

# Dirichlet Harmonic Shape Compression with Feature Preservation for Parameterized Surfaces

Yang Liu, Balakrishnan Prabhakaran, Xiaohu Guo

University of Texas at Dallas

---

## Abstract

*With the rapid advancement of 3D scanning devices, large and complicated 3D shapes are becoming ubiquitous, and require large amount of resources to store and transmit them efficiently. This makes shape compression a demanding technique in order for the user to reduce the data transmission latency. Existing shape compression methods could achieve very low bit-rates by sacrificing shape quality. But none of them guarantees the preservation of salient feature lines that users care. In addition, many 3D shapes come with parametric information for texture mapping purposes. In this paper we describe a spectral method to compress the geometric shapes equipped with arbitrary valid parametric information. It guarantees to preserve user-specified feature lines while achieving a high compression ratio. By applying the spectral shape analysis – Dirichlet Manifold Harmonics, in the 2D parametric domain, this method provides a progressive compression mechanism to trade-off between bit-rate and shape quality. Experiments show that this method provides very low bit-rate with high shape-quality and still guarantees the preservation of user-specified feature lines.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

---

## 1. Introduction

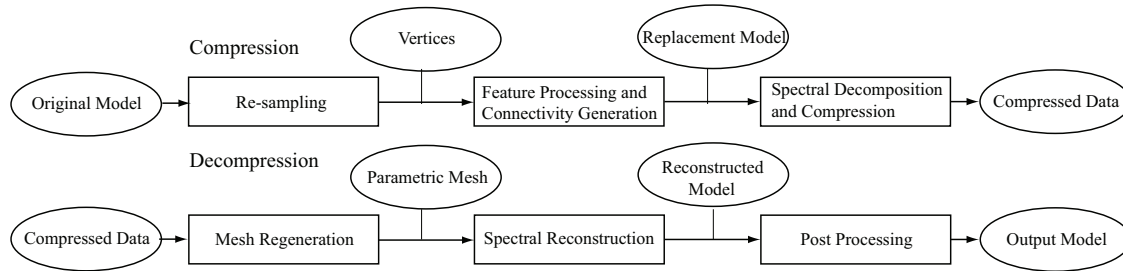
Nowadays we can routinely acquire large, detailed, and arbitrary topology 3D surfaces with millions of vertices. The rapid advancement of internet and distributed computing technologies gives rise to the pressing need for efficient transmission of these 3D geometric shapes, which will require shape compression to reduce the transmission latency and progressive representations to transform the 3D shapes into streams manageable by the networks.

3D shapes represented as triangular meshes typically contain three parts of information: geometric information, connectivity information, and parameter information. Parameter information is often used for texture mapping purposes. When the original mesh is considered as just one instance of the surface geometry that is to be compressed, shape compression has to be considered rather than mesh compression. Shape compression based on semi-regular or regular remeshing [KSS00, GGH02] assumes that the connectivity and parametric information are not important for representation of geometry. Thus their common goal is to reduce these two components as much as possible. This is achieved

through semi-regular/regular remeshing of an input irregular mesh, and can typically achieve a very high compression rate of the newly generated geometric model.

By focusing on the geometry compression, spectral coding approaches [KG00, KG01] use the eigenvectors of the Laplacian matrix derived from mesh connectivity graph to transform the three Cartesian geometry vectors  $(x, y, z)$ . Similar to traditional Fourier transform coding, a good approximation to the geometry vectors may be obtained by using just a linear combination of a small number of basis vectors. Thus they can typically achieve very low bit-rate by sacrificing shape quality. However, none of the existing remeshing and spectral compression method guarantees the preservation of salient feature lines that users care.

In this paper, we take advantages of the remeshing strategy by regenerating the regular connectivity for the 3D shape, to avoid the storage of connectivity information. We perform such remeshing in the parameter space by utilizing the parametric information of the meshes, and compute the Dirichlet Manifold Harmonics as a spectral analysis tool, to band-limit the high-frequency details of the geometry and



**Figure 1:** Scheme of our compression and decompression.

achieve a very high compression ratio. The Dirichlet Manifold Harmonic Transformation incorporates a novel boundary handling strategy so that both the boundary and feature lines of the original model can be preserved, while the spectral reconstruction guarantees that the decompressed 3D shape will be smooth for the non-feature parts. Our spectral compression method provides a progressive mechanism to trade-off between bit-rate and shape quality. Our experiments show that this method can guarantee the fidelity of user-specified feature lines and achieve very low bit-rate with high geometric shape quality.

## 2. Related Works

Alliez and Gotsman [AG03] categorized existing 3D mesh compression techniques as progressive techniques and single-rate techniques, depending on whether the model is decoded during or only after the transmission. Progressive techniques [TGHL98, COLR99, PR00, KSS00, AD01] aim for the best trade-off between data-size and approximation accuracy, while loss-less single-rate techniques [Dee95, TGHL98, Ros99, GGH02] aim for removing redundant information.

Progressive Mesh [Hop96] transforms triangular meshes into a stream of spatial refinements applied to a simple coarse mesh. The spatial refinements include both connectivity and geometry information of the shapes. Pajarola and Rossignac [PR00] extended it further by grouping refinements into batches, and proposed the Compressed Progressive Mesh (CPM). Khodakovskiy et al. [KSS00] used a multi-resolution analysis tool called wavelets to distinguish large-scale and fine-grain features. It only preserves geometry information and eliminates parameter and connectivity by using semi-regular remeshing.

Spectral compression of mesh geometry proposed by Karni and Gotsman [KG00] is another type of progressive compression technique. It employs the eigenvectors of graph Laplacian operator as bases to perform spectral compression for triangular mesh. It achieves very low bit-rates by discarding the high-frequency information. As a spectral method, it has smooth reconstruction due to the smooth nature of low-frequency basis functions. By parameterizing the mesh onto

the unit disk, a fixed set of spectral bases [KG01] can be used to further reduce the decoding time.

Geometry Image (GIM) [GGH02] is a shape compression technique, which builds completely regular re-meshing for models with arbitrary topology using a rectangular grid. Because the grid implies connectivity information, such information is not stored explicitly. GIM handles parameter information as well as geometry information, which is different from the above mentioned techniques. GIM requires the parameterization of arbitrary surfaces to be on rectangular parameter domain. Thus it could benefit from existing image compression techniques. However, this restriction makes it hard to find an optimal parameterization, especially when the model is topologically complicated. An improper parameterization may result in extreme dense or sparse vertices over the shape. Sander et al. [SWG\*03] tried to overcome this problem by using multi-charts of arbitrary shapes instead of a single rectangle. This reduces parameterization distortion and increases geometric fidelity when regularly sampled on grids. When the multi-charts are packed into a single texture image, however, traditional image compression techniques will not be suitable. Since without paying special attention to the chart boundaries during image compression, it will be hard to guarantee the cross-boundary connectivity and continuity of the overall shape.

