Shape correspondence using anisotropic Chebyshev spectral CNNs

Qinsong Li[1], Shengjun Liu[*1,3], Ling Hu[*2], Xinru Liu[1]
[1]Institute of Engineering Modeling and Scientific Computing, Central South University
[2]School of Mathematics and Computational Science, Hunan First Normal University
[3]State Key Laboratory of High Performance Manufacturing Complex, Central South University

Abstract

Establishing correspondence between shapes is a very important and active research topic in many domains. Due to the powerful ability of deep learning on geometric data, lots of attractive results have been achieved by convolutional neural networks (CNNs). In this paper, we propose a novel architecture for shape correspondence, termed Anisotropic Chebyshev spectral CNNs (ACSCNNs), based on a new extension of the manifold convolution operator. The extended convolution operators aggregate the local features of signals by a set of oriented kernels around each point, which allows to much more comprehensively capture the intrinsic signal information. Rather than using fixed oriented kernels in the spatial domain in previous CNNs, in our framework, the kernels are learned by spectral filtering, based on the eigen-decompositions of multiple Anisotropic Laplace-Beltrami Operators. To reduce the computational complexity, we employ an explicit expansion of the Chebyshev polynomial basis to represent the spectral filters whose expansion coefficients are trainable. Through the benchmark experiments of shape correspondence, our architecture is demonstrated to be efficient and be able to provide better than the state-of-the-art results in several datasets even if using constant functions as inputs.

1. Introduction

Establishing correspondence between shapes is a fundamental problem in geometry processing, computer graphics and vision, referring to a wide variety of applications such as texture mapping, animation, etc.

In the past few years, many methods have been proposed for shape correspondence, mainly including minimum-distortion based methods [13, 49], embedding methods [30, 22], soft correspondence [33, 45] and functional mapping based methods [36, 39, 32].

Compared to the above handcrafted methods, recently, deep learning methods seem to be able to achieve more appealing success. Inspired by the remarkable success of applying Convolutional Neural Networks (CNNs) in image recognition tasks, a few approaches have been proposed to extend convolution to geometry processing and graphics communities. Most achievements obtained by deep learning methods over the last years heavily rely on properties of the convolution operation in convolutional neural networks. First attempts among them still focus on treating geometric data as Euclidean structures, like [38] used CNNs to be applied to range images obtained from multiple views of 3D objects, [51] used a view-based representation to find correspondence between nonrigid shapes and [52] proposed volumetric CNNs to rasterize the volumetric representation of 3D shapes. However, such Euclidean representations may lose significant information of 3D shapes and are not intrinsic, suffering the disadvantage of huge training sets and computational complexity when addressing deformed shapes. Therefore, a set of methods brought together under the term geometric deep learning [11] emerged, aiming to define convolution operations for deep neural networks that can directly process non-Euclidean input data such as graphs and manifolds.

Existing works in geometric deep learning can be broadly categorized into two different subsets: spatial filtering and spectral filtering. The former spatial filtering performs convolution in local Euclidean neighborhoods w.r.t. local positional relations between points, represented for example as polar or Cartesian coordinates [29, 34, 19]. However, constructing such local coordinates sometimes may be considerable time consuming (e.g. polar coordinates need to compute geodesic distances between point pairs) or heavily depend on shape representations. The latter subset, spectral filtering [12, 17, 23], based on spectral graph theory [14], provides a well-defined localization operator on graphs or manifolds via convolutions implemented in the spectral domain [43], analogously to Fourier domain filtering of traditional signals, where eigenvalues of a graph's Laplacian are acted as frequencies of signals defined on graphs.

[*] Corresponding authors: shjliu.cg@gmail.com; 18246514@qq.com

Compared to the spatial filtering, spectral filtering can be flexibly applied in various shape representations provided equipped with Laplace-Beltrami Operator (LBO) discretization, and generally gets higher computational efficiency. However, in the recent construction of such type of CNNs, most of the efforts are paid on designing desirable expansions of the spectral filters based on graph Laplacian, such as Spectral CNN [12], Chebyshev Spectral CNN (ChebyNet) [17], Simplified Chebyshev Net[23], CayleyNets [26], etc. For shape correspondence, these methods still seem to be inefficient to analyze the overall structure information of shapes due to the important directional information ignored during the learning process.

Main contributions. In this paper, we propose a novel CNN architecture for manifolds, called Anisotropic Chebyshev spectral CNNs (ACSCNNs). We first extend the general graph convolution operators to an anisotropic version for manifolds, via filtering on the eigenvalues and eigenfunctions of multiple Anisotropic Laplace-Beltrami Operators (ALBO) [8], each of them is determined by one rotation angle around the normals in tangent planes of the underlying shape. Such mechanism can also be equally treated as performing convolution operators for signals with plenty of oriented kernels defined on manifolds, which allows to comprehensively incorporate the local information of signals from multiple directions around each point on the shape. As Chebyshev polynomials based filters are proved to be naturally spatial localized and the convolution computation is not costly due to it alleviating the cost of explicitly computing the graph Fourier transform [17], we then use Chebyshev polynomials to represent our spectral filters where the expansion coefficients are learned under shape correspondence tasks.

Compared with two manifold deep learning methods that are most relevant to us, Anisotropic Diffusion descriptor (ADD) [8] integrates the shape directional information provided by eigen-decomposition of the ALBO to learn optimized task-dependent spectral descriptors while we utilize them to construct anisotropic convolution operators, with totally different goals and learning architectures. Anisotropic CNN (ACNN) [7] constructs convolution patches using fixed anisotropic heat kernels as spatial weighting functions and they only use the first few of eigenfunctions (low-frequency) of the ALBO. Conversely, our work is more efficient and superior as our kernels are trainable and all the eigenfunctions implicitly used. Extensive benchmark experiments across several challenging datasets demonstrate that our architecture outperforms the state-of-the-art results in shape correspondence even using constant functions as inputs.

2. Related works

2.1. Shape correspondence

Shape correspondence is a well-studied area of computer graphics. Related algorithms can be broadly categorized into two families: handcrafted methods and deep learning methods. We review the former in this subsection and the latter in the next one.

