

# Laplace-Beltrami Eigenfunctions

## Towards an algorithm that “understands” geometry

Bruno Lévy  
INRIA - ALICE  
Villers-lès-Nancy, France  
Bruno.Levy@inria.fr

### Abstract

*One of the challenges in geometry processing is to automatically reconstruct a higher-level representation from raw geometric data. For instance, computing a parameterization of an object helps attaching information to it and converting between various representations. More generally, this family of problems may be thought of in terms of constructing structured function bases attached to surfaces.*

*In this paper, we study a specific type of hierarchical function bases, defined by the eigenfunctions of the Laplace-Beltrami operator. When applied to a sphere, this function basis corresponds to the classical spherical harmonics. On more general objects, this defines a function basis well adapted to the geometry and the topology of the object.*

*Based on physical analogies (vibration modes), we first give an intuitive view before explaining the underlying theory. We then explain in practice how to compute an approximation of the eigenfunctions of a differential operator, and show possible applications in geometry processing.*

### 1. Motivations

These last few years, mesh parameterization is a topic for which much time and effort has been devoted to. Floater in his recent survey [11] reviews a large number of methods for objects with disc topology. More recently, the geometry processing community started to study methods that could be applied to objects of arbitrary topology, based for instance on holomorphic one-forms [15]. Such an interest for parameterization methods is justified by the fact that it facilitates attaching attributes to surfaces. For instance, such attributes may be physical properties simulated by a PDE solver. Adapting a finite-element method to this representation of surfaces is easy, and is often referred to as the *master element* method (see e.g. [21]). However, recent advances

in PDE solving and in calculus suggest that a parameterization may not be the best representation of geometry and attributes attached to the geometry. For instance, to construct a hierarchical representation, the CHARMS method presented in [14] suggests to use *function basis* refinement instead of *finite element* refinement. This function-centric view leads to simpler computations. The *external calculus* and its discrete counterparts are also leading towards this direction (see e.g. Desbrun and Schroeder’s course at Siggraph 2005).

For this reason, rather than seeking for a global parameterization of the objects, we will investigate algorithms that create a *function basis* over an object of arbitrary topology. Surprisingly, as shown further, the underlying basic principles (Laplace operator) are present in a wide variety of disciplines of computer graphics and geometry processing.

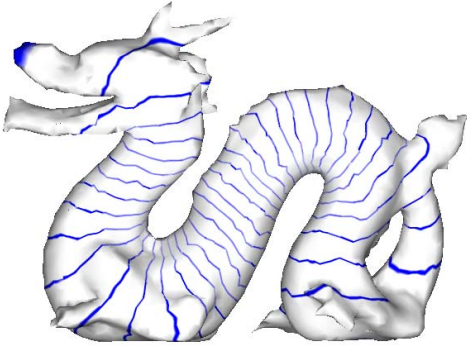
The remainder of this paper is organized as follows. Section 2 reviews spectral analysis used in the discrete setting. Section 3 explains the links with the continuous setting and the Laplace-Beltrami operator. Section 4 presents some possible applications in geometry processing.

### 2. The discrete setting: Graph Laplacians

Spectral graph theory was used for instance in [16] to compute an ordering of vertices in a mesh that facilitates out-of-core processing. Such a natural ordering can be derived from the Fiedler eigenvector of the Graph Laplacian. The Graph Laplacian  $L = (a_{i,j})$  is a matrix defined by:

$$\begin{aligned} a_{i,j} &= w_{i,j} > 0 && \text{if } (i,j) \text{ is an edge} \\ a_{i,i} &= -\sum_j w_{i,j} \\ a_{i,j} &= 0 && \text{otherwise} \end{aligned}$$

where the coefficients  $w_{i,j}$  are weights associated to the edges of the graph. One may use the uniform weighting  $w_{i,j} = 1$  or more elaborate weightings, computed from the embedding of the graph.



**Figure 1. The Fiedler vector gives a natural ordering of the nodes of a graph. The displayed contours show that it naturally follows the shape of the dragon.**

The first eigenvector of the Graph Laplacian is  $(1, 1 \dots 1)$  and its associated eigenvalue is 0. The second eigenvector is called the Fiedler vector and has interesting properties, making it a good permutation vector for numerical computations [9, 10]. It has many possible applications, such as finding natural vertices ordering for streaming meshes [16]. Figure 1 shows what it looks like for a snake-like mesh (it naturally follows the shape of the mesh).

