Eurographics Symposium on Geometry Processing 2014
Thomas Funkhouser and Shi-Min Hu
(Guest Editors)

Volume 33 (2014), Number 5

Compressed Manifold Modes for Mesh Processing

T. Neumann[1,2], K. Varanasi[3,4], C. Theobalt[3], M. Magnor[2] and M. Wacker[1]

[1]HTW Dresden, Germany [2]TU Braunschweig, Germany
[3]Max-Planck-Institut für Informatik, Saarbrücken, Germany [4]Technicolor Research, France

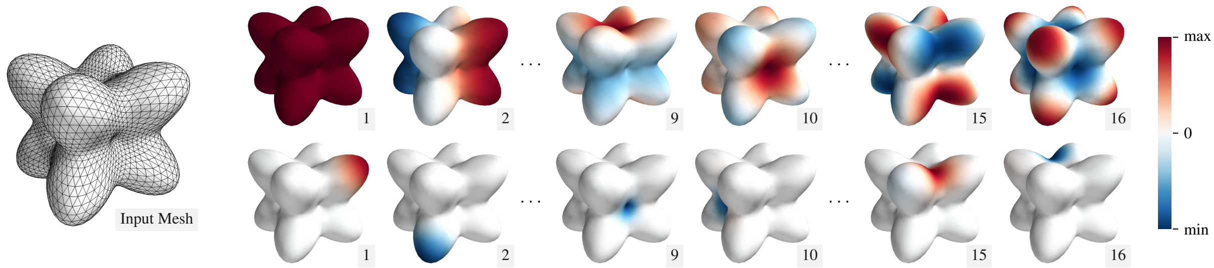


Figure 1: Comparison of different types of eigenfunctions of the Laplace-Beltrami operator of the input mesh (leftmost). *Top Row:* the manifold harmonics have global support. *Bottom Row:* the proposed compressed manifold modes (CMMs) have local support and are confined to specific local features like protrusions and ridges. Here, 8 of the CMMs were found for the 8 protrusions at the corners (1 to 8, only 2 shown here), 6 concentrate at each of the dents (2 shown here), and 12 CMMs automatically form at the valleys between the protrusions.

Abstract

This paper introduces compressed eigenfunctions of the Laplace-Beltrami operator on 3D manifold surfaces. They constitute a novel functional basis, called the compressed manifold basis*, where each function has local support. We derive an algorithm, based on the alternating direction method of multipliers (ADMM), to compute this basis on a given triangulated mesh. We show that compressed manifold modes identify key shape features, yielding an intuitive understanding of the basis for a human observer, where a shape can be processed as a collection of parts. We evaluate compressed manifold modes for potential applications in shape matching and mesh abstraction. Our results show that this basis has distinct advantages over existing alternatives, indicating high potential for a wide range of use-cases in mesh processing.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Hierarchy and geometric transformations

1. Introduction

The eigenfunctions of the Laplace-Beltrami operator of a 3D surface define a basis, known as the manifold harmonic basis (MHB). This basis can be understood as a generalization of the Fourier spectrum for functions defined on a general manifold surface. The Laplace-Beltrami operator captures all the intrinsic properties of the shape and is invariant to extrinsic shape transformations such as isometric deformations. As such, its eigenfunctions constitute a compact and elegant basis for spectral shape processing that is independent of the actual shape representation and parameterization. In fact, the MHB is unique and characteristic of the geometric and topological properties of the shape. Several successful applications have been proposed that took advantage of these desirable properties, for spectral geometry filtering, compression, and surface editing [VL08, BKP*10, LZ10]. The MHB provides an efficient and robust formulation for estimating shape correspondences, e.g. within the functional maps framework [OBCS*12]. Similar to the Fourier spectrum, the manifold harmonics (MHs) have global spatial support; i.e. each function acts on all the vertices of the mesh. This means that the functions are not easily interpretable, they act on all parts of the mesh. For example, the shape of a hand can be understood as the palm and five fingers, which is not directly represented within the MHB basis. The global support of MHs also renders them very sensitive to topological noise due to large holes and occlusions that often appear in scanned meshes. This reduces the utility of the MHB for practical shape matching problems in the real world.

In this paper, we aim to produce an intrinsic shape basis that tackles these defects. The proposed basis inherits the beneficial property of orthogonality and shape-awareness from the MHB. Yet, our new basis functions have local support, where the locality can be easily controlled. We call this basis the *compressed manifold basis (CMB)* and the individual basis functions *compressed manifold modes (CMM)*. In summary, our contributions are:

1. We extend the theory of compressed eigenfunctions of differential operators, that was recently presented in the context of computational physics [OLCO13], to the setting of general manifold surfaces in computer graphics.
2. We show how to adapt the alternating direction method of multipliers to compute the compressed manifold basis on a triangle mesh, for a given discretization of the Laplace-Beltrami operator.

We show that the compressed manifold modes automatically identify key shape features of the underlying mesh. They automatically group confined local regions like protrusions and ridges into separate basis functions, Fig. 1. Due to their unique spatial locality, they are robust to significant geometric and topological noise, such as what happens due to partial scans. Thus, the CMB can be considered as a tool for robust shape analysis and matching. At the same time, CMB is an orthogonal basis and can reconstruct any function defined on the shape, up to an arbitrary degree of precision. We qualitatively evaluate the CMB towards developing potential applications in shape matching, shape approximation and feature detection. A reference implementation of the algorithm as well as all scripts to reproduce the results in this paper are available at
`https://github.com/tneumann/cmm`

2. Background and Related Work

Before introducing the new compressed manifold basis (CMB), let us first give some background on the manifold harmonic basis (MHB) and on their applications in various areas in computer graphics. We will also review recent results on obtaining compactly supported eigenfunctions.

Manifold Harmonic Basis The Laplace-Beltrami operator Δ on a 2D manifold surface embedded in 3D space induces a set of eigenfunctions ϕ_k which satisfy the classical equations

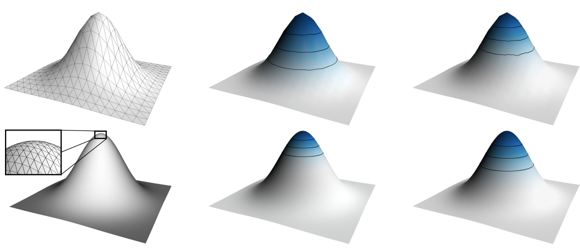
$$-\Delta\phi_k = \lambda_k\phi_k\,, \quad k \in \mathbb{N},\, \lambda_k \in \mathbb{R}\,, \tag{1}$$

where λ_k are the eigenvalues of the operator. The eigenfunctions form a basis that is called the manifold harmonic basis (MHB) [VL08]. In most practical applications, only a set of $K \in \mathbb{N}$ eigenfunctions corresponding to the smallest eigenvalues is required.

