LAPPED SPECTRAL DECOMPOSITION FOR 3D TRIANGLE MESH COMPRESSION

P. Rondao-Alface, B. Macq

Laboratoire de Télécommunications
et Télédétection
Université catholique de Louvain
Louvain-La-Neuve, Belgium
{rondao/macq}@tele.ucl.ac.be

F. Cayre, F. Schmitt and H. Maître

Ecole Nationale Supérieure
des Télécommunications
Département TSI - CNRS URA 820
Paris, France
cayre@tsi.enst.fr, {schmitt/maitre}@enst.fr

ABSTRACT

Spectral decomposition of mesh geometry has first been introduced by Taubin for geometry processing purposes. It has been extended to address transmission issues by Karni & Gotsman. Such decompositions give rise to pseudo-frequential information of the geometry defined over the mesh connectivity. For large meshes a piecewise decomposition has to be applied in order to restrict the complexity of the transform. In this paper, we propose to introduce overlap for its spectral representation. We show visual gains obtained in compression and progressive transmission of mesh geometry.

1. INTRODUCTION

Over the past decade, very fruitful efforts have converged to offer tools devoted to the processing of meshes. Meshes are considered 2D surfaces defined in the 3D space. Applications using such data range from collaborative design, art heritage and biomedical to game industry. This paper reviews the mesh geometry spectral decomposition. We improve the technique to progressive transmission purposes.

Spectral decomposition has been used first for geometry compression by Karni and Gotsman[5]. They further improved their method to achieve lower computational time by using fixed basis decomposition. Unfortunately, in a progressive transmission point of vue, these methods based on mesh partitioning lead to artifacts on the borders of the submeshes. We extend their fixed basis concept to introduce spatial overlapping in the spectral representation. However, state-of-the-art mesh compression[1] requires proper connectivity encoding, which we do not address here.

2. SPECTRAL SPACE, MESH SPECTRUM AND BASIS DECOMPOSITION

In this section we define how to compute the spectrum of the mesh geometry by using the Laplacian operator. This may be seen as the equivalent of the DCT for meshes. Taubin first originated this research line when he was investigating filtering[9]. The foundations of this representation relies on the fact that performing a Fourier analysis is equivalent to diagonalize a Laplacian matrix. We first have to define the local Laplacian operator for meshes, and then to derive the transform equations.

2.1. Mesh geometry decomposition

Taubin[9] proposed an original way to apply Fourier analysis to discrete 3D meshes by using the Laplacian operator, and Karni & Gotsman[5] adapted it to compression purposes. Our work follows this track. A mesh consists of a set V of N vertices ($|V| = N$) and a set $E \subset V^2$ of edges. The geometry is specified in Cartesian coordinates $V = (X, Y, Z) \subset \mathbb{R}^3$. In other words, geometry is defined as three vectors (X, Y, Z) of dimension N defined on every node of the mesh graph. One can add per-vertex or per-face attributes such as color, transparency, texture, etc. We focus here on geometrical data defined over the connectivity graph of the mesh and do not treat the per-vertex attributes which could also be transformed using this decomposition. Defining the *star* $\{i^*\}$ of a vertex $v_i \in V$ as:

$$\forall v_j \in V \quad j \in \{i^*\} \quad \Longleftrightarrow \quad (v_i, v_j) \in E \tag{1}$$

and denoting d_i the degree of vertex v_i (i.e : $d_i = |\{i^*\}|$), Taubin defines the Laplacian matrix of the mesh as the $N \times N$ matrix L of generic term:

$$L_{ij} = \begin{cases} 1 & \text{if } i = j \\ -d_i^{-1} & \text{if } j \in \{i^*\} \text{ and } d_i \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The eigenvectors of L form an orthogonal basis of $\mathbb{R}^N$, so one can consider the eigenvalues as pseudo-frequencies of the geometry defined over the mesh graph[1]. These eigenvalues e_i , $0 \leq i \leq N-1$, are bounded by 0 and 2. The transformed vectors of the geometry are obtained by projecting the three X, Y, Z vectors over the basis functions. The eigenvalues are ordered by decreasing magnitude. We thus obtain the following projection system:

$$diag(e_0, ..., e_{N-1}) = B^{-1} L B \tag{3}$$

The columns of B are the basis functions (eigenvectors of L) and the columns of B^{-1} are the dual basis functions. As a result of the transform applied to V, we obtain three vectors of dimension N called spectra or spectral components (P, Q, R):

$$P = BX, \quad Q = BY, \quad R = BZ \tag{4}$$

An exact reconstruction is performed by:

$$X = B^{-1}P, \quad Y = B^{-1}Q, \quad Z = B^{-1}R \tag{5}$$

[1]In the case of a 2D square mesh, the L_{ij} terms are equal to 1, $-1/4$ and 0 providing a discrete approximation of the Laplacian operator.