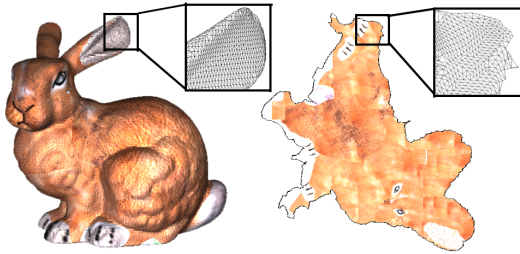
## 3. Scheme of Compression and Decompression

As shown in figure 1, the scheme of this work consists of the following procedures for compression and decompression. During the compression process, the re-sampling stage (section 3.2) replaces the original connectivity of the input surface with regular connectivity which could be generated with much less information in decompression. To preserve important features of the input model, we encode the feature lines and rebuild the connectivity on top of the regular grids (section 3.3). Dirichlet Manifold Harmonic Transform (sections 3.4 and 4) is proposed to transform the geometric information into the spectral domain and encode the transformed spectral coefficients, while satisfying the boundary and feature constraints. During the decompression process (section 3.5), similar process of mesh regeneration will be applied to

generate regular connectivity with the feature and boundary edges. The Inverse Dirichlet Manifold Harmonic Transform will be applied to reconstruct the surface while preserving the feature lines (section 4.5). After that the patches of the model will be stitched together along the cutting boundaries and features as a post-processing step. These procedures are elaborated as follows.

### 3.1. Assumptions about Input Model

The input surface  $\mathcal{M}_o = (M_o, P_o, M_{o,p})$  is assumed to be a valid triangular mesh embedded in  $\mathbb{R}^3$ , with valid parameterization in  $\mathbb{R}^2$ . Here  $M_o$  denotes the triangular mesh in  $\mathbb{R}^3$ ,  $M_{o,p}$  denotes the parametric mesh in  $\mathbb{R}^2$ ,  $P_o : M_o \rightarrow M_{o,p}$  is the parametric mapping (see figure 2). Note that  $P_o$  should be a bijection. We consider  $P_o$  to be a *valid parameterization* if the triangles in  $M_{o,p}$  do not overlap with each other. Suppose  $x$ ,  $y$  and  $z$  are the coordinates of the vertices in  $M_o$ . Since the parameterization is a bijection, we can consider the surface  $\mathcal{M}$  as 3 coordinate functions  $\mathbf{x}$ ,  $\mathbf{y}$  and  $\mathbf{z}$  defined on  $M_{o,p}$ . This method relies on spectrally compressing these three coordinate functions.



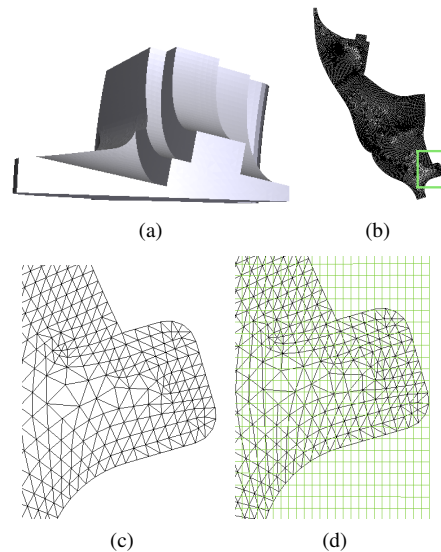
**Figure 2:** Geometric mesh  $M_o$  and parametric mesh  $M_{o,p}$  of the Bunny model (35,190 vertices) with texture mapping.

Different from the geometry image approach [GGH02], this method does not have any additional requirement for the parameterization. For example, the boundaries of the parametric mesh  $M_{o,p}$  do not need to form a rectangular shape, and we can have arbitrary number of parameterization charts. Each chart will be handled independently in our compression method.

### 3.2. Re-sampling

We avoid the encoding of connectivity information by replacing it with the regular connectivity in the parametric domain. Thus the only information we need to encode is reduced to the boundary curves of the parametric mesh. The procedure of parametric re-sampling is employed to create the vertices in a regular manner. This procedure takes 2 integer parameters: the division number  $d_u$  along  $u$  axis and the division number  $d_v$  along  $v$  axis.

As shown in figure 3, a grid  $G$  consisting of  $d_u \times d_v$  vertices is created in the parametric space. Denote the parametric vertices of  $G$  as  $v_{gp}$ . The grid is created so that it could just tightly cover  $M_{o,p}$  according to the max/min parametric coordinates. The division numbers  $d_u$  and  $d_v$  are chosen such that the sampling density of the grid is equivalent to the maximal density of  $M_{o,p}$ , so that there will be sufficient sampling points to capture the geometry of the original mesh. A more detailed discussion is given in section 5.1. All  $v_{gp}$  that fits inside  $M_{o,p}$  are kept to build a replacement model  $\mathcal{M}$ . Denote the set of these eligible vertices as  $V_{gp}$ . Because  $\mathcal{M}_o$  has a valid parameterization, there is no overlapping triangles in  $M_{o,p}$ .  $\forall v_{gp} \in V_{gp}$ ,  $x(v_{gp})$ ,  $y(v_{gp})$ ,  $z(v_{gp})$  are linearly interpolated using existing coordinate functions of  $x_o$ ,  $y_o$ ,  $z_o$ .



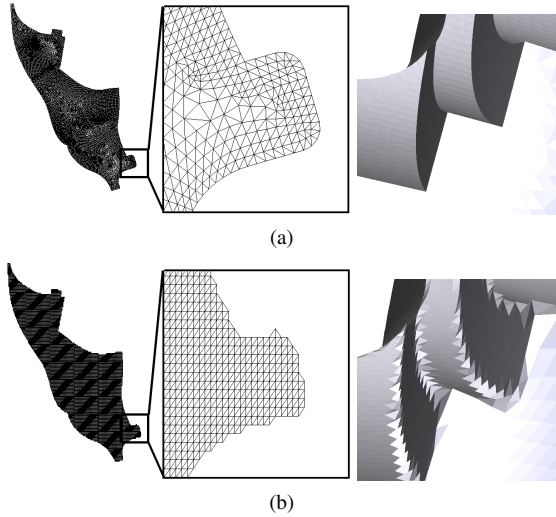
**Figure 3:** Re-sampling: (a) the Fandisk model (6,699 vertices), (b) the parametric mesh, (c) closer look at the green box in (b), and (d) the same area during the re-sampling.

