EUROGRAPHICS 2008 / G. Drettakis and R. Scopigno
(Guest Editors)

Volume 27 (2008), Number 2

Spectral Geometry Processing with Manifold Harmonics

B. Vallet[1] and B. Lévy[1]

[1]ALICE team, INRIA

Abstract

We present an explicit method to compute a generalization of the Fourier Transform on a mesh. It is well known that the eigenfunctions of the Laplace Beltrami operator (Manifold Harmonics) define a function basis allowing for such a transform. However, computing even just a few eigenvectors is out of reach for meshes with more than a few thousand vertices, and storing these eigenvectors is prohibitive for large meshes. To overcome these limitations, we propose a band-by-band spectrum computation algorithm and an out-of-core implementation that can compute thousands of eigenvectors for meshes with up to a million vertices. We also propose a limited-memory filtering algorithm, that does not need to store the eigenvectors. Using this latter algorithm, specific frequency bands can be filtered, without needing to compute the entire spectrum. Finally, we demonstrate some applications of our method to interactive convolution geometry filtering. These technical achievements are supported by a solid yet simple theoretic framework based on Discrete Exterior Calculus (DEC). In particular, the issues of symmetry and discretization of the operator are considered with great care.

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

1. Introduction

3D scanning technology easily produces computer representations from real objects. However, the acquired geometry often presents some noise that needs to be filtered out. More generally, it may be suitable to enhance some details while removing other ones, depending on their sizes (spatial frequencies). In his seminal paper, Taubin [Tau95] showed that the formalism of signal processing can be successfully applied to geometry processing. His approach is based on the similarity between the eigenvectors of the graph Laplacian and the basis functions used in the discrete Fourier transform. This Fourier function basis enables a given signal to be decomposed into a sum of sine waves of increasing frequencies. He used this analogy as a theoretical tool to design and analyse approximations of low-pass filters. Several variants of this approach were then suggested, as discussed below.

In this paper, instead of only using Fourier analysis as a theoretical tool to analyse approximations of filters, our idea is to explicitly implement its generalization to surfaces of arbitrary topology, and use this to achieve interactive general convolution filtering. Our processing pipeline is similar to the one described in [PG01] for point sets. The main difference is that no resampling nor segmentation is needed: the Fourier transform is computed *directly* on the mesh. The pipeline is outlined in Figure 1:

- A: given a triangulated mesh with n vertices, compute a function basis $H^k, k = 1 \dots m$ that we call the *Manifold Harmonics Basis* (MHB). The k^{th} element of the MHB is a piecewise linear function given by its values H_i^k at vertices i of the surface;
- B: once the MHB is computed, transform the geometry into frequency space by computing the Manifold Harmonic Transform (MHT) of the geometry, that is to say three vectors of coefficients $[\tilde{x}_1, \tilde{x}_2, \dots \tilde{x}_m]$, $[\tilde{y}_1, \tilde{y}_2, \dots \tilde{y}_m]$, and $[\tilde{z}_1, \tilde{z}_2, \dots \tilde{z}_m]$.
- C: apply a frequency space filter $F(\omega)$ by multiplying each $(\tilde{x}_k, \tilde{y}_k, \tilde{z}_k)$ by $F(\omega_k)$, where ω_k denotes the frequency associated with H^k;
- D: finally, transform the object back into geometric space by applying the inverse MHT.

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

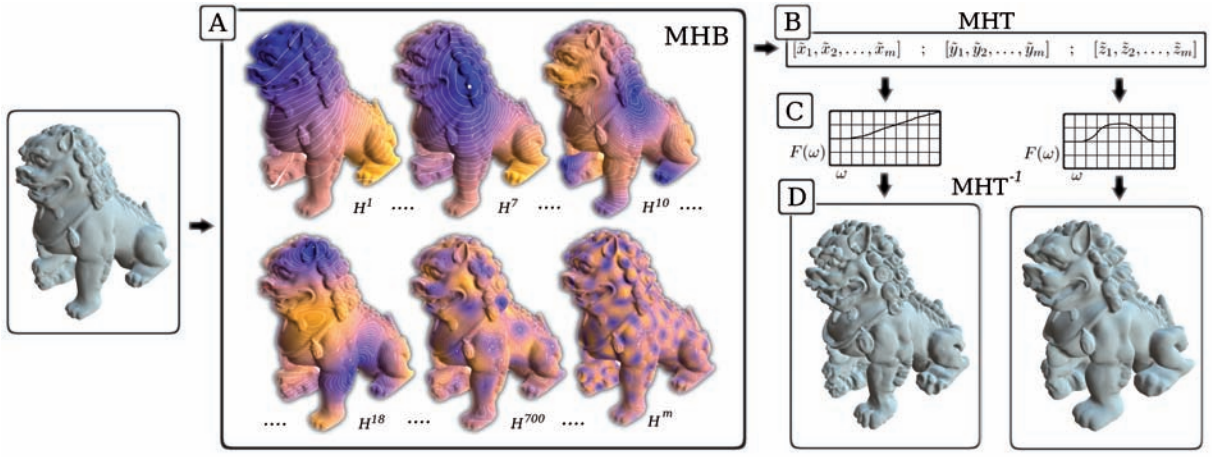

Figure 1: *Processing pipeline of our method: A: Compute the Manifold Harmonic Basis (MHB) of the input triangulated mesh. B: Transform the geometry into frequency space by computing the Manifold Harmonic Transform (MHT). C: Apply the frequency space filter on the transformed geometry. D: Transform back into geometric space by computing the inverse MHT.*

Note that this approach shares some similarities with signal processing with spherical harmonics or with the Discrete Cosine Transform. The main difference is that our approach does not require any resampling of the original surface. However, our MHB depends on the surface, and therefore requires to be precomputed (step A). Once the MHB is known, the subsequent stages of the pipeline can be very efficiently computed. This allows the solution to be interactively updated when the $F(\omega)$ filter is modified by the user.