The handcrafted family asserts a certain geometric assumption and pursuits for by some numerical schemes. Given a pair of shapes, the traditional solution to build correspondence between them involves minimization of a distortion criterion which can be roughly divided into two categories: pointwise descriptor similarity [4, 41, 46, 31] and pairwise relations [13, 15, 47]. In the former case, modeling assumptions attempt to characterize the action of a class of deformations on some geometric quantities commonly referred to as descriptors. Such geometric quantities often encode local geometric information in the vicinity of a point on the shape (point-wise descriptors) such as normal orientation [42], curvature [37], and heat [46] or wave propagation properties [4]. Matches are obtained via nearest neighbor search or, when injectivity is required by solving a linear assignment problem. Another type of geometric quantities is the global relation between pairs of points (pair-wise descriptors), which include geodesic [18, 9, 50], diffusion [15, 10, 49], etc. The use of pair-wise descriptors leads to a quadratic assignment problem.

There exists another branch of works trying to establish correspondence based on functional map framework [36, 39, 35, 32] which is originally proposed by [36]. Such methods assume as input a set of corresponding functions (derived from several geometric features). Then by estimating a functional map matrix, it allows to transfer real functions across the two shapes, which can be converted to a point-wise map.

2.2. Geometric deep learning

As of today, most recent works attempt to formulate the correspondence problem as a learning problem [29, 34, 19] and these methods hold the record of performance on deformable correspondence benchmarks. More related work refers to the survey [1]. Here we only review those most closely related to ours.

Deep learning on manifolds. By considering the general form of heat kernel signature [46] and wave kernel signature [4], [28] introduced a learning architecture to construct optimized task-specific descriptors, where the coefficients of spectral filters which are represented as linear combinations of B-spline basis are learned. Later, [8] generalized this work to an anisotropic version via learning a set of optimal oriented spectral filters, based on the eigen-decomposition of the ALBO [2, 8]. Moreover, a variety of works focus on generalizing CNNs to manifolds. To incorporate the local information of the signals defined on such the non-Euclidean data structure, the spatial filtering methods attempt to directly construct patch operators on manifolds or graphs on the spatial domain and then learn the template weightings by minimizing a task-specific cost function. [29] introduced the Geodesic Convolutional Neural Networks (GCNN) by using a local geodesic system of polar coordinates proposed by [24] to extract "patches". [6] generalized GCNN by using the Windowed Fourier Transform on graphs [44] to construct convolutional patches from the spectral domain, called Localized Spectral CNN (LSCNN). The same author [7] improved this approach further, introducing Anisotropic CNN (ACNN) via using anisotropic heat kernels as spatial weighting functions, which allows to align extracted patches based on the local principal curvatures of the input mesh. Mixture Model Networks (MoNet) [34] unified previous methods, e.g., classical Euclidean CNN, Geodesic CNN, ACNN, and introduced a new type of kernel in parametric construction. Based on the framework of MoNet, [19] defined another type of continuous kernel functions with B-splines.

Deep learning on graphs. Literature [12] utilized the spectral definition of the convolution theorem on graphs to define convolution operators, where the eigenvectors of the graph Laplacian act as Fourier basis. As an extension, [21] suggested using spline interpolation for smoothing kernels in the spectral domain. [23] proposed an simplified ChebyNet [17] by considering only the one-neighborhood for one filtering application. A filter based on the Caley transform was proposed as an alternative for the Chebyshev approximation by [26]. Together with a trainable zooming parameter, this results in a more stable and flexible spectral filter.

3. Foundation

We model shape as a connected smooth compact two-dimensional manifold (surface) X (possible with boundaries) embedded into $\mathbb{R}^3$. Denote the tangent plane of point $x \in X$ as T_xX and each element in T_xX is called the tangent vector at x. A Riemannian metric is an inner product on the tangent plane $\langle \cdot, \cdot \rangle_{T_xX} : T_xX \times T_xX \rightarrow \mathbb{R}$, which is smoothly depending on x and describes how the manifold locally differs from a plane. We denote the space of square integrable real functions on the manifold X by $L^2(X) = \left\{ f : X \rightarrow \mathbb{R}, \int_X f(x)^2 dx < \infty \right\}$, where dx is the area element induced by the above Riemannian metric. Here $< f, g >_X = \int_X f(x)g(x)dx$ expresses the standard inner product on the shape.

As we know, given the Riemannian metric, the second fundamental form of shape X, a 2×2 matrix, its eigenvalues, κ_M and κ_m, are called the principal curvatures and their corresponding eigenvectors $\mathbf{V}_M(x)$ and $\mathbf{V}_m(x)$ constitute an orthonormal basis on the tangent plane T_xX.

3.1. Signal processing on manifolds

Researchers [48] found that the eigenvalues (or spectrum) and eigenfunctions of the Laplace-Beltrami Operator (LBO) behave similarly to the frequency and Fourier basis in the Euclidean space respectively. Then this discovery launched the research on geometry processing and shape analysis from the point of signal processing view.

The Laplace-Beltrami Operator (LBO) (or Laplacian) on shape X is defined as

$$\Delta_X f(x) = -\text{div}_X(\nabla_X f(x)) \tag{1}$$

where $\nabla_X f$ and $\text{div}_X f$ is the intrinsic gradient and the divergence of $f(x) \in L^2(X)$ respectively. Since the LBO is a positive-semidefinite operator, it admits a real eigendecomposition

$$\Delta_X \phi_k(x) = \lambda_k \phi_k(x)$$

with countable non-negative eigenvalues $0{=}\lambda_0 \leq \lambda_1 \leq \ldots$. Eigenfunctions $\phi_0, \phi_1, \ldots$ are orthonormal w.r.t the standard inner product, i.e. $\langle \phi_i, \phi_j \rangle_X = \int_X \phi_i(x)\phi_j(x)dx = \delta_{ij}$ and form an orthonormal basis for $L^2(X)$.