Consider the set of parametric vertices  $V_{gp} = \{v_{gp}\}$  and the set of corresponding 3D vertices  $V_g = \{v_g\}$  created above. It's obvious that  $|V_{gp}| = |V_g|$  holds and each parametric vertex  $v_{gp}$  corresponds to one and only one geometric vertex  $v_g$ . During decompression, if the boundary curves of  $M_{o,p}$  are preserved precisely, these grid parametric vertices  $\{v_{gp}\}$  can be re-generated exactly in a regular manner in  $\mathbb{R}^2$ .

### 3.3. Feature Processing and Connectivity Building

The re-sampling grid generated above do not guarantee to capture the boundary information exactly, neither the sharp feature lines, as shown in figure 4. Building the replacement model  $\mathcal{M}$  using  $V_g$  and  $V_{gp}$  only may introduce inevitable visual distortion. Besides all boundary edges of the parameterization charts, the user may also specify some edges in the original model  $\mathcal{M}_o$  as important feature lines. In this step we

try to incorporate these boundary and feature lines into the regular mesh, and rebuild the connectivity of  $\mathcal{M}$  so that it can capture the boundary and feature lines of  $\mathcal{M}_o$  exactly.

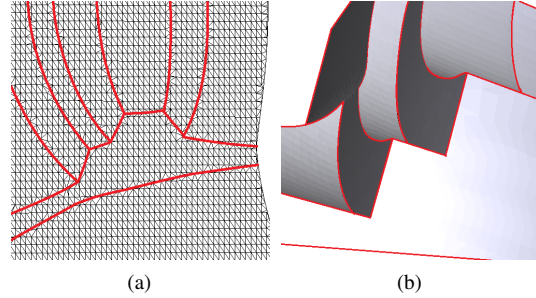


**Figure 4:** (a) The parametric mesh of the Fandisk model and a closer look at the sharp features. (b) The re-sampled grid vertices could capture neither boundary nor sharp features.

Note that there are some automatic algorithms [HPW05] available for computing the salient feature lines. All these edges and their parametric vertices are stored along with their 3D geometric coordinates for decompression. In the rest of this paper, both feature lines and chart boundary edges will be treated uniformly. We denote the set of boundary and feature vertices in the parametric domain as  $V_{pf}$ . We can preserve all the boundary and feature information in the replacement model  $\mathcal{M}$  by preserving  $V_{pf}$  exactly in the compression data.

We take the boundary and feature edges as constraints and create the connectivity (incorporating the regular grid  $G$ ) using the Constrained Delaunay Triangulation (using CGAL library) in the parametric space. Then the replacement model  $\mathcal{M}$  is created, as shown in figure 5. Here  $\mathcal{M}$  has the same geometry of  $\mathcal{M}_o$  except for the regular connectivity that could be re-generated in decompression.

Recall that the grid-sampled vertices  $\{v_{pg}\}$  are generated using parameters of  $d_u$  and  $d_v$  together with the max/min parametric coordinates of the feature vertices  $\{v_{pf}\}$ . Since the feature information will be encoded precisely, we can guarantee to produce the identical re-sampled parametric vertices and hence their connectivity during decompression. Thus these connectivity and non-feature vertex information do not need to be encoded for decompression, which helps to lower the bit-rate significantly. After the procedures described in sections 3.2 and 3.3, the replacement model  $\mathcal{M}$



**Figure 5:** Constructing the connectivity of the Fandisk model with features: (a) closer look at the constructed parametric mesh and (b) closer look at the constructed 3D model. Red lines denote the feature lines.

with regular connectivity and feature information is created.  $\mathcal{M}$  will be compressed and decompressed instead of  $\mathcal{M}_o$  in following procedures of this method. Denote  $\mathcal{M} = (M, P, M_p)$  similarly as  $\mathcal{M}_o$ .

### 3.4. Spectral Decomposition and Compression

We are using spectral analysis to compress the geometric information of the replacement model  $\mathcal{M}$ . Consider the coordinate functions  $x$ ,  $y$  and  $z$  defined over  $M_p$  which is the parametric mesh of the replacement model  $\mathcal{M}$ . By applying Dirichlet Manifold Harmonic Transformation (DMHT) which will be introduced in section 4, these three functions will be transformed into spectra of  $\{c_{x,i}\}$ ,  $\{c_{y,i}\}$  and  $\{c_{z,i}\}$ . Here  $c_i$  stands for the spectral coefficients of a specific function defined over  $M_p$ . They are presented from low frequency to high frequency respectively. Theoretically there are  $n$  coefficients for each function where  $n$  is the number of non-feature and non-boundary vertices in  $M_p$ . Low-frequency coefficients correspond to large-scale descriptions of the shape while high-frequency ones correspond to fine-scale details. These coefficients will be encoded instead of the geometric coordinates for each vertex.

During decompression, the Inverse Dirichlet Manifold Harmonic Transformation (I-DMHT) is used to reconstruct the geometric information for all vertices. Theoretically, I-DMHT could transfer spectra back to the spatial domain without any loss. That is, given the full spectra, I-DMHT could reconstruct the functions precisely. However, similar to Karni and Gotsman's approach [KG00], only low frequency coefficients are encoded for decompression in order to reduce the bit-rate. We can either balance between less storage bits and more details of the model, or provide a progressive compression strategy since high frequency details of the shape can be superimposed independently and successively. Note that the details of our DMHT and I-DMHT are introduced in section 4.

### 3.5. Decompression

As mentioned above, the original connectivity information is discarded. Only the boundary edges and the feature lines are preserved in the code. By repeating the same procedures in sections 3.2 and 3.3, we can reconstruct the identical parametric mesh  $M_{p, re} = M_p$ . Geometric information is encoded as spectral coefficients of the three coordinate functions  $x$ ,  $y$  and  $z$ . Reconstructing these functions depends on the spectral analysis tool of I-DMHT which takes the parametric mesh  $M_{p, re}$  as domain. Thus it is necessary to re-build the identical connectivity information and hence the identical domain mesh for I-DMHT, in order to ensure that the spectral basis functions are identical as the ones used by DMHT during compression.

