SHAPE APPROXIMATION FOR EFFICIENT PROGRESSIVE MESH COMPRESSION

K. Mamou[1,2]*, C. Dehais*[1] *F. Chaeib*[2]*, F. Ghorbel*[2]

(1) FittingBox *

(2) Groupe de Recherche GRIFT/ENSI †

ABSTRACT

This paper presents an original lossless multi-resolution 3D mesh compression approach, referred to as Shape Approximation-based Progressive Mesh (SAPM). The SAPM codec statically compresses the mesh connectivity and exploits it to build a smooth approximation of the original mesh, which is then decimated in order to derive a progressive mesh hierarchy exploited in order to efficiently predict and progressively transmit the geometry approximation errors. The SAPM codec supports both spatial and quality scalabilities, while exhibiting high rate-distortion performances. Experimental evaluation shows that the proposed technique is 40-50% more efficient than the state-of-the-art lossless multi-resolution 3D mesh compression techniques.

Index Terms— Compression, quality scalability, spatial scalability, progressive mesh, shape approximation.

1. INTRODUCTION

The number of network-based 3D graphics services has been exponentially growing during the last five years (e.g. on-line gaming, virtual worlds and virtual, e-medicine, try-on applications[1]). With the increasing processing power of CPUs/GPUs, the development of powerful 3D modeling softwares and the proliferation of 3D capturing devices, the generated 3D contents are becoming more and more rich and realistic. Thus, efficiently exchanging such complex 3D models via heterogenous networks with limited bandwidth and between different terminals (e.g. smart phones, PCs or PDAs) is becoming a crucial challenge for various industrial applications (e.g. photo-realistic rendering, medical visualization, and computational simulation). Furthermore, for collaborative, medical and engineering applications, recovering the original mesh connectivity is a strong requirement. Therefore, the design of efficient 3D compression technologies especially targeting dense 3D meshes and supporting lossless connectivity encoding while providing advanced functionalities such as spatial and quality scalabilities has been widely studied in the literature (see [1] for a detailed review).

In his pioneering work [2], Hoppe introduced the *Progressive Mesh* (PM) approach for multi-resolution 3D mesh coding. The PM technique represents the original mesh M as a base mesh M_0 together with a set of *vertex split* refinement operations (*cf.* Figure 1). The coarse mesh M_0 is obtained by successively decimating M with edge-collapse decimation operations (*cf.* Figure 1). The set of refinement operations, denoted $(vsplit_l)_{l\in\{1,\ldots,L\}}$ (L being the number of LODs), is computed by storing at each step, the position/attribute of the decimated vertex v_t, the index of the refined vertex v_s and the local indices of its left and right vertices v_l and v_r (*cf.* Figure 1).

The PM representation is particularly well suited for progressive transmission and rendering. Furthermore, it provides high quality intermediate LODs since the decimation path is chosen to minimize the shape distortions. However, as pointed out in [3], the compression rates of the PM approach are heavily penalized by the high amount of bits required to locate the vertex split operations (i.e. 9-14 bits per vertex). The local indices of the vertices v_l and v_r are encoded by using 5 bpv on average. With a total of 15-19 bpv, the PM technique encodes the connectivity information 10-20 times less efficiently than the static mesh encoder [4].

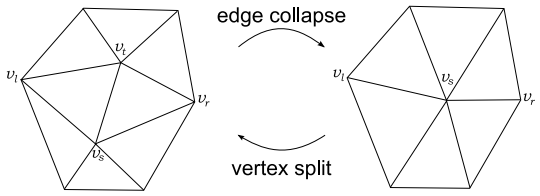

Fig. 1: Edge collapse and vertex-split operations.

Since its introduction in [2], the PM principle has been constantly exploited to design more compact multi-resolution representations for 3D meshes. As discussed in [1], the different progressive codecs have achieved higher compression rates (i.e. 2-10 bpv for the connectivity information) by introducing either a coarser refinement operation or a partially or totally constrained decimation strategy. Note however that the reported gains are obtained at the cost of lower LODs quality. In fact, the topological and geometric constraints underlying the state-of-the-art approaches lead, especially for non-uniformly sampled meshes, to sub-optimal or even poor

*2 Rue de la Roquette, 75011 Paris-France.