Due to the harmonic properties of the eigenvalues and eigenfunctions of the LBO, the inner product $\hat{f}(\lambda_k) = \langle f, \phi_k \rangle_X$ is called as manifold Fourier transform (coefficient) and function $f \in L^2(x)$ (or inverse manifold fourier tranform) can be expressed as

$$f(x) = \sum_{k \geq 0} \langle f, \phi_k \rangle_X \phi_k(x) = \sum_{k \geq 0} \hat{f}(\lambda_k)\phi_k(x).$$

Analogous to classical signal processing, the above transform enables the formulation of fundamental operations such as filtering. The frequency filtering of signal f is defined as $\hat{f}^{out}(\lambda_k) = \hat{f}(\lambda_k)\hat{g}(\lambda_k)$, where $\hat{g}(\lambda)$ is the spectral filter of kernel $g(x)$. Then taking an inverse manifold Fourier transform for $\hat{f}^{out}$, we can get the filtered signal f^{out}.

Inferred from the above spectral filtering manner, we equivalently define the *Convolution Theorem* on manifolds as

$$(f * g)(x) = \sum_{k \geq 0} \hat{f}(\lambda_k)\hat{g}(\lambda_k)\phi_k(x). \tag{2}$$

Formula (2) enforces the property that convolution in the surface domain is equivalent to multiplication in the manifold spectral domain.

3.2. Anisotropic Laplace-Beltrami Operator

The LBO stated in the above section (Formula (1)), however, is isotropic, as it is insensitive to the underlying shape orientation around each point. In the work of

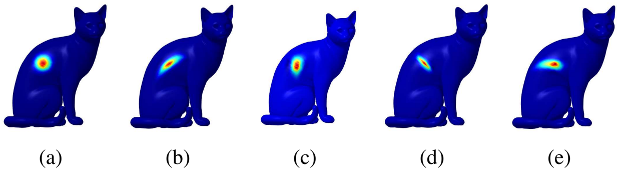


Figure 1. Anisotropic kernels $g_{\alpha\theta,x}(y)$ centered at same point x with different rotation angles θ. Filter $\hat{g}(\lambda) = T_5(\lambda)$ is the Chebyshev polynomial with order 5 (Formula (9)). The respective anisotropic levels and rotation angles are (a) $\alpha = 0, \theta = 0$, (b) $\alpha = 10, \theta = 0$, (c) $\alpha = 10, \theta = \frac{\pi}{4}$, (d) $\alpha = 10, \theta = \frac{\pi}{2}$, (e) $\alpha = 10, \theta = \frac{3\pi}{4}$.

Andreux[2], by changing the diffusion speed along with the directions of principal curvature on the surface, they defined an Anisotropic LBO as

$$\Delta_X f(x) = -\mathrm{div}_X(\mathbf{D}(x)\nabla_X f(x)),$$

here $\mathbf{D}(x)$ is a thermal conductivity tensor acting on the intrinsic gradient direction in the tangent plane and can be represented by an orthogonal basis $\mathbf{V}_M(x)$ and $\mathbf{V}_m(x)$. ALBO allows modeling a heat flow that is position and direction-dependent. Specifically, the authors considered an anisotropy along the maximum curvature direction, i.e.

$$\mathbf{D}_\alpha(x) = \begin{bmatrix} \frac{1}{1+\alpha} & \\ & 1 \end{bmatrix}, \tag{3}$$

where parameter α controls the anisotropic level.

Instead of using only a single anisotropic diffusion kernel along the principal curvature direction, [8] extended the above anisotropic Laplacian [2] with multiple anisotropic angles. They defined the thermal conductivity tensor as

$$\mathbf{D}_{\alpha\theta}(x) = \mathbf{R}_\theta \mathbf{D}_\alpha(x) \mathbf{R}_\theta{}^{\mathrm{T}},$$

where $\mathbf{R}_\theta$ is a rotation around surface normals with angle θ in the tangent plane. Finally, ALBO are defined with two parameters, as shown in the following

$$\Delta_{\alpha\theta} f(x) = -\mathrm{div}_X(\mathbf{R}_\theta \mathbf{D}_\alpha(x) \mathbf{R}_\theta{}^{\mathrm{T}} \nabla_X f(x)) \tag{4}$$

If we consider all the possible rotations $\theta \in [0, 2\pi]$, and set the principal curvature direction as the reference $\theta = 0$, the ALBO will be intrinsic. Therefore, we will adapt this setting in our framework to analyze deformed shapes.

In this paper, to completely capture the shape information, we use the ALBO in Formula (4). While keeping the desirable properties of the standard LBO, ALBO opens the possibility to effectively replace the omnipresent Laplace-Beltrami Operator in many shape analysis methods.

4. Methods

Formula (2) shows that convolution on manifolds can be realized by multiplication in the spectral domain if given desirable filters $\hat{g}(\lambda)$. However, traditional convolution operators on manifolds are based on the eigen-decomposition of LBO or graph Laplacian, and thus the resulted kernels $g(x)$ are also homogeneously diffused on surfaces, insensitive to the intrinsic shape direction information. However, such information plays a very important role in those applications that require a high quality of shape geometric description. To overcome this shortage, we introduce the Anisotropic convolution operator based on the eigen-decomposition of ALBO.

