terms of Precision-Recall) of shape retrieval on the FAUST dataset using different descriptors. Higher curve corresponds to better performance.

[11] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Comp. Harmonic Analysis*, 21(1):5–30, 2006. 1

[12] É. Corman, M. Ovsjanikov, and A. Chambolle. Supervised descriptor learning for non-rigid shape matching, 2014. 1

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005. 1

[14] J. Digne et al. The level set tree on meshes. In *Proc. 3DPVT*, 2010. 1

[15] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *PAMI*, 25(10):1285–1295, 2003. 1

[16] H. Fang et al. From captions to visual concepts and back. *arXiv:1411.4952*, 2014. 1

[17] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980. 1

[18] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. CVPR*, 2006. 5

[19] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *PAMI*, 21(5):433–449, 1999. 1

[20] A. Kanezaki et al. Learning similarities for rigid and non-rigid object detection. In *Proc. BMVC*, 2014. 1

[21] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv:1412.2306*, 2014. 1

[22] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. *TOG*, 30(4):1–12, 2011. 6

[23] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. *PNAS*, 95(15):8431–8435, 1998. 3

[24] I. Kokkinos et al. Intrinsic shape context descriptors for deformable shapes. In *Proc. CVPR*, 2012. 1, 2, 3, 4, 6

[25] I. Kokkinos and A. Yuille. Scale invariance without scale selection. In *Proc. CVPR*, 2008. 4

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012. 1

[27] Y. LeCun et al. Backpropagation applied to handwritten zip code recognition. *Neural Comp.*, 1(4):541–551, 1989. 1

[28] B. Lévy. Laplace-Beltrami eigenfunctions towards an algorithm that "understands" geometry. In *Proc. SMI*, 2006. 1

[29] R. Litman and A. M. Bronstein. Learning spectral descriptors for deformable shape correspondence. *PAMI*, 36(1):170–180, 2014. 1, 2, 3, 4, 6

[30] R. Litman et al. Supervised learning of bag-of-features shape descriptors using sparse coding. *CGF*, 33(5):127–136, 2014. 1

[31] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1

[32] S. Manay et al. Integral invariants for shape matching. *PAMI*, 28(10):1602–1618, 2006. 1

[33] J. Matas et al. Robust wide-baseline stereo from maximally stable extremal regions. *IVC*, 22(10):761–767, 2004. 1

[34] R. Osada et al. Shape distributions. *TOG*, 21(4):807–832, 2002. 1

[35] M. Ovsjanikov et al. Functional maps: a flexible representation of maps between shapes. *TOG*, 31(4):1–11, 2012. 1, 6

[36] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993. 2

[37] E. Rodolà et al. Dense non-rigid shape correspondence using random forests. In *Proc. CVPR*, 2014. 1, 5, 6

[38] P. Sermanet et al. OverFeat: Integrated recognition, localization and detection using convolutional networks. In *Proc. ICLR*, 2014. 1

[39] J. Shotton et al. Real-time human pose recognition in parts from single depth images. *Comm. ACM*, 56(1):116–124, 2013. 1

[40] D. I. Shuman, B. Ricaud, and P. Vandergheynst. Vertex-frequency analysis on graphs. *arXiv:1307.5708*, 2013. 5

[41] E. Simo-Serra et al. Fracking deep convolutional image descriptors. *arXiv:1412.6537*, 2014. 5

[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 1

[43] P. Skraba et al. Persistence-based segmentation of deformable shapes. In *Proc. NORDIA*, 2010. 1

[44] H. Su et al. Multi-view convolutional neural networks for 3D shape recognition. In *Proc. ICCV*, 2015. 1

[45] J. Sun, M. Ovsjanikov, and L. J. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *CGF*, 28(5):1383–1392, 2009. 1, 2, 4, 6

[46] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *Proc. ECCV*, 2006. 4

[47] T. Windheuser et al. Optimal intrinsic descriptors for non-rigid shape analysis. In *Proc. BMVC*, 2014. 1

[48] Z. Wu et al. 3D ShapeNets: A deep representation for volumetric shape modeling. In *Proc. CVPR*, 2015. 1

[49] A. Zaharescu et al. Surface feature detection and description with applications to mesh matching. In *Proc. CVPR*, 2009. 1

[50] M. D. Zeiler. ADADELTA: An adaptive learning rate method. *arXiv:1212.5701*, 2012. 6