†University of Manouba, 2010 Manouba-Tunisia.

[1]www.fittingbox.com

approximations at low bitrates.

To overcome such limitations, in our Shape Approximation-based Progressive Mesh (SAPM) approach, PM hierarchy is built at the decoder side instead of explicitly encoding it into the bitstream. In order to build high quality LODs, the PM representation is computed by applying the Quadratic Error Metric (QEM) simplification technique [5] to a smooth approximation of M. The SAPM technique supports both quality scalability (i.e. progressively refining the precision of coordinates/attributes as the bitstream is decoded) and spatial scalability (i.e. adapting the mesh resolution to the terminal rendering performances and to the available bandwidth), while exhibiting high rate-distortion performances (*cf.* Section 3).

The remainder of this paper is structured as follows. Section 2 describes the proposed technique. The compression performances of the SAPM codec are objectively evaluated in Section 3. Finally, Section 4 concludes the paper and suggests possibilities for future work.

2. SHAPE APPROXIMATION-BASED PROGRESSIVE MESH

Figure 2 presents the synopsis of both the SAPM encoder and decoder.

As in [6], the SAPM encoder exploits the fact that the connectivity information represents less than 20% of the entire compressed bitstream and therefore compresses it statically by using the single rate encoder [4]. It is then used in order to progressively compress the geometry information. More precisely, the SAPM technique exploits the mesh connectivity in order to derive a smooth approximation of the original mesh M, denoted M^*. M^* is entirely described by: (1) the connectivity information, (2) a small set of N control points denoted $\mathcal{C} = (c_k)_{k\in\{1,...,N\}}$ and (3) the set of indices $\mathcal{S}$ of the salient edges (*cf.* Section 2.1). The positions $(P_{c_k})_{k\in\{1,...,N\}}$ of control points are quantized and arithmetically encoded together with the indices $\mathcal{C}$ and $\mathcal{S}$. A progressive mesh hierarchy, denoted $PM(M^*)$, is then constructed by decimating M^* as described in [5], allowing only half-edge collapse operations. $PM(M^*)$ is then exploited in order to predict the approximation errors of M^* w.r.t. M (*cf.* Section 2.2). Finally, the predicted approximation errors $(e_v)_{v\in\{1,...,V\}}$ (V being the number of vertices) are arithmetically encoded and progressively transmitted to the decoder.

The SAPM decoder proceeds as follows. First, the mesh connectivity, the positions $(P_{c_k})_k$ and the indices $\mathcal{C}$ and $\mathcal{S}$ are decoded and exploited to compute the approximation M^*. M^* is then simplified by applying [5] to build exactly the same progressive mesh structure $PM(M^*)$ computed by the encoder. Note here that the PM hierarchy is obtained at no extra cost since it is entirely derived from M^*. Furthermore, $PM(M^*)$ produces high quality LODs since it is generated by the QEM simplification strategy, which minimizes the shape distortions without being penalized by any topological or geometric constraints. Finally, the predicted approximation errors $(e_v)_v$ are progressively decoded and used to reconstruct the different LODs as described in Section 2.2.