Contributions

The main contribution of this paper is an efficient numerical mechanism to compute the eigenfunctions of the Laplacian that overcomes the current limits (thousands vertices) by several orders of magnitude (up to a million vertices). The eigenfunctions are computed band by band based on spectral transforms and an efficient eigensolver, and an out-of-core implementation that can compute thousands of eigenvectors for meshes with up to a million vertices. We also propose a limited-memory filtering algorithm, that does not need to store the eigenvectors. Once computed, the MHB allows for interactive filtering on meshes which we demonstrate applied to geometry. These computations use a clean, rigorous yet simple DEC-based framework which clarifies the issues of weighting and symmetry of the discrete cotangent Laplacian. In particular, the symmetry guaranties that the eigenfunctions form an orthonormal "Manifold Harmonics Basis" (MHB) with positive eigenvalues.

Previous Work

The discretization of the Laplacian operator plays a central role in geometry processing and has been extensively studied, motivated by the large number of its applications, that include parameterization, remeshing, compression, reconstruction and minimal surfaces. The eigenfunctions of the Laplacian are known to define a Fourier-like function basis [ZvKD07], that can be used to define spectral analysis on manifolds. We will now review some of the most significant definitions and applications of discrete Laplacians, focusing mainly on the works related with spectral analysis and geometric filtering.

Combinatorial Laplacians: A combinatorial Laplacian solely depends on the connectivity of the mesh. Spectral analysis of the combinatorial graph Laplacian was first used by Taubin [Tau95] to approximate low pass filters. A basis of eigenfunctions of the graph Laplacian was used by Karni *et al.* [KG00] for geometry compression. Zhang [Zha04] studies several variants of combinatorial laplacians and their properties for spectral geometry processing and JPEG-like mesh compression. However, as pointed-out in [MDSB03], the analogy between the graph Laplacian and the discrete cosine transform supposes a uniform sampling of the mesh. Moreover, different embeddings of the same graph yield the same eigenfunctions, and two different meshings of the same object yield different eigenfunctions which is problematic as shown in Figure 2.

Geometric Laplacians and applications: More geometry can be injected into the definition of a discrete Laplacian through the ubiquitous cotan weights [PP93, MDSB03]. These weights can also be derived from Finite Element Modeling (FEM) such as done in [WBH*07], and they converge to the continuous Laplacian under certain conditions as explained in [HPW06] and [AFW06]. Reuter *et al.* [RWP05] also use FEM to compute the spectrum (i.e. the eigenval-

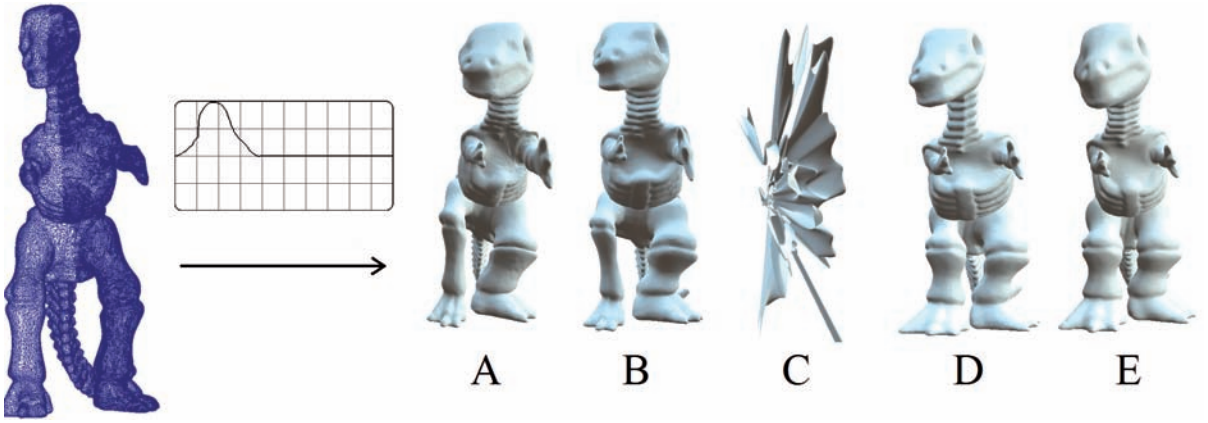

Figure 2: *Filtering an irregularly sampled surface (twice denser on the right half) with different discrete laplacians. Combinatorial Laplacian (A), unweighted cotan* $cot(\beta_{ij}) + cot(\beta'_{ij})$ *(B) and symmetrized weighted cotan* $(cot(\beta_{ij}) + cot(\beta'_{ij}))/(A_i + A_j)$ *(D) produce deformed results (right leg is bigger). Weighted cotan* $(cot(\beta_{ij}) + cot(\beta'_{ij}))/A_i$ *are not symmetric which does not allow for correct reconstruction (C). Only our symmetric weights* $(cot(\beta_{ij}) + cot(\beta'_{ij}))/\sqrt{A_i A_j}$ *are fully mesh-independent (E).*

ues) of a mesh, which provides a signature for shape classification. The cotan weights were also used in [DBG*06] to compute an eigenfunction to steer a quad-remeshing process. The cotan weights alone are still dependent on the sampling as shown in Figure 2-B, so they are usually weighted by the one ring or dual cell area of each vertex, which makes them loose their symmetry. As a consequence, they are improper for spectral analysis (2-C). An empirical symmetrization was proposed in [Lev06] (see Figure 2-D). We clarify these issues based on a rigorous DEC formulation, and recover symmetry by expressing the operator in a proper basis. This ensures that its eigenfunctions are both geometry aware and orthogonal (Figure 2-E). Note that a recent important proof [WMKG07] states that a "perfect" discrete Laplacian that satisfies all the properties of the continuous one cannot exist on general meshes. This explains the large number of definitions for a discrete Laplacian, depending on the desired properties.

Laplacian based Geometric Filtering: A Laplacian-based geometric filtering method was recently proposed [KR05] and was applied to a wide class of filters (e.g. band-exaggeration). This method combines explicit and implicit schemes to reach the different frequency bands involved in the filter. In comparison, our method can use arbitrary user-defined filters, and offers in addition the possibility of changing the filter interactively.

Geometric filtering through energy minimization: Other approaches to geometric filtering are based on energy minimization (e.g. [Mal92]). These methods are called *discrete fairing* in [Kob97, KCVS98], in reference to their continuous-setting counterparts [BW90]. Recently, a method was proposed [NISA06] to optimize both inner fairness (triangle shapes) and outer fairness (surface smoothness), by using a combination of the combinatorial Laplacian and the discrete Laplace-Beltrami operator.