More insight on the Fiedler vector is given by the following alternative definition. The Fiedler vector  $u = (u_1 \dots u_n)$  is the solution of the following constrained minimization problem:

$$\begin{aligned} \text{Minimize: } F(u) &= u^t L u = \sum_{i,j} w_{i,j} (u_i - u_j)^2 \\ \text{Subject to: } \sum_i u_i &= 0 \quad \text{and} \quad \sum_i u_i^2 = 1 \end{aligned} \quad (1)$$

In other words, given a graph embedded in some space, and supposing that the edge weight  $w_{i,j}$  corresponds to the lengths of the edges in that space, the Fiedler vector  $(u_1 \dots u_n)$  defines a (1-dimensional) embedding of the graph on a line that tries to respect the edge lengths of the graph.

This naturally leads to the question of whether embedding in higher-dimensional spaces can be computed (for instance, computing a 2-dimensional embedding of a surface corresponds to the classical surface parameterization problem). This general problem is well known by the *automatic learning* research community as a *Manifold learning* problem, also called *dimension reduction*, see for instance Martin Law's web page <http://www.cse.msu.edu/~lawhiu/manifold/>.

One of the problems in manifold learning is extracting from a set of input (e.g. a set of images of the same ob-

ject) some meaningful parameters (e.g. camera orientation and lighting conditions), and sort these images with respect to these parameters. From an abstract point of view, the images leave in a high-dimensional space (the dimension corresponds to the number of pixels of the images), and one tries to *parameterize* this image space. The first step constructs a graph, by connecting each sample to its nearest neighbors, according to some distance function. Then, different classes of methods have been defined, we quickly review the most popular ones:

*Local Linear Embedding* [25] tries to create an embedding that best approximates the barycentric coordinates of each vertex relative to its neighbors. In a certain sense, Floater's Shape Preserving Parameterization (see [11]) is a particular case of this approach.

*Isomap* [27] computes the geodesic distances between each pair of vertex in the graph, and then uses MDS (*multidimensional scaling*) [29] to compute an embedding that best approximates these distances. Multidimensional scaling simply minimizes an objective function that measures the deviation between the geodesic distances in the initial space and the Euclidean distances in the embedding space (GDD for Geodesic Distance Deviation), by computing the eigenvectors of the matrix  $D = (d_{i,j})$  where  $d_{i,j}$  denotes the geodesic distance between vertex  $i$  and vertex  $j$ . This is a multivariate version of Equation 1, that characterizes the Fiedler vector (in the univariate setting). Isomaps and Multidimensional scaling were used to define parameterization algorithms in [32], and more recently in the *ISO-charts* method [31], used in Microsoft's DirectX combined with our packing algorithm presented in [2].

At that point, we understand that the eigenvectors play an important role in determining meaningful parameters. Just think about the simple linear case: in PCA (principal component analysis), the eigenvectors of the covariance matrix characterize the most appropriate hyperplane on which the data should be projected. In dimension reduction, we seek for eigenvectors that will fit non-linear features. For instance, in MDS, these eigenvectors are computed in a way that makes the embedding space mimic the global metric structure of the surface, captured by the matrix  $D = (d_{i,j})$  of all geodesic distances between all pairs of vertices in the graph.

Instead of using the dense matrix  $D$ , methods based on the Graph Laplacian only use local neighborhoods (one-ring neighborhoods). As a consequence, the used matrix is sparse, and extracting its eigenvectors requires lighter computations. Note that since the Graph Laplacian is a symmetric matrix, its eigenvectors are orthogonal, and can be used as a vector basis to represent functions. This was used in [19] to define a compact encoding of mesh geometry. The basic idea consists in encoding the topology of the mesh together with the coefficients that define the geometry pro-

jected onto the basis of eigenvectors. The decoder simply recomputes the basis of eigenvectors and multiplies them with the coefficients stored in the file. A survey of spectral geometry compression and its links with graph partitioning is given in [12]. Spectral graph theory also enables to exhibit ways of defining valid graph embeddings. For instance, Colin-de-verdière's number [4] was used in [13] to construct valid spherical embeddings of genus 0 meshes. Other methods that use spectral graph theory to compute graph embeddings are reviewed in [20]. Spectral graph theory can also be used to compute topological invariants (e.g. Betti numbers), as explained in [8]. As can be seen from this short review of spectral graph theory, the eigenvectors and eigenvalues of the graph Laplacian contain both geometric and topological information.

However, as explained in [31], only using the connectivity of the graph may lead to highly distorted mappings. Methods based on MDS solve this issue by considering the matrix  $D$  of the geodesic distances between all possible pairs of vertices. However, we think that it is also possible to inject more geometry in the Graph Laplacian approach, and understand how the global geometry and topology of the shape may emerge from the interaction of local neighborhoods.

This typically refers to notions from the *continuous* setting, i.e. functional analysis and operators. The next section shows its link with the Laplace-Beltrami operator, that appears in the wave equation (Helmholtz's equation). We will also exhibit the link between the so-called *stationary waves* and spectral graph theory.