Applications Due to their compactness, encoding efficiency, isometric invariance, and computational efficiency, manifold harmonics can be found in countless applications in computer graphics [LZ10]. Vallet and Lévy [VL08] present several mesh filtering applications and introduced an algorithm that is able to compute the MHB on very large meshes. The recently proposed functional map framework [OBCS*12] uses maps between functions on shapes, for example to transfer segmentations or to estimate correspondences between deformed shapes. Various point signatures based on Laplace-Beltrami eigenfunctions are successfully used in shape matching [Rus07, SOG09]. Other applications include mesh parametrization [MTAD08], shape segmentation [dGGV08] and compression of meshes [KG00]. The MHs can even help in accurate facial tracking [BWP13].

Sparsity-inducing norms in graphics The main mathematical tool used in this paper is sparsity-inducing regularization. This concept has been very popular in compressed sensing and machine learning as well as in image- and signal-processing. Recently, applications in computer graphics appeared. For example, Deng et al. [DBD*13] show local modification of constrained architectural meshes using sparse regularization methods. Neumann et al. [NVW*13] propose a variant of sparse PCA to automatically extract local deformation components from a dataset of meshes, e.g. mesh sequences captured from human facial performances. Pokrass et al. [PBB*13] show that sparse regularization can also help to improve correspondence-estimation between deformed shapes. Rustamov [Rus11] proposes a multiresolution kernel which is centered locally around a specific point on the mesh, also using the sparsity-inducing ℓ_1 norm. Due to a similar sparsity inducing objective, these multi resolution kernels look similar to some of the first compressed modes presented here. However, they look very different for higher number of compressed basis functions K. Another important difference is that the multi-resolution kernels [Rus11] are defined with respect to some given central point in the mesh and do not form a basis.

Compressed Modes Ozoliņš, Lai, and Osher [OLCO13] propose to find *compressed* eigenfunctions of a general differential operator. To this end, they add a sparsity inducing ℓ_1-norm into a variational formulation of a problem of



(a) Input (b) Unweighted (c) Weighted

Figure 2: Our method correctly handles the area-weighted Laplacian for computing compressed eigenfunctions. This is shown for a low-resolution (top row, 512 triangles) and a high-resolution (bottom row, 32000 triangles) mesh. (b) Without area-weighted Laplacian, eigenfunctions are resolution-dependent. (c) Our algorithm gives resolution-independent eigenfunctions.

type (1) . Given a parameter $\mu \in \mathbb{R}^+$ that controls the sparsity, they arrive at

$$\min_{\phi_k} \sum_{k=1}^{K} \langle \phi_k \, , \, \Delta\phi_k \rangle + \mu |\phi_k|_1 \ , \quad \text{s.t. } \langle \phi_k \, , \, \phi_j \rangle = \delta_{kj} \ , \qquad (2)$$

where δ_{kj} is the Kronecker delta, here used to enforce orthogonality of the eigenfunctions. Interestingly, those compressed eigenfunctions were proven to have *compact support* [BCO14]: They are non-zero only in a confined region of the domain. The size of the compact support can be controlled by μ: a large value for μ will result in smaller support regions. Until now, compressed eigenfunctions have been predominantly employed on 1D and 2D domains for applications in physics and partial differential equations [OLCO13]. Our approach extends this work and provides compressed Laplace-Beltrami eigenfunctions on three-dimensional mesh surfaces and for 3D shape processing.

3. Computing Compressed Manifold Modes

As a preliminary, we begin by clarifying how we discretize both the Laplace-Beltrami operator Δ and the eigenvalue equation (1) , as well as the minimization problem (2) . With those building blocks in place, we then derive an algorithm that is able to compute compressed manifold modes by solving (2) .

3.1. Discretization

In this paper, we concentrate on triangle meshes. A very popular discretization of the Laplacian Δ for a triangle mesh with N vertices may be realized as a sparse matrix $\mathbf{L} \in \mathbb{R}^{N\times N}$ with the cotangent weights [MDSB02]. Along with the cotangent weights in $\mathbf{L}$, which only respect angles between edges, we also use the lumped mass matrix $\mathbf{D}$ containing the vertex areas along its diagonal. Notice that our discretization corresponds to the finite element formulation of the Laplace-Beltrami operator [LZ10], which leads to a generalized eigenvalue problem as explained in the following.

With the Laplace-Beltrami operator defined, we discretize the eigenfunctions. The manifold harmonic basis (MHB) is defined by [VL08]

$$-\mathbf{L}\Phi_k = \lambda \mathbf{D}\Phi_k \, , \qquad (3)$$

This can be solved using an off-the-shelf sparse iterative eigensolver or by using the efficient band-by-band computation method presented in [VL08]. Given a mesh with N vertices, we assemble the first K eigenvectors into a matrix $\Phi \in \mathbb{R}^{N\times K}$, where each column Φ_k is one eigenvector.

For computing the compressed manifold basis, we extend the variational formulation (2) from [OLCO13] to be applicable to triangle meshes. Here, we especially have to consider the area-matrix $\mathbf{D}$. Without $\mathbf{D}$, the eigenbasis is not independent of the mesh resolution, as demonstrated in Fig. 2. Our discretization of (2) thus reads

$$\min_{\Phi} \mathrm{Tr}(\Phi^\top \mathbf{L}\,\Phi) + \mu\|\Phi\|_1 \ , \quad \text{s.t. } \Phi^\top \mathbf{D}\,\Phi = \mathbf{I} \, . \qquad (4)$$

The next section explains an efficient algorithm that solves this minimization problem.