4.1. Anisotropic convolution operator

Given an anisotropic level α and a rotation angle $\theta \in [0, 2\pi)$, we first compute the eigen-decomposition of the corresponding ALBO, by solving following equation

$$\Delta_{\alpha\theta}\phi_{\alpha\theta,k}(x) = \lambda_{\alpha\theta,k}\phi_{\alpha\theta,k}(x), k = 0, 1, ...$$

Since ALBO is semi-definite positive, it has non-negative eigenvalues $\lambda_{\alpha\theta,k} \geq 0$ and a set of orthogonal eigen-functions, i.e., $\{\phi_{\alpha\theta,k}\}_{k=0}^{\infty}$ satisfying

$$\langle \phi_{\alpha\theta,i}, \phi_{\alpha\theta,j} \rangle_X = \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases}$$

Now, we similarly introduce the Anisotropic convolution operator with Formula (2). Firstly, we build an oriented localized kernel function $g_{\alpha\theta,x}(y)$ centered at point x from the spectral domain. It is generated from the convolution with a Kronecker delta function δ_x using an appropriate filter $\hat{g}(\lambda)$ (How to choose will be detailed discussed in the next section), i.e.

$$g_{\alpha\theta,x}(y) = \sum_{k\geq 0} \hat{g}(\lambda_{\alpha\theta,k})\phi_{\alpha\theta,k}(x)\phi_{\alpha\theta,k}(y). \tag{5}$$

For clearly clarifying, we show a series of such kernels $g_{\alpha\theta,x}(y)$ centered at same point x but with different anisotropic levels α and rotation angles θ in Figure 1. Note that, when $\alpha = 0, \theta = 0$, the kernel is isotropic.

Similarly to the classical convolution, our anisotropic convolution operator is defined as

$$(f * g)_{\alpha\theta}(x) = \int_{y\in X} f(y)g_{\alpha\theta,x}(y)dy, \tag{6}$$

We can see that the kernel $g_{\alpha\theta,x}(y)$ acts as a weighting function that incorporates the local information of signal f along the orientation θ of point x.

Actually, rather than using the spatial representation to perform anisotropic convolution (Formula (6)) which is generally highly computing expensive, we further deduce its spectral manner to overcome this issue. Substituting Formula (5) into (6), we have

$$\begin{aligned}(f*g)_{\alpha\theta}(x) &= \int_{y\in X} f(y)g_{\alpha\theta,x}(y)dy \\ &= \int_{y\in X} f(y)\sum_{k\geq 0}\hat{g}(\lambda_{\alpha\theta,k})\phi_{\alpha\theta,k}(x)\phi_{\alpha\theta,k}(y)dy \\ &= \sum_{k\geq 0}\hat{g}(\lambda_{\alpha\theta,k})\phi_{\alpha\theta,k}(x)\int_{y\in X} f(y)\phi_{\alpha\theta,k}(y)dy \\ &= \sum_{k\geq 0}\hat{g}(\lambda_{\alpha\theta,k})\hat{f}(\lambda_{\alpha\theta,k})\phi_{\alpha\theta,k}(x)\end{aligned} \tag{7}$$

here $\hat{f}(\lambda_{\alpha\theta,k}) = \langle f, \phi_{\alpha\theta,k}\rangle_X$, called as anisotropic Fourier transform.

Such convolution operator incorporates the local information of signal f along the orientation θ of point x. Hence, to comprehensively extract all direction features, we finally define our *anisotropic convolution operator* as

$$(f*g)(x) = \int_0^{2\pi}(f*g)_{\alpha\theta}(x)d\theta \tag{8}$$

Note that, in this work, we just consider single value of the anisotropic lever α. Its multi-applied situation will be our future work. From Formula (7), we can see that the design of filter $\hat{g}(\lambda)$ is the key ingredient in the convolution construction.

4.2. Chebyshev filters

Compared to using fixed anisotropic heat kernels in [7], our work intends to learn task-dependent kernels by learning their parameterized filters $\hat{g}(\lambda)$. In practical applications, the selection of filter $\hat{g}(\lambda)$ is very important w.r.t their localization in space as well as their learning complexity. As stated in [12, 17], these issues can be overcome with the use of a polynomial filter. In our framework, we intend to adopt Chebyshev polynomials to represent filter $\hat{g}(\lambda)$, due to its smooth and stable recurrence relation which will avoid explicitly eigen-decomposition of ALBO and hence greatly reduce the learning complexity [20, 17]. We will detailedly discuss in their discretization.

Given the Chebyshev polynomial $T_s(\lambda)$ of order s, it can be computed by the stable recurrence relation

$$T_s(\lambda) = 2\lambda T_{s-1}(\lambda) - T_{s-2}(\lambda), T_0 = 1, T_1 = \lambda. \tag{9}$$

These polynomials form an orthogonal basis for $L^2([-1,1], d\lambda/\sqrt{1-\lambda^2})$ and this Hilbert space of square integrable functions w.r.t the measure $d\lambda/\sqrt{1-\lambda^2}$. Thus, our filter $\hat{g}(\lambda)$ can be parametrized as the truncated expansion

$$\hat{g}(\lambda) = \sum_{s=0}^{S-1} c_s T_s(\lambda) \tag{10}$$

of order S.

4.3. Discretization

Theoretically, our anisotropic convolution operator can be constructed on any discrete representation of surface provided they have accurate discretized ALBO. In this paper, we concentrate on triangular meshes and other representations can be deduced similarly.

Given a triangular mesh $M(V, E, F)$, where set V includes N vertices, E and F are the sets of edges and triangles, the function f on mesh M is represented as a vector $\mathbf{f} \in \mathbb{R}^N$.

Setting an anisotropic level α and a rotation angle θ, according to [8], the discrete ALBO can be represented as a sparse matrix $\mathbf{L}_{\alpha\theta} = \mathbf{A}^{-1}\mathbf{B}_{\alpha\theta} \in \mathbb{R}^{N\times N}$, where the element of mass matrix $\mathbf{A}$ and stiffness matrix $\mathbf{B}_{\alpha\theta}$ are referred to this paper.

Then, we compute the eigen-decomposition of ALBO as solving a generalized eigen-problem

$$\mathbf{B}_{\alpha\theta}\boldsymbol{\phi}_{\alpha\theta,k} = \lambda_{\alpha\theta,k}\mathbf{A}\boldsymbol{\phi}_{\alpha\theta,k}.$$

Since the eigenvectors $\{\boldsymbol{\phi}_{\alpha\theta,k}\}_{k=0}^{N-1}$ are $\mathbf{A}$-orthogonal, the inner product of two functions $\mathbf{f}$ and $\mathbf{g}$ is discretized as $\langle\mathbf{f},\mathbf{g}\rangle_{\mathbf{A}} = \mathbf{f}^{\mathrm{T}}\mathbf{A}\mathbf{g}$.