### 3. The Continuous Setting: Laplace Beltrami

Since we aim at constructing a function basis, we need some notions from functional analysis, quickly reviewed here. A similar review in the context of light simulation is given in [1].

#### 3.1 Functional analysis

In a way similar to what is done for vector spaces, we need to introduce a dot product (or inner product) to be able to define function bases and project functions onto those bases. This corresponds to the notion of Hilbert space, outlined below.

##### Hilbert spaces

Given a surface  $S$ , let  $X$  denote the space of real-valued functions defined on  $S$ . Given a norm  $\|\cdot\|$ , the function space  $X$  is said to be *complete* with respect to the norm if Cauchy sequences converge in  $X$ , where a Cauchy sequence is a sequence of functions  $f_1, f_2, \dots$  such that

$\lim_{n,m \rightarrow \infty} \|f_n - f_m\| = 0$ . A complete vector space is called a *Banach space*.

The space  $X$  is called a *Hilbert space* in the specific case of the norm defined by  $\|f\| = \sqrt{\langle f, f \rangle}$ , where  $\langle \cdot, \cdot \rangle$  denotes an inner product. A possible definition of an inner product is given by  $\langle f, g \rangle = \int_S f(x)g(x)dx$ , which yields the  $L_2$  norm.

One of the interesting features provided by this additional level of structure is the ability to define function bases and project onto these bases using the inner product. Using a function basis  $(\Phi_i)$ , a function  $f$  will be defined by  $f = \sum \alpha_i \Phi_i$ . Similarly to the definition in geometry, a function basis  $(\Phi_i)$  is *orthonormal* if  $\|\Phi_i\| = 1$  for all  $i$  and  $\langle \Phi_i, \Phi_j \rangle = 0$  for all  $i \neq j$ . Still following the analogy with geometry, given the function  $f$ , one can easily retrieve its coordinates  $\alpha_i$  with respect to an orthonormal basis  $(\Phi_i)$  by *projecting*  $f$  onto the basis, i.e.  $\alpha_i = \langle f, \Phi_i \rangle$ .

##### Operators

We now give the basic definitions related with operators. Simply put, an operator is a function of functions (i.e. from  $X$  to  $X$ ). An operator  $L$  applied to a function  $f \in X$  is denoted by  $Lf$ , and results in another function in  $X$ . An operator  $L$  is said to be *linear* if  $L(\lambda f) = \lambda Lf$  for all  $f \in X, \lambda \in \mathbb{R}$ . An *eigenfunction* of the operator  $L$  is a function  $f$  such that  $Lf = \lambda f$ . The scalar  $\lambda$  is an *eigenvalue* of  $L$ . In other words, the effect of applying the operator to one of its eigenfunctions means simply *scaling* the function by the scalar  $\lambda$ .

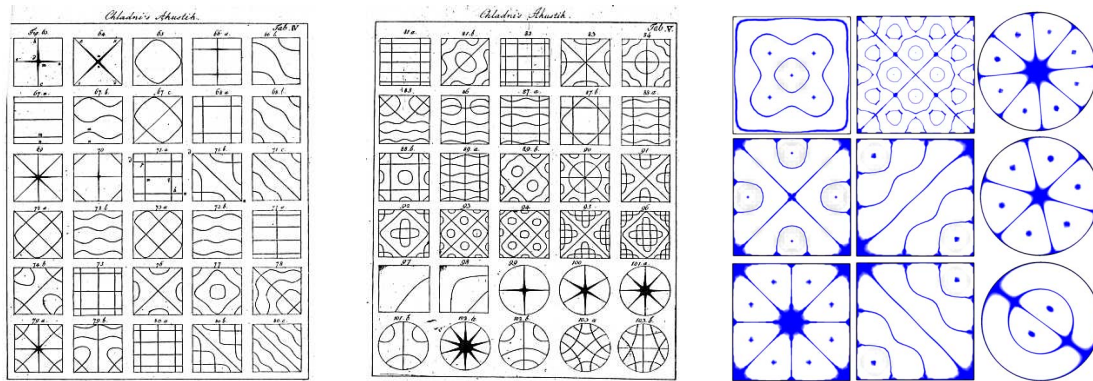
A linear operator  $L$  is said to be *Hermitian* (or with Hermitian symmetry)<sup>1</sup> if  $\langle Lf, g \rangle = \langle f, Lg \rangle$  for each  $f, g \in X$ . An important property of Hermitian operators is that their eigenfunctions associated to different eigenvalues have real eigenvalues and are orthogonal. This latter property can be easily proven as follows, by considering two eigenfunctions  $f, g$  associated with the different eigenvalues  $\lambda, \mu$  respectively:

$$\begin{aligned} \langle Lf, g \rangle &= \langle f, Lg \rangle \\ \langle \lambda f, g \rangle &= \langle f, \mu g \rangle \\ \lambda \langle f, g \rangle &= \mu \langle f, g \rangle \end{aligned}$$

which gives the result ( $\langle f, g \rangle = 0$ ) since  $\lambda \neq \mu$ .