3.2. Reformulation using ADMM

In this section, we reformulate the optimization problem (4) so that it can be solved efficiently using the alternating direction method of multipliers (ADMM). ADMM is a mathematical framework for convex optimization, interested readers are referred to the comprehensive article by Boyd et al. [BPC*11]. ADMM can solve problems that involve two functions and constraints,

$$f(\mathbf{x}) + g(\mathbf{z}) \ , \quad \text{s.t. } \mathbf{A}\,\mathbf{x} + \mathbf{B}\,\mathbf{z} = \mathbf{c} \, . \qquad (5)$$

Our optimization problem (4) can be reformulated in the above form (5) . To this end, we replace the orthogonality constraint using an indicator function

$$\iota(\Phi) = \begin{cases} 0 & \text{if } \Phi^\top \mathbf{D}\,\Phi = \mathbf{I} \\ \infty & \text{otherwise} \, . \end{cases} \qquad (6)$$

This lets us transform the optimization objective (4) into a sum of three functions $\mathrm{Tr}(\Phi^\top \mathbf{L}\,\Phi) + \mu\|\Phi\|_1 + \iota(\Phi)$. Since extensions of ADMM for more than two functions are still being heavily researched and not necessarily guaranteed to converge [CHYY13], we adopt a splitting strategy introduced recently [WHML13]. The idea is to use one of the three functions as the "main function" f in (5) , and group the remaining functions into g, cf. (5) . For computing CMMs, we found that choosing ι as the main function works best. We then arrive at the following reformulation of (4)

$$\begin{aligned} \min_{\Phi,\mathbf{S},\mathbf{E}} \ & \iota(\Phi) + \mathrm{Tr}(\mathbf{E}^\top \mathbf{L}\ \mathbf{E}) + \mu\|\,\mathbf{S}\|_1 \ , \\ \text{s.t. } & \Phi = \mathbf{S} \ , \ \Phi = \mathbf{E} \, . \end{aligned} \qquad (7)$$

Notice how the two separate coupling constraints force the variable Φ to be equal to $\mathbf{S}$ and $\mathbf{E}$. If those constraints are fulfilled exactly then we arrive again at (4) . The equivalence to the standard formulation of ADMM (5) can be seen by first substituting $\mathbf{x} = \Phi$ and $f = \iota$. Variable $\mathbf{z}$ and function g are then block-separable with

$$g\left(\begin{bmatrix}\mathbf{E}\\ \mathbf{S}\end{bmatrix}\right) = \begin{bmatrix}\mathrm{Tr}(\mathbf{E}^\top \mathbf{L}\ \mathbf{E})\\ \mu\|\mathbf{S}\|_1\end{bmatrix} . \tag{8}$$

Finally, the constraint matrices from (5) may be written as

$$\mathbf{A} = \begin{bmatrix}\mathbf{I}\\ \mathbf{I}\end{bmatrix} , \quad \mathbf{B} = \begin{bmatrix}-\mathbf{I} & 0\\ 0 & -\mathbf{I}\end{bmatrix} , \quad \mathbf{c} = \mathbf{0} . \tag{9}$$

3.3. Numerical Algorithm

Rephrasing of our original minimization problem (4) into (7) enables us to apply the ADMM method [BPC*11, Section 3.1.1]. To this end, we introduce the dual variable $\mathbf{U} \in \mathbb{R}^{2N\times K}$ which consists of two blocks $\mathbf{U} = [\mathbf{U}_E; \mathbf{U}_S]$, corresponding to two auxiliary variables $\mathbf{E}$ and $\mathbf{S}$. Given an initial guess for Φ, we set $\mathbf{E} = \mathbf{S} = \Phi$ and $\mathbf{U} = \mathbf{0}$. The algorithm then comes down to iterating the following steps

$$\Phi \leftarrow \arg\min_{\Phi}\ \iota(\Phi) + \frac{\rho}{2}\left\|\begin{bmatrix}\Phi\\ \Phi\end{bmatrix} - \begin{bmatrix}\mathbf{E}\\ \mathbf{S}\end{bmatrix} + \begin{bmatrix}\mathbf{U}_E\\ \mathbf{U}_S\end{bmatrix}\right\|_2^2 \tag{10}$$

$$\mathbf{E} \leftarrow \arg\min_{\mathbf{E}}\ \mathrm{Tr}(\mathbf{E}^\top \mathbf{L}\ \mathbf{E}) + \frac{\rho}{2}\|\Phi - \mathbf{E} + \mathbf{U}_E\|_2^2 \tag{11}$$

$$\mathbf{S} \leftarrow \arg\min_{\mathbf{S}}\ \mu\|\mathbf{S}\|_1 + \frac{\rho}{2}\|\Phi - \mathbf{S} + \mathbf{U}_S\|_2^2 \tag{12}$$

$$\mathbf{U} \leftarrow \mathbf{U} + \begin{bmatrix}\Phi\\ \Phi\end{bmatrix} - \begin{bmatrix}\mathbf{E}\\ \mathbf{S}\end{bmatrix} . \tag{13}$$

The variable ρ is called the penalty parameter. Usually, updates (11) and (12) are done in a single step involving $g(\mathbf{z})$ in (5) , but due to the independent blocks in (8) they can be separated.

We now explain how the individual update steps, which are small optimization problems themselves, can be performed efficiently. The objective of the minimization (10) can be transformed into $\iota(\Phi) + \frac{\rho}{2}\|\Phi - \frac{1}{2}(\mathbf{S}+\mathbf{E})\|_2^2 - 2\,\mathrm{Tr}(\mathbf{S}+\mathbf{E}) + \frac{1}{2}\|\mathbf{S}+\mathbf{E}\|_2^2$. Ignoring terms that don't change the minimum allows reduction of (10) to

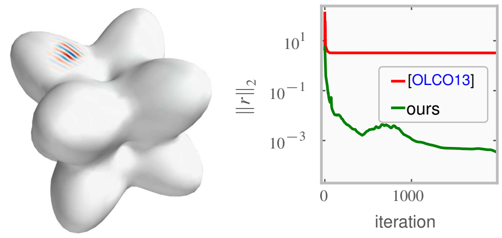
$$\Phi \leftarrow \arg\min_{\Phi}\|\Phi - \mathbf{Y}\|_2^2 \quad \text{s.t. } \Phi^\top \mathbf{D}\,\Phi = \mathbf{I} , \tag{14}$$

with $\mathbf{Y} = \frac{1}{2}(\mathbf{S} - \mathbf{U}_S + \mathbf{E} - \mathbf{U}_E)$. Since $\mathbf{D}$ contains the vertex area weights, it is a positive diagonal matrix and trivial to invert. We can then substitute $\Phi = \mathbf{D}^{-1/2}\Psi$ to yield

$$\Phi \leftarrow \mathbf{D}^{-1/2}\left(\arg\min_{\Psi\ ,\ \Psi^\top\Psi=\mathbf{I}}\ \|\mathbf{D}^{-1/2}\Psi - \mathbf{Y}\|_2^2\right) . \tag{15}$$

Based on the SVD factorization of $(\mathbf{D}^{1/2}\mathbf{Y})^\top(\mathbf{D}^{1/2}\mathbf{Y}) = \mathbf{V}\mathbf{W}\mathbf{V}^\top$, a closed-form solution is found as

$$\Phi \leftarrow \mathbf{D}^{-1/2}\left(\mathbf{Y}\mathbf{V}\mathbf{W}^{-1/2}\mathbf{V}^\top\right) . \tag{16}$$



(a) Φ_{20} with [OLCO13] (b) primal residual

Figure 3: Convergence of our method (green) vs the direct extension of [OLCO13] to 3D meshes (red). Some of the eigenfunctions computed by [OLCO13] show oscillations as visible in (a), which do not appear with our method (compare with our result in Fig. 1). (b) Plotting the primal residual across the iterations shows that [OLCO13] obtains infeasible solutions while our method converges.