Spectral transform: To directly implement the spectral transform on manifolds, several methods consist in putting the input surface in one-to-one correspondence with a simpler domain [ZBS04], or to partition it into a set of simpler domains [LSS*98, PG01] on which it is easier to define a frequency space. Note that these methods generally need to resample the geometry, with the exception of Mousa *et al.* [MCA06] who directly compute the Spherical Harmonic Transform of a star-shaped mesh. It is also possible to extract the frequencies from a progressive mesh [LSS*98] and avoid resampling the geometry by using irregular subdivision [GSS99]. Finally, an original approach is presented in [SCOIT05] that computes "geometry aware" basis functions, defined as solutions of some least-squares problems. Our method computes the frequency-space basis functions as eigenvectors of a symmetric matrix, for a surface of arbitrary topology without needing any resampling nor segmentation.

The rest of the paper is organized as follows. We first recall some notions on the Fourier Analysis (Section 2.1). A Manifold Harmonics Basis (MHB) is then built through a DEC formulation, and its relations with the classical discrete Laplacian is explained (Section 2.2). Equipped with this new tool, it is then simple to generalize spectral analysis and Fourier transform to arbitrary manifolds. This defines the Manifold Harmonics Transform (MHT) that transforms from geometric space into frequency space, and the inverse MHT (Section 3). We finally explain how to efficiently compute the MHB in practice, and how to implement scalable

spectral geometry processing (Section 4). We conclude by presenting some applications and results.

2. Spectral Analysis on Manifolds

Manifold harmonics (also called *shape harmonics*) are defined as the eigenfunctions of the Laplace operator. This section starts by recalling the familiar Fourier analysis to justify the choice of the eigenfunctions of the Laplace operator to generalize this setting to arbitrary manifolds. We will then define the Laplace operator in the DEC setting and justify a symmetric weighting ensuring the orthogonality of the MHB.

2.1. Fourier Analysis

Given a square-integrable periodic function $f : x \in [0,1] \mapsto f(x)$, or a function f defined on a closed curve parameterized by normalized arclength, it is well known that f can be expanded into an infinite series of sines and cosines of increasing frequencies:

$$f(x) = \sum_{k=1}^{\infty} \tilde{f}_k H^k(x) \quad ; \quad \left\{ \begin{array}{lcl} H^{2k} & = & \sin(kx) \\ H^{2k+1} & = & \cos(kx) \end{array} \right. \tag{1}$$

where the coefficients $\tilde{f}_k$ of the decomposition are given by:

$$\tilde{f}_k = < f, H^k > = \int_0^1 f(x) H^k(x) dx \tag{2}$$

and where $< .,. >$ denotes the inner product (i.e. the "dot product" for functions defined on $[0,1]$). The "Circle harmonics" basis H^k is orthonormal with respect to $< .,. >$: $< H^k, H^k > = 1$, $< H^k, H^l > = 0$ if $k \neq l$.

The set of coefficients $\tilde{f}_k$ (Equation 2) is called the Fourier Transform (FT) of the function f. Given the coefficients $\tilde{f}_k$, the function f can be reconstructed by applying the inverse Fourier Transform FT^{-1} (Equation 1). Our goal is now to generalize these notions to arbitrary manifolds. To do so, we can consider the functions H^k of the Fourier basis as the eigenfunctions of $-\partial^2/\partial x^2$: the eigenfunctions H^{2k} (resp. H^{2k+1}) are associated with the eigenvalues k^2:

$$-\frac{\partial^2 H^{2k}(x)}{\partial x^2} = k^2 \sin(kx) = k^2 H^{2k}(x)$$

To extended this construction to arbitrary manifolds, we need to generalize both inner product $< .,. >$ and second-order derivative $\partial^2/\partial x^2$ to arbitrary manifolds. We now explain how Discrete Exterior Calculus (DEC) helps doing so.

2.2. DEC for spectral analysis

For a complete introduction to DEC we refer the reader to [DKT05], [Hir03] and to [AFW06] for proofs of convergence. We quickly introduce the few notions and notations that we are using to define the inner product $< .,. >$ and generalized second-order derivative (i.e. Laplacian operator).

A k-simplex s_k is the geometric span of $k+1$ points. For instance, 0, 1 and 2-simplices are points, edges and triangles respectively. In our context, a mesh can be defined as a 2-dimensional simplicial complex S, i.e. a collection of n_k k-simplices ($k = 0,1,2$), with some conditions to make it manifold. A discrete k-form ω^k on S is given by a real value $\omega^k(s_k)$ associated with each oriented k-simplex (that corresponds to the integral of a smooth k-form over the k-simplex). The set $\Omega^k(S)$ of k-forms on S is a vector space of dimension n_k. With a proper numbering of the k-simplices, ω^k can be assimilated to a vector of size n_k, and linear operators from $\Omega^k(S)$ to $\Omega^l(S)$ can be assimilated to (n_k, n_l) matrices.

The *exterior derivative* $d_k : \Omega^k(S) \rightarrow \Omega^{k+1}(S)$ is defined by the signed adjacency matrix: $(d_k)_{s_k, s_{k+1}} = \pm 1$ if s_k belongs to the boundary of s_{k+1}, with the sign depending on their respective orientations.

DEC provides $\Omega^k(S)$ with a L_2 *inner product*:

$$< \omega_1^k, \omega_2^k > = (\omega_1^k)^T \star_k \omega_2^k \tag{3}$$

where $\star_k$ is the so-called *Hodge star*. As a matrix, the Hodge star is diagonal with elements $|s_k^*|/|s_k|$ where s_k^* denotes the circumcentric dual of simplex s_k, and $|.|$ is the simplex volume. In particular, for *v*ertices, *e*dges and *t*riangles:

$$(\star_0)_{vv} = |v^*| \; ; \; (\star_1)_{ee} = \frac{|e^*|}{|e|} = \cot\beta_e + \cot\beta'_e \; ;$$

where β_e and β'_e denote the two angles opposite to e.