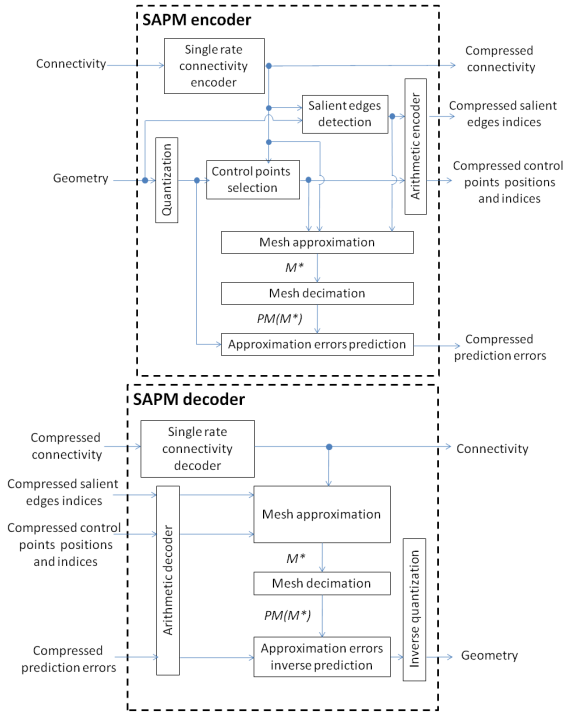

Fig. 2: Synopsis of the SAPM encoder and decoder.

2.1. Laplacian-based Mesh Approximation

The SAPM codec extends the Laplacian-based approximation technique described in [6] to meshes with salient features. In this work, we have considered as salient all the edges exhibiting a dihedral angle higher than $\frac{\pi}{6}$. We define the Laplacian matrix L as follows:

$$\forall (i,j) \in \{1,...,V+C\} \times \{1,...,V\},$$

$$L_{i,j} = \begin{cases} 1 & \text{if } (j=i) \\ -\frac{\alpha}{\alpha\times|i_s^*|+\beta\times|i_n^*|} & \text{if } (i\leq V) \text{ and } (j\in i_s^*) \\ -\frac{\beta}{\alpha\times|i_s^*|+\beta\times|i_n^*|} & \text{if } (i\leq V) \text{ and } (j\in i_n^*) \\ 1 & \text{if } (i>V) \text{ and } (j\in\mathcal{C}) \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

where,

- i_s^* is the set of topological neighbours of the vertex i sharing with it either a boundary edge (i.e. adjacent to exactly one triangle) or a salient edge (i.e. it belongs to $\mathcal{S}$),
- i_n^* is the set of the neighbours of the vertex i not belonging to i_s^*,

- $|i_n^*|$ and $|i_s^*|$ are the number of elements of i_n^* and i_n^* respectively,
- α and β are the weights associated to the special edges (i.e. salient or boundary edges) and to the non-special ones, respectively.

As in [6], the $V \times 3$ matrix of approximated vertex positions, denoted $P^* = (P_v^*(k))_{v\in\{1,\dots,V\}}^{k\in\{1,2,3\}}$, is computed by solving the following sparse linear system:

$$(L^T L) \times P = B. \quad (2)$$

The $V \times 3$ matrix B is given by:

$$B_{i,k} = \begin{cases} P_i(k) & \text{if } (i \in \mathcal{C}) \\ 0 & \text{otherwise} \end{cases}, \quad (3)$$

where $P_i(1)$, $P_i(2)$ and $P_i(3)$ represent the original cartesian x, y and z coordinates of the vertex i, respectively.

Note that if the weights α and β are equal or if there are no special edges, we obtain exactly the definition of the Laplacian matrix proposed in [6]. In this work, α was set to 100 and β to 1. This implies that a vertex i located on a mesh boundary or on a salient edge is a 100 times more influenced by its special neighbours i_s^* than by the non-special ones (i.e. i_n^*). As illustrated in Figure 3, this modified version of L better preserves the salient features of the mesh.

The set of control points $\mathcal{C}$ is iteratively chosen in order to minimize at each iteration the maximal approximation error. The algorithm proceeds as follows. The first control point is randomly chosen. Then, at each iteration, the linear system (2) is solved and the vertex with the maximal approximation error is added to $\mathcal{C}$. This process is iterated until the number of control points reaches 1% of V.

(a) Original (b) Approach [6] (c) Proposed

Fig. 3: Shape approximation of the "Sword" model (a): (b) uniform Laplacian vs. (c) salient features aware Laplacian.