A proof of the last step appears in various sources, e.g. in [LO14]. Thus, step (10) requires an SVD of a $K\times K$ matrix - the number of vertices N is irrelevant (usually $K \ll N$).

To solve the minimization problem (11) , we set its derivative to zero, which gives a closed form solution

$$\left(\rho\,\mathbf{I} - \mathbf{L} - \mathbf{L}^\top\right)\mathbf{E} = \rho\,(\Phi + \mathbf{U}_E) , \tag{17}$$

so we have to solve K very sparse linear systems in each iteration. By prefactorizing $(\rho\,\mathbf{I} - \mathbf{L} - \mathbf{L}^\top)$, e.g. using Cholesky factorization, this step can be significantly accelerated. Updating the factorization is only necessary when ρ changes.

Finally, the third step of updating $\mathbf{S}$ in (12) is separable for each entry $\mathbf{S}_{ij}$. It has a simple closed-form solution that may be expressed concisely using the proximal operator (denoted **prox**) of the ℓ_1 norm [BJMO12],

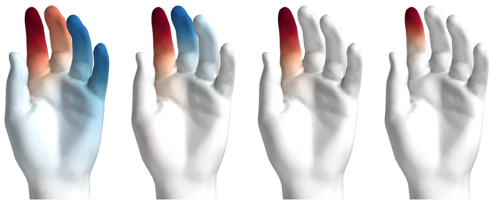
$$\mathbf{S}_{ij} \leftarrow \mathbf{prox}_{\frac{\mu}{\rho}\|\cdot\|_1}(v) = \mathrm{sgn}(v)\,\max\left(|v| - \frac{\mu}{\rho},\ 0\right) , \tag{18}$$

where we substitute $v = \Phi_{ij} + (\mathbf{U}_S)_{ij}$ for brevity.

3.4. Convergence

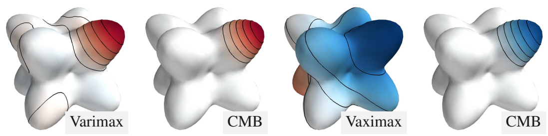
ADMM is guaranteed to converge for a convex f and g. However, the minimization task in (4) contains the *non-convex* constraint of orthogonality of the eigenvectors, so convergence to a global minimum cannot be guaranteed. Nevertheless, the proposed method finds local minima which are suitable for practical applications.

To monitor convergence we use the primal residual $\|r\|_2$ and the dual residual $\|s\|_2$, cf. [BPC*11]. When those values fall below a numerical threshold, our method stops. To automatically set the parameter ρ, our method adopts the adjustment strategy discussed in [BPC*11, Section 3.4.1]. This means that the only parameters for our method are μ which



(a) $\mu = 0$ (b) $\mu = \frac{1}{100} \cdot N$ (c) $\mu = 1 \cdot N$ (d) $\mu = 10 \cdot N$

Figure 4: The parameter μ controls the locality of the CMMs. A large μ will give small local support. N is the number of vertices in the mesh.



(a) $K = 26$ (b) $K = 26$ (c) $K = 4$ (d) $K = 4$

Figure 5: (a) Varimax manifold harmonics show residual oscillations on the whole shape as indicated by the black isolines. (b) In contrast, the proposed CMMs are truly sparse and zero almost on the whole shape. This effect is amplified when only $K = 4$ instead of $K = 26$ basis functions are computed: in this case, (c) the Varimax harmonics show global support while in (d) the proposed CMMs are still sparse.

(a) initialization (b) iteration 50 (c) iteration 200

Figure 6: For low K, our method may converge to different local minima depending on the initialization. Here, each row shows computation of a basis with one compressed mode ($K = 1$). (a) random or user-given initialization, (b) compressed mode after 50 and (c) after 200 iterations.

sets the sparsity and controls the locality, and K which is the number of CMMs to compute. We also compare the convergence of our method with our reimplementation of the algorithm in [OLCO13] for 3D meshes. We set $\mathbf{D} = \mathbf{I}$, because [OLCO13] cannot handle arbitrary $\mathbf{D}$, and run the algorithm on the mesh from Fig. 1. In this example only our algorithm is able to find a feasible solution, as can be seen in Fig. 3. We believe that this behavior is caused mainly by our splitting strategy that groups the two convex functions together and selects the non-convex ι as "main function" f.

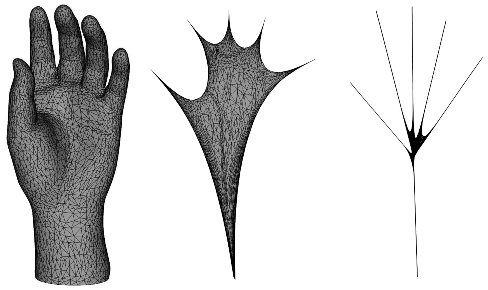
4. Experiments and Results

Comparison to Varimax Varimax (or Orthomax) finds a unitary transformation of the eigenspace that aims to localize the basis, through maximizing the second order moments. An example application in computer graphics can be found in [SBCBG11]. Applying the Varimax method on the MHB indeed localizes the functions, but global oscillations remain on the mesh: The eigenfunctions are often not exactly zero and are thus not sparse (compare Fig. 5a to Fig. 5b). Also, for small K the locality of Varimax diminishes completely (Fig. 5c) while the CMMs are always local (Fig. 5d).

Influence of Varying Sparsity The locality of the CMMs can be controlled by varying the parameter μ, which is not possible with Varimax. This is demonstrated in Fig. 4: large μ will result in smaller local support. To additionally make the parameter μ invariant to the number of vertices N we usually multiply it by N.

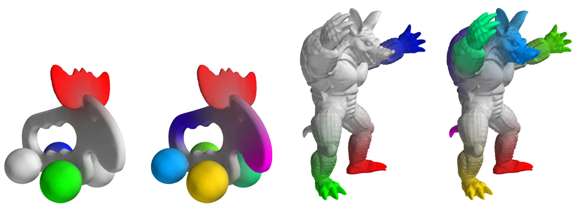
Initialization Our method can be initialized with a random Φ. We typically observe that the algorithm converges to the same set of basis functions, although their ordering might be different. For example, we always found CMMs for the five fingers and the palm if we compute six CMMs on a hand. Similarly, the set of eigenfunctions in Fig. 1 were always the same, but their ordering was different because some CMMs have eigenvalues that are close to each other. When K is very small, e.g. $K = 1$, we observe that different initializations lead to different local minima, see Fig. 6. Instead of giving random initialization, we can also give a user-drawn scribble as initialization, which might be useful for certain applications. Other initialization strategies are also possible, for example in Fig. 8 we used Varimax, because we wanted to have reproducible results even though K is quite low.

