Eurographics Symposium on Geometry Processing 2015
Mirela Ben-Chen and Ligang Liu
(Guest Editors)

Volume 34 (2015), Number 5

Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks

D. Boscaini[1], J. Masci[1], S. Melzi[2], M. M. Bronstein[1,3], U. Castellani[2] and P. Vandergheynst[4]

[1]Institute of Computational Science, Faculty of Informatics, University of Lugano (USI), Switzerland
[2]Department of Informatics, University of Verona, Italy
[3]Perceptual Computing, Intel, Israel
[4]Department of Electrical Engineering, EPFL, Lausanne, Switzerland

Abstract

In this paper, we propose a generalization of convolutional neural networks (CNN) to non-Euclidean domains for the analysis of deformable shapes. Our construction is based on localized frequency analysis (a generalization of the windowed Fourier transform to manifolds) that is used to extract the local behavior of some dense intrinsic descriptor, roughly acting as an analogy to patches in images. The resulting local frequency representations are then passed through a bank of filters whose coefficient are determined by a learning procedure minimizing a task-specific cost. Our approach generalizes several previous methods such as HKS, WKS, spectral CNN, and GPS embeddings. Experimental results show that the proposed approach allows learning class-specific shape descriptors significantly outperforming recent state-of-the-art methods on standard benchmarks.

Categories and Subject Descriptors (according to ACM CCS): Computational Geometry and Object Modeling [I.3.5]: — Feature Measurement [I.4.7]: — Learning [I.2.6]: —

1. Introduction

Shape descriptors are commonly used in a wide range of geometry processing applications, such as correspondence, segmentation, labeling, and retrieval. A shape descriptor is a method for describing the local behavior of the surface around some point, which is captured by a multi-dimensional vector. The set of descriptors at all the points of the surface can be thought of as a vector field thereon. Typically, one wishes a descriptor that is *discriminative* (highlighting distinctive attributes), *robust* (invariant with respect to noise and deformations) *compact* (using a small number of dimensions), and *computationally-efficient*.

Previous work There is a plethora of literature on geometric shape descriptors, and we refer the reader to a recent survey for a comprehensive overview [L*13]. Descriptors like spin images [JH99], shape distributions [OFCD02], integral volume descriptors [MCH*06], and multi-scale features [PKG03] are based on extrinsic structure of the shape and therefore invariant under Euclidean transformations, but not under deformations. One of the first works to deal with deformations was Elad and Kimmel [EK03] employing multi-dimensional scaling to represent the geodesic distance metric in the Euclidean space. Other descriptors based on geodesic were proposed in [HK03], while [BCG08] used conformal factors.

Spectral descriptor try to exploit the geometry arising from the eigenfunctions and eigenvalues of the Laplace-Beltrami operator of the surface [BBG94, CL06, Lév06]; popular methods include shapeDNA [RWP06], global point signature (GPS) [Rus07], heat kernel signatures (HKS) [SOG09, GBAL09], wave kernel signatures (WKS) [ASC11], and heat kernel maps [OMMG10].

Another class of approaches try to bring successful models like SIFT [Low04] or shape context [BMP00], from images to surfaces [SB11, KBLB12]. Following the recent image processing trend of learning invariant structure rather than trying to hand-craft them, several learning frameworks have been proposed in the geometry processing community as well, for applications such as correspondence [RRBW*14], retrieval [LBBC14], labelling and segmentation [KHS10, HSG13]. Several methods for learning descriptors have appeared very

recently [LB14, COC14, MBBV15]. The main advantage of learning methods is, instead of trying to model the noise or shape variability axiomatically, one learns them from examples. In particular, learning methods allow creating *class-specific* descriptors that address fine-grained differences between shapes in the class [LBBC14]. Human shapes are an important and challenging class example, as they exhibit rich geometric variability.

A particularly successful learning model recently regaining popularity in the computer vision and pattern recognition communities are convolutional neural networks (CNN) [LBD*89], whose main strength is the ability to learn hierarchical abstractions from large collections of data with little prior knowledge. CNNs with three-dimensional convolutions have been applied for classification and retrieval of volumetric rigid shapes in a very recent work of [WSK*15]. In order to apply CNNs for the analysis of deformable shapes, one has to define convolution in an intrinsic manner on a Riemannian manifold, a rather difficult task due to the lack of shift invariance on non-Euclidean domains. We are aware of two recent attempts to extend the CNN framework to non-Euclidean domains. Bruna et al. [BZSL14] proposed a spectral formulation of CNNs on graphs. Masci et al. [MBBV15] proposed a generalization of CNNs to triangular meshes using a local geodesic charting technique of [KBLB12] in order to define non-Euclidean 'patches'.

Contribution This paper is a continuation of the previous effort [MBBV15] to generalize the convolutional neural network model to non-Euclidean domains for deformable shape analysis applications, in particular, for constructing class-specific dense intrinsic shape descriptors. The main novelty of our present approach is an alternative generalization of the convolution. Instead of the patch operator used in [MBBV15] that is defined on triangular meshes, we use the vertex-frequency analysis framework of [SRV13]. The core of this construction is a windowed Fourier transform, allowing to capture local context around a point on a surface and represent it in the frequency domain. Combined with a learning framework similar to that of [MBBV15], we are able to learn discriminative and robust descriptors that are specific to a given class of deformable shapes. The significant advantage of the spectral framework is that it is intrinsic by construction and works only with eigenvalues and eigenfunctions of the Laplacian, thus naturally allowing to deal with shapes in any representation, e.g. meshes, point clouds, or graphs, as opposed to the previous construction of [MBBV15] limited to meshes only. We show experimentally that our descriptor can be efficiently computed, is highly discriminative and robust, and compares favorable to previous methods.

2. Background

Manifold We model a 3D shape as a connected smooth compact two-dimensional surface (manifold) X, possibly with a boundary ∂X. Locally around each point x the manifold is homeomorphic to the *tangent plane* T_xX. The *exponential map* $\exp_x\colon T_xX \to X$ maps tangent vectors onto the surface. A *Riemannian metric* is an inner product $\langle\cdot,\cdot\rangle_{T_xX}\colon T_xX \times T_xX \to \mathbb{R}$ on the tangent space depending smoothly on x.

Laplace-Beltrami operator (LBO) Let us denote by $L^2(X)$ the space of square-integrable real functions on X and by $\langle f,g\rangle_{L^2(X)} = \int_X f(x)g(x)dx$ the standard inner product on $L^2(X)$, where dx is the area element induced by the Riemannian metric. Given a smooth function $f \in L^2(X)$, the composition $f\circ\exp_x\colon T_xX \to \mathbb{R}$ is a function on the tangent plane. We define the *Laplace-Beltrami operator* (LBO) as a positive semidefinite operator $\Delta_X\colon L^2(X) \to L^2(X)$ given by

$$\Delta_X f(x) = \Delta(f\circ\exp_x)(0), \tag{1}$$

where Δ is the Euclidean Laplacian operator on the tangent plane. 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 surface.