2.2. Approximation Errors Prediction

The SAPM encoder compresses the predicted approximation errors in the reverse order of $PM(M^*)$. At each step, the predicted approximation error e_v associated with the vertex v is computed as follows:

$$e_v = (P_v - \frac{1}{|v^*|}\sum_{w\in v^*} \hat{P}_w) - (P_v^* - \frac{1}{|v^*|}\sum_{w\in v^*} P_w^*), \quad (4)$$

where v^* represents the set of topological neighbours of v in the current LOD of $PM(M^*)$ and $(\hat{P}_w)_{w\in v^*}$ are the positions reconstructed by the decoder as described in the following (*cf.* equation (7)).

The obtained error e_v is then decomposed into a normal component e_v^n and two tangential ones e_v^t and e_v^r, defined by:

$$e_v^n = e_v \cdot n_v^*, e_v^t = e_v \cdot t_v^*, e_v^r = e_v r_v^*, \quad (5)$$

where n_v^* is the normal of M^* at vertex v and r and t are two vectors chosen to form a direct orthonormal basis with n_v^*.

Finally, e_v^n, e_v^t and e_v^r are quantized and arithmetically encoded. As noted in [7], for fine meshes, the normal component e_v^n contains more information than the tangential ones (i.e. e_v^t and e_v^r). Therefore, we quantize more finely e_v^n than e_v^t and e_v^r.

The SAPM decoder progressively decompresses the approximation prediction errors starting from the lower LOD to the higher one. Here, at each step, the three components $(\hat{e}_v^n, \hat{e}_v^t, \hat{e}_v^r)$ are arithmetically decoded, de-quantized and then used to reconstruct the approximation error $\hat{e}_v$, as follows:

$$\hat{e}_v = \hat{e}_v^n.\hat{n}_v^* + \hat{e}_v^t.t_v^* + \hat{e}_v^r.r_v^*. \quad (6)$$

Finally, the decoded positions $(\hat{P}_v)_v$ are given by:

$$\hat{P}_v = \hat{e}_v + \frac{1}{|v^*|}\sum_{w\in v^*} \hat{P}_w + (P_v^* - \frac{1}{|v^*|}\sum_{w\in v^*} P_w^*). \quad (7)$$

Note that by encoding/decoding the vertices in the reverse order of $PM(M^*)$, the SAPM encoder/decoder guaranty that when processing the vertex v, the positions $(\hat{P}_w)_{w\in v^*}$ of all its neighbours have already been reconstructed. Furthermore, by exploiting the progressive mesh structure $PM(M^*)$, the SAPM codec directly supports the spatial scalability functionality. The quality scalability is obtained by reconstructing all the vertex positions, while setting to zero the non-decoded predicted approximation errors $\hat{e}_v$ in equation (7).

2.3. Computational complexity

The SAPM decoding complexity is determined by the mesh decimation procedure which is in $O(Vlog(V))$. The encoding times are dominated by the control point selection procedure, which requires solving the linear system (2) N times. Here, the computational complexity is quadratic with the number of vertices. As a reference, on a 2.4 GHz Intel Core 2 Duo CPU with 3 Gbytes of RAM, the SAPM decoder processes 10K vertices per second on average. The SAPM encoder compresses a mesh with 20K vertices in 3 minutes.

3. EXPERIMENTAL RESULTS

The compression performances of the SAPM approach have been objectively evaluated on a data set of nine 3D models

with different complexities (6K-200K vertices), topologies and geometries. As distortion measure, we have considered the L^2 error evaluated by the MESH tool[2]. The bitrates are reported in bpv.

Figure 4 and Table 1 compare the rate-distortion curves and the compression performances of the proposed SAPM technique to those of the single rate approach [4] and to the multi-resolution codecs [7] and [8]. The bitrates reported in Table 1 correspond to compression distortions equivalent to a 12 bits uniform quantization.

As shown in Figure 4, the SAPM technique offers LODs with a better quality starting from 2 bpv. For bitrates lower than 2 bpv, the rate-distortion performances of SAPM are almost equivalent to those of [8]. In fact, for this bitrates range, the binary stream is dominated by the statically compressed connectivity information, which slightly reduces the compression performances of SPAM. Note however that by sending the entire mesh connectivity at the beginning of the transmission, the SAPM fully supports the quality scalability functionnality, which is not the case of [8] and [7].