Shape Approximation Since both MHs and CMMs form a basis, it is possible to use them for encoding the actual mesh coordinates. This process was described for MHs in [VL08]. Specifically, we have the transformation to "fre-



(a) Input Mesh (b) MH reconstr. (c) CMM reconstr.

Figure 7: Reconstruction of vertex positions from a small number of $K = 6$ (b) manifold harmonics and (c) our proposed compressed manifold modes. See text for details.

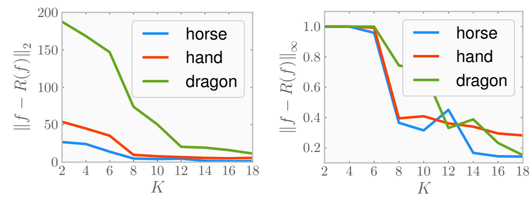


(a) $K = 3$ (b) $K = 7$ (c) $K = 3$ (d) $K = 7$

Figure 8: Soft segmentation using CMMs for two meshes and different number of CMMs K. Protrusions and geodesic extremities are segmented, grey areas are not covered by any of the first CMMs.

mesh	basis	K	μ	error	size	time
hand (868)	MHB	10	5	9.34	68 KB	0.03 s
	CMB	10	-	14.39	43 KB	3 s
	MHB	30	5	4.00	203 KB	0.08 s
	CMB	30	-	3.97	156 KB	29 s
fertility (4994)	MHB	10	5	45.02	390 KB	0.18 s
	CMB	10	-	17.54	216 KB	44 s
	MHB	40	5	7.92	1.5 MB	0.39 s
	CMB	40	-	6.93	933 KB	560 s
bunny (34834)	MHB	10	1.25	36.89	2.7 MB	1.4 s
	CMB	10	-	121.28	1.7 MB	419 s
	MHB	40	1.25	20.49	10.6 MB	3.5 s
	CMB	40	-	20.68	8.2 MB	1357 s

Table 1: Quantitative comparison of the proposed compressed manifold basis (CMB) to the manifold harmonic basis (MHB). The number of vertices N is given in brackets below the mesh name. See text for further details.



(a) reconstruction error (b) coverage

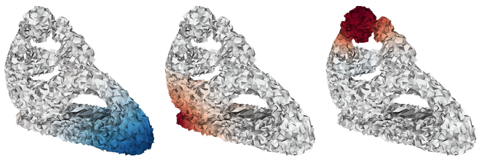
Figure 9: Reconstruction error of a constant function on the mesh for varying K on three different meshes. (a) shows that with increasing K the function can be better reproduced and (b) measures ℓ_∞ norm of that error, a value below 1 means that all vertices are covered by at least one CMM.

quency space" by $\hat{\mathbf{x}} = \mathbf{D}\Phi^\top \mathbf{x}$, and its inverse transformation by $\mathbf{x} = \Phi\hat{\mathbf{x}}$. Here, $\mathbf{x}$ are the x, y and z coordinates of the vertices. For $K \ll N$, applying the transform followed by its inverse gives an approximation of the mesh. Comparing those approximations can help better understanding the properties of the CMB compared to the MHB. MHs show better compression guarantees and retain the rough shape of the whole mesh (Fig. 7b). When K is increased, the MH approximation will move all the vertices towards the input mesh. In contrast, the CMM approximation is more abstract and almost looks like a skeleton of the mesh (Fig. 7c). Each time K is increased, a local change is done to the approximated mesh by changing only the vertices of a local feature, for example by placing the finger tip. Another instructive example is given in Fig. 8. Here, two things can be clearly seen: First, the CMMs automatically form at high-curvature regions and topological protrusions of the mesh. To comply with the sparsity regularization, the first K CMMs ignore flat regions or regions connecting the geodesic extremities.

Time and Space Complexity for Reconstruction In Table 1 we quantitatively evaluate the CMB in comparison to the MHB for different number of components K and for different meshes. The reconstruction error of the vertex coordinates (ℓ_2 norm of the difference between reconstruction and ground truth) approaches that of the MHB when the number of basis functions K goes up. It is also interesting to compare the required memory size for the MHB, stored as a dense array (64bit double), and to the CMB Φ stored as a sparse matrix in CSR format. The CMB is smaller since Φ contains many zeros. The runtimes in Table 1 were measured on a Linux system with Intel i7 3.4 GHz CPU and 16 GB RAM. We compare our unoptimized Python/NumPy implementation for computing the CMB versus the highly optimized ARPACK eigensolver to compute the MHB.

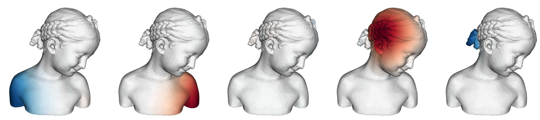
Full Support CMMs are sparse and local, thus it cannot be guaranteed that all the vertices are covered by at least one CMM. In Fig. 9 we empirically show that with an increasing number of compressed modes K, more and more vertices are covered. To this end, we compute the CMM basis on the mesh (with varying K) and reproduce the constant function $f = 1$ on the mesh. The closest least-squares reconstruc-

(a) no noise

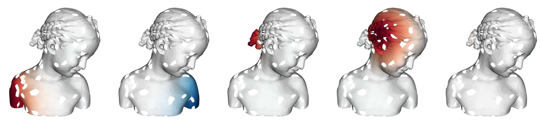


(b) 50% gaussian noise on vertex positions

Figure 10: Insensitivity to noise. (a) 3 out of $K = 6$ CMMs and (b) 3 CMMs after adding gaussian noise to the vertex positions of the mesh. Up to sign flip, the CMM basis functions align well between the original and the noisy mesh.