Finally the *Laplace de Rham* operator on 0-forms is given by: $\Delta = -\star_0^{-1} d_1^T \star_1 d_0$ and its coefficients are:

$$\Delta_{ij} = -\frac{\cot(\beta_{ij}) + \cot(\beta'_{ij})}{|v_i^*|} \quad ; \quad \Delta_{ii} = -\sum_j \Delta_{ij}$$

For surfaces with borders, if the edge ij is on the border, the term $\cot(\beta'_{ij})$ vanishes and the dual cell v_i^* is cropped by the border. This matches the FEM formulation (see [HL88] or [VL07]) with Neumann boundary conditions, with the difference that FEM has a mass matrix (corresponding to $\star_0$) which is not diagonal and which terms consist of one ring areas (whereas $\star_0$ contains dual cell areas). This comes from the fact that DEC is based on a mixed Finite Elements/Finite Volumes scheme where the elements are the same as in FEM but the volumes are the dual cells. This difference has litte impact, especially as the mass matrix is often diagonalized (lumped mass approximation) in FEM formulations.

Remark: The matrix Δ corresponds to the standard discrete Laplacian, except for the sign. The sign difference comes from the distinction between Laplace-Beltrami and Laplace de Rham operators.

The so-defined Laplacian Δ apparently looses the *symmetry* of its continuous counterpart ($\Delta_{ij} \neq \Delta_{ji}$). This makes the

Figure 3: *Some functions of the Manifold Harmonic Basis (MHB). Note the similarity with the sine products used by the DCT.*

eigenfunction basis no longer orthonormal, which is problematic for our spectral processing purposes (Figure 2-C). To recover symmetry, consider the canonical basis (ϕ_i) of 0-forms: $\phi_i = 1$ on vertex i and $\phi_i = 0$ on other vertices. This basis is orthogonal but not normal with respect to the inner product defined in Equation 3 ($< \phi_i, \phi_i >= (\phi_i)^T \star_0 \phi_i \neq 1$). However, since the Hodge star $\star_0$ is a diagonal matrix, one can easily normalize (ϕ_i) as follows:

$$\bar{\phi}_i = \star_0^{-1/2} \phi_i$$

In this orthonormalized basis $(\bar{\phi}_i)$, the Laplacian $\bar{\Delta}$ is symmetric, and its coefficients are given by:

$$\bar{\Delta} = \star_0^{-1/2} \Delta \star_0^{-1/2} \quad ; \quad \bar{\Delta}_{ij} = -\frac{\cot \beta_{ij} + \cot \beta'_{ij}}{\sqrt{|v_i^*||v_j^*|}} \tag{4}$$

2.3. The Manifold Harmonic Basis (MHB)

Given these definitions, we can now compute the MHB, defined to be the set of eigenvectors of $\bar{\Delta}$ expressed in the canonical basis:

1. Assemble the discrete Laplacian $\bar{\Delta}$ (see Equation 4);
2. Compute its eigenvectors $(\bar{H}^k)$ (see Section 4);
3. Map them into the canonical basis: $(H^k) = (\star_0^{-1/2} \bar{H}^k)$.

The set of so-obtained (H^k) vectors is called the Manifold Harmonics basis (MHB). An example is shown in Figure 3. Note that by construction, the MHB is orthonormal with respect to the inner product. This property is important to define the associated transforms, as explained below.

3. The Manifold Harmonic Transform

Transforming 0-forms between the canonical basis (ϕ^k) *i.e., geometric space* and the MHB (H^k) *i.e., frequency space* will be called the Manifold Harmonic Transform (MHT). It is also easy to define the inverse transform (MHT^{-1}). They generalize the notion of Fourier transform to simplicial complexes. We now derive the expressions of MHT and MHT^{-1}, and apply them to geometry filtering.

3.1. MHT and inverse MHT

We consider the geometry x (resp. y,z) of the simplicial complex S, that can be seen as a piecewise linear function. In other words, x is a linear combination of the canonical basis functions ϕ^i:

$$x = \sum_{i=1}^{n} x_i \phi^i \tag{5}$$

where x_i denotes the x coordinate at vertex i. Computing the MHT means projecting the function x into the MHB, i.e. finding the coefficients $\tilde{x}_k$ such that:

$$x = \sum_{k=1}^{m} \tilde{x}_k H^k \tag{6}$$

Since the MHB is orthonormal, we can easily project (5) and (6) onto each $H^{k'}$ with the inner product:

$$\begin{array}{rcl} < x, H^{k'} > & = & \sum_{i=1}^{n} x_i < \phi^i, H^{k'} >= x^T \star_0 H^{k'} \\ < x, H^{k'} > & = & \sum_{k=1}^{m} \tilde{x}_k < H^k, H^{k'} >= \tilde{x}_{k'} \end{array}$$

By equating both expressions, we finally obtain the expression of $\tilde{x}_k$:

$$\tilde{x}_k = x^T \star_0 H^k = \sum_{i=1}^{n} x_i (\star_0)_{ii} H_i^k \tag{7}$$

Using these equations, we know how to transform 0-forms between geometric space (x) and frequency space $(\tilde{x})$. This defines the MHT and inverse MHT:

- MHT: $(x_i) \rightarrow (\tilde{x}_k)$ (Equation 7)
- MHT^{-1}: $(\tilde{x}_k) \rightarrow (x_i)$ (Equation 6)

Figure 4 shows the geometry reconstructed from the MHT of a surface using a different number m of MHT coefficients. As can be seen, the first H^k functions capture the general shape of the functions and the next ones correspond to the details.

3.2. Filtering

Once the geometry is converted into the MHB, each component $(\tilde{x}_k, \tilde{y}_k, \tilde{z}_k)$ of the MHT corresponds to an individual spatial frequency ω_k. In the case of a closed curve (Section

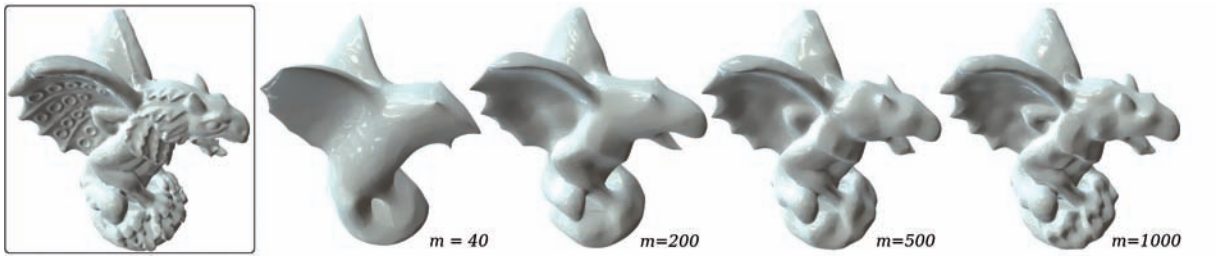

Figure 4: *Reconstructions obtained with an increasing number of MH functions.*

2.1), we have $-\partial^2 \sin(\omega x)/\partial x^2 = \omega^2 \sin(\omega x)$. Therefore, the spatial frequency ω_k corresponds to the square root of the eigenvalue λ_k. In the case of a manifold, similar results can be obtained, that connect $\omega_k = \sqrt{\lambda_k}$ with the length of the nodal sets [DF88].