Spectral analysis on manifolds The LBO of a compact manifold admits an eigendecomposition $\Delta_X\phi_k = \lambda_k\phi_k$ with a countable set of real eigenvalues $\lambda_1 \le \lambda_2 \le \dots$ and the corresponding eigenfunctions $\phi_1,\phi_2,\dots$ form an orthonormal basis on $L^2(X)$. Consequently, a function $f \in L^2(X)$ can be represented as the *Fourier series*

$$f(x) = \sum_{k\ge1} \langle f,\phi_k\rangle_{L^2(X)}\phi_k(x), \tag{2}$$

where the analysis $\hat{f}_k = \langle f,\phi_k\rangle_{L^2(X)}$ can be regarded as the forward Fourier transform and the synthesis $\sum_{k\ge1}\hat{f}_k\phi_k(x)$ is the inverse one; the eigenvalues $\{\lambda_k\}_{k\ge1}$ play the role of frequencies. The first eigenvalue $\lambda_1 = 0$ corresponds to a constant eigenvector ('DC component'). The Laplacian eigenbasis is a generalization of the classical Fourier basis to non-Euclidean domains, and one can easy verify that $e^{i\omega x}$ are eigenfunctions of the Euclidean Laplacian operator $-\frac{d^2}{dx^2}e^{i\omega x} = \omega^2 e^{i\omega x}$.

The *generalized convolution* of f and g on the manifold can be defined by analogy to the classical case as the inverse transform of the product of forward transforms,

$$\begin{aligned}(f\star g)(x) &= \sum_{k\ge1}\langle f,\phi_k\rangle_{L^2(X)}\langle g,\phi_k\rangle_{L^2(X)}\phi_k(x)\\ &= \sum_{k\ge1}\hat{f}_k\hat{g}_k\phi_k(x),\end{aligned} \tag{3}$$

and is in general *non-shift-invariant*.

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

$$(\Delta_X + \partial_t)\,u(x,t) = 0, \quad u(x,0) = u_0(x), \tag{4}$$

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 surface has a

boundary, appropriate boundary conditions must be added). The solution of (4) is obtained by applying the *heat operator* $H^t = e^{-t\Delta_X}$ to the initial condition,

$$u(x,t) = H^t u_0(x) = \int_X u_0(x') h_t(x,x') dx', \qquad (5)$$

where $h_t(x,x')$ is called the *heat kernel*. Since H^t has the same eigenfunctions as Δ_X with the eigenvalues $\{e^{-t\lambda_k}\}_{k\geq 1}$, the solution of (4) can be expressed a generalized convolution (3),

$$\begin{aligned} u(x,t) &= H^t u_0(x) = \sum_{k\geq 1} \langle u_0, \phi_k \rangle_{L^2(X)} e^{-t\lambda_k} \phi_k(x) \\ &= \int_X u_0(x') \underbrace{\sum_{k\geq 1} e^{-t\lambda_k} \phi_k(x) \phi_k(x')}_{h_t(x,x')} dx', \end{aligned} \qquad (6)$$

where the coefficients $e^{-t\lambda}$ play the role of a transfer function corresponding to a low-pass filter sampled at frequencies $\{\lambda_k\}_{k\geq 1}$.

3. Spectral descriptors

In this section, we provide a concise overview of several spectral shape descriptors that will be important for our further discussion. This overview is by no means exhaustive, and we refer the reader to the cited related works and references therein for a more complete picture.

Global Point Signature (GPS) Rustamov [Rus07] proposed the *global point signature (GPS)* embedding, a dense shape descriptor constructed using scaled LBO eigenfunctions,

$$\mathbf{f}(x) = (\lambda_1^{-1/2}\phi_1(x), \dots, \lambda_Q^{-1/2}\phi_Q(x))^\top, \qquad (7)$$

thus associating each point x with a Q-dimensional descriptor (see [BBG94,CL06] for earlier constructions in the theoretical math community).

Due to an inherent ambiguity in the definition of the LBO eigenbasis, GPS descriptors cannot be matched in a simple-minded manner. First, an eigenfunction is defined up to sign, $\Delta_X(\pm\phi_i) = \lambda_i(\pm\phi_i)$. Second, if an eigenvalue with non-trivial multiplicity is present in the spectrum of Δ_X, any rotation in the corresponding subspace produces valid eigenfunctions. Third, noise and non-isometric deformations may alter the eigenvalues and eigenfunctions of the LBO. Trying to cope with these ambiguities, several techniques have been proposed trying to match GPS descriptors (see, e.g. [MHK*08]).

Heat/Wave Kernel Signature (HKS/WKS) Several popular spectral shape descriptor take a generic form of the diagonal of a parametric kernel diagonalized by the LBO eigenbasis. Notable examples include the *heat kernel signature (HKS)* [SOG09, GBAL09] and the *wave kernel signature (WKS)* [ASC11]. More specifically, such methods construct at each point a descriptor

$$\mathbf{f}(x) = \sum_{k\geq 1} \boldsymbol{\tau}(\lambda_k) \phi_k^2(x) \qquad (8)$$

expressed by a bank of transfer functions $\boldsymbol{\tau}(\lambda) = (\tau_1(\lambda), \dots, \tau_Q(\lambda))^\top$. Such descriptors have several appealing properties making their use popular in numerous applications. First, they are intrinsic and hence invariant to isometric deformations of the manifold by construction. Second, they are dense. Third, (8) can be efficiently computed using the first few eigenvectors and eigenvalues of the Laplace-Beltrami operator.

HKS uses low-pass transfer functions $\tau_t(\lambda) = e^{-t\lambda}$ for various values of the parameter $t \in \{t_1, \dots, t_Q\}$, giving rise to the *autodiffusivity function* $h_t(x,x)$, whose physical interpretation is the amount of heat remaining at point x after time t. A notable drawback of HKS is poor spatial localization, which is a consequence of the uncertainty principle: good localization in the Fourier domain (large value of t) results in a bad localization in the spatial domain.

WKS uses band-bass transfer functions $\tau_\nu(\lambda) = \exp\left(\frac{\log\nu - \log\lambda}{2\sigma^2}\right)$ for various values of the parameter $\nu \in \{\nu_1, \dots, \nu_Q\}$. The physical interpretation of WKS is the probability to find a quantum particle at point x, given that it has an initial log-normal energy distribution with mean value ν and variance σ. Typically, WKS exhibits oscillatory behavior and has a better localization compared to HKS.

Optimal spectral descriptors (OSD) Litman and Bronstein [LB14] used parametric transfer functions expressed as

$$\tau_q(\lambda) = \sum_{m=1}^{M} a_{qm} \beta_m(\lambda) \qquad (9)$$

in some fixed (e.g. B-spline) basis $\beta_1(\lambda), \dots, \beta_M(\lambda)$, where a_{qm} ($q = 1, \dots, Q, m = 1, \dots, M$) are the parametrization coefficients. Plugging (9) into (8) one can express the qth component of the spectral descriptor as

$$f_q(x) = \sum_{k\geq 1} \tau_q(\lambda_k)\phi_k^2(x) = \sum_{m=1}^{M} a_{qm} \underbrace{\sum_{k\geq 1} \beta_m(\lambda_k)\phi_k^2(x)}_{g_m(x)}, \qquad (10)$$

where $\mathbf{g}(x) = (g_1(x), \dots, g_M(x))^\top$ is a vector-valued function referred to as *geometry vector*, dependent only on the intrinsic geometry of the shape. Thus, (8) 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 [LB14] is to *learn* the optimal parameters $\mathbf{A}$ by minimizing a task-specific loss. Given a training set consisting of a pair of geometry vectors $\mathbf{g}, \mathbf{g}^+$ representing knowingly similar points (*positives*), and $\mathbf{g}, \mathbf{g}^-$ representing knowingly dissimilar points (*negatives*), one tries to find $\mathbf{A}$ such that $\|\mathbf{f} - \mathbf{f}^+\| = \|\mathbf{A}(\mathbf{g} - \mathbf{g}^+)\|$ is as small as possible and $\|\mathbf{f} - \mathbf{f}^-\| = \|\mathbf{A}(\mathbf{g} - \mathbf{g}^-)\|$ is as large as possible. The authors