Denote matrix $\mathbf{U}_{\alpha\theta} = (\boldsymbol{\phi}_{\alpha\theta,0}, \boldsymbol{\phi}_{\alpha\theta,1}, \cdots, \boldsymbol{\phi}_{\alpha\theta,N-1}) \in \mathbb{R}^{N\times N}$ and $\Lambda_{\alpha\theta} = \mathrm{diag}(\lambda_{\alpha\theta,0}, \lambda_{\alpha\theta,1}, \cdots, \lambda_{\alpha\theta,N-1})$, we have $\mathbf{L}_{\alpha\theta} = \mathbf{U}_{\alpha\theta}\Lambda_{\alpha\theta}\mathbf{U}_{\alpha\theta}^{\mathrm{T}}\mathbf{A}$, and the anisotropic Fourier coefficient vector $\hat{\mathbf{f}} = \mathbf{U}_{\alpha\theta}^{\mathrm{T}}\mathbf{A}\mathbf{f}$.

Now, we give the discretization of the anisotropic convolution theorem in Formula (7)

$$(\mathbf{f}*\mathbf{g})_{\alpha\theta} = \mathbf{U}_{\alpha\theta}\hat{g}(\Lambda_{\alpha\theta})\mathbf{U}_{\alpha\theta}^{\mathrm{T}}\mathbf{A}\mathbf{f}.$$

If using a polynomial filter, we can avoid expensive eigen-decomposition in the above convolution operator and get

$$(\mathbf{f}*\mathbf{g})_{\alpha\theta} = \hat{g}(\mathbf{L}_{\alpha\theta})\mathbf{f},$$

which only involves the multiplication of sparse matrixes and vectors.

Specially, for Chebshev polynomial filters (Formula (10)), we should first get a scaled anisotropic Laplacian matrix $\tilde{\mathbf{L}}_{\alpha\theta} = 2\lambda_{N-1}^{-1}\mathbf{L}_{\alpha\theta} - \mathbf{I}$ (here $\mathbf{I}$ is a $N\times N$ identity matrix) whose eigenvalues lie in the interval $[-1, 1]$. Then we have

$$(\mathbf{f}*\mathbf{g})_{\alpha\theta} = \hat{g}(\tilde{\mathbf{L}}_{\alpha\theta})\mathbf{f} = \sum_{s=0}^{S-1} c_{\alpha\theta,s}T_s(\tilde{\mathbf{L}}_{\alpha\theta})\mathbf{f}.$$

From the recurrence relation Formula (9), we know $T_0(\tilde{\mathbf{L}}_{\alpha\theta})\mathbf{f} = \mathbf{f}$, $T_1(\tilde{\mathbf{L}}_{\alpha\theta})\mathbf{f} = \tilde{\mathbf{L}}_{\alpha\theta}\mathbf{f}$, and $T_s(\tilde{\mathbf{L}}_{\alpha\theta})\mathbf{f} = 2\tilde{\mathbf{L}}_{\alpha\theta}T_{s-1}(\tilde{\mathbf{L}}_{\alpha\theta})\mathbf{f} - T_{s-2}(\tilde{\mathbf{L}}_{\alpha\theta})\mathbf{f}$.

After choosing an appropriate value of α, we use L equally-spaced rotation angles, i.e. $\theta = \theta_1, \theta_2, \cdots, \theta_L$, where $\theta_l = 2(l-1)\pi/L$, $l = 1, 2, \cdots, L$, then compute the convolution along each angle, i.e. $(\mathbf{f} * \mathbf{g})_{\alpha\theta}$. Finally we compute our anisotropic convolution by

$$\mathbf{f} * \mathbf{g} = \sum_{l=1}^{L} (\mathbf{f} * \mathbf{g})_{\alpha\theta_l} = \sum_{l=1}^{L} \sum_{s=0}^{S-1} c_{\alpha\theta_l,s} T_s(\tilde{\mathbf{L}}_{\alpha\theta_l})\mathbf{f}. \quad (11)$$

In the next section, we will design CNN architectures to learn the filter coefficient set $\{c_{\alpha\theta_l,s}\}$ for shape correspondence.

5. Implementation and results

In our framework, shape correspondence refers to the task of labeling each vertex of a given shape to the corresponding vertex of a reference shape.

Datasets. We use four public domain datasets: FAUST[5], SCAPE[3], SHREC'16 Topology[25], SHREC'16 Partial(cuts)[16] to test our performance on shape correspondence. These datasets contain plenty of 3D shapes ranging from human bodies to animals, with non-rigid deformations and parts of them suffering topological changes or missing. All the ground-truth point-wise correspondences between shapes are provided.

Evaluation. We take the Princeton benchmark protocol [22] to evaluate the correspondence quality, which exhibits the percentage of matches that are at most r normalized geodesic distance from the ground-truth correspondence on the reference shape. Note that, we treat the symmetric points of the ground-truth as incorrect correspondence.

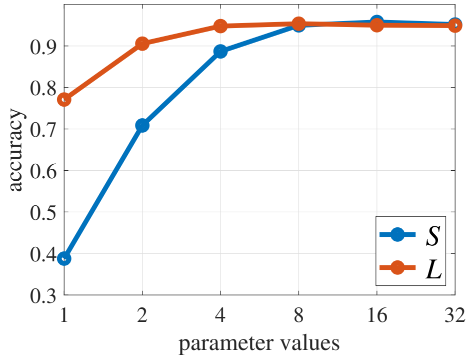


Figure 2. Demonstration of correspondence accuracy ($r = 0$) with different parameter values on FAUST dataset.

Table 1. Performance comparisons on FAUST dataset.