A frequency-space filter is a function $F(\omega)$ that gives the amplification to apply to each spatial frequency ω. Since all frequencies are separated by the MHT, applying a filter $F(\omega)$ to the geometry becomes a simple product in frequency space. The filtered coordinate x_i^F (resp y_i^F, z_i^F) at vertex i is then given by:

$$x_i^F = \sum_{k=1}^{m} F(\omega_k)\tilde{x}_k H_i^k = \sum_{k=1}^{m} F(\sqrt{\lambda_k})\tilde{x}_k H_i^k$$

In practice, we stop computing the MHB at a given "cutoff" frequency $\omega_m = \sqrt{\lambda_m}$ (inverse of 10 times the average edge length in our experiments). Therefore, smaller geometric details are not represented in the MHT. However, it is possible to keep track of them, by storing in each vertex the difference x_i^{hf} (resp. y_i^{hf}, z_i^{hf}) between the original geometry and its projection onto the MHB:

$$x_i^{hf} = \sum_{k=m+1}^{\infty} \tilde{x}_k H_i^k = x_i - \sum_{k=1}^{m} \tilde{x}_k H_i^k$$

The frequency-space filter can be applied to the high-frequency components of the signal, by re-injecting them into the inverse MHT, as follows:

$$x_i^F = \sum_{k=1}^{m} F(\omega_k)\tilde{x}_k H_i^k + f^{hf} x^{hf} \tag{8}$$

In this equation, the term f^{hf} denotes the average value of the filter F on $[\omega_m, \omega_M]$, where ω_M denotes the maximum (Nyquist) frequency of the mesh (half the inverted edge length). The high-frequency component behaves like a wave packet that can be filtered as a whole, but that cannot be considered as independent frequencies.

Figure 5 demonstrates low-pass, enhancement and band-exaggeration filters. Note that unlike existing methods, ours does not need to re-scale the model. Moreover, only the filtered inverse MHT (Equation 8) depends on the filter F. As a consequence, after storing the MHB and the MHT coefficients, the solution can be updated interactively when the user changes the filter F.

We now proceed to explain how to compute the coefficients H_i^k of the MHB and the associated eigenvalues λ_k.

4. Numerical Solution Mechanism

Computing the MHB means solving for the eigenvalues λ_k and eigenvectors $\bar{H}^k$ for the symmetric positive semi-definite matrix $\bar{\Delta}$:

$$\bar{\Delta}\bar{H}^k = \lambda_k \bar{H}^k \tag{9}$$

However, eigenvalues and eigenvectors computations are known to be extremely computationally intensive. To reduce the computation time, Karni *et al.* [KG00] partition the mesh into smaller charts, and [DBG*06] use multiresolution techniques. In our case, we need to compute multiple eigenvectors (typically a few thousands). This is known to be currently impossible for meshes with more than a few thousand vertices [WK05]. In this section, we show how this limit can be overcome by several orders of magnitude.

To compute the solutions of large sparse eigenproblems, several iterative algorithms exist. The publicly available library ARPACK (used in [DBG*06]) provides an efficient implementation of the Arnoldi method. Yet, two characteristics of eigenproblem solvers hinder us from using them directly to compute the MHB for surfaces with more than a few thousand vertices:

- first of all, we are interested in the lower frequencies, i.e. eigenvectors with associated eigenvalues lower than ω_m^2. Unfortunately, iterative solvers perform much better for the other end of the spectrum. This can be explained in terms of filtering as lower frequencies correspond to higher powers of the smoothing kernel, which may have a poor condition number;
- secondly, we need to compute a large number of eigenvectors (typically a thousand), and it is well known that computation time is superlinear in the number of requested

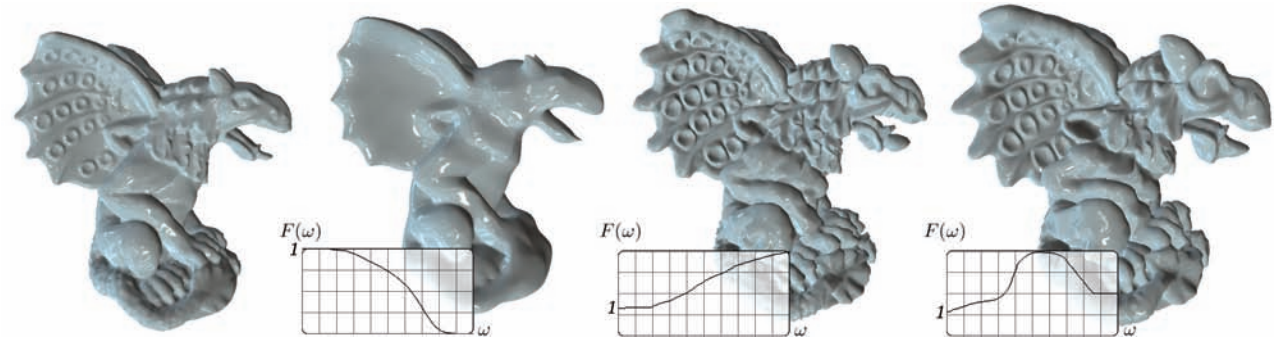

Figure 5: *Low-pass, enhancement and band-exaggeration filters. The filter can be changed by the user, the surface is updated interactively.*

eigenvectors. In addition, if the surface is large (a million vertices), the MHB does not fit in system RAM.