Because our decompression method guarantees the precise reconstruction of the original boundary for each connected charts, for large models we can segment it into smaller charts and process each chart individually. After decompression these charts will be seamlessly glued together to reconstruct the large model. Note that in Karni and Gotsman's spectral compression approach [KG00], there is no guarantee that the boundaries of the segments can meet together after decompression. Thus we provide an out-of-core strategy for handling spectral compression of gigantic polygon meshes in a piece-by-piece manner [IG03]. Of course the details of such out-of-core spectral compression is beyond the scope of this paper, and will be investigated in our future work.

## 4. Dirichlet Manifold Harmonic Transform

The spectral analysis tool used in our compression method is called the Dirichlet Manifold Harmonic Transform (DMHT). It is used to analyze functions defined over 2-manifold triangular mesh. The DMHT relies on basis functions as a spectral analysis tool. The basis function used here in DMHT is called the Dirichlet Manifold Harmonic Basis (DMHB). Note that in this compression method we apply DMHT over the flat parametric mesh  $M_p$  rather than  $M$  although DMHT also works for  $M$  in  $\mathbb{R}^3$ . So in the following processing edges, angles, triangles and areas of  $M_p$  are used rather than those of  $M$ .

### 4.1. Building Dirichlet Manifold Harmonic Basis

Given an open 2-manifold surface embedded in  $\mathbb{R}^3$  represented by a triangular mesh  $M_p$ , the Discrete Laplace-Beltrami Operator (LBO)  $L$  could be constructed using Finite Element Method (FEM) [RWP06]. Suppose  $M_p$  contains  $m$  boundary vertices  $V_{bd} = \{v_{bd,j}\}$  and  $n$  non-boundary vertices  $V_{nbd} = \{v_{nbd,i}\}$ .  $L$  is an  $(n+m) \times (n+m)$  matrix. Here

$$L = D^{-1}Q, \quad (1)$$

where  $D$  is a diagonal matrix with positive elements and  $Q$  is a symmetric matrix.  $Q_{i,j} = (\cot(\beta_{i,j}) + \cot(\beta'_{i,j}))/2$  for  $i \neq j$  and  $Q_{i,i} = -\sum_j Q_{i,j}$ .  $D_{i,i} = (\sum_{t \in St(i)} |t|)/3$ . Here  $t, t'$  are the triangles that share the edge  $(i, j)$ ,  $|t|$  and  $|t'|$  denote their areas,  $\beta_{i,j}, \beta'_{i,j}$  denote the two angles opposite to the edge  $(i, j)$  in  $t$  and  $t'$ , and  $St(i)$  denotes the set of triangles incident to vertex  $i$ . Without any lose of generality, it is assumed that all boundary vertices are listed after non-boundary vertices in any vector-form function or matrix-form operator.

Consider  $L = D^{-1}Q$  which is an  $(n+m)$ -square matrix. We can rewrite it as:

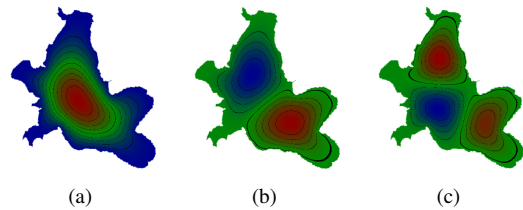
$$L = \begin{bmatrix} D_n & \mathbf{0}_{n,m} \\ \mathbf{0}_{m,n} & D_m \end{bmatrix}^{-1} Q = \begin{bmatrix} D_n^{-1} & \mathbf{0}_{n,m} \\ \mathbf{0}_{m,n} & D_m^{-1} \end{bmatrix} \begin{bmatrix} Q_n & R_{m,n}^T \\ R_{m,n} & Q_m \end{bmatrix}. \quad (2)$$

Here  $D_n, D_m$  are  $n \times n$ - and  $m \times m$ -submatrices of  $D$ .  $Q_n, Q_m$  and  $R_{m,n}$  are  $n \times n$ -,  $m \times m$ - and  $m \times n$ -submatrices of  $Q$ .

By applying the Dirichlet boundary condition on all boundary vertices, i.e.,  $f(v_{bd}) = 0$ , Dirichlet eigenvectors  $H^i$  and corresponding eigenvalues  $\lambda_i$  could be solved. Here the eigen-problem is defined as:

$$\begin{bmatrix} D_n^{-1} & \mathbf{0}_{n,m} \\ \mathbf{0}_{m,n} & D_m^{-1} \end{bmatrix} \begin{bmatrix} Q_n & R_{m,n}^T \\ R_{m,n} & Q_m \end{bmatrix} \begin{bmatrix} H_n \\ \mathbf{0}_m \end{bmatrix} = -\lambda H_n. \quad (3)$$

It could be rewritten as  $D_n^{-1}Q_n H_n = -\lambda H_n$ . Here the tradition of adding minus sign “-” is followed to ensure that  $\lambda \geq 0$  holds. Without losing generality,  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  is assumed. Because Dirichlet eigen-problem does not have the constant value solution as Neumann eigen-problem, we have  $\lambda_i > 0$ . Figure 6 shows the first three Dirichlet eigenvectors on the parametric mesh of Bunny model. Note that the isocontours of the Dirichlet eigenvectors are parallel with the domain boundaries.



**Figure 6:** Dirichlet eigenvectors of  $L$  on the Bunny model with original boundary and no additional feature: (a)  $H^1$ , (b)  $H^2$  and (c)  $H^3$ . The values range from negative minima (blue) to positive maxima (red).

Recall that the discrete LBO  $L$  is defined as  $L = D^{-1}Q$ , where  $D$  is a diagonal matrix with positive elements and  $Q$  is a symmetric matrix. The inner product of two functions  $f$  and  $g$  defined over  $M_p$  is then defined as:

$$\langle f, g \rangle = \mathbf{f}^T \mathbf{D} \mathbf{g}, \quad (4)$$

where

$$\mathbf{f} = \{f(v_1), \dots, f(v_{n+m})\}^T, \quad (5)$$

$$\mathbf{g} = \{g(v_1), \dots, g(v_{n+m})\}^T \quad (6)$$

are vector forms of  $f$  and  $g$ . Then the inner product of  $H^i$  and  $H^j$  can be rewritten as:

$$\langle H^i, H^j \rangle = (H^i)^T D H^j \quad (7)$$

$$= \begin{bmatrix} H_n^i \\ \mathbf{0}_m \end{bmatrix}^T \begin{bmatrix} D_n & \mathbf{0}_{n,m} \\ \mathbf{0}_{m,n} & D_m \end{bmatrix} \begin{bmatrix} H_n^j \\ \mathbf{0}_m \end{bmatrix} \quad (8)$$

$$= (H_n^i)^T D_n H_n^j. \quad (9)$$

It is obvious that equation 9 also holds for any two functions satisfying Dirichlet condition defined over  $M_p$ . Such kind of functions are called *Dirichlet functions*.