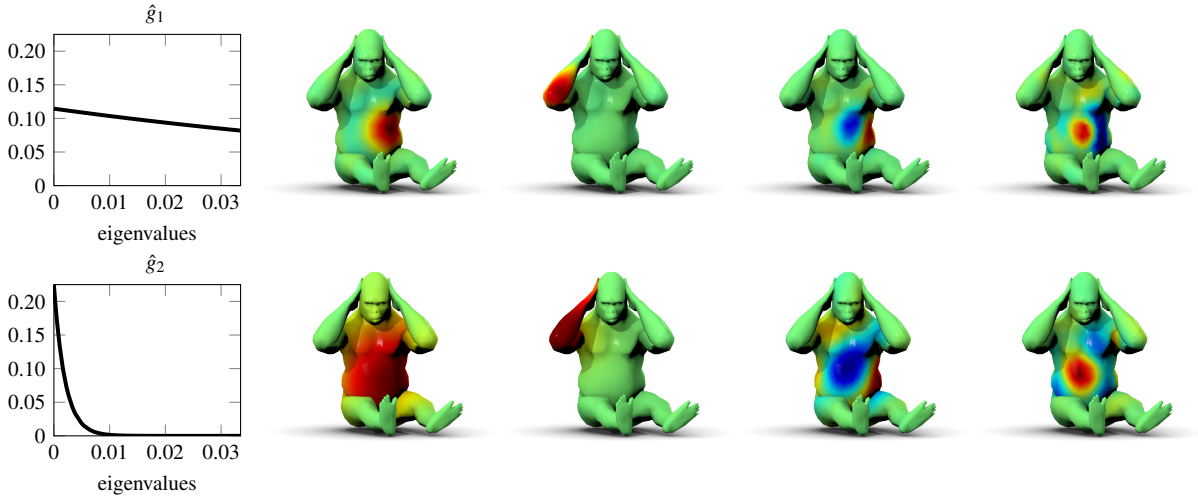

Figure 1: Examples of different WFT atoms $g_{x,k}$ using different windows (top and bottom rows; window Fourier coefficients are shown on the left), shown in different localizations (second and third columns) and modulations (fourth and fifth columns).

show that the problem boils down to a simple Mahalanobis-type metric learning.

4. Windowed Fourier transform

A central piece to our construction of shape descriptors is the notion of *vertex-frequency analysis* or *windowed Fourier transform (WFT)*, generalizing these constructions from classical signal processing to non-Euclidean domains. Here, we follow the approach of Shumann et al. [SRV13] for the generalization of the WFT in the spectral domain. We note that in principle other methods for hierarchical and local frequency analysis on graphs or manifolds can be used instead of the presented construction, including wavelets [CL06] or compressed modes [NVT*14].

Classical WFT The main idea of classical WFT is to analyze the frequency content of a signal that is localized by means of multiplication by a window. Given a function $f \in L^2(\mathbb{R})$ and some 'mother window' g localized at zero, one computes the WFT as

$$(Sf)_{x,\omega} = \int_{\mathbb{R}} f(x')g(x'-x)e^{-ix'\omega}dx'. \qquad (11)$$

Note that the WFT has two indices: spatial location x of the window and frequency ω of the signal in that window. Alternatively, it can be presented as an inner product with a translated an modulated window, $(Sf)_{x,\omega} = \langle f, M_\omega T_x g\rangle_{L^2(\mathbb{R})}$, where T_x and M_ω denote the translation and modulation operators, respectively.

Translation operator in the Euclidean setting is simply $(T_{x'}f)(x) = f(x-x')$. In order to generalize it to manifolds, translation to point x' can be replaced by convolution with a delta-function centered at x', yielding

$$\begin{aligned}(T_{x'}f)(x) &= (f * \delta_{x'})(x)\\ &= \sum_{k\geq 1} \langle f,\phi_k\rangle_{L^2(X)}\langle \delta_{x'},\phi_k\rangle_{L^2(X)}\phi_k(x)\\ &= \sum_{k\geq 1} \hat{f}_k\phi_k(x')\phi_k(x), \qquad (12)\end{aligned}$$

where convolution is understood in the generalized sense of equation (3). Note that such a translation is not shift-invariant in general, i.e., the window would change when moved around the manifold (see Figure 1).

Modulation operator in the classical case is a multiplication by a basis function $(M_\omega f)(x) = e^{i\omega x}f(x)$. In the Fourier domain, the action of modulation amounts to translation $(\widehat{M_{\omega'}f})(\omega) = \hat{f}(\omega-\omega')$. In the generalized case, the modulation is defined in exactly the same way,

$$(M_k f)(x) = \phi_k(x)f(x), \qquad (13)$$

where the eigenvalue λ_k corresponding to the eigenfunction ϕ_k plays the role of 'frequency'.

Manifold WFT Combining the two operators together, we have the modulated and translated window (transform 'atom'; see examples in Figure 1) expressed as

$$g_{x',k}(x) = (M_k T_{x'} g)(x) = \phi_k(x)\sum_{l\geq 1}\hat{g}_l\phi_l(x')\phi_l(x). \qquad (14)$$

Note that the 'mother window' is defined here in the frequency domain by the coefficients $\hat{g}_l$. We thus readily have the WFT of a signal $f \in L^2(X)$

$$(Sf)_{x,k} = \langle f, g_{x,k}\rangle_{L^2(X)} = \sum_{l\geq 1}\hat{g}_l\phi_l(x)\langle f,\phi_l\phi_k\rangle_{L^2(X)}, \qquad (15)$$

which can be regarded as a meta-descriptor: given some dense descriptor f (e.g. one of the components of HKS, WKS, or a geometry vector (10)), we construct $D(x)f = ((Sf)_{x,1},\dots,(Sf)_{x,K})^\top$ taking the first K frequencies of the WFT. The WFT allows to capture the local context of a signal on the manifold, making it roughly analogous to taking the values of the signal in a small 'patch'; here $D(x)$ acts as a position-dependent 'patch operator' representing the local structure of f around point x in the frequency domain.

Special cases We would like to point out the following special cases of the WFT, which show that this framework can be considered as a generalization of several previous approaches.

Case I: when $\hat{g}_k = \delta_{k1}$, we simply have $g_{x',k}(x) = \phi_k(x)\phi_1(x')\phi_1(x)$. Since the first LBO eigenvector is constant, the atom (up to scaling) is $g_{x',k}(x) \propto \phi_k(x)$, i.e., the standard LBO eigenbasis element independent on the location x'. The WFT thus reduces to a simple Fourier transform (2). This result is an intuitive consequence of the uncertainty principle: when the window is perfectly localized in the frequency domain, it is perfectly delocalized in the spatial domain.

Case II: when $f \equiv 1$, the WFT contains information only about the geometric structure of the manifold. In this setting,

$$(S1)_{x,k} = \sum_{l\geq 1} \hat{g}_l \phi_l(x) \underbrace{\langle \phi_k, \phi_l \rangle_{L^2(X)}}_{\delta_{kl}} = \hat{g}_k \phi_k(x), \qquad (16)$$

and for a particular choice of $\hat{g}_k = \lambda_k^{-1/2}$ we get Rustamov's GPS descriptor (7).