As a consequence, considering the eigenfunctions of an Hermitian operator is a possible way of defining an orthonormal function basis associated to a given function space  $X$ . The next section shows this method applied to the Laplace-Beltrami operator. Before entering the heart of the matter, we will first consider the historical perspective.

<sup>1</sup>the general definition of Hermitian operators concerns complex-valued functions, we only consider here real-valued functions.



**Figure 2. Left: In the late 1700's, the physicist Ernst Chladni was amazed by the patterns formed by sand on vibrating metal plates. Right: numerical simulations obtained with our approach.**

### 3.2 Chladni plates

In 1787, the physicist Ernst Chladni published the book entitled “Discoveries Concerning the Theories of Music”. In this book, he reports his observations obtained when putting a thin metal plate into vibration using a bow, and spreading sand over it. Sand accumulates in certain zones, forming surprisingly complex patterns (see Figure 2).

This behavior can be explained by the theory of *stationary waves*. When the metal plate vibrates, some zones remain still, and sand naturally concentrates in these zones. This behavior can be modeled as follows, by the spatial component of Helmholtz’s wave propagation equation:

$$\Delta f = \lambda f \quad (2)$$

In this equation,  $\Delta$  denotes the Laplace-Beltrami operator on the considered object. In Cartesian 2D space,  $\Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2$ . We are seeking for the eigenfunctions of this operator. To better understand the meaning of this equation, let us first consider a vibrating circle. This corresponds to the univariate case on the interval  $[0, 2\pi]$  with cyclic boundary conditions (i.e.  $f(0) = f(2\pi)$ ). In this setting, the Laplace-Beltrami operator simply corresponds to the second order derivative. Recalling that  $\sin(\omega x)'' = -\omega^2 \sin(\omega x)$ , our eigenfunctions are simply  $\sin(Nx)$ ,  $\cos(Nx)$  and the constant function, where  $N$  is an integer. Note that the so-constructed function basis is the one used in Fourier analysis.

From the spectrum of the Laplace-Beltrami operator, it is well known that one can extract the area of  $S$ , the length of its border and its genus. This leads to the question asked by Kac in 1966: *Can we hear the shape of a drum ?* [18]. The answer to this question is “no”: one can find drums that have the same spectrum although they do not have the same shape [3] (they are then referred to as *isospectral* shapes).

However, the spectrum contains much information, which led to the idea of using it as a signature for shape matching and classification, as explained in the “shape DNA” approach [24]. Note that the term DNA is somewhat too strong since one may expect to be able to reconstruct the shape from its DNA. To be able to do so, one rather needs the *eigenfunctions* and the series of coefficients obtained by projecting the shape onto these eigenfunctions.

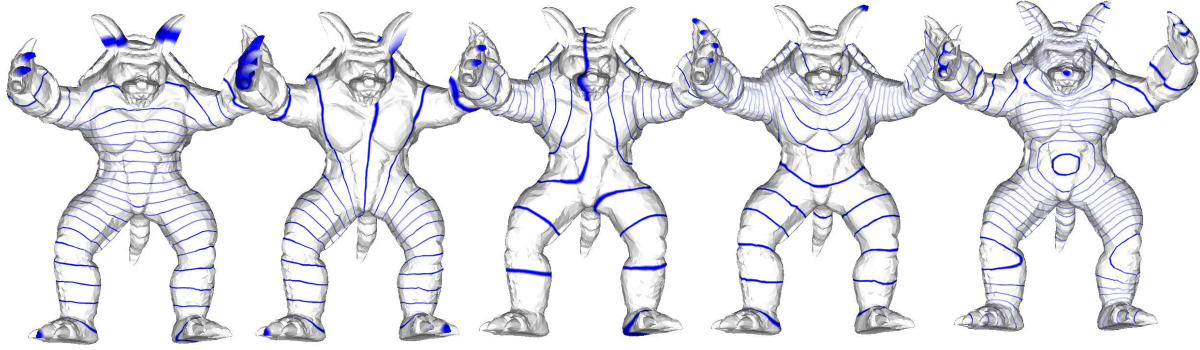
For this reason, we are going now to take a look at these eigenfunctions. Mathematicians mostly studied bounds and convergence of the spectrum. However, some results are known about the *geometry* of the eigenfunctions [17]. More precisely, we are interested in the so-called *nodal sets*, defined to be the zero-set of an eigenfunction. Intuitively, they correspond to the locations that do not move on a Chladni plate, where sand accumulates (see Figure 2). A nodal set partitions the surface into a set of *nodal domains* of constant sign. The nodal sets and nodal domains are characterized by the following theorems:

1. the  $n$ -th eigenfunction has at most  $n$  nodal domains
2. the nodal sets are curves intersecting at constant angles

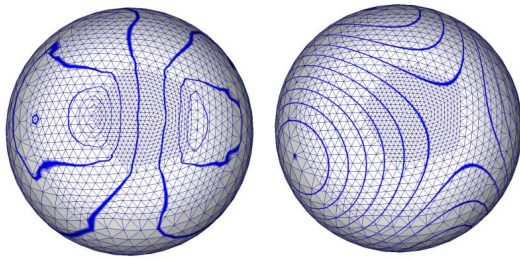
Besides their orthogonality, these properties make eigenfunction an interesting choice for function bases. Theorem 1 exhibits their multi-resolution nature, and from theorem 2, one can suppose that they are strongly linked with the geometry of the shape. Note also that these theorems explain the appearance of Chladni plates. This may also explain the very interesting re-meshing results obtained by Dong *et al* [6], that use a Morse-smale decomposition of one of the eigenfunctions.

In the case of simple objects, a closed form of the eigenfunctions can be derived. This made it possible to retrieve the patterns observed by Chladni in the case of a square and a circle. For curved geometry, Chladni could not make the





**Figure 3. Contours of the first eigenfunctions. Note how the protrusions and symmetries are captured. The eigenfunctions combine the geometric and topological information contained in the shape signal.**



**Figure 4. Contours of the 4th eigenfunction, computed from the Graph Laplacian (left) and cotangent weights (right) on an irregular mesh.**

experiment, since sand would not remain in the nodal set. However, one can still study the eigenfunctions. For instance, on a sphere, the eigenfunction correspond to *spherical harmonics* (see e.g. [17]), often used in computer graphics to represent functions defined on the sphere (such as radiance fields or Bidirectional Reflectance Distribution Functions). In other words, on a sphere, the eigenfunctions of the Laplace-Beltrami operator define an interesting hierarchical function basis. One can now wonder whether the same approach could be used to create function bases on more complex geometries. In the general case, a closed form cannot be derived, and one needs to use a numerical approach, as explained in the next section.

### 3.3 Numerical Solution Mechanism

Given a function basis  $(\Phi_i)$ , using the Galerkin method to solve a PDE means *projecting* the PDE onto the function

basis. In our specific problem, we want to find an approximation of the eigenfunctions of Laplace-Beltrami (Equation 2). The projected equation is given by:

$$\forall i, \quad \langle \Delta f, \Phi_i \rangle = \lambda \langle f, \Phi_i \rangle$$

At this point, it is possible to use a high-order  $(\Phi_i)$  function basis (e.g. polynomials), and use the divergence formula to transform the equation into a generalized eigenvalue problem, as done in [24].

In our context, to avoid the right-hand side matrix multiply in the generalized eigenvalue problem and the overhead in the computations, we use a discrete Laplacian operator (see e.g. [23, 5, 28] for possible definitions). We used Desbrun's formulation. However, special care needs to be taken: since these approaches define *discrete* rather than *discretized* Laplacians, they do not keep all the properties of their continuous counterparts. In our case, since we want to compute orthogonal function bases, we want to keep the Hermitian symmetry of the operator, that will be translated into a symmetric matrix. To ensure the symmetry of the matrix, we simply use  $0.5(L + L^t)$ , where  $L$  is the discrete Laplacian given in [5]. The eigenvalues and eigenvectors of the sparse matrix  $0.5(L + L^t)$  are found by the ARPACK solver (see <http://www.caam.rice.edu/software/ARPACK/>).

Figure 3 shows the first eigenfunctions computed on a decimated version of the Armadillo model. The contours of the eigenfunctions nicely follow the protrusions of the object and respect the symmetries. Figure 4 shows the result obtained with a combinatorial Laplacian (left) and with a discrete Laplacian that takes the geometry into account (right). Note how the irregular mesh does not influence the result when the geometric terms are used.