Because  $\{H^i\}$  are Dirichlet eigen-functions, we have:

$$LH^i = D^{-1}QH^i = \begin{bmatrix} D_n^{-1}Q_nH_n^i \\ \mathbf{r}_m \end{bmatrix} = \begin{bmatrix} \lambda_i H_n^i \\ \mathbf{r}_m \end{bmatrix}, \quad (10)$$

where  $\mathbf{r}_m$  is an  $m$ -vector that we do not care about. Then we have:

$$(H^i)^T QH^j = (H_n^i)^T Q_n H_n^j \quad (11)$$

$$= (H_n^i)^T D_n D_n^{-1} Q_n H_n^j \quad (12)$$

$$= \lambda_j (H_n^i)^T D_n H_n^j \quad (13)$$

$$(H^i)^T QH^j = (H_n^i)^T Q_n D_n^{-1} D_n H_n^j \quad (14)$$

$$= (D_n^{-1} Q_n H_n^i)^T D_n H_n^j \quad (15)$$

$$= \lambda_i (H_n^i)^T D_n H_n^j. \quad (16)$$

Thus when  $\lambda_i \neq \lambda_j$ , we have  $\langle H^i, H^j \rangle = (H_n^i)^T D_n H_n^j = 0$ . That is,  $\{H^i\}$  are orthogonal to each other, and can serve as bases for our spectral analysis. They are referred as the Dirichlet Manifold Harmonics Basis (DMHB). The corresponding eigenvalues  $\lambda_i$  could be considered as frequencies of the bases.

## 4.2. Spectral Decomposition of Dirichlet Functions

It is easy to verify that any Dirichlet function  $f$  defined over  $M_p$  can be represented as a linear combination of  $\{H^i\}$ . That is:  $\forall f|_{f(v_{bd})=0}, \exists c_1, \dots, c_n$ ,

$$\mathbf{f} = \sum_{i=1}^n c_i H^i, \quad (17)$$

where  $\mathbf{f}$  is the vector-form of  $f$  evaluated at the vertices. Without losing generality, we assume that  $\{H^i\}$  are orthonormal. That is,  $\langle H^i, H^j \rangle = \delta_{i,j}$  where  $\delta_{i,j}$  is the Kronecker delta. Then it's obvious that for any Dirichlet function  $f$  defined over  $M_p$ , we have

$$c_i = \langle f, H^i \rangle = \mathbf{f}^T D H^i. \quad (18)$$

This could serve as a spectral analysis tool for any Dirichlet function over the domain of  $M_p$ .  $\{c_i\}$  are referred as spectral

coefficients. The inverse of this transformation is to reconstruct function  $f_r$  using  $\{c_i\}$ :

$$\mathbf{f}_r = \sum_{i=1}^n c_i H^i. \quad (19)$$

## 4.3. Spectral Decomposition of Non-Dirichlet Functions

We just show that any Dirichlet function over  $M_p$  can be decomposed completely using DMHB. But this type of decomposition does not work for non-Dirichlet functions. In such cases, separating Dirichlet and non-Dirichlet components of the arbitrary function  $f$  is necessary:  $f = f_{dirichlet} + f_{nondirichlet}$ .

One naive way is defining  $f_{dirichlet}(v) \equiv 0$  for vertices in  $V_{bd}$  and  $f_{nondirichlet}(v) \equiv 0$  for other vertices. It is obvious that this naive method creates sharp discontinuity near the boundary edges for the Dirichlet component  $f_{dirichlet}$ . This may decrease the performance of this spectral transformation tool. In this work we propose the following approach to represent the non-Dirichlet component using a harmonic function.

Denote the function to be processed as  $f$  and its vector-form as  $\mathbf{f}$ . By solving the boundary value problem related to Discrete LBO  $L$ , a harmonic function  $f_{harmonic}$  is computed from:

$$\begin{cases} Lf_{harmonic}(v) = 0 & \text{if } v \in V_{in} \\ f_{harmonic}(v) = f(v) & \text{if } v \in V_{bd} \end{cases} \quad (20)$$

This problem has  $n$  unknown variables and  $n$  equations. So it is well-determined, and has only one valid solution  $f_{harmonic}$ .

Thus the original function  $f$  could be represented as  $f = f_{harmonic} + f_{dirichlet}$ . Then:

$$f_{dirichlet} = f - f_{harmonic} \quad (21)$$

$$c_i = \langle f_{dirichlet}, H^i \rangle = \mathbf{f}_{dirichlet}^T D H^i \quad (22)$$

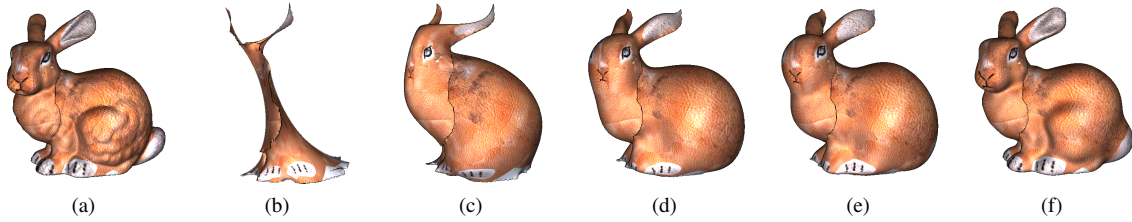
$$\mathbf{f} = \mathbf{f}_{harmonic} + \sum_{i=1}^n c_i H^i. \quad (23)$$

Here  $f_{harmonic}$  is referred as the *harmonic component* of  $f$  and  $f_{dirichlet}$  as the *Dirichlet component*.  $\{c_i\}$  are the spectral coefficients of  $f_{dirichlet}$ . Thus any function  $f$  defined over the open triangular mesh could be transferred into the spectral coefficients  $\{c_i\}$ .