Case III: when $f = \delta_x$, the DC frequency of the WFT has the form of (8),

$$(S\delta_x)_{x,1} = \sum_{l\geq 1} \hat{g}_l \phi_l^2(x), \qquad (17)$$

and in particular for $\hat{g}_l = e^{-t\lambda_l}$ we obtain the HKS and for $\hat{g}_l = \exp\left(\frac{\log \nu - \log \lambda_l}{2\sigma^2}\right)$ the WKS at point x, respectively.

Discretization In the discrete setting, the surface X is sampled at N points $x_1,\dots,x_N$. On these points, we construct a triangular mesh (V,E,F) with vertices $V = \{1,\dots,N\}$, 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.

A function on the surface is represented by an N-dimensional vector $\mathbf{f} = (f(x_1),\dots,f(x_N))^\top$. The inner product is discretized as $\langle \mathbf{f},\mathbf{g}\rangle = \mathbf{f}^\top \mathbf{A}\mathbf{g}$, where $\mathbf{A} = \mathrm{diag}(a_1,\dots,a_N)$ and $a_i = \frac{1}{3}\sum_{jk:ijk\in F} A_{ijk}$ denotes the local area element at vertex i and A_{ijk} denoting the area of triangle ijk.

To discretize the LBO as an $N\times N$ matrix $\mathbf{L} = \mathbf{A}^{-1}\mathbf{W}$, we use the classical cotangent formula [PP93, MDSB03],

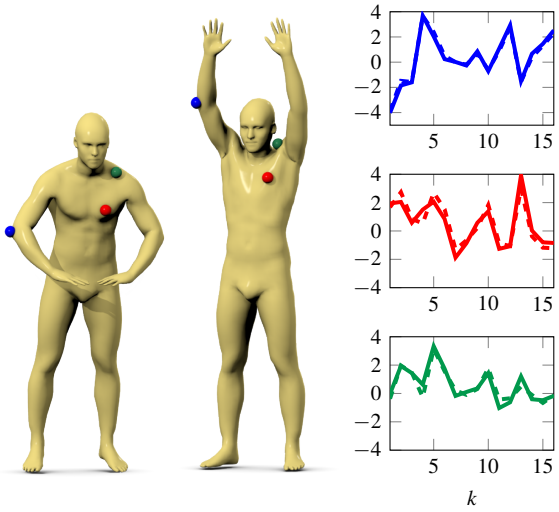

Figure 2: WFTs computed on two different poses of the same shape at three points marked in red, blue, and green. Solid lines represent the WFT of the shape on the left, dashed lines the ones of the shape on the right.

according to which

$$w_{ij} = \begin{cases} (\cot\alpha_{ij} + \cot\beta_{ij})/2 & ij \in E; \\ -\sum_{k\neq i} w_{ik} & i = j; \\ 0 & \text{else;} \end{cases} \qquad (18)$$

where α_{ij},β_{ij} denote the angles $\angle ikj, \angle jhi$ of the triangles sharing the edge ij.

The first $K \leq N$ eigenfunctions and eigenvalues of the LBO operator are computed by performing the generalized eigendecomposition $\mathbf{W}\mathbf{\Phi} = \mathbf{A}\mathbf{\Phi}\mathbf{\Lambda}$, where $\mathbf{\Phi} = (\boldsymbol{\phi}_1,\dots,\boldsymbol{\phi}_K)$ is an $N\times K$ matrix containing as columns the discretized eigenfunctions and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1,\dots,\lambda_K)$ is the diagonal matrix of the corresponding eigenvalues. Note that the eigenvectors are $\mathbf{A}$-orthonormal, i.e. $\mathbf{\Phi}^\top\mathbf{A}\mathbf{\Phi} = \mathbf{I}$.

The discretized WFT is computed as

$$S\mathbf{f} = (\mathbf{f}\boxdot\mathbf{\Phi})^\top\ \mathbf{A}\ \mathbf{\Phi}\hat{\mathbf{g}}\boxdot\mathbf{\Phi}^\top), \qquad (19)$$

where $\hat{\mathbf{g}}$ is the K-dimensional vector representing the window in the frequency domain, $\mathbf{f}$ is the N-dimensional vector representing the input function, and $(\mathbf{a}\boxdot\mathbf{B})_{ij} = a_i b_{ij}$ denotes element-wise multiplication of a vector and matrix, replicating the vector along the second dimension (`repmat` in MATLAB). The resulting WFT is a matrix of size $K\times N$.

5. Localized spectral CNN

The main goal of this paper is to extend the convolutional neural networks (CNN) to non-Euclidean domains. Convolutional neural networks [LBD*89] are hierarchical architectures built of alternating convolutional-, pooling- (non-linear averaging), and fully connected layers. The parameters of

different layers are learned by minimizing some task-specific cost function. In image analysis applications, the input into the CNN is a function representing pixel values given on a Euclidean domain (plane); due to shift-invariance the convolution can be thought of as passing a template across the plane and recording the correlation of the template with the function at that location. One of the major problems in applying the CNN paradigm to non-Euclidean domains is the lack of shift-invariance, making it impossible to think of convolution as correlation with a fixed template: the template now has to be location-dependent.

Here, we propose using the WFT as a mechanism for extracting local 'patches' from functions defined on manifolds. The spatial support of the 'patch' depends on the choice of the window g. Note that in the definition of the WFT the geometric structure of the manifold is captured by the Laplace-Beltrami eigenfunctions. As a result, the same framework can be used for any shape representation (e.g. mesh, point cloud, etc.): the specific representation of the shape influences only the construction of the Laplace-Beltrami operator.

We refer to our approach as *localized spectral CNN* (LSCNN). For the sake of simplicity, the neural network architecture considered in the following consists of only two layers (comparable with the SN1 architecture in [MBBV15]). The first layer is a *fully connected layer*, producing outputs as weighted sums of the inputs, followed by a non-linear function. The second layer applies the WFT to extract the local structure of the input around each point. Since each input dimension might contain features of different scale, we employ a different window for each input dimension. The WFTs are then passed through a bank of filters applied in the frequency domain, producing the outputs used as the descriptor dimensions. As the input to the first layer, any intrinsic descriptor can be used (specifically, we use geometry vectors defined in equation (10)). All the parameters of the layers (weights, windows coefficients, and filters) are variables that are found by means of supervised learning.

Fully connected layer Let us be given a P-dimensional input $\mathbf{f}^{\text{in}}(x) = (f_1^{\text{in}}(x), \dots, f_P^{\text{in}}(x))$. The fully connected layer produces a Q-dimensional output defined as

$$f_q^{\text{out}}(x) = \xi\left(\sum_{p=1}^{P}\sum_{k=1}^{K} w_{qp} f_p^{\text{in}}(x)\right), \quad q = 1, \dots, Q, \quad (20)$$

where $\xi(t) = \max(0,t)$ is the *ReLU activation function*. Note that without ReLU, if the inputs are geometry vectors, learning the weights of the fully connected layer is equivalent to the OSD [LB14]. Fixing weights corresponding to low- or band-pass filters, the fully connected layer implements the HKS and WKS, respectively.