Method	Refinement	Input	Accuracy (r=0)	Accuracy (r=0.01)
GCNN [29]		SHOT	66.61 %	74.98 %
ACNN [7]	FM[36]	SHOT	62.40 %	83.31 %
MoNet [34]	PMF[49]	SHOT	88.20 %	92.35 %
SpiralNet [27]		SHOT	93.06 %	96.32 %
ACSCNN		SHOT	98.06 %	99.26 %
SplineCNN [19]		**1**	99.12 %	99.37 %
ACSCNN		**1**	98.98 %	99.64 %
ACSCNN	PMF[49]	**1**	99.56 %	99.87 %

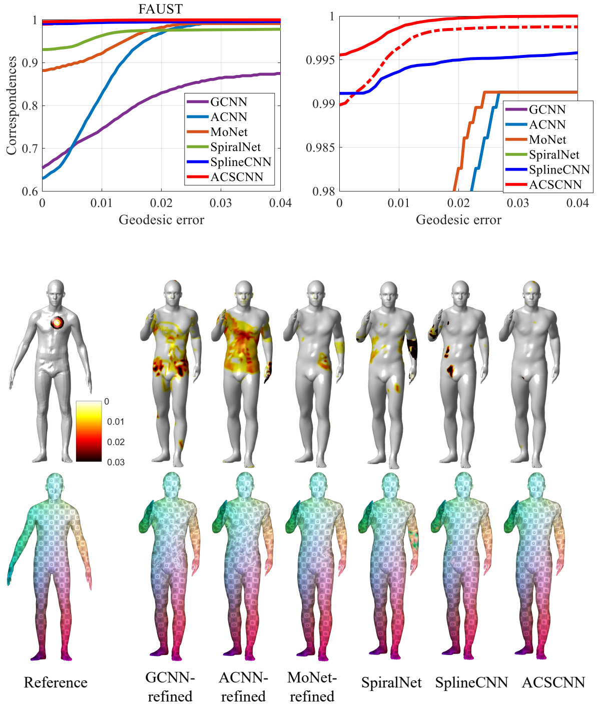


Figure 3. Performance comparisons on FAUST. The top row shows the correspondence accuracy of each method. Dashed lines show the results of our method without refinements. For clarity, the top left parts of these lines have been magnified in the right figure. The middle row shows the visualization of correspondence error, where the hotter colors refer to larger errors. The results of texture and color transfers are shown in the third row.

Settings. According to our anisotropic convolution (Formula (11)), we should determine a set of appropriate values for three parameters, e.g., S (the order of the Chebyshev Polynomials), L (the number of rotation angles) and α (the anisotropic level). Theoretically, the higher order of Chebyshev polynomials can represent more complex filters and more rotation angles can capture more different direction information when α is fixed. Both of them can lead to better performance (Figure 2) and more time consuming meanwhile. We need to make a trade-off between time costs and performance. In all the following experiments, we set the number of rotation angle $L = 8$, anisotropic level $\alpha = 10$ and the order of Chebyshev filters $S = 15$ empirically and use the symmetric normalization laplacian matrix. We use

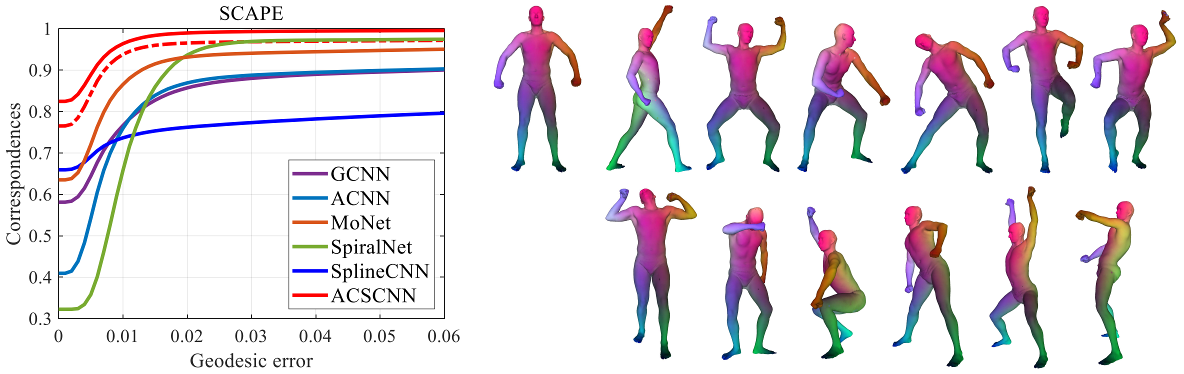


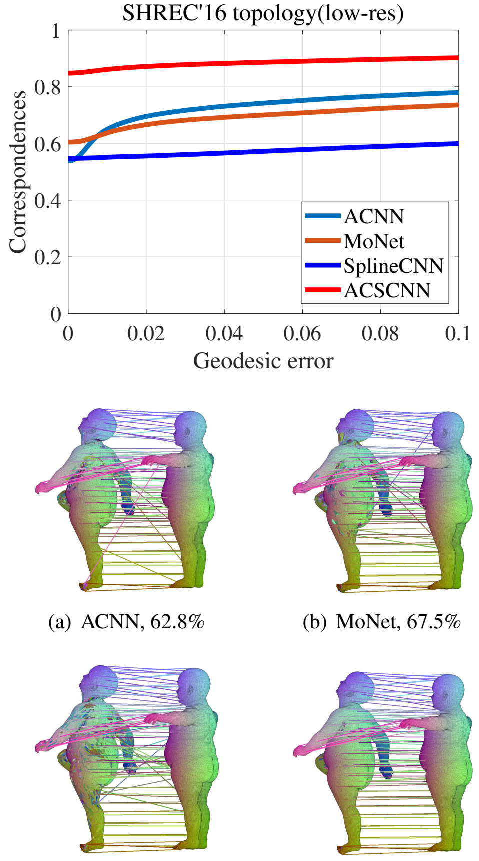
Figure 4. Performance comparisons on SCAPE. Here, the red solid and dotted line respectively show the results of our ACSCNN using SHOT and constant vector **1** as inputs. The right part is the color transfer (corresponding points are shown in the same color) result of our method where the leftmost is the reference shape.

constant vector $\mathbf{1} \in \mathbb{R}^{N\times 1}$ and the network architecture provided by SplineCNN [19] as our default inputs and architecture. For overall comparison, in some experiments we also consider SHOT [42], a 544-dimensional point descriptor as inputs, which widely used by other learning correspondence methods [29, 7, 34, 27], followed by the network architecture provided by ACNN [7] (with fewer convolution layers than SplineCNN).