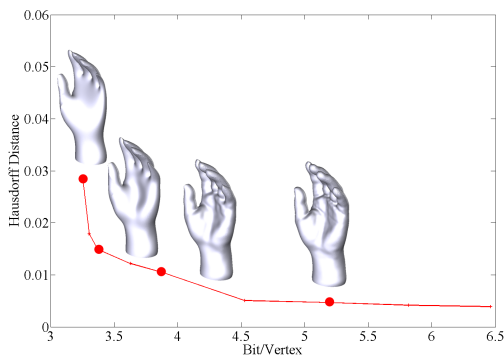
As shown in figure 8, decomposition of the Bunny shape using harmonic component and DMHT performs very well for maintaining the continuity across the chart boundaries. Figure 7 shows our compression performance on the Hand model – the relation curve between the bit-rates and decomposition errors (measured in Hausdorff distance).

## 4.4. Dirichlet vs. Neumann Boundary Condition

For open surfaces, there are two commonly used boundary conditions to solve the eigenvectors of LBO  $L$ : Dirichlet



**Figure 8:** (a) The replacement model of the original Bunny (35,190 vertices) re-sampled using 251x251 grid. (b) The harmonic component of the replacement model. (c) The model reconstructed using 10 coefficients: 2.12 bit/vertex. (d) The model reconstructed using 50 coefficients: 2.17 bit/vertex. (e) The model reconstructed using 100 coefficients: 2.24 bit/vertex. (f) The model reconstructed using 500 coefficients: 2.78 bit/vertex.



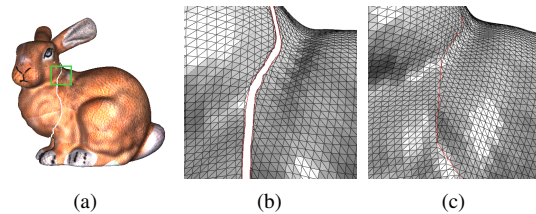
**Figure 7:** Performance of our Dirichlet Harmonic Compression on the Hand model (37,234 vertices).

boundary condition and Neumann boundary condition. The Dirichlet boundary condition specifies  $f \equiv 0$  on the boundary of the surface domain where  $f$  is the solution. It is used in this paper, as mentioned above, to solve the eigen-problem of  $L$  in discrete form.

The Neumann condition, however, does not enforce  $f$  to have specific value at boundary, but requires its derivative in the normal direction of the boundary curve to be zero. On closed manifold surfaces, it is obvious that Dirichlet eigenvectors are the same as Neumann eigenvectors. Neumann eigenvectors  $\{H^i\}$  of  $L$  is used in Manifold Harmonics [RWP06, VL08] as the bases for spectral analysis on closed triangular mesh. For open surfaces, simply solving eigenvectors of  $L$  using the FEM approach [RWP06, VL08] implies the Neumann condition. Unlike Dirichlet eigenvectors, spectral analysis using Neumann eigenvectors does not need to compute the harmonic component, since the geometry of the boundary curves are not zero – they can be computed from the combination of the eigenvectors.

Recall that Neumann condition requires eigen-functions to have derivatives of zero in the normal direction of the boundary curve. In the case of the open triangular mesh,

Neumann eigenvectors  $\{H^i\}$  tend to have similar values for vertices near boundary in the normal direction of the boundary curve. If we reconstruct geometry coordinate functions using  $\{H^i\}$ , the vertices near boundary tend to have similar reconstructed values along the normal direction of the boundary curve. In other words, the boundary of reconstructed model tends to shrink inwards unless the model is reconstructed with full-spectrum. In our compression method high-frequency coefficients are dropped to reduce data size. Thus there would be cracks along the matching boundaries due to shrinking of the surface if we use Neumann eigenvectors instead of Dirichlet eigenvectors, as shown in figure 9.



**Figure 9:** Neumann boundary condition: (a) reconstructed Bunny model using Neumann eigenvectors. There is a crack due to mismatched boundary edges. (b) A closer look at the green square in (a). (c) A closer look at the replacement model of the same area.

#### 4.5. Preserving Feature Lines

In this compression method, we take the parametric mesh  $M_p$  as the domain to analyze geometric coordinate functions  $x$ ,  $y$  and  $z$ . Recall that in the procedure elaborated in section 3.3 all the feature edges are presented as constraints to build the connectivity for re-sampled model. This can guarantee that the re-sampled model  $\mathcal{M}$  holds the specified features of  $M_o$ . But this does not ensure that the reconstructed model  $\mathcal{M}_{re}$  will hold the same feature information for these edges, because we will truncate some high-frequency coefficients

after DMHT. This means that the sharp features tend to be lost if there is no special handling.

Recall that the information of chart boundaries and the feature edges are encoded precisely before computing DMHB. When we are handling the harmonic component and Dirichlet component, we can treat the feature edges in the same way as chart boundaries. That is, the procedure elaborated in section 4.3 is performed with the boundary vertices of  $V_{bdf} = V_{bd} \cup V_f$  rather than  $V_{bd}$ , where  $V_{bd}$  is the chart boundary vertex set and  $V_f$  is the feature vertex set. Although  $V_f$  is treated as boundary vertices, they may be located inside the chart. Note that  $M_p$  may be divided into several separated regions by these feature edges, and will be processed and encoded individually.

## 5. Experimental Details and Results

In our experiments, the input 3D surfaces are assumed to have valid parameterizations. If a model does not come with parametric information, we use the ABF++ [SLMB05] algorithm provided by the Graphite tool to parameterize them. Since we do not have any other requirements on the parameterization except for the assumptions defined in section 3.1, any other parameterization methods [GGH02, SWG\*03, GY03] can be used.

The Hausdorff distance  $d_H$  between the decompressed mesh and the original mesh is used as a metric for shape quality after lossy compression. In our experiments all the spectral coefficients are quantized to 16-bit float numbers.

Experiments are conducted on a Windows platform with Intel 2.66GHz CPU and 4GB memory. The eigen problem is solved in MATLAB. It costs about 20-40 minutes for most models. Other operations are written in Python language, which can be computed in several seconds.