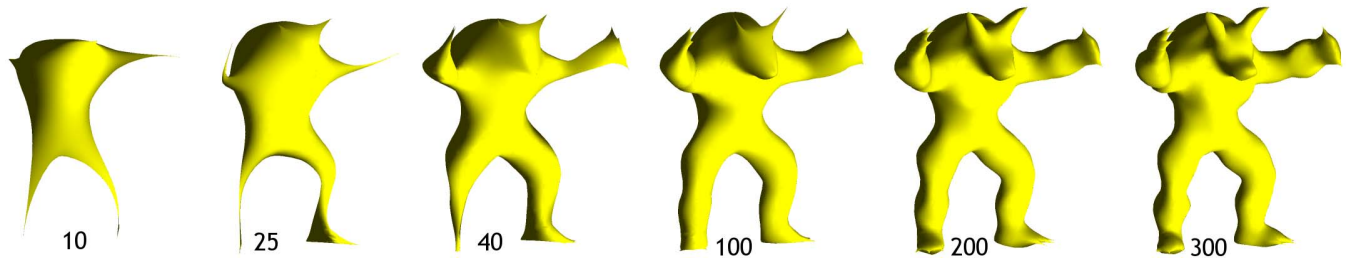


Figure 6. Geometric reconstructions obtained with increasing number of eigenfunctions.

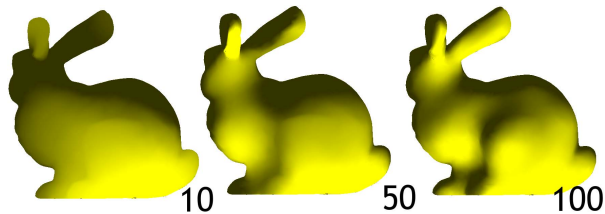


Figure 5. Reconstructing a signal (here the normal vector) using the eigenfunction basis. The variations of the normal vector over the bunny are well approximated by 100 coefficients.

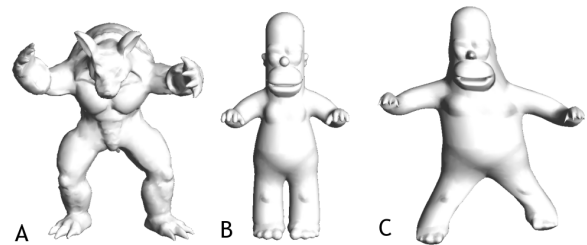


Figure 7. Pose transfer: the 5 first coefficients of the Armadillo (A) were copied to Homer (B) and adapted its pose to the Armadillo (C)

#### 4. Possible Applications

**Signal processing on surfaces:** Once the eigenfunction basis is computed, it can be used to represent various functions on the surface. Figure 5 shows how a signal (here the normal vector) can be compressed in this basis. The normal vector on the whole bunny is well approximated by 100 coefficients. This is similar to the way spherical harmonics can approximate functions on the sphere, with the difference that the function basis is adapted to the object on which functions need to be defined. This representation of the signal is interesting from a signal processing point of view (for instance, filtering can be done in real-time, since a convolution is replaced by a product in frequency space). However, this is not an efficient way of *compressing* a signal, since the representation also requires 100 coefficients per vertex to represent the function basis.

**Geometry processing:** If the geometry is also considered to be a signal, it can also be projected onto the eigenfunction basis (Figure 6). The figure shows the reconstructed geometry with an increasing number of coefficients. This is similar to Taubin's approach, described in his seminal paper [26]. To completely implement Taubin's view of geometry processing, one can construct a hierarchical

function space, defined over a chart-based parameterization of the surface (see e.g. [22]). More recent work attempt to use a more natural function space, defined by spherical harmonics, mapped onto the surface thanks to a spherical parameterization [30]. Besides the limitation to genus 0 geometry, this may also introduce distortion and an uneven sampling of the function space near protrusion. In contrast, the eigenfunction basis is well adapted to the geometry of the object, and can be defined for objects of arbitrary topology. Figure 6 shows that non-shrinking filtering can be easily obtained, by simply selecting the components that correspond to low frequencies.

**Registration and pose transfer:** One may wonder what our parameter-space looks like. In a certain sense, we completely get rid of referencing any geometry, the parameter-space is an abstract manifold, that only know the *neighborhoods*. This characteristic is interesting when trying to define common parameterizations for two shapes that do not share the same discretization. Since the protrusions have an important influence on the low frequencies, they naturally define a common parameter space for the two shapes. As a consequence, the coefficients corresponding to geometry projected onto the eigenfunction basis are compatible.

Figure 7 shows an example of this idea applied to pose transfer, performed as follows:

1. Compute the eigenfunction bases  $(\Phi_i)$  of the Armadillo and  $(\Phi'_i)$  of Homer;
2. Project the geometry onto each eigenfunction  $\Phi_i$  and  $\Phi'_i$

$$\begin{aligned}\alpha_i^x &\leftarrow \sum_j x_j \Phi_i(j) \\ \alpha_i^y &\leftarrow \sum_j y_j \Phi_i(j) \\ \alpha_i^z &\leftarrow \sum_j z_j \Phi_i(j)\end{aligned}$$

$$\begin{aligned}\beta_i^x &\leftarrow \sum_j x'_j \Phi'_i(j) \\ \beta_i^y &\leftarrow \sum_j y'_j \Phi'_i(j) \\ \beta_i^z &\leftarrow \sum_j z'_j \Phi'_i(j)\end{aligned}$$

where  $p_j = (x_j, y_j, z_j)$  denote the vertices of the Armadillo and  $q_j = (x'_j, y'_j, z'_j)$  denote the vertices of Homer;