Ohbuchi[8] proposed to use a Kirchhoff matrix[2] (also known as *combinatorial* Laplacian) to derive the spectral information. Such a matrix exhibits almost the same properties as Taubin's one and enables faster computations.

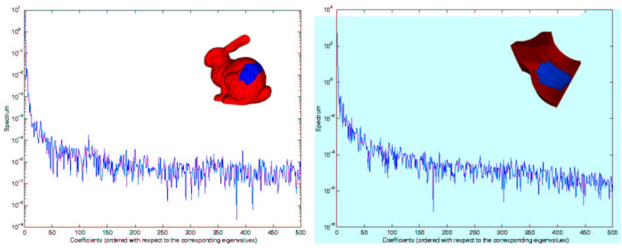

Fig. 1. Spectra (log scale) of the dark grey patch for models *bunny* and *fandisk* (optimal basis).

2.2. Mesh spectra

The sum of the powers of the signal over the three pseudo-frequency axis defines the power spectrum of the Laplacian of V (coefficients sorted with respect to eigenvalues):

$$S_i = \|P_i\|^2 + \|Q_i\|^2 + \|R_i\|^2 \ , \ 0 \le i \le N-1 \qquad (6)$$

It is illustrated in Fig. 1 for two patches extracted from the models *bunny* and *fandisk*.

3. FIXED BASIS DECOMPOSITION, PARTITIONING AND OVERLAPPING

For large meshes (a typical limit today is about 500 vertices), the computation of the eigenvalues may become unstable, and very time-consuming. This computation is generally done in $O(N^3)$, but it may be reduced to $O(N)$ in the case of sparse matrices. For these reasons, meshes have to be partitioned into several sub-meshes (patches) . It is the case in Fig. 1 where the only patches in dark grey are considered to compute the spectra. They are limited to 500 nodes. But, since the eigenvalues depend on the mesh connectivity, one has to perform matrix diagonalization for every patch. In order to improve the efficiency of the method, it may be useful to perform the decomposition on a fixed basis for all the patches. We use the same idea as in an earlier work[6]: a Tutte projection[10] is performed to map the original geometry on a fixed basis for every patch. In the following, we first describe Tutte projection and vertex mapping, which are the two stages for translating from arbitrary to fixed connectivity basis functions. Finally we complement this approach with the presentation of our new patch augmentation scheme by spatial overlapping. Using a fixed basis, we divided by 3 the overall processing time. This comes from avoiding diagonalizing tens of 500×500 matrices.

3.1. Tutte projection

Changing the connectivity of a mesh into a fixed one is generally a difficult problem. A Tutte projection allows to assign 2D coordinates to vertices with respect to their connectivity. The Tutte projection is an iterative process: the vertices of the patch border are fixed on a circle, and 2D coordinates of the inner vertices are updated until every vertex is at a fixed distance from the center of mass of its neighborhood. This process converges to an equilibrium: the vertices are mapped on the disk while respecting their connectivity. Finding a solution in the Tutte disk will translate into a solution in the original connectivity representation. Such a projection is also known as harmonic projection.