### 5.1. Re-sampling Rate

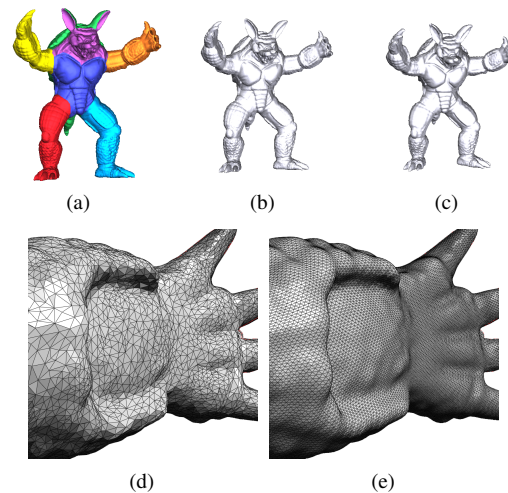
In this work two integer parameters of  $d_u$  and  $d_v$  are used to control the re-sampling rate. They determine how dense the vertices will be in the replacement model  $\mathcal{M}$ . The major concern in deciding re-sampling rate is the shape quality of the replacement model  $\mathcal{M}$ , because under-sampling may introduce too much distortion. The density of grid vertices on the shape depends on parameterization as well as  $d_u$  and  $d_v$  because the regular grid is created in parameter space. In the areas of high density in the parametric mesh, users may need to increase both  $d_u$  and  $d_v$  to achieve satisfying re-sampling density at that specific area. In such cases, users may consider adding additional cuts to the shape so that it would be easier to find a better parameterization and reduce  $d_u$  and  $d_v$  at the same time.

Although over-sampled models does not affect the final compressed data size, it does mean larger matrix eigen-problem to solve in Dirichlet Manifold Harmonics. This may

result in more computational resources. Users need to balance it while deciding  $d_u$  and  $d_v$ . In our current work, the division numbers  $d_u$  and  $d_v$  are chosen such that the sampling density of the grid is equivalent to the maximal density of  $M_{o,p}$ , so that there will be sufficient sampling points to capture the geometry of the original mesh. We will consider using multi-resolution and adaptive sampling grids to accommodate high variations of sampling density of the parametric mesh in our future work.

### 5.2. Choosing Bit-Rate

Large models may contain several charts that would be compressed separately, like the Armadillo model shown in figure 10. There will be no crack between the boundaries of adjacent charts. Those charts may have different bit-rates. By assigning higher bit-rates to charts with more details, users could get lower overall distortion on the shape. By assigning higher bit-rates to charts that people cares more, better visual quality may be achieved.



**Figure 10:** The Armadillo model (168,170 vertices): (a) The original model contains 7 charts. (b) The model compressed with CPM (7.5 bit/vertex). (c) The model compressed with our Dirichlet Harmonic Compression (7.2 bit/vertex). (d) Closer look at the CPM compressed model. (e) Closer look at the model compressed using our approach. Dirichlet Harmonic Compression presents a smoother reconstruction despite the lower bit-rate.

### 5.3. Comparison with Other Approaches

We compared the results of our Dirichlet Harmonic Compression with Karni and Gotsman's approach [KG00], since it also uses spectral analysis to differentiate large-scale and fine-grain descriptions of the shape. As shown in figure 11 and 12, when there are boundaries in the original model,

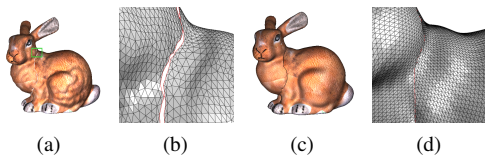


Karni and Gotsman’s method tends to create cracks near matching boundaries and suffers from errors near open boundaries even using a higher bit-rate, since there is no particular boundary handling strategy in their method.

Figure 13 shows the comparison of the compression performance on the Bunny model between our method and Karni and Gotsman’s method.

We also compare our results with Compressed Progressive Mesh (CPM) [PR00] since our compression is also progressive in nature. Figures 10(d) and 10(e) show the comparative results with CPM for the compression of the Armadillo model. Note that our method tends to provide smoother reconstruction due to the essence of spectral analysis. It is easy to prove that the eigen functions of LBO minimizes the smoothness function:  $S(f) = \int_{\mathcal{M}} \|\nabla f(\mathbf{x})\|^2 = \int_{\mathcal{M}} f \Delta_{\mathcal{M}} f = \langle \Delta_{\mathcal{M}} f, f \rangle$ , where  $\|\nabla f\|$  is an estimate of how far apart  $f$  maps nearby points. The sampling rate does not affect bit-rate nor smoothness of the reconstructed model. Figure 14 shows the comparison of the compression performance on the Hand model between the three different approaches. Note that Karni and Gotsman’s method generates large cracks along the boundaries when the bit-rate is low.

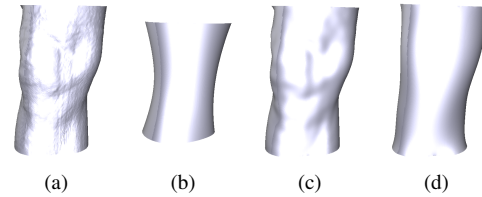
Compared with both of the other two methods [KG00, PR00], our Dirichlet harmonic compression can preserve user-specified feature lines accurately. This is useful especially for models with sharp edges, like the Fandisk model shown in figure 15. Note that our method packs the texture mapping information too while other methods do not.



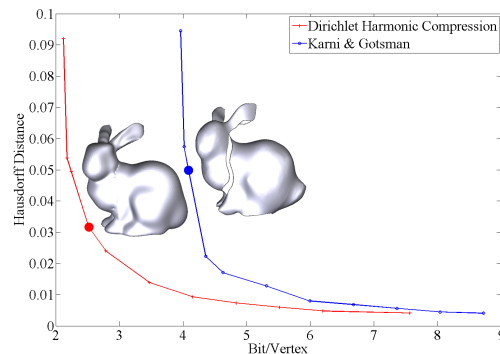
**Figure 11:** (a) The Bunny model (35,190 vertices) with boundaries reconstructed using Karni and Gotsman’s method with 3,000 frequency coefficients (8 bit/vertex). There are cracks near the boundaries. (b) A closer look at green box in (a). (c) The identical model reconstructed using our Dirichlet Harmonic Compression with 1,000 frequency coefficients (3.5 bit/vertex). There is no crack along the boundaries (denoted as red lines). (d) A close look at the same area of (c).

#### 5.4. Ridge Effect

As shown in figure 8, undesirable ridge may appear along the matching boundaries, especially when the bit-rate is low. It is caused by imprecise derivatives of the reconstructed surface along the boundaries. That is, the  $C_1$  continuity across matching boundary is not guaranteed. Enforcing the  $C_1$  continuity across artificial boundaries will be the focus of our future work.