(a) original mesh with $K = 5$ CMMs



(b) $K = 5$ CMMs computed after adding 200 small holes

Figure 11: Up to ordering and sign flip, the CMMs align even after adding topological noise in form of small holes. Notice that one basis function is located on the backside of the mesh, 3rd image in (a) and 5th image in (b).

tion of f can be quickly computed by $R(f,\Phi) = \Phi \mathbf{D} \Phi^\top f$. We then measure the reconstruction error using the ℓ_2 norm, $\|f - R(f,\Phi)\|_2$, which as expected decreases as the number of modes K increases, Fig. 9a. To check if all the vertices are covered, we measure the maximum error, in other words the ℓ_∞ norm of the error, $\|f - R(f,\Phi)\|_\infty$. If this is 1, then this means that there is some vertex on the mesh that is not covered. As can be seen in 9b the vertices are completely covered already for relatively small K. Note that this also depends on μ: a CMB with small local support requires more basis functions to cover every vertex on the mesh.

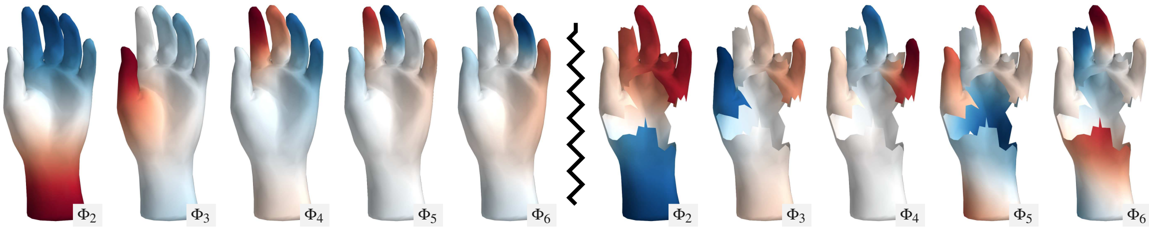
Insensitivity to noise Since the MH basis is based on the intrinsic Laplace operator, it is robust to various kinds of noise. We now empirically show that the CMM basis functions share this property with the MH basis. In Fig. 10, we added gaussian noise with $\sigma = 50\%$ of the average edge length - the CMMs align up to sign flip. In another experiment, we randomly removed vertices as well as their neighboring triangles to observe how the CMB reacts to changes in topology. In Fig. 11 one can see that up to ordering and sign flip, the CMMs align well. In both experiments, we observed that the ordering aligns much more robust when a good initialization is used. Therefore we used Varimax for initialization here. We will discuss insensitivity to partial meshes and to isometric deformations in the following section.

5. Applications in Shape Matching

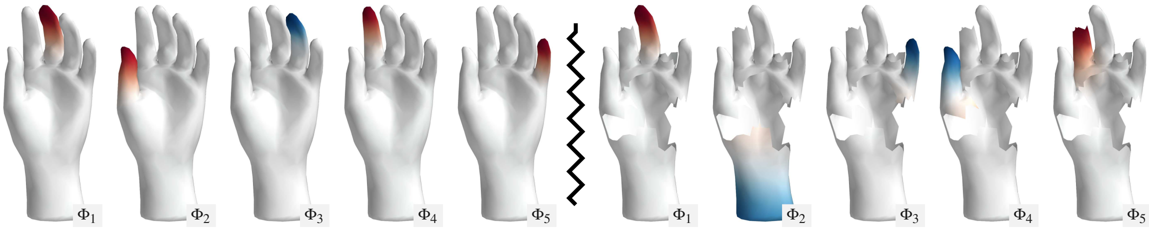
Typically, shape matching first involves robust feature detection and matching, and then a geometry aware regularization that extends this correspondence to the entire shape. For practical shape correspondence, both these stages have to be robust to geometric and topological surface noise that occurs inevitably in real world scanning systems. Occlusions and partial scans make this problem even more challenging.

Ovsjanikov et al. [OBCS*12] use the manifold harmonic basis to propose the elegant framework of functional maps. Often, the manifold harmonic basis between two shapes do not exactly correspond to each other. The functional map still gives a full correspondence of the two shapes even in this case, but only if the sparse input correspondences (point or region features) that are required to construct the map are correct. On the other hand, if the shapes are related by an isometry, the manifold harmonics align well. Then, the functional map yields a diagonal correspondence matrix (or at least diagonal after permutation). Pokrass et al. [PBB*13] use this idea for regularizing the functional map to be sparse and close to a diagonal. However, this is limited to situations where the manifold harmonics can be aligned between the shapes. One important case where this assumption fails is with large holes and severe topological noise due to partial scans. We show an example in Fig. 12 where the mesh of a hand is badly corrupted by holes. “Dense” harmonics of the MHB are severely damaged by these holes and no longer align with the original mesh. However, up to a sign flip and ordering, CMMs align robustly with those of the original mesh. At the same time, the CMMs align very well in the case of purely isometric deformations, which is demonstrated in Fig. 13. Notice that in both examples, the same set of CMMs is recovered even though completely different random initializations were used for the different meshes.

Because of their spatial locality and robustness, compressed manifold modes can be used as shape features in the first stage of shape matching. But since the CMB provides an orthogonal basis that is invariant to isometric deformations, they can also be used in the second stage of shape-aware regularization. In this paper, we tested this second aspect by using the CMB to replace the MHB in the functional map framework. In the discrete setting the functional correspon-



(a) 5 MH's computed separately on the hand (left) and the partial hand (right)



(b) 5 of the proposed CMMs computed separately on the hand (left) and the partial hand (right)

Figure 12: Basis functions of a hand mesh (left side) and a partial shape of the hand emulating typical scanning artifacts (right side). (a) The MHB basis functions do not correspond very well between partial meshes. (b) For every CMB on the full hand, left, we can find a CMM on the partial hand, right. One exception is the ring finger which is completely missing on the right side. Thus, up to sign flip and ordering, the CMMs align very well.