Convolutional layer Next, the output of the fully connected layer acts as the input into the convolutional layer; we denote the input again by $\mathbf{f}^{\text{in}}(x)$ and its dimension by P. For each input dimension, we use a different window. The family of P windows is parametrized in some fixed interpolation basis in the frequency domain as in (9),

$$\gamma_p(\lambda) = \sum_{m=1}^{M} b_{pm}\beta_m(\lambda), \quad p = 1, \dots, P, \quad (21)$$

where the $P \times M$ matrix (b_{pm}) of weights defines the windows. Figure 3 shows an example of the estimated windows after the learning. The WFT of the pth input dimension uses the respective pth window,

$$(Sf_p^{\text{in}})_{x,k} = \sum_{l \geq 1} \gamma_p(\lambda_l)\phi_l(x)\langle f_p^{\text{in}}, \phi_l\phi_k\rangle_{L^2(X)}, \quad (22)$$

producing at each point a K-dimensional vector for each of the P input dimensions. Our goal is to produce a Q-dimensional output, and for this purpose, the WFTs are passed through a bank of filters. The qth dimension of the output is given by

$$f_q^{\text{out}}(x) = \sum_{p=1}^{P}\sum_{k=1}^{K} a_{qpk}|(Sf_p^{\text{in}})_{x,k}|, \quad q = 1, \dots, Q. \quad (23)$$

The output of the convolutional layer is used as our final LSCNN descriptor.

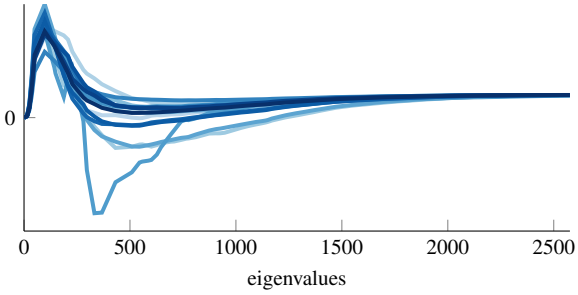

Figure 3: Example of a family of windows $\hat{g}_1, \dots, \hat{g}_P$ learned by the LSCNN on the FAUST dataset.

Loss function The LSCNN comprising the fully-connected and convolutional layer is a parametric hierarchical system $\mathbf{f}_\Theta(x)$ producing a Q-dimensional descriptor at each point x (here $\Theta = \{(w_{qp}), (b_{pm}), (a_{qpk})\}$ denotes the set of learnable parameters). Given a training set of knowingly similar and dissimilar pairs of points on pairs of shapes, respectively positives $\mathcal{T}^+ = \{(x, x^+)\}$ and negatives $\mathcal{T}^- = \{(x, x^-)\}$, we aim at estimating the optimal task-specific parameters of the descriptor minimizing the aggregate loss

$$\ell(\Theta) = (1-\mu)\ell_+(\Theta) + \mu\,\ell_-(\Theta) \quad (24)$$

where

$$\ell_+(\Theta) = \frac{1}{|\mathcal{T}^+|}\sum_{(x,x^+)\in\mathcal{T}^+} \|\mathbf{f}_\Theta(x) - \mathbf{f}_\Theta(x^+)\|_2^2, \quad (25)$$

$$\ell_-(\Theta) = \frac{1}{|\mathcal{T}^-|}\sum_{(x,x^-)\in\mathcal{T}^-} \max\{0, M - \|\mathbf{f}_\Theta(x) - \mathbf{f}_\Theta(x^-)\|^2\},$$

are the positive and negative losses, respectively, μ is a parameter governing their trade-off, and M is a margin mapping the negatives apart.

We stress that HKS, WKS, and OSD descriptors are obtained by a particular choice of the parameters Θ. Thus, if the training set is designed well and training is performed correctly, our descriptor can perform only better than the above.

Comparison to ShapeNet Masci et al. [MBBV15] introduced ShapeNet, a generalization of CNN to triangular meshes based on geodesic local patches. The core of this method is the construction of local geodesic polar coordinates using a procedure previously employed for intrinsic shape context descriptors [KBLB12]. The patch operator $(D(x)f)(\theta,\rho)$ in ShapeNet maps the values of the function f around vertex x into the local polar coordinates θ,ρ, leading to the definition of the *geodesic convolution*

$$(f*a)(x)=\sum_{\rho,\theta} a(\theta+\Delta\theta)(D(x)f)(\theta,\rho), \tag{26}$$

which follows the idea of multiplication by template, but is defined up to arbitrary rotation $\Delta\theta\in[0,2\pi)$ due to the ambiguity in the selection of the origin of the angular coordinate. In the ShapeNet convolutional layer, the outputs corresponding to all the rotations of the templates are produced and then a maximum is taken,

$$f_q^{\text{out}}=\max_{\Delta\theta}\sum_{p=1}^{P} f_p^{\text{in}}*a_{\Delta\theta,qp}, \tag{27}$$

where $a_{\Delta\theta}(\theta,\rho)=a(\theta+\Delta\theta,\rho)$ denotes the coefficients of the template rotated by $\Delta\theta$, and the convolution is in the sense of equation (26).

We note the following main drawbacks of this construction. First, the charting method relies on a fast marching-like procedure requiring a triangular mesh. The method is relatively insensitive to the triangulation, but may fail if the mesh is very irregular. Second, the radius of the geodesic patches must be sufficiently small compared to the convexity radius of the shape, otherwise the resulting patch is not guaranteed to be a topological disk. In practice, this limits the size of the patches one can safely use, or requires an adaptive radius selection mechanism. In contrast, the proposed localized spectral CNN is free of these limitations: it can work with any shape representation, provided one can compute the discretized Laplace-Beltrami operator and its eigenfunctions and eigenvalues for this representation; since the patch operator is constructed in the frequency domain using the WFT, there is also no issue related to the topology of the patch.

6. Results

Datasets We used two public-domain datasets of scanned human shapes in different poses: SCAPE [A*05] and FAUST [BRLB14], the latter being the most recent and particularly challenging, given a high variability of non-isometric deformations as well as significant variability between different human subjects. The meshes in SCAPE were resampled to 12.5K vertices, whereas for FAUST we used the registration meshes without further pre-processing. In addition we scaled all shapes to have unit geodesic diameter. In both datasets, groundtruth point-wise correspondence between the shapes was known for all points.

Methods and Settings In all our experiments, we used $K=300$ LBO eigenfunctions and eigenvalues computed using MATLAB `eigs` function. For OSD and our descriptor, we used $M=150$-dimensional geometry vectors as inputs, computed according to (9)–(10) using B-spline bases [LB14]. We compared the performance of the proposed approach to HKS [SOG09], WKS [ASC11], OSD [LB14], spectral CNN (SCNN) [BZSL14], and ShapeNet (SN1) [MBBV15] using the code and settings provided by the respective authors. To make the comparison fair, all the descriptors were $Q=16$-dimensional as in [LB14].

Our descriptor was tested in two configurations. *LSCNN1*, consisting of a fully connected layer (reducing the dimensionality of the 150-dimensional input to 16 dimensions), followed by a convolutional layer using a fixed WFT Gaussian window $\gamma(\lambda)=e^{-\frac{\lambda^2}{2\sigma^2}}$ with $\sigma=10^{-5}$. In this configuration the parameters of the network that are learned are $\Theta=\{(w_{qp}),(a_{qpk})\}$. *LSCNN2* is similar to LSCNN1, with the difference that now the WFT windows are also learned. We use 16 filters (one per dimension), each represented by the B-spline coefficients. In this configuration, the free parameters are $\Theta=\{(w_{qp}),(b_{pm}),(a_{qpk})\}$. Furthermore, as a 'sanity check', we also used a configuration without the convolutional layer, comprising only a fully connected + ReLU layer (referred to as NN1). This architecture is compatible with the OSD, with the addition of a non-linearity at the output.