The compression performances reported in Table 1 clearly show that the proposed SAPM progressive geometry encoding strategy outperforms the simple predictors considered by the single rate codec [4] and the progressive techniques [7] and [8], with average gains ranging from 30 to 50%.

4. CONCLUSION & PERSPECTIVES

In this article, we have presented an original multi-resolution 3D mesh compression technique, referred to as SAPM. The SAPM approach exploits a shape approximation-based strategy in order to progressively compress the geometry information, while supporting both spatial and quality scalabilities. The conducted experimental evaluation shows that the SAPM codec outperforms the the state-of-the-art lossless multi-resolution codecs, with 40 to 50% average gains.

Future work will concern the design of a low-complexity control points selection procedure, as well as an efficient compression strategy for mesh attributes such as normals and texture coordinates.

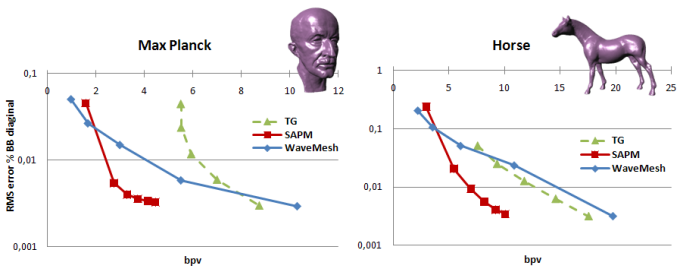

Fig. 4: Rate-distortion curves: SAPM vs. the techniques [4] and [8].

[2]http://mesh.berlios.de

Model	V	SAPM bpv	[4] bpv	[8] bpv	[7] bpv
”Fandisk”	6K	11.4	13.4	18.26	25
”Mannequin”	12K	7.8	13.1	14.45	19.9
”Horse”	20K	10	17.5	19.7	20.9
”Dino”	24K	15.1	21.7	18.8	20.5
”Dilo”	27K	6.2	11.3	11.1	19.8
”Screwdiver”	27K	9.2	12.6	15.7	21.8
”Bunny”	35K	11.62	12.45	16.3	19.8
”Feline”	50K	12.1	16.6	19.2	19.9
”Max Planck”	200K	4.5	8.7	10.3	13.35
Average gain	-	0%	31%	40%	52%

Table 1: Compression performances: SAPM vs. the codecs [4], [8] and [7].

5. REFERENCES

[1] J. Peng, C.-S. Kim, and C.-C.J. Kuo, “Technologies for 3D mesh compression: A survey,” *Journal of Visual Communication and Image Representation*, vol. 16, no. 6, pp. 688–733, December 2005.

[2] H. Hoppe, “Progressive meshes,” in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 99–108.

[3] G. Taubin, A. Gueziecand, W. Horn, and F. Lazaru, “Progressive forest split compression,” in *International Conference on Computer Graphics and Interactive Techniques*, 1998, p. 123132.

[4] C. Touma and C. Gotsman, “Triangle mesh compression,” in *Proceedings of Graphics Interface’98*, 1998, pp. 26–34.

[5] M. Garland and P.S. Heckbert, “Surface simplification using quadric error metrics,” in *International Conference on Computer Graphics and Interactive Techniques archive*, 1997, pp. 209 – 216.

[6] D. Chen, D. Cohen-Or, O. Sorkine, and S. Toledo, “Algebraic analysis of high-pass quantization,” *ACM Transactions on Graphics*, vol. 24, pp. 1259–1282, 2005.

[7] P. Alliez and M. Desbrun, “Progressive compression for lossless transmission of triangle meshes,” in *International Conference on Computer Graphics and Interactive Techniques*, 2001, p. 195202.

[8] S. Valette and R. Prost, “A wavelet-based progressive compression scheme for triangle meshes : Wavemesh,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 2, pp. 123–129, 2004.