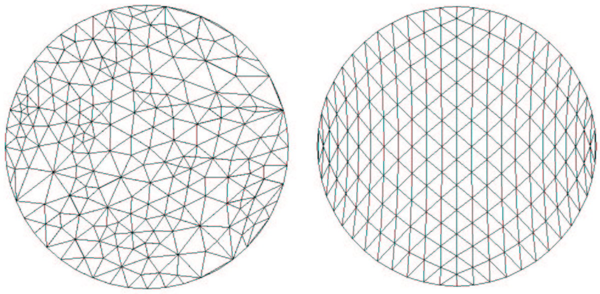

Fig. 2. *Left*: A patch after Tutte projection. The vertices are located at the center of their neighbors. The vertices still have original connectivity but they now have 2D-only coordinates. The grid contains 196 vertices. *Right*: Regular grid on which to map the Tutte projected vertices. The grid contains 196 vertices

3.2. Vertex mapping

The problem amounts to construct a one-to-one correspondence between vertices of the Tutte projection and of a regular grid. To assign every arbitrary connectivity to a fixed one is an assignment problem which is known as bipartite matching. This may be turned into a unit capacity flow problem. The costs of the bipartite graph are the distances between a vertex in the Tutte projection and every vertex in the regular grid. Several methods already exist to solve this problem[3]. We choose the augmenting path algorithm[4], which provides a solution within a couple of seconds for a 500-vertices patch. To have a fixed basis, we just need to add a multiplication by a permutation matrix in the projective system (Eq. 3), because vertex mapping only renumbers the vertices. This step sets the overall decomposition algorithm complexity to $O(N^2)$ (computation of distances between vertices of the Tutte projection and those of the regular grid). The heuristic we used has shown to perform within less than $O(N^2)$.
The main advantage of this method is the use of only one fixed basis for every patch, thus saving both memory and processing time. However, the major drawback is that decorrelation is of lower quality than if performed on the optimal basis. Since vertex mapping changes the connectivity, the geometry is still well reconstructed, but on an associated 6-regular connectivity. There is still decorrelation, but it is not performed on the mesh original connectivity. This drawback is well illustrated within the transmission problem, where the quality of the progressive reconstructed mesh is still significally improved up to the end of the bitstream.

3.3. Partitioning and patch augmentation by overlapping

For partitioning the mesh into patches, we initially used the MeTiS software[7]. This general purpose approach takes the connectivity

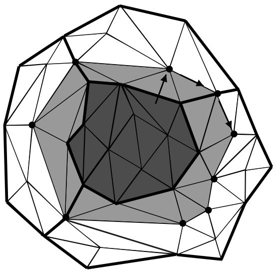

Fig. 3. Patch augmentation by overlapping : in this exemple the mesh is partitioned in 4 patches. The patch being augmented is represented in dark grey. The vertices selected for the overlapping on the neighboring patches are choosen in a spiral way as shown by the arrows. The patch resulting from the augmentation is represented in dark and light gray.

into account regardless of the geometry[2]. This partitioning method provides patches with approximately 400 vertices. A regular mesh has an amount of vertices given by the square of an integer which for small size patches leads to values like $14^2 = 196$, $15^2 = 225$, $16^2 = 256$, etc. For computational reasons we fixed this number to $23^2 = 529$. Therefore we have to increase the amount of vertices of any patch from some 400 or so 529. In their original work[6], Karni and Gotsman introduce some new vertices located in the center of existing triangles. This has no effect on the original geometry.

We choose to augment the patches without any new vertex but rather by adding vertices located in adjacent patches, thus yielding redundancy in the representation because of the overlapping between patches. Vertices are added in a spiral way as described in Fig. 3 .

In practice, it turns out that the Laplacian operator tends to smooth the geometry, preferably pushing the vertices in the interior of the patches. With this added redundancy, acting like a LOT[3] for meshes, we improve the quality of the reconstructed mesh, at a reduced extra cost. Vertices belonging to more than one patch are reconstructed by computing the mean of the geometrical values from the patches they belong to.

4. MESH GEOMETRY COMPRESSION AND PROGRESSIVE TRANSMISSION

In the previous section, it was explained how to decompose geometry over the mesh connectivity and how to reconstruct it. From a compression or transmission point of view, we have to make sure that connectivity has already been decoded before beginning decoding any spectral coefficient. If connectivity is sent first, the original patches yielding the basis functions may be easily retrieved. Then, one sends the geometrical information either with respect to the geometrical relevance of the coefficients or patch by patch. Compression is achieved by quantization of the spectral coefficients followed by entropy coding.

[2]The patches created with this software are generally not homotopic to a disk, which is a condition of optimality for the Tutte projection. Taking the geometry and topological information into account for partitioning leads to better results.

[3]*Lapped* Orthogonal Transform.