4.1. Band-by-band computation of the MHB

We address both issues by applying spectral transforms to the eigenproblem. To get the eigenvectors of a spectral band centered around a value λ_S, we start by shifting the spectrum by λ_S, by replacing $\bar{\Delta}$ with $\Delta_S = \bar{\Delta} - \lambda_S Id$. Then, we can swap the spectrum by inverting Δ_S. This is called the *Shift-Invert* spectral transform, and the new eigenproblem to solve is given by:

$$\Delta_S^{-1} \bar{H}^k = \mu_k \bar{H}^k \tag{10}$$

The fact that Δ_S may be singular is in practice not a problem as the iterative solver we use to solve (10) does not ask for Δ_S^{-1} explicitly. Instead, it outputs some vectors $\vec{v}$ and asks in return for the result of the multiplication $\vec{x} = \Delta_S^{-1}\vec{v}$. Hence we only need to compute a sparse *indefinite* Cholevsy factorization of Δ_S using sparse direct solvers (TAUCS, SUPERLU). Thanks to this factorization, it is very fast to solve $\Delta_S \vec{x} = \vec{v}$ by simple back-substitution each time the iterative solver asks for a multiplication with Δ_S^{-1}. For large models (a million vertices), we used the sparse OOC (out-of-core) symmetric indefinite factorization [MIT06] implemented in the future release of TAUCS, kindly provided by S. Toledo.

It is easy to check that (9) and (10) yield the same eigenvectors, and that the eigenvalues are related by $\lambda_k = \lambda_S + 1/\mu_k$. Our iterative solver returns the high end of the spectrum (largest μ's), corresponding to a band of eigenvalues of $\bar{\Delta}$ centered around λ_S. It is then possible to split the MHB computation into multiple bands, and obtain a computation time that is linear in the number of computed eigenpairs. Based on the considerations above, we derive the following "band-by-band" algorithm:

(1) $\lambda_S \leftarrow 0 \quad ; \quad \lambda_{last} \leftarrow 0$
(2) **while**$(\lambda_{last} < \omega_m^2)$
(3) compute the Cholevsky factorization of $\Delta_S = \bar{\Delta} - \lambda_S Id$
(4) find the 50 first eigenpairs $(\bar{H}^k, \mu_k)$ of Δ_S^{-1}
(5) **for** $k = 1$ **to** 50
(6) $\lambda_k \leftarrow \lambda_S + 1/\mu_k$
(7) **if** $(\lambda_k > \lambda_{last})$ **write**$(\bar{H}^k, \lambda_k)$
(8) **end** $//for$
(9) $\lambda_S \leftarrow \max(\lambda_k) + 0.4(\max(\lambda_k) - \min(\lambda_k))$
(10) $\lambda_{last} \leftarrow \max(\lambda_k)$
(11) **end** $//while$

If the MHB does not fit in RAM, the new eigenpairs of each band of can be stream-writen into a file (Line 7). Since the eigenvalues are centered around the shift λ_S, the shift for the next band is given by the last computed eigenvalue plus slightly less than half the bandwidth to ensure that the bands overlap and that we are not missing any eigenvalue (Line 9). If the bands do not overlap, we recompute a larger band until they do. Note that this is different from the shift-invert spectral transform implemented by ARPACK, dedicated to iterative solvers. Ours makes use of the factorization of the matrix, resulting in much better performances.

4.2. Limited-memory MH Filtering

After computing the MHB (Section 4.1) the filtering algorithm (Section 3.2) allows for interactive spectral geometry processing. However, for large meshes, storing the MHB may be prohibitive. Fortunately, it is possible to compute the MHB and the filter *all together* (at the cost of loosing interactivity). In addition, computations can be reordered in a way that requires storing only one eigenvector at a time:

(1) $x^F \leftarrow x$
(2) **for each** eigenpair(H^k, ω_k)
(3) $x^F \leftarrow x^F + (F(\omega_k) - 1) < x, H^k > H^k$
(where $< x, H^k > = \tilde{x}_k = x^T \star_0 H^k$)
(4) **end for**

	dino Fig. 2	drago Fig. 1	drago1*	drago2**	drago3** Fig. 6
n	56K	150K	244K	500K	1M
m	447	315	667	800	1331
MHB	77 s	160 s	9 m	2 h 21 m	6 h
MHT	0.34 s	0.65 s	18 s	32 s	76 s
MHT^{-1}	0.53 s	1.02 s	4 s	48 s	85 s
LM-filt	18 s	41 s	135 s	28 m	1h

Table 1: *Timings for the different phases of the algorithm. For each data set, we give the number of vertices n, the number of computed eigenfunctions m, and the timings for the MHB, MHT and inverse MHT with filtering (Intel T7600 2.33 GHz). (*) using OOC MHT (**) using both OOC factorization and OOC MHT. The limited-memory (LM-filt) example uses a band-exaggeration filter that spans 1/4th of the object's spectrum.*

In a certain sense, this algorithm *replaces* a given frequency in the mesh by its filtered version. Once again, this is made possible by the *orthonormality* of the MHB. In practice, the combined MHB+filtering algorithm is obtained by replacing line (7) of the MHB computation algorithm with line (3) of the algorithm above (instead of streaming the eigenpair to a file, we replace the corresponding frequency in the mesh). Interestingly, since our limited-memory algorithm changes a specific frequency *in-place* (i.e. without storing the MHB), it is possible to limit the computation of the MHB to the frequencies changed by the filter, by using the shift-invert spectral transform.

5. Results, Discussion and Conclusions

We have experimented the OOC factorization combined with the streamed band-by-band eigenvectors algorithm for computing up to a thousand eigenvectors on a mesh with one million vertices. We have also implemented an OOC version of the MHT, filtering and inverse MHT, that reads one frequency band at a time and accumulates its contribution.

We have experimented our filtering method with objects of different sizes. The timings are reported in table 1. The last row (limited-memory version) uses a band-exaggeration filter that spans $1/4$th of the object's spectrum. As can be seen, since we only compute the concerned eigenpairs, this results in better performances (as compared to summing $MHB + MHT + MHT^{-1}$ rows). For smoothing purposes only, our method does not outperform local approaches. However, for filters that involve low frequencies, performances are better than Geofilter [KR05], since this latter method needs to compute the inverse of a power of the Laplacian matrix, that is not sparse anymore for lower ends of the spectrum. For large models, e.g. the 1M vertices dragon (Figure 6), performance is still slow (1h). However, note that no other existing method would allow to reach and act on the lower-end of the spectrum (that corresponds to large geometric details). Our MH-based filtering can be applied to objects of arbitrary topology. Figure 7 shows a low-pass filter used to remove high-frequency noise from a sphere and from a genus 4 object. The low-pass filter nearly preserves the symmetry of the sphere. Figure 8 and the video show how our method implements an interactive version of *geofilter* [KR05]. In addition, since our filter explicitly implements the spectral transform, it does not introduce the shrinking effect usually encountered with existing ones.