Training Each dataset was split into disjoint training, validation, and test sets. On the FAUST dataset subjects 1–7 were used for training (10 poses per subject, a total of 70 shapes), subject 8 (10 shapes) for validation, and subject 9–10 for testing (total of 20 shapes). On SCAPE, we used shapes 20–29 and 50–70 for training (total 31 shapes), five different shapes for validation, and 40 remaining shapes for testing. The positive and negative sets of vertex pairs required for training were generated on the fly, to keep the storage requirements for the training algorithm, via uniform stochastic sampling. Each point on the first shape has only a single groundtruth match (given by the one-to-one correspondence) and is assigned to one out of $N-1$ possible negatives: first, sample two shapes, then form the positive set with all corresponding points, and finally, form the negative set with first shape vertices and a random permutation of the ones of the second shape. This strategy differs from [LB14] who considered only points on the same shape. The advantage of our sampling strategy is

that it allows learning invariance also across several poses and subjects.

LSCNN was implemented in Theano [B*10] and trained until convergence using Adadelta [Zei12], a stochastic first order method with automatic adjustment of the learning rate (step size). Training was performed for 250 epochs, each epoch consisting of 100 updates (stochastic gradient descent steps). In each update of the training, we used N positive and negative pairs, where N is the number of shape vertices.

Timing Typical training times for the more complex descriptor (LSCNN2) are around two hours on a NVIDIA TITAN Black GPU board and, at test time, the system is able to produce a throughput of approximately 30K vertices per second. The pre-computation of the LB operator and its eigendecomposition takes around 10s for a shape with 7K vertices.

Similarity map Figures 4 (compare to Figure 2 in [LB14] and Figures 5–6 in [MBBV15]) 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. Figure 5 shows another example of LSCNN on point clouds, where the WFT was computed using the graph Laplacian. Our approach shows a good trade-off between localization (similar to HKS) and accuracy (less spurious minima than WKS and OSD), as well as robustness to different kinds of noise.

Descriptor evaluation We evaluated the descriptor performance using the *cumulative match characteristic* (CMC) and the *receiver operator characteristic* (ROC). 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 correspondence quality possible with our descriptors was evaluated using the Princeton protocol [KLF11], plotting the percentage of nearest-neighbor matches that are at most r-geodesically distant from the groundtruth correspondence.

The performance evaluation is depicted in Figures 6–9. We observe that NN1 (fully connected layer+ReLU) outperforms the OSD, which we attribute to the non-linearity. We see further significant improvement from using a convolutional layer (LSCNN1 and LSCNN2). Furthermore, we observe that LSCNN generalizes better to data from a different dataset (transfer learning from FAUST to SCAPE and vice versa) compared to ShapeNet.

7. Conclusions

In this paper, we proposed a new generalization of convolutional neural networks to non-Euclidean domains using the windowed Fourier transform for representing local shape structures. In our localized spectral CNN, both the transform window and a bank of filters that are applied to the transform are learned. Using this approach, we were able to create class-specific descriptors that are expressive, localized, and robust.

Limitations The construction of class-specific descriptors tacitly assumes that all shapes in the class share some common geometric structure, and their Laplacian eigenbasis, up to known ambiguities, do not differ arbitrarily. We hypothesize that if one tries to deal with a class that is too broad (e.g. all mechanical objects, or all living things), the performance advantage of our method over 'hand-crafted' descriptors such as HKS and WKS will diminish, and it is likely that we will learn these descriptors (as they are a particular configuration of our network).

Extensions The spectral formulation of our framework allows application to a broad range of geometric structures, such as point clouds or even abstract graphs and networks. Constructing an analogy of successful convolutional neural networks on such domains has been elusive so far, as there is no clear notion of a local 'patch' and its representation. We believe that our approach could be the right path towards this goal.

Acknowledgement

D. B., J. M., and M. B. are supported by the ERC Starting grant No. 307047.

References

[A*05] ANGUELOV D., ET AL.: SCAPE: Shape completion and animation of people. *TOG 24*, 3 (2005), 408–416. 7

[ASC11] AUBRY M., SCHLICKEWEI U., CREMERS D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. ICCV* (2011). 1, 3, 7

[B*10] BERGSTRA J., ET AL.: Theano: a CPU and GPU math expression compiler. In *Proc. SciPy* (June 2010). 8

[BBG94] BÉRARD P., BESSON G., GALLOT S.: Embedding riemannian manifolds by their heat kernel. *Geometric & Functional Analysis 4*, 4 (1994), 373–398. 1, 3

[BCG08] BEN-CHEN M., GOTSMAN C.: Characterizing shape using conformal factors. In *Proc. 3DOR* (2008). 1

[BMP00] BELONGIE S., MALIK J., PUZICHA J.: Shape context: A new descriptor for shape matching and object recognition. In *Proc. NIPS* (2000). 1

[BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR* (2014). 7

[BZSL14] BRUNA J., ZAREMBA W., SZLAM A., LECUN Y.: Spectral networks and locally connected networks on graphs. In *Proc. ICLR* (2014). 2, 7

[CL06] COIFMAN R. R., LAFON S.: Diffusion maps. *Applied and Computational Harmonic Analysis 21*, 1 (2006), 5–30. 1, 3, 4

[COC14] CORMAN E., OVSJANIKOV M., CHAMBOLLE A.: Supervised descriptor learning for non-rigid shape matching. In *Proc. NORDIA* (2014). 2