Results. FAUST dataset is composed of 10 poses of 10 people (100 shapes in total), each having nearly 7K vertices. For this dataset, we use the first 80 shapes for training, the rest 20 shapes for test and the first shape as the reference. Table 1 and Figure 3 detailedly list the performance statistic of our ACSCNN and recent state-of-the-art approaches, using the architectures provided by respective authors. We compare them in different situations, including whether or not using refinements and different inputs.

The top part of Table 1 shows the results of each method using SHOT as inputs. Obviously, our method outperforms all competitors even not followed by refinement. In the second part, we just employ a very simple input, constant vector **1** to test their performance. It can be seen that even using such a rough input, our ACSCNN still achieves very high accuracy.

In Figure 3, we further show their qualitative and quantitative comparisons on this dataset. The results clearly clarify our superiority to other methods. Note that, SplineCNN also gets very desirable performance on this dataset, however, due to its simplification on pseudo-coordinates, the correspondence accuracy will drop significantly on the following more challenging datasets.



(a) ACNN, 62.8%

(b) MoNet, 67.5%

(c) SplineCNN, 66.9%

(d) ACSCNN, 90.2%

Figure 5. Performance comparisons on the SHREC'16 Topology dataset. The top part shows the correspondence accuracy of each method (GCNN and SpiralNet failed), followed by their visualization comparisons in the bottom, where we use straight lines to connect the correspondence point pairs. The number is the correspondence accuracy of each method at $r = 0$ and the shape of T pose is the reference.

SCAPE dataset contains 71 registered meshes from a human with different poses, each mesh has around 12.5K vertices. Randomly select 70 percent of the shapes for training and the rest for testing, the correspondence accuracy is shown in the left of Figure 4. As SpiralNet can not be directly applied on such large meshes, we simplify the ev-

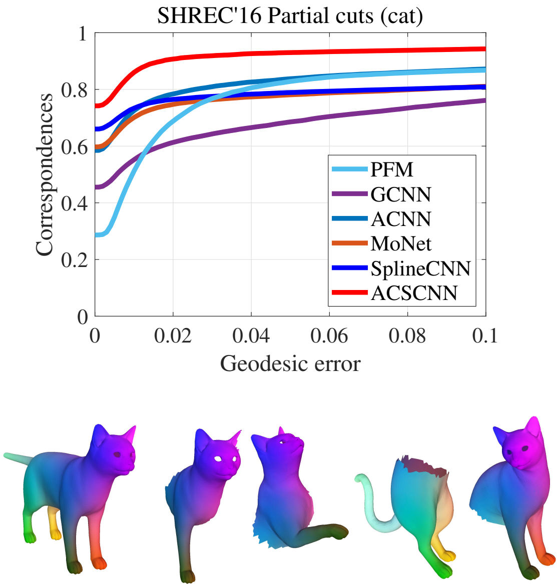


Figure 6. Performance comparisons on SHREC'16 Partial cuts (cat) dataset. The top part shows the correspondence accuracy of each method, the bottom part is the color transfer result of our method where the leftmost is the reference shape

ery original meshes into half number of the vertices for this method. Two input and architecture settings of our method are used in this experiment and the results demonstrate that our method significantly outperforms all other methods under both input settings. Furthermore, the right part of Figure 4 shows the visualization of color transfer with our method on this dataset.

Moreover, to test the robustness of our method, we also build correspondence on SHREC'16 Topology dataset. This challenging dataset contains 25 shapes of a kid with around 12K vertices, undergoing near-isometric deformation in addition to large topological shortcuts. The first 16 shapes are used for training and the left 9 for testing. In this experiment, both GCNN and SpiralNet fail due to the topology noises. The quantitative and visualization comparisons are shown in Figure 5, where our method still achieves a good result and is superior to other methods.

In addition, we also test the correspondence accuracy on SHREC'16 Partial cuts meshes in Figure 6. A total of 45 cat shapes are used, with the first 30 shapes for training and the rest for testing. All of the shapes lose large parts of the bodies. Note that, for this dataset, we additionally compare with the Partial functional Map [40] method, the winner of the SHREC'16 Partial Correspondence contest. Our architecture still significantly outperforms all other competitors.

Complexity. In practice, some steps can be precomputed once for each shape, including the construction of L anisotropic Laplacian matrixes ($N \times N$ sparse matrix, $O(N)$ non-zero elements), normalizing and scaling them. The computation of these precomputation steps costs $O(N^2)$ operations for each Laplacian matrix. Our Chebyshev polynomials based spectral filtering avoids full eigendecomposition of Laplacian matrix firstly, which costs $O(N^3)$ operations. And the adoption of Chebyshev polynomials reduces the spectral filtering from dense matrix multiplication to sparse matrix multiplications. The former costs $O(N^2)$, and the latter only costs $O(L \times S \times N) \ll O(N^2)$. Because L and S are user-defined constant, so $O(L \times S \times N) = O(N)$ holds, which resulted in linear computational complexity.

Timing. All experiments are tested on a PC with Intel(R) Core i7-4790 CPU at 3.6GHz, 16G RAM and Nvidia GeForce RTX 2080 Ti (11G). In the following, the typical timings are for FAUST dataset with $N = 6.9K$ vertices for each shape. The pre-computations took 4.8s for each shape. And it took 27.78s for training (80 shapes) per epoch and 3.80s for testing (20 shapes). So for testing, it only took 0.19s to predict all vertices' labels for each shape.