Conclusions

In this paper, we have presented a method for filtering functions defined on manifolds. Our method and associated numerical solution mechanism may find applications in various contexts, e.g. segmentation, mesh watermarking or reconstruction. We have explained how a DEC framework allows to recover the symmetry of discrete Laplace operators and the orthogonality of its eigenvectors. We used this theoretical framework to define the Manifold harmonic transform and inverse. After precomputing the MHB, the resulting filter is interactive and does not encounter the shrinking effect obtained with classical schemes. On the practical side, we have overcome the current size limits of spectral geometry processing by several orders of magnitudes, by making it usable for meshes with up to $10^5 - 10^6$ vertices. With our limited-memory MH filtering algorithm, storage space is no-longer a limit. However, processing time for the MHB starts to be expensive (hours) beyond 10^6 vertices. This will be optimized in future works, by introducing multiresolution in our solution mechanism.

Another limitation of our method concerns objects with creases. It is well known that low-pass filters based on Fourier-like methods cannot preserve the creases. Using the eigenfunctions of an anisotropic version of the Laplace operator may improve the frequency localization of the creases and therefore better preserve them when filtering.

Our solver applied to the uniform Laplacian can be also used

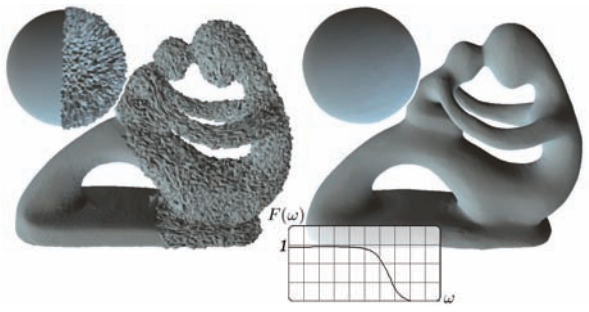

Figure 7: *Left: a sphere and a genus-4 model with random noise added. Right: the low-pass filtered result.*

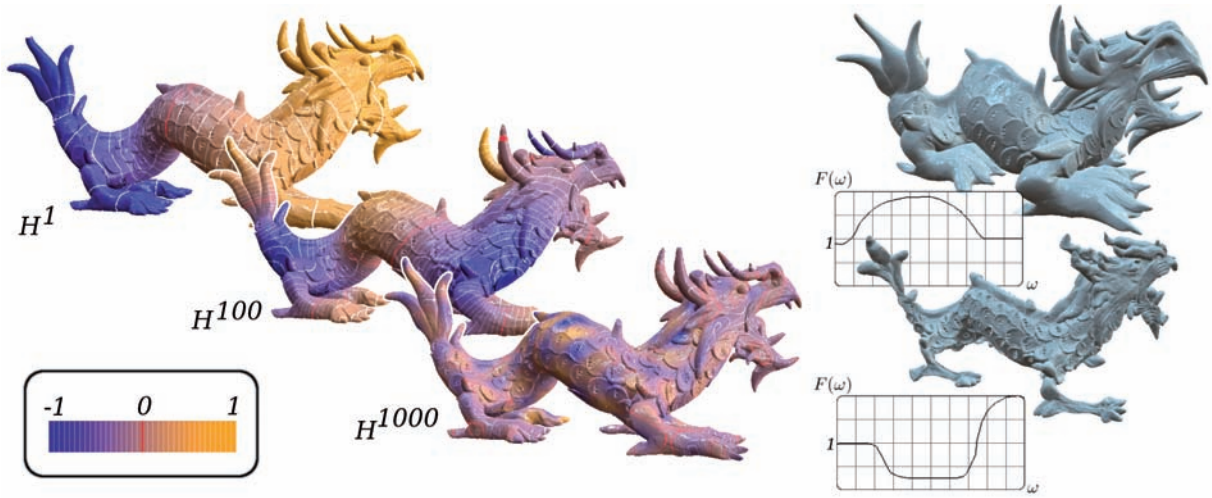

Figure 6: *The MHB computed on 1M vertices (XYZ dragon) and limited-memory MH-filtering (that does not store the MHB).*

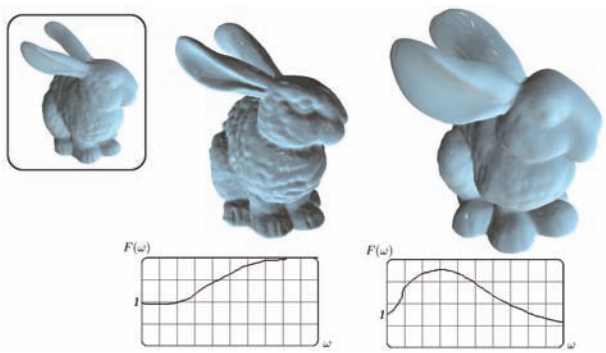

Figure 8: *Filtering Stanford's bunny. Results similar to geofilter are obtained, with the addition of interactivity, and without any shrinking effect.*

to implement an efficient version of Karni *et. al.*'s Spectral Mesh Compression [KG00]. Our method cannot be directly applied to mesh compression as we took particular care making our Laplacian geometry dependant. In this case, a purely combinatorial Laplacian is required as in Karni *et. al.*'s Spectral Mesh Compression [KG00]. Moreover, it turns out that because the MHB is not spatially localized, many MHT coefficients (several thousands) are required to accurately reconstruct the geometry, even with a geometry aware Laplacian (see Figure 4). Besides Karni's initial concern of reducing computation time, we think that partitioning also partially fixes the problem of spatial localization (this is also why JPEG uses small blocks instead of applying the DCT to the whole image) at the expense of losing continuity. This leads to forecast that defining *Manifold Wavelets* localized in both frequency and spatial domains [GKS02] will be an exciting research avenue in the future.