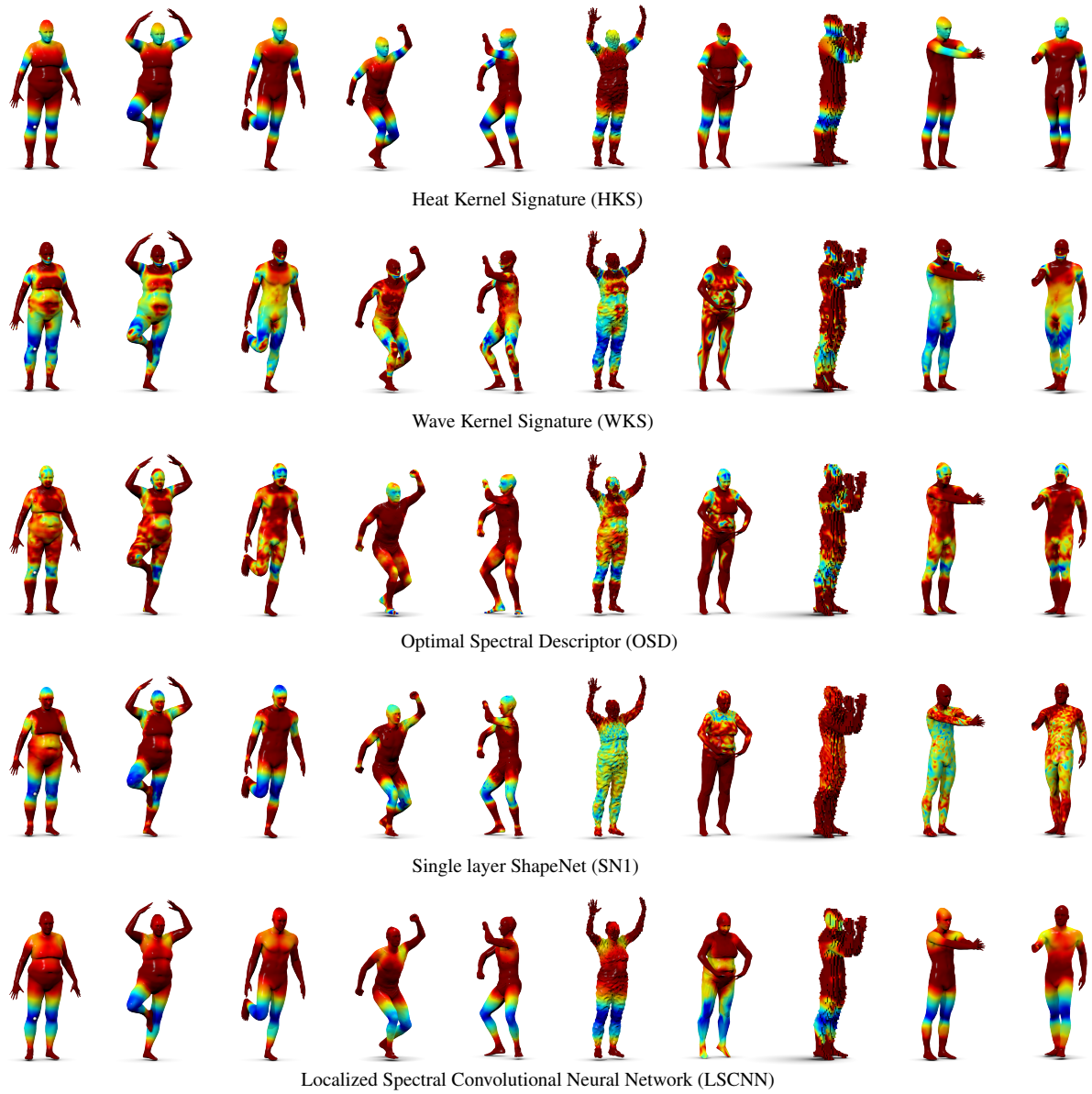

Figure 4: Distance map in the descriptor space. A point on the reference shape (leftmost) is compared to all other points on the same and on other shapes. Shown left-to-right: reference shape from FAUST dataset, different pose of the same shape, different subject in the same dataset, two shapes from SCAPE dataset, Gaussian noise, heavy subsampling, voxelization noise, topological noise (glued fingers and missing parts). Small distances in the descriptor space correspond to cold colors.

[EK03] ELAD A., KIMMEL R.: On bending invariant signatures for surfaces. *PAMI 25*, 10 (2003), 1285–1295. 1

[GBAL09] GEBAL K., BÆRENTZEN J. A., ANÆS H., LARSEN R.: Shape analysis using the auto diffusion function. *CGF 28*, 5 (2009), 1405–1413. 1, 3

[HK03] HAMZA A. B., KRIM H.: Geodesic object representation and recognition. In *Proc. DGCI* (2003). 1

[HSG13] HUANG Q., SU H., GUIBAS L.: Fine-grained semi-supervised labeling of large shape collections. *TOG* (2013). 1

[JH99] JOHNSON A. E., HEBERT M.: Using spin images for efficient object recognition in cluttered 3D scenes. *PAMI 21*, 5 (1999), 433–449. 1

[KBLB12] KOKKINOS I., BRONSTEIN M. M., LITMAN R., BRONSTEIN A. M.: Intrinsic shape context descriptors for deformable shapes. In *Proc. CVPR* (2012). 1, 2, 7

[KHS10] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3D mesh segmentation and labeling. *TOG 29*, 3 (2010). 1

[KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. *TOG 30*, 4 (2011), 1–12. 8

Figure 5: Distance map in the descriptor space computed using LSCNN on point clouds. A point on the reference shape (leftmost) is compared to all other points on the same and on other shapes (four from SCAPE and four from FAUST datasets). Small distances in the descriptor space correspond to cold colors.

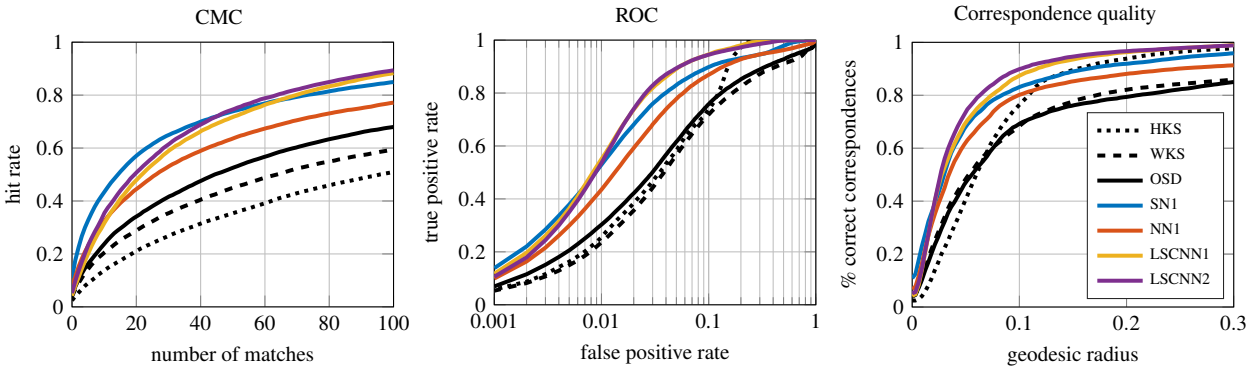

Figure 6: Performance of descriptors trained on a subset of FAUST dataset and tested on a disjoint subset of FAUST dataset.

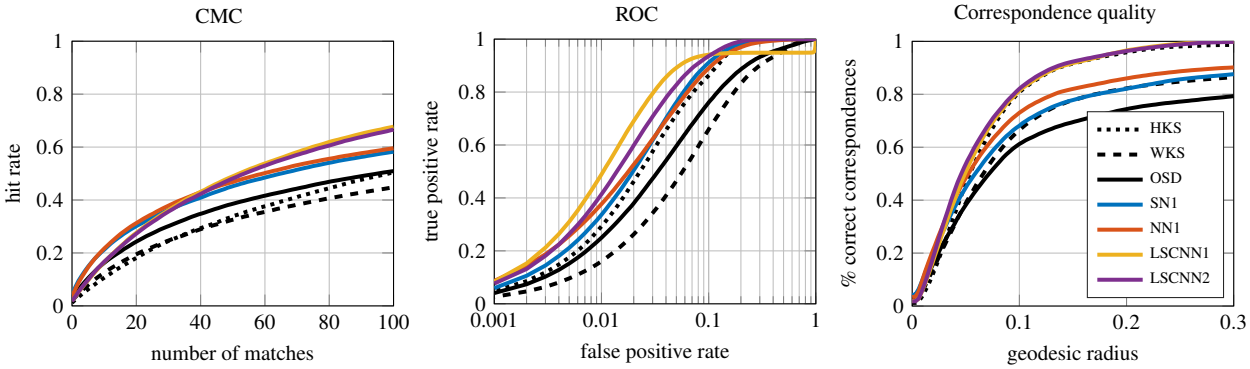

Figure 7: Performance of descriptors trained on a subset of FAUST dataset and tested on SCAPE dataset.