6. Conclusions

We introduce a novel convolution neural network termed Anisotropic Chebyshev spectral CNNs, based on the anisotropic convolution operator, an extension of manifold convolution operator proposed by us. Our anisotropic convolution operator aggregates local features by a set of oriented patch operators which are defined in the spectral domain by filtering on the eigen-decompositions of multiple anisotropic Laplacian operators with different orientations. Compared to previous works, such type of anisotropic convolution allows to more comprehensively capture the intrinsic local information of signals due to its direction considering. As to reduce the computational complexity, we employ Chebyshev polynomials to represent the filters where the coefficients are trainable. Through extensive benchmark experiments of shape correspondence, we show our architecture can improve state-of-the-art results in several datasets. Being a generic convolutional model, we plan to use ACSCNN to solve many other shape analysis tasks, such as shape segmentation and classification in the future. The Pytorch-based implementation to replicate our results will be released upon the publication.

Acknowledgments

We would acknowledge the anonymous reviewers for their valuable comments. This work was supported by the Hunan Science Fund for Distinguished Young Scholars under Grant 2019JJ20027, the Natural Science Foundation of China under Grant 61572527, 61602524 and the Hunan Provincial Innovation Foundation For Postgraduate under Grant CX20190092.

References

[1] Eman Ahmed, Alexandre Saint, Abd El Rahman Shabayek, Kseniya Cherenkova, Rig Das, Gleb Gusev, Djamila Aouada, and Björn Ottersten. Deep learning advances on different 3d data representations: A survey. *arXiv preprint arXiv:1808.01462*, 1, 2018.

[2] Mathieu Andreux, Emanuele Rodola, Mathieu Aubry, and Daniel Cremers. Anisotropic laplace-beltrami operators for shape analysis. In *European Conference on Computer Vision*, pages 299–312. Springer, 2014.

[3] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: Shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, page 408–416, New York, NY, USA, 2005. Association for Computing Machinery.

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

[5] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. Faust: Dataset and Evaluation for 3D Mesh Registration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014.

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

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

[9] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172, 2006.

[10] Alexander M Bronstein, Michael M Bronstein, Ron Kimmel, Mona Mahmoudi, and Guillermo Sapiro. A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *International Journal of Computer Vision*, 89(2-3):266–286, 2010.

[11] 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.

[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] Qifeng Chen and Vladlen Koltun. Robust nonrigid registration by convex optimization. In *IEEE International Conference on Computer Vision*, pages 2039–2047, 2015.

[14] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[15] Ronald R Coifman, Stephane Lafon, Ann B Lee, Mauro Maggioni, Boaz Nadler, Frederick Warner, and Steven W Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the national academy of sciences*, 102(21):7426–7431, 2005.

[16] Luca Cosmo, Emanuele Rodolà, Michael M Bronstein, Andrea Torsello, Daniel Cremers, and Y Sahillioglu. Shrec'16: Partial matching of deformable shapes. *Proc. 3DOR*, 2(9):12, 2016.

[17] 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.

[18] Asi Elad and Ron Kimmel. On bending invariant signatures for surfaces. *IEEE Transactions on pattern analysis and machine intelligence*, 25(10):1285–1295, 2003.

[19] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 869–877, 2018.

[20] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

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

[22] Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM Transactions on Graphics (TOG)*, 30(4):1–12, 2011.

[23] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[24] Iasonas Kokkinos, Michael M Bronstein, Roee Litman, and Alex M Bronstein. Intrinsic shape context descriptors for deformable shapes. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 159–166. IEEE, 2012.

[25] Z Lähner, Emanuele Rodolà, MM Bronstein, Daniel Cremers, Oliver Burghard, Luca Cosmo, Andreas Dieckmann, Reinhard Klein, and Y Sahillioglu. Shrec'16: Matching of deformable shapes with topological noise. *Proc. 3DOR*, 2:11, 2016.

[26] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

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

[28] 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.

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

[30] Diana Mateus, Radu Horaud, David Knossow, Fabio Cuzzolin, and Edmond Boyer. Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[31] Simone Melzi, Maks Ovsjanikov, Giorgio Roffo, Marco Cristani, and Umberto Castellani. Discrete time evolution process descriptor for shape analysis and matching. *ACM Transactions on Graphics (TOG)*, 37(1):1–18, 2018.

[32] Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Trans. Graph.*, 38(6), Nov. 2019.

[33] Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4):417–487, 2011.

[34] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[35] D. Nogneng, S. Melzi, E. Rodolà, U. Castellani, M. Bronstein, and M. Ovsjanikov. Improved functional mappings via product preservation. *Computer Graphics Forum*, 37(2):179–190, 2018.

[36] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30, 2012.

[37] Helmut Pottmann, Johannes Wallner, Qi-Xing Huang, and Yong-Liang Yang. Integral invariants for robust geometry processing. *Computer Aided Geometric Design*, 26(1):37–60, 2009.

[38] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016.

[39] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG)*, 37(6):1–16, 2018.

[40] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. *Computer Graphics Forum*, 36(1):222–236, 2017.

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

[42] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014.

[43] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

[44] David I Shuman, Benjamin Ricaud, and Pierre Vandergheynst. Vertex-frequency analysis on graphs. *Applied & Computational Harmonic Analysis*, 40(2):260–291, 2016.

[45] Justin Solomon, Andy Nguyen, Adrian Butscher, Mirela Ben-Chen, and Leonidas Guibas. Soft maps between surfaces. *Computer Graphics Forum*, 31(5):1617–1626, 2012.

[46] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009.

[47] Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. Feature correspondence via graph matching: Models and global optimization. In *European Conference on Computer Vision*, pages 596–609. Springer, 2008.

[48] B. Vallet and B. Lévy. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum*, 27(2):251–260, 2008.

[49] Matthias Vestner, Zorah Lähner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodola, Alex Bronstein, Michael Bronstein, Ron Kimmel, et al. Efficient deformable shape correspondence via kernel matching. In *2017 International Conference on 3D Vision (3DV)*, pages 517–526. IEEE, 2017.

[50] Matthias Vestner, Roee Litman, Emanuele Rodolà, Alex Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3327–3336, 2017.

[51] Lingyu Wei, Qixing Huang, Duygu Ceylan, Etienne Vouga, and Hao Li. Dense human body correspondences using convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1544–1553, 2016.

[52] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.