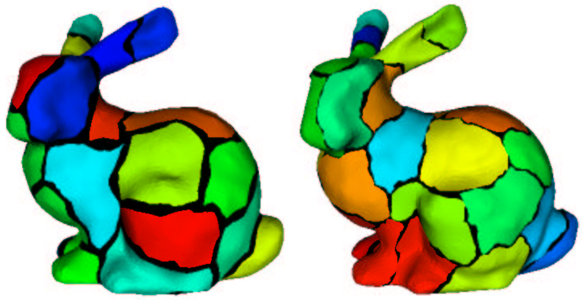

Fig. 4. *Left*: *Bunny* model with 500-vertices patches. 5% of the spectral coefficients are used for reconstruction. No patch overlapping. $D_{vis} = 7.84 \times 10^{-6}$. *Right*: *Bunny* with 400-vertices patches, augmented to 480 vertices (20% overlapping). 5% of the spectral coefficients were used for reconstruction. The quality of the reconstruction is improved, particularly on the border of the patches (cells drawn in black). $D_{vis} = 7.01 \times 10^{-6}$. See Eq. 8 for visual distance definition.

With a correct bitstream structure, it becomes easy to progressively send the spectral coefficients in the spectra order. However, in the case of on-the-fly decoding of the geometry, the first approximations of the models are often of poor quality. We address this issue with our patch augmentation strategy: the geometry keeps relatively smooth because overlapping was introduced in the decomposition. The low frequencies are better captured. In order to qualify the methodology, we have to define a metric to evaluate our results. We follow Karni & Gotsman proposal[5].

4.1. Visual metric

We first need to define a geometrical Laplacian for every vertex. This Laplacian will have to penalize the increase of the distance of a vertex to the center of its neighbors (i.e: the barycentric position). Let l_{ij} be the distance of vertex v_i to vertex v_j. The geometrical local Laplacian operator is defined as:

$$GL(v_i) = v_i - \frac{\sum_{j\in\{i^*\}} l_{ij}^{-1} v_j}{\sum_{j\in\{i^*\}} l_{ij}^{-1}} \quad (7)$$

The local Laplacian represents the local smoothness of the mesh at vertex v_i. The visual distance between a mesh M and another mesh M' with the same connectivity must combine global distances between vertices and distances between local Laplacians. Such a visual distance D_{vis} then takes into account raw PSNR-like information and local smoothness difference:

$$\begin{aligned} D_{vis}(M, M') &= \frac{1}{2|V|}\Big(\sum_{i=0}^{N-1} \|v_i - v'_i\| \\ &+ \sum_{i=0}^{N-1} \|GL(v_i) - GL(v'_i)\|\Big) \quad (8) \end{aligned}$$

This metric has the advantage to differentiate between random noise addition on the vertices and poor reconstruction quality: the eye better accomodates on a smooth object rather than on a noisy mesh. The metric increases in case of local disturbances which generally results in a poor quality to the human eye (see Fig. 4).

Unfortunately, the order of magnitude of the visual metric depends on the mesh. It will not be the same for *bunny* and *fandisk* (resp. 10^{-6} and 10^{-5}).

4.2. Geometric compression

Partitioning was performed by MeTiS[7] algorithm. The vertex mapping heuristic makes use of the augmenting path method[4]. The P, Q, R vectors are quantized and entropy coded. A typical range of quantization for geometrical information is from 10 to 16 bits per coordinate. Actually, it turns out that spectral coefficients need more precision in their representation than spatial coefficients. At this stage we used a uniform scalar quantizer and a quick hand-made Huffman coder. If performances are the final goal of the process, more work may be done to better compress the data. Fig. 5 presents the rate/distortion curve for model *venusc* and *head*. We naturally used optimal basis for compression. We found that our algorithm better performs when using 14 bits for uniform quantization.