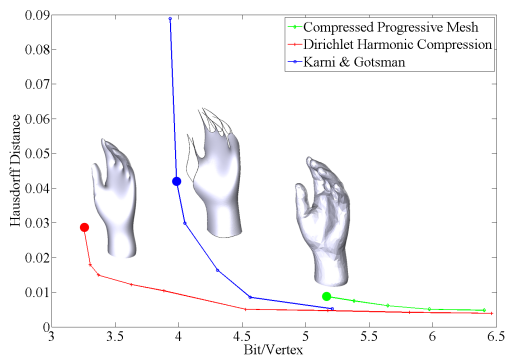
**Figure 12:** (a) The Knee model (37,888 vertices) with boundaries on the top and the bottom; (b) the Knee model reconstructed using Karni and Gotsman’s method with 10 frequency coefficients (3.99 bit/vertex); (c) the Knee model reconstructed with 700 frequency coefficients (2.59 bit/vertex) using our method. (d) the Knee model reconstructed with 10 frequency coefficients (1.71 bit/vertex) using our method. Note that the shrinking effect makes the reconstructed model in (b) shorter.



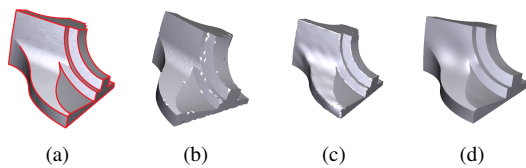
**Figure 13:** Performances for compressing the Bunny model (35,190 vertices): our method vs. Karni and Gotsman’s approach [KG00].

## 6. Conclusion

In this paper we described a spectral shape compression method for triangular mesh based on the spectral analysis tool called Dirichlet Manifold Harmonics. It provides guarantee of fidelity for user-specified feature lines. By re-sampling the shape in parameter space, storage of connectivity information is avoided. By dropping the high frequency details, this method could provide a trade-off between lower bit-rate and higher shape-quality. Our experiments show that it could achieve much lower bit-rate than Karni and Gotsman’s approach [KG00], higher shape-quality than the Compressed Progressive Mesh [PR00], as well as guaranteeing the fidelity of user-specified feature lines that most of other shape compression methods do not provide.



**Figure 14:** Performances for compressing the Hand model (37,234 vertices): our method vs. CPM [PR00], and Karni and Gotsman's approach [KG00].



**Figure 15:** Preserving feature lines: (a) The original *Fandisk* model with 6,699 vertices. User-specified feature lines are marked in red. (b) The model compressed with CPM [PR00]: 18.6 bit/vertex,  $d_H = 0.005684$ . (c) The model compressed with Karni and Gotsman's approach [KG00]: 14.6 bit/vertex,  $d_H = 0.007618$ . (d) The model compressed with our Dirichlet Harmonic Compression: 13.8 bit/vertex,  $d_H = 0.004754$ .

## Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive comments. Yang Liu and Xiaohu Guo are partially supported by National Science Foundation under Grant No. CCF-0727098.

## References

- [AD01] ALLIEZ P., DESBRUN M.: Progressive compression for lossless transmission of triangle meshes. In *SIGGRAPH'01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM, pp. 195–202. 2
- [AG03] ALLIEZ P., GOTSMAN C.: Recent advances in compression of 3d meshes. In *Proceedings of the Symposium on Multiresolution in Geometric Modeling* (2003), Springer-Verlag, pp. 3–26. 2
- [COLR99] COHEN-OR D., LEVIN D., REMEZ O.: Progressive compression of arbitrary triangular meshes. In *VIS'99: Proceedings of the conference on Visualization '99* (Los Alamitos, CA, USA, 1999), IEEE Computer Society Press, pp. 67–72. 2
- [Dee95] DEERING M.: Geometry compression. In *SIGGRAPH'95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), ACM, pp. 13–20. 2
- [GGH02] GU X., GORTLER S. J., HOPPE H.: Geometry images. *ACM Transactions on Graphics* 21, 3 (2002), 355–361. 1, 2, 3, 8
- [GY03] GU X., YAU S.-T.: Global conformal surface parameterization. In *SGP'03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 127–137. 8
- [Hop96] HOPPE H.: Progressive meshes. *SIGGRAPH'96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), 99–108. 2
- [HPW05] HILDEBRANDT K., POLTHIER K., WARDETZKY M.: Smooth feature lines on surface meshes. In *SGP'05: Proceedings of the 3rd Eurographics symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2005), Eurographics Association, p. 85. 4
- [IG03] ISENBURG M., GUMHOLD S.: Out-of-core compression for gigantic polygon meshes. In *SIGGRAPH'03: Proceedings of the 30th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2003), ACM, pp. 935–942. 5
- [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* (2000), pp. 279–286. 1, 2, 4, 5, 8, 9, 10
- [KG01] KARNI Z., GOTSMAN C.: 3D mesh compression using fixed spectral bases. In *Proceedings of Graphics Interface* (2001), pp. 1–8. 1, 2
- [KSS00] KHODAKOVSKY A., SCHRÖDER P., SWELDENS W.: Progressive geometry compression. 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. 271–278. 1, 2
- [PR00] PAJAROLA R., ROSSIGNAC J.: Compressed progressive meshes. *IEEE Transactions on Visualization and Computer Graphics* 6 (2000), 79–93. 2, 9, 10
- [Ros99] ROSSIGNAC J.: Edgebreaker: Connectivity compression for triangle meshes. *IEEE Transactions on Visualization and Computer Graphics* 5, 1 (1999), 47–61. 2
- [RWP06] REUTER M., WOLTER F.-E., PEINECKE N.: Laplace-Beltrami spectra as “shape-DNA” of surfaces and solids. *Computer-Aided Design* 38, 4 (2006), 342–366. 5, 7
- [SLMB05] SHEFFER A., LÉVY B., MOGILNITSKY M., BOGOMYAKOV A.: ABF++: fast and robust angle based flattening. *ACM Trans. Graph.* 24, 2 (2005), 311–330. 8
- [SWG\*03] SANDER P. V., WOOD Z. J., GORTLER S., SNYDER J., HOPPE H.: Multi-chart geometry images. In *SGP'03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 146–155. 2, 8
- [TGHL98] TAUBIN G., GUÉZIEC A., HORN W., LAZARUS F.: Progressive forest split compression. In *SIGGRAPH'98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), ACM, pp. 123–132. 2
- [VL08] VALLET B., LÉVY B.: Spectral geometry processing with manifold harmonics. *Computer Graphics Forum (Proceedings of Eurographics)* 27, 2 (2008), 251–260. 7