Acknowledgments

We gratefully thank all the reviewers of this paper and especially the senior reviewer for helping us to enhance the quality of this paper. We wish to thank Matthias Zwicker for the suggestion of studying manifold learning, Ramsey Dyer for discussions about orthogonality, Sivan Toledo for providing us with his latest TAUCS package. We are also grateful to Michela Spagnulo for encouraging us to continue in this direction, and Michael Garland for discussions on the eigenfunctions. We thank Raphaelle Chaine and Remi Allegre for some of the mesh models. This work was funded by AIM@Shape (FP6 European Network of Excellence), ARC GEOREP (INRIA grant) and Geometric Intelligence (Microsoft Research grant).

References

[AFW06] ARNOLD D. N., FALK R. S., WINTHER R.: Finite element exterior calculus, homological techniques, and applications. *Acta Numerica 15* (2006).

[BW90] BLOOR M., WILSON M.: Using partial differential equations to generate free-form surfaces. *Computer-Aided Design*, 22 (1990), 202–212.

[DBG*06] DONG S., BREMER P.-T., GARLAND M., PASCUCCI V., HART J. C.: Spectral surface quadrangulation. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), ACM Press, pp. 1057–1066.

[DF88] DONNELLY H., FEFFERMAN C.: Nodal sets of eigenfunctions on riemannian manifolds. *Invent. Math. 93*, 1 (1988), 161–183.

[DKT05] DESBRUN M., KANZO E., TONG Y.: Dis-

crete differential forms for computational modeling. *Siggraph '05 course notes on Discrete Differential Geometry, Chapter 7* (2005).

[GKS02] GRINSPUN E., KRYSL P., SCHRÖDER P.: CHARMS: a simple framework for adaptive simulation. In *SIGGRAPH Proceedings* (July 2002), vol. 21, pp. 281–290.

[GSS99] GUSKOV I., SWELDENS W., SCHRÖDER P.: Multiresolution signal processing for meshes. *Computer Graphics Proceedings (SIGGRAPH 99)* (1999), 325–334.

[Hir03] HIRANI A.: Discrete exterior calculus. *PhD thesis* (2003).

[HL88] HILDEBRANDT S., LEIS R.: Finite elements for the beltrami operator on arbitrary surfaces. *Partial differential equations and calculus of variations, Lecture Notes in Mathematics 1357* (1988), 142–155.

[HPW06] HILDEBRANDT K., POLTHIER K., WARDETZKY M.: On the convergence of metric and geometric properties of polyhedral surfaces. *Geom Dedicata* (2006).

[KCVS98] KOBBELT L., CAMPAGNA S., VORSATZ J., SEIDEL H.: Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH Conference Proceedings* (1998), pp. 105–114.

[KG00] KARNI Z., GOTSMAN C.: Spectral compression of mesh geometry. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 279–286.

[Kob97] KOBBELT L.: Discrete fairing. In *Proceedings of the Seventh IMA Conference on the Mathematics of Surfaces* (1997), pp. 101–131.

[KR05] KIM B., ROSSIGNAC J.: GeoFilter: Geometric Selection of Mesh Filter Parameters. *Computer Graphics Forum 24*, 3 (2005), 295–302.

[Lev06] LEVY B.: Laplace-beltrami eigenfunctions: Towards an algorithm that understands geometry. In *IEEE International Conference on Shape Modeling and Applications* (2006).

[LSS*98] LEE A. W. F., SWELDENS W., SCHRÖDER P., COWSAR L., DOBKIN D.: Maps: Multiresolution adaptive parameterization of surfaces. *Computer Graphics Proceedings (SIGGRAPH 98)* (1998), 95–104.

[Mal92] MALLET J.: Discrete Smooth Interpolation. *Computer Aided Design 24*, 4 (1992), 263–270.

[MCA06] MOUSA M., CHAINE R., AKKOUCHE S.: Direct spherical harmonic transform of a triangulated mesh. *GRAPHICS-TOOLS 11*, 2 (2006), 17–26.

[MDSB03] MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, Hege H.-C., Polthier K., (Eds.). Springer-Verlag, Heidelberg, 2003, pp. 35–57.

[MIT06] MESHAR O., IRONY D., TOLEDO S.: An out-of-core sparse symmetric indefinite factorization method. *ACM Transactions on Mathematical Software 32* (2006), 445–471.

[NISA06] NEALEN A., IGARASHI T., SORKINE O., ALEXA M.: Laplacian mesh optimization. In *Proceedings of ACM GRAPHITE* (2006), pp. 381–389.

[PG01] PAULY M., GROSS M.: Spectral processing of point sampled geometry. In *SIGGRAPH Proceedings* (2001).

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

[RWP05] REUTER M., WOLTER F.-E., PEINECKE N.: Laplace-spectra as fingerprints for shape matching. In *SPM '05: Proceedings of the 2005 ACM symposium on Solid and physical modeling* (New York, NY, USA, 2005), ACM Press, pp. 101–106.

[SCOIT05] SORKINE O., COHEN-OR D., IRONY D., TOLEDO S.: Geometry-aware bases for shape approximation. *IEEE Transactions On Visualization And Computer Graphics* (2005).

[Tau95] TAUBIN G.: A signal processing approach to fair surface design. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), ACM Press, pp. 351–358.

[VL07] VALLET B., LEVY B.: *Manifold Harmonics*. Tech. rep., 2007.

[WBH*07] WARDETZKY M., BERGOU M., HARMON D., ZORIN D., GRINSPUN E.: Discrete quadratic curvature energies. *Computer Aided Geometric Design (CAGD)* (2007).

[WK05] WU J., KOBBELT L.: Efficient spectral watermarking of large meshes with orthogonal basis functions. In *The Visual Computer* (2005).

[WMKG07] WARDETZKY M., MATHUR S., KALBERER F., GRINSPUN E.: Discrete laplace operators: No free lunch. *Eurographics Symposium on Geometry Processing* (2007).

[ZBS04] ZHOU K., BAO H., SHI J.: 3d surface filtering using spherical harmonics. In *Computer-Aided Design 36* (2004), p. 363Ű375.

[Zha04] ZHANG H.: Discrete combinatorial laplacian operators for digital geometry processing. In *Proc. SIAM Conference on Geometric Design and Computing* (2004).

[ZvKD07] ZHANG H., VAN KAICK O., DYER R.: Spectral methods for mesh processing and analysis. 1–22.