[L*13] LIAN Z., ET AL.: A comparison of methods for non-rigid 3D shape retrieval. *Pattern Recognition 46*, 1 (2013), 449–461. 1

[LB14] LITMAN R., BRONSTEIN A. M.: Learning spectral descriptors for deformable shape correspondence. *PAMI 36*, 1 (2014), 170–180. 2, 3, 6, 7, 8

[LBBC14] LITMAN R., BRONSTEIN A., BRONSTEIN M., CASTELLANI U.: Supervised learning of bag-of-features shape descriptors using sparse coding. *CGF 33*, 5 (2014), 127–136. 1, 2

[LBD*89] LECUN Y., BOSER B., DENKER J. S., HENDERSON D., HOWARD R. E., HUBBARD W., JACKEL L. D.: Backpropagation applied to handwritten zip code recognition. *Neural Computation 1*, 4 (1989), 541–551. 2, 5

[Lév06] LÉVY B.: Laplace-Beltrami eigenfunctions towards an algorithm that "understands" geometry. In *Proc. SMI* (2006). 1

[Low04] LOWE D. G.: Distinctive image features from scale-invariant keypoints. *IJCV 60*, 2 (2004), 91–110. 1

[MBBV15] MASCI J., BOSCAINI D., BRONSTEIN M. M., VANDERGHEYNST P.: ShapeNet: Convolutional neural networks on non-Euclidean manifolds. *arXiv:1501.06297* (2015). 2, 6, 7, 8

[MCH*06] MANAY S., CREMERS D., HONG B.-W., YEZZI A. J., SOATTO S.: Integral invariants for shape matching. *PAMI 28*, 10 (2006), 1602–1618. 1

[MDSB03] MEYER M., DESBRUN M., SCHRÖDER P., BARR

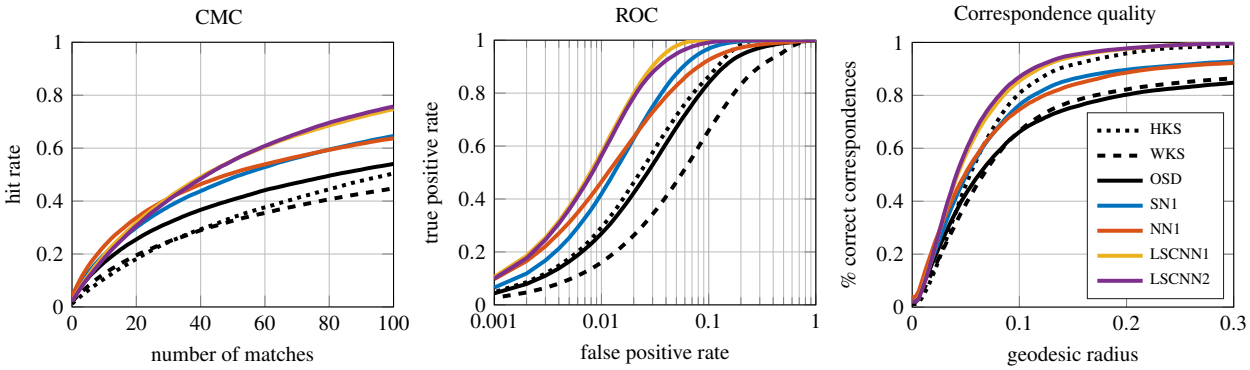

Figure 8: Performance of descriptors trained on a of SCAPE dataset and tested on a disjoint subset of SCAPE dataset.

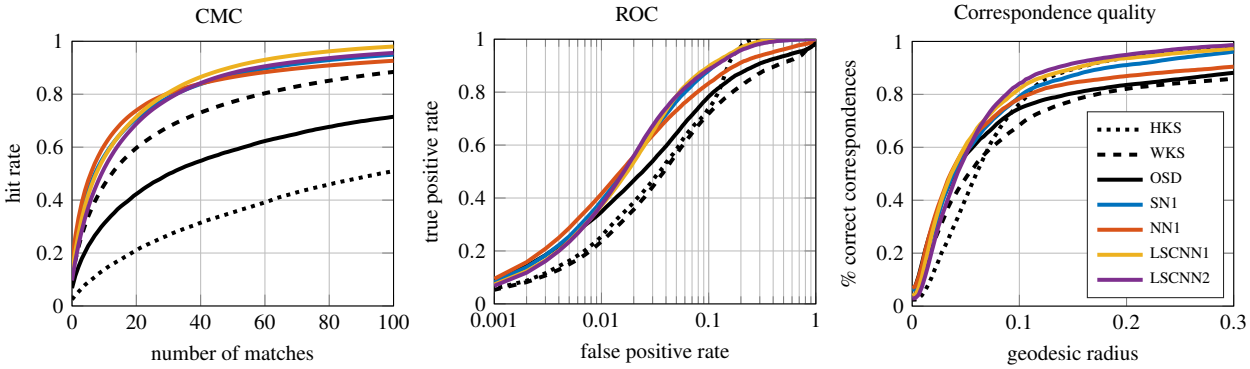

Figure 9: Performance of descriptors trained on a subset of SCAPE dataset and tested on FAUST dataset.

A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization&Mathematics* (2003), 35–57. 5

[MHK*08] MATEUS D., HORAUD R., KNOSSOW D., CUZZOLIN F., BOYER E.: Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In *Proc. CVPR* (2008). 3

[NVT*14] NEUMANN T., VARANASI K., THEOBALT C., MAGNOR M., WACKER M.: Compressed manifold modes for mesh processing. *Computer Graphics Forum 33*, 5 (2014), 35–44. 4

[OFCD02] OSADA R., FUNKHOUSER T., CHAZELLE B., DOBKIN D.: Shape distributions. *TOG 21*, 4 (2002), 807–832. 1

[OMMG10] OVSJANIKOV M., MÉRIGOT Q., MÉMOLI F., GUIBAS L.: One point isometric matching with the heat kernel. *Computer Graphics Forum 29*, 5 (2010), 1555–1564. 1

[PKG03] PAULY M., KEISER R., GROSS M.: Multi-scale feature extraction on point-sampled surfaces. *CGF 22*, 3 (2003), 281–289. 1

[PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2*, 1 (1993), 15–36. 5

[RRBW*14] RODOLÀ E., ROTA BULÒ S., WINDHEUSER T., VESTNER M., CREMERS D.: Dense non-rigid shape correspondence using random forests. In *Proc. CVPR* (2014). 1

[Rus07] RUSTAMOV R. M.: Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Proc. SGP* (2007). 1, 3

[RWP06] REUTER M., WOLTER F.-E., PEINECKE N.: Laplace–beltrami spectra as 'shape-dna'of surfaces and solids. *Computer-Aided Design 38*, 4 (2006), 342–366. 1

[SB11] SIPIRAN I., BUSTOS B.: Harris 3D: a robust extension of the harris operator for interest point detection on 3D meshes. *Visual Computer 27*, 11 (2011), 963–976. 1

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L. J.: A concise and provably informative multi-scale signature based on heat diffusion. *CGF 28*, 5 (2009), 1383–1392. 1, 3, 7

[SRV13] SHUMAN D. I., RICAUD B., VANDERGHEYNST P.: Vertex-frequency analysis on graphs. *arXiv:1307.5708* (2013). 2, 4

[WSK*15] WU Z., SONG S., KHOSLA A., YU F., ZHANG L., TANG X., XIAO J.: 3d shapenets: A deep representation for volumetric shape modeling. In *Proc. CVPR* (2015). 2

[Zei12] ZEILER M. D.: ADADELTA: An adaptive learning rate method. *arXiv:1212.5701* (2012). 8