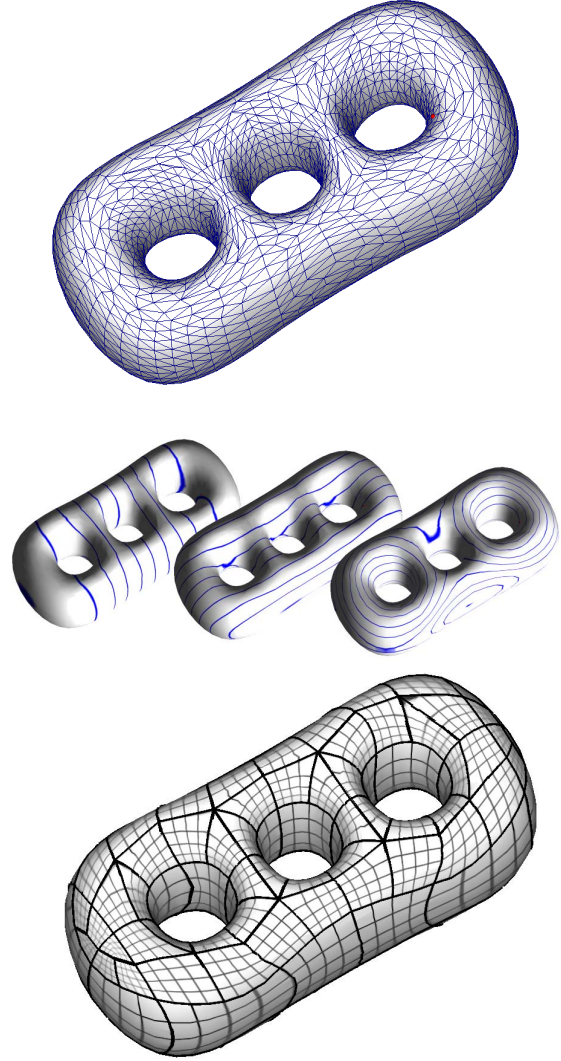
3. Reconstruct the signal for each vertex  $j$ , using the  $\alpha$ 's for the low frequencies and the  $\beta$ 's for the high frequencies:

$$\begin{aligned}x_j &\leftarrow \sum_{i=1}^5 \alpha_i^x \Phi_i(j) + \sum_{i=6}^n \beta_i^x \Phi_i(j) \\ y_j &\leftarrow \sum_{i=1}^5 \alpha_i^y \Phi_i(j) + \sum_{i=6}^n \beta_i^y \Phi_i(j) \\ z_j &\leftarrow \sum_{i=1}^5 \alpha_i^z \Phi_i(j) + \sum_{i=6}^n \beta_i^z \Phi_i(j)\end{aligned}$$

As shown in the figure, this process successfully transfers small deformations. This method works provided that the eigenfunctions that correspond to the lower frequencies match, in other words, provided that the  $\alpha$ 's and the  $\beta$ 's are expressed in the same "language". We think that deformations of larger scale may be transferred by a similar approach, by projecting local coordinates (e.g. Laplacian coordinates) onto the eigenfunctions rather than simply projecting the Cartesian coordinates  $(x, y, z)$  as we did.

**Segmentation and parameterization:** as suggested in [6], it is possible to use the eigenfunctions to steer a re-meshing algorithm. Figure 8 shows how eigenfunctions can be used to segment a model, prior to parameterizing the charts. The model is segmented by cutting several zero-sets of eigenfunctions. The main difficulty will be to determine which eigenfunction should be used (eigenfunctions were manually selected in this example).

Finally, as suggested by Durand *et. al* in [7], light transport can be studied in frequency space. Based on Durand's remark, we think that one possible application of projecting the geometry and the lighting onto a basis of eigenfunctions may lead to define tools for combined geometry/lighting processing tools.



**Figure 8. Top: a genus 3 object; Center: three eigenfunctions; Bottom: computing a segmentation from the nodal sets of these eigenfunctions (thick black lines), and parameterizing the charts (thin lines). Non-square charts are then subdivided from their center, and Floater's Mean Values Coordinates parameterization is applied to all the charts.**

In future works, we will study practical methods for computing the function basis for large models. Our current implementation is limited to a few thousand vertices. Multi-resolution methods may be an efficient way of overcoming this limit. About the theoretical aspects, we think that the eigenfunctions capture the interaction between the topology and the geometry at a fundamental level. Intuitively, the Laplace-Beltrami operator is a diffusion operator, that spreads the function it is applied to over small neighborhoods. The eigenfunctions make it possible to understand how these *local* neighborhoods interact and how *global* properties emerge from this interaction. We think that exterior calculus and co-homology may explain how these properties emerge. Another aspect we plan to study is the definition of hierarchical function bases with *local* supports, in order to support adaptive refinement.