dence between two shapes, S_1, resp. S_2, can be given as a simple $K \times K$ matrix $\mathbf{C}$. If both shapes are equipped with the basis then we have matrices $\Phi_{(S1)}$ and correspondingly $\Phi_{(S2)}$. Those can be seen as K-dimensional point clouds (in eigenspace), and the matrix $\mathbf{C}$ aligns those point clouds by $\Phi_{(S2)} = \Phi_{(S1)}\mathbf{C}^{\top}$. Thus, $\Phi_{(S1)}\mathbf{C}^{\top}$ is a rotated basis such that the basis vectors (columns) overlap with those in $\Phi_{(S2)}$. A necessary condition for $\mathbf{C}$ to be an area-preserving vertex-to-vertex map is that it must be orthogonal [OBCS*12, Theorem 5.1]. To illustrate CMMs in a shape matching scenario, we take two shapes with ground-truth correspondence differing by isometric deformations. Those given correspondences are then used to find the optimal orthogonal $\mathbf{C}$ for different number of basis functions K. We then measure how much error (in geodesic distance) the resulting map between the basis functions introduces, depending on the number of eigenfunctions K. This is exactly the same experiment as performed in [OBCS*12, Fig. 3], which we apply to the new CMB. Here, we use the unweighted Laplacian for both. We find that, despite the CMB not being as information-dense as the MHB, it achieves similar geodesic error with a fairly low number of basis functions K. Around 35 compressed modes are good enough to meet the accuracy of dense harmonics, Fig. 14. More interesting is the phenomenon that the correspondence matrix $\mathbf{C}$ is much sparser for the CMB, compare Fig. 14b to Fig. 14c. Notice that at no point during the estimation of $\mathbf{C}$ we enforced this sparsity. A direct implication of this is that the matrix $\mathbf{C}$ for the CMB is closer to a permutation. For visualization, we choose the permutation that yields a diagonalized correspondence matrix, similar to Pokrass et al [PBB*13]. This is a trivial step and we show the final diagonalized correspondence matrix. For matching partial shapes, this effect is even more apparent, see Fig. 14, middle and bottom row. The respective correspondence matrix $\mathbf{C}$ with CMB is again much closer to a permutation matrix, whereas that of the MHB shows more confusion and is much denser. We think this is a strong indicator that CMMs can be extremely useful for shape matching that is robust against large holes or even for partial shape matching.

6. Discussion and Future Work

It is possible to use CMMs as skinning weights for mesh editing applications. But it is desirable to produce skinning weights that are positive and that sum up to one for all the vertices on the shape [JBK*12]. In contrast to bounded biharmonic weights [JBPS11], CMMs cannot meet those requirements and do not fulfill exact interpolation constraints. But while bounded biharmonic weights also achieve some level of sparsity, they require the user to provide a set of point constraints or a skeleton rig. This is not needed for CMMs. The CMB automatically provides meaningful "areas" that can be used to restrain certain edits. CMMs can regularize sparse input positional constraints for applications such as posing a shape and key-framing an animation, or tracking a deformable object in video. Additionally, the CMB can be used for projecting certain deformation ener-

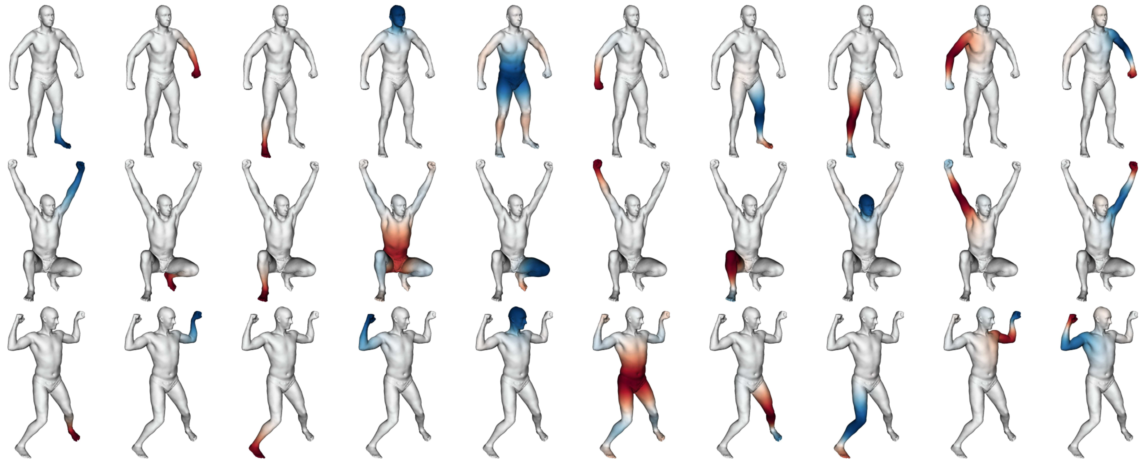


Figure 13: Compressed manifold modes are insensitive to isometric deformations, up to ordering and sign flip. Each row shows $K = 10$ CMMs on a different mesh from the SCAPE database [ASK*05].

gies, thereby reducing their dimensionality and thus computationally simplifying and regularizing the deformation process. These ideas as well as the connection with bounded biharmonic weights [JBPS11] for skinning deformation would be worth exploring in the future.

The Laplace-Beltrami eigenfunctions are related to the heat kernel operator for heat diffusion on the mesh surface. This heat kernel has several applications, particularly in shape matching [SOG09]. Investigating the effect of sparsity and ℓ_1 minimization in the heat diffusion framework was partly done in [Rus11]. CMMs can help in further investigations in the future. Local support was proven for compressed modes in the plane [BCO14], we showed empirically that this also seems to be the case on manifolds. Proving theoretically that the obtained functions are connected on the manifold is a related question open for future research.

A final point we would like to discuss is the non-convexity of our objective function in (2) which leads to a certain dependence on the initialization. [LLO14] recently proposed a convex relaxation of (2) . However, this approach involves optimizing over an $N \times N$ matrix, which means that it is infeasible for meshes with large number of vertices N. So this is still an avenue for future work.

7. Conclusion

In this paper, we introduced compressed manifold modes (CMMs) on mesh surfaces, a novel orthogonal basis for general manifold surfaces in 3D. The compressed basis functions have local support around key shape features that are automatically detected. Our paper presented a complete mathematical derivation of a numerical algorithm that extracts this basis from a given 3D mesh. We empirically demonstrated desirable properties of the CMMs, such as full support and robustness to noise and partial scans. We qualitatively evaluated the CMMs for potential applications such as shape matching and shape approximation. Our results indicate high potential for a wider range of applications in the future.

References

[ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: SCAPE: shape completion and animation of people. *ACM Trans. Graph. (Proc. SIGGRAPH) 24*, 3 (2005). 9

[BCO14] BAREKAT F., CAFLISCH R., OSHER S.: On the support of compressed modes. *UCLA CAM Report 14-14* (2 2014). 3, 9

[BJMO12] BACH F. R., JENATTON R., MAIRAL J., OBOZINSKI G.: Optimization with sparsity-inducing penalties. *Found. Trends Mach. Learn. 4*, 1 (2012). 4

[BKP*10] BOTSCH M., KOBBELT L., PAULY M., ALLIEZ P., UNO LEVY B.: Smoothing. In *Polygon Mesh Processing*. AK Peters, 2010. 2

[BPC*11] BOYD S., PARIKH N., CHU E., PELEATO B., ECKSTEIN J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn. 3*, 1 (2011). 3, 4