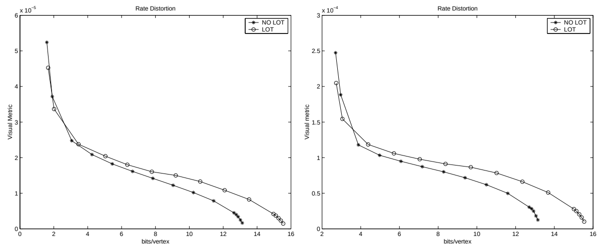

Fig. 5. Rate-distortion curves : *venus* (left) and *head* (right) models for compression (optimal basis). Uniform quantization precision was 16 bits. We could not distinguish the reconstructed model from $D_{vis} = 1.3 \times 10^{-5}$ (*venus*) and $D_{vis} = 1.4 \times 10^{-4}$ (*head*)on. Overlapping results approximately in a 1 bit/vertex overhead, but provides much better quality at the beginning of the reconstruction.

4.3. Progressive transmission

We simulated on-the-fly decoding during progressive transmission. In order to better represent the geometrical artifacts of this transform, we used a CAD object (which has flat areas and orthogonal adjacent surfaces), see Fig. 6. In the case of such a CAD model like *fandisk*, one must wait for the whole data to get rid of spurious waves on the flat areas. This demonstrates that spectral decomposition used towards compression of CAD data is a bad choice. However, we focused on models coming from natural signals, like animals, faces and so on.

5. CONCLUSION AND FUTURE WORK

Our goal was to validate the spectral decomposition of mesh geometry as a powerful tool to both compress and transmit the geometry. As we got more and more background with this tool, we were able to define the major orientations of our future work. We need better partitioning where all patches project on a disk and maybe the use of an harmonic projection which would lead to better decorrelation. We believe all processings would take advantage to be done

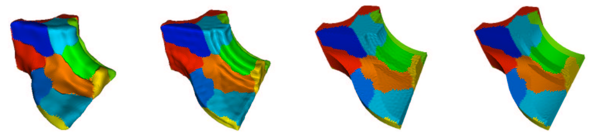

Fig. 6. Various stages of on-the-fly mesh geometry decoding. We took 20% overlapping (480 vertices per patch after augmentation). On CAD objects, many coefficients are needed to better reconstruct the geometry. $D^{5\%}_{vis} = 7.7 \times 10^{-5}$, $D^{15\%}_{vis} = 4.56 \times 10^{-5}$, $D^{75\%}_{vis} = 1.3 \times 10^{-5}$, $D^{90\%}_{vis} = 0.62 \times 10^{-5}$. The model is well reconstructed from $D_{vis} = 5 \times 10^{-6}$ on.

in a patch-dependent basis. These are some of the improvements we are currently working on.

6. ACKNOWLEDGEMENTS

The authors would like to acknowledge the suggestions of Isabelle Bloch (ENST/TSI) and Vincent Blondel (UCL/INMA).

7. REFERENCES

[1] P. Alliez, M. Desbrun *Progressive compression for lossless transmission of triangle meshes*, Proceedings of SIGGRAPH'01, pp. 198-205, 2001.

[2] B. Bollabas *Modern graph theory*, Springer, 1998.

[3] B.V. Cherkassky, A.V.Goldberg, J.C. Setubal *Augment or Push? A computational study of bipartite matching and unit capacity flow algorithms*, NEC Research Institute TR # 98-036R, 1998.

[4] R. Jonker, A. Volgenant *A shortest augmenting path for dense and sparse linear assigment problems*, Computing, 38, pp. 325-340, 1987.

[5] Z. Karni, C. Gotsman *Spectral compression of mesh geometry*, Computer Graphics (Proceedings of SIGGRAPH), pp. 279-286, 2000.

[6] Z. Karni, C. Gotsman *3D mesh compression using fixed spectral basis*, Proceedings of Graphics Interface, pp. 1-8, Ottawa, June 2001.

[7] G. Karypis, V. Kumar, *MeTiS: A software package for partitioning unstuctured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. Version 4.0*, Univ. of Minnesota, Dept. of Computer Science, 1998.

[8] R. Ohbuchi, S. Takahashi, T. Miyazawa, A. Mukaiyama *Watermarking 3D polygonal meshes in the mesh spectral domain*, Proceedings of Graphics Interface, pp. 9-18, Ottawa, June 2001.

[9] G. Taubin, T. Zhang, G. Golub, *Optimal surface smoothing as filter design*, IBM Technical Report RC-20404, March 1996.

[10] W.T. Tutte, *How to draw a graph*, Proceedings of the London Mathematical Society, Vol. 10, pp. 304-332, 1960.