## Acknowledgment

We want to thank the AIM@Shape European network of excellence for funding this research, the SMI organizing committee for inviting this talk, Matthias Zwicker for his suggestion of studying dimension reduction and manifold learning, and Nicolas Ray for his intuition of “finding the natural vibrations” of shapes that did inspire the Chladni plate analogy.

## References

- [1] J. Arvo. The Role of Functional Analysis in Global Illumination. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 115–126, New York, NY, 1995. Springer-Verlag.
- [2] L. Bruno, P. Sylvain, R. Nicolas, and M. Jrome. Least squares conformal maps for automatic texture atlas generation. In ACM, editor, *SIGGRAPH conf. proc.*, 2002.
- [3] B. Cipra. You can't always hear the shape of a drum. *What's Happening in the Mathematical Sciences*, 1, 1993.
- [4] Y. C. de Verdiere. Sur un nouvel invariant des graphes et un critere de planarite. *J. of Combinatorial Theory*, 50, 1990.
- [5] M. Desbrun. Applied geometry: Discrete differential calculus for graphics. *Comput. Graph. Forum*, 23(3), 2004.
- [6] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart. Quadrangulating a mesh using laplacian eigenvectors. Technical report, June 2005.
- [7] F. Durand, N. Holzschuch, C. Soler, E. Chan, and F. X. Sillion. A frequency analysis of light transport. *ACM Trans. Graph.*, 24(3):1115–1126, 2005.
- [8] J. Feidman. Computing betti numbers via combinatorial laplacians. In *Proc. 28th Sympos. Theory Comput.*, pages 386–391. ACM, 1996.
- [9] M. Fiedler. Algebraic connectivity of graphs. *Czech. Math. Journal*, 23:298–305, 1973.
- [10] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czech. Math. Journal*, 25:619–633, 1975.
- [11] M. S. Floater and K. Hormann. *Surface parameterization: a tutorial and survey*. Springer, 2004.
- [12] C. Gotsman. On graph partitioning, spectral analysis, and digital mesh processing. In *Shape Modeling International*, pages 165–174, 2003.
- [13] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3d meshes, 2003.
- [14] E. Grinspun, P. Krysl, and P. Schroder. Charms: A simple framework for adaptive simulation, 2002.
- [15] X. Gu and S.-T. Yau. Global conformal parameterization. In *Symposium on Geometry Processing*, pages 127–137, 2003.
- [16] M. Isenburg and P. Lindstrom. Streaming meshes. In *IEEE Visualization*, page 30, 2005.
- [17] D. Jakobson, N. Nadirashvili, and J. Toth. Geometric properties of eigenfunctions.
- [18] M. Kac. Can you hear the shape of a drum? *Amer. Math. Monthly*, 73, 1966.
- [19] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *SIGGRAPH*, pages 279–286, 2000.
- [20] Y. Koren. On spectral graph drawing, 2002.
- [21] G. Lecot, B. Levy, L. Alonso, and J.-C. Paul. Master-element vector irradiance for large tessellated models. In *Graphite Conference Proceedings*, 2005.
- [22] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. MAPS: Multiresolution adaptive parameterization of surfaces. *Computer Graphics*, 32(Annual Conference Series):95–104, 1998.
- [23] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.
- [24] M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace-beltrami spectra as “shape-dna” of surfaces and solids. *CAD Journal*, 2005.
- [25] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.
- [26] G. Taubin. A signal processing approach to fair surface design. *Computer Graphics*, 29(Annual Conference Series):351–358, 1995.
- [27] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.
- [28] G. Xu. Convergent discrete laplace-beltrami operators over triangular surfaces. In *GMP*, pages 195–204, 2004.
- [29] F. W. Young. Multidimensional scaling. *Encyclopedia of Statistical Sciences*, 5:649–658, 1985.
- [30] K. Zhou, H. Bao, and J. Shi. 3d surface filtering using spherical harmonics. *Computer-Aided Design*, 36(4):363–375, 2004.
- [31] K. Zhou, J. Snyder, B. Guo, and H.-Y. Shum. Iso-charts: Stretch-driven mesh parameterization using spectral analysis. In *Symposium on Geometry Processing*, pages 47–56, 2004.
- [32] G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *IEEE Transactions on Visualization and Computer Graphics*, 8(2), 2002.