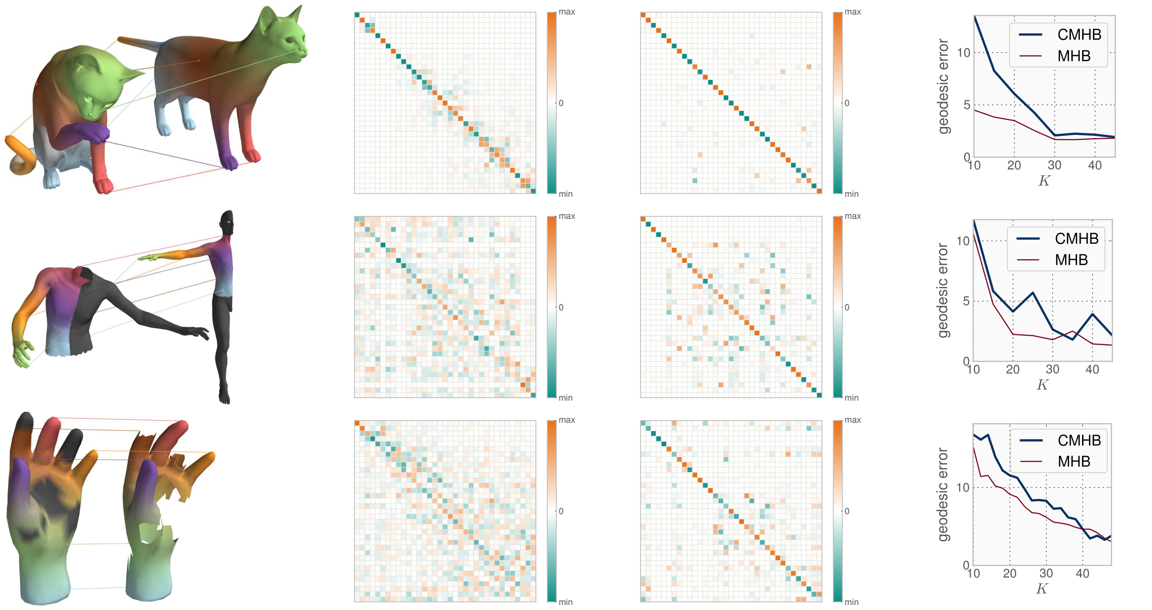
[BWP13] BOUAZIZ S., WANG Y., PAULY M.: Online modeling for realtime facial animation. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 4 (July 2013). 2

[CHYY13] CHEN C., HE B., YE Y., YUAN X.: The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. *preprint* (2013). 3

[DBD*13] DENG B., BOUAZIZ S., DEUSS M., ZHANG J., SCHWARTZBURG Y., PAULY M.: Exploring local modifications for constrained meshes. *Comp. Graph. Forum (Proc. EG) 32*, 2 (2013). 2

[dGGV08] DE GOES F., GOLDENSTEIN S., VELHO L.: A hierarchical segmentation of articulated bodies. In *Proc. Symp. Geom. Proc.* (2008). 2

[JBK*12] JACOBSON A., BARAN I., KAVAN L., POPOVIĆ J., SORKINE O.: Fast automatic skinning transformations. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4 (2012). 8



(a) Input Shapes (b) **C** from MHB (c) **C** from CMB (d) Functional Map Error

Figure 14: Given two meshes and ground truth vertex-to-vertex correspondence, column (a), we compute the functional representation of this map between the two shapes. Column (b) shows the functional map between the MHB. In column (c), the functional map between the CMMs clearly shows much more sparsity. In (d) we plot the mapping error of the maps between MHB and CMB. With increasing number of basis functions K the error of the CMB approaches that of the MHB. Notice how the effect of sparsity in the correspondence matrix is retained even in cases of partial correspondences between meshes with huge holes, middle and bottom row.

[JBPS11] Jacobson A., Baran I., Popović J., Sorkine O.: Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph. (Proc. SIGGRAPH) 30*, 4 (2011), 78:1–78:8. 8, 9

[KG00] Karni Z., Gotsman C.: Spectral compression of mesh geometry. In *Proc. SIGGRAPH* (2000). 2

[LLO14] Lai R., Lu J., Osher S.: Density matrix minimization with ℓ_1 regularization. *ArXiv e-prints* (Mar. 2014). 9

[LO14] Lai R., Osher S.: A splitting method for orthogonality constrained problems. *J. Sci. Comput. 58*, 2 (2014). 4

[LZ10] Levy B., Zhang R. H.: Spectral geometry processing. In *ACM SIGGRAPH Course Notes* (2010). 2, 3

[MDSB02] Meyer M., Desbrun M., Schröder P., Barr A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*. 2002. 3

[MTAD08] Mullen P., Tong Y., Alliez P., Desbrun M.: Spectral conformal parameterization. *Comp. Graph. Forum (Symp. Geom. Proc.) 27*, 5 (2008). 2

[NVW*13] Neumann T., Varanasi K., Wenger S., Wacker M., Magnor M., Theobalt C.: Sparse localized deformation components. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 32*, 6 (Nov. 2013). 2

[OBCS*12] Ovsjanikov M., Ben-Chen M., Solomon J., Butscher A., Guibas L.: Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4 (July 2012). 2, 7, 8

[OLCO13] Ozoliņš V., Lai R., Caflisch R., Osher S.: Compressed Modes for Variational Problems in Mathematics and Physics. *Proc. Natl. Acad. Sci.* (2013). in press. 2, 3, 4, 5

[PBB*13] Pokrass J., Bronstein A. M., Bronstein M. M., Sprechmann P., Sapiro G.: Sparse modeling of intrinsic correspondences. *Comp. Graph. Forum (Proc. EG) 32*, 2 (2013). 2, 7, 8

[Rus07] Rustamov R. M.: Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proc. Symp. Geom. Proc.* (2007). 2

[Rus11] Rustamov R. M.: Multiscale biharmonic kernels. *Comp. Graph. Forum 30*, 5 (2011). 2, 9

[SBCBG11] Solomon J., Ben-Chen M., Butscher A., Guibas L.: Discovery of intrinsic primitives on triangle meshes. *Comp. Graph. Forum (Proc. EG) 30*, 2 (2011). 5

[SOG09] Sun J., Ovsjanikov M., Guibas L.: A concise and provably informative multi-scale signature based on heat diffusion. *Comp. Graph. Forum (Symp. Geom. Proc.) 28*, 5 (2009). 2, 9

[VL08] Vallet B., Lévy B.: Spectral geometry processing with manifold harmonics. *Computer Graphics Forum 27*, 2 (2008). 2, 3, 5

[WHML13] Wang X., Hong M., Ma S., Luo Z.-Q.: Solving Multiple-Block Separable Convex Minimization Problems Using Two-Block Alternating Direction Method of Multipliers. *ArXiv e-prints* (Aug. 2013). 3