

AN EFFICIENT SUBDIVISION INVERSION FOR WAVEMESH-BASED PROGRESSIVE COMPRESSION OF 3D TRIANGLE MESHES

Sébastien Valette*, Jarek Rossignac** and Rémy Prost*

*CREATIS, INSA Lyon, France

** GVU Center, Georgia Institute of Technology, USA

ABSTRACT

Wavemesh is a powerful scheme for 3D triangular mesh processing. In sharp contrast with other approaches using wavelets for mesh compression which apply only to meshes having subdivision connectivity, Wavemesh can simplify, approximate, and compress meshes even if they do not respect this constraint, with unmatched results for progressive lossless compression when compared to other approaches. We propose in this paper an improvement for our scheme : higher efficiency for meshes with large subdivision connectivity sets, as shown by experimental results. Also, in some cases, wavemesh can even perform better than mono-resolution approaches in terms of connectivity compression.

1. INTRODUCTION

Digital models of three-dimensional shapes are essential in numerous application domains, including CAD, entertainment, medicine, geosciences, and architecture. In most cases, these models are represented as 3D triangular meshes, with the native support provided by graphics processors. However, the use of large and complex models can become an issue when the end-user faces storage or transmission (bandwidth limitation) issues. Data compression comes as a solution for these problems by two different approaches: The model can be compressed in a mono resolution bitstream, or in a progressive bitstream. Mono resolution approaches [8, 13, 12] often give the best compression rates, but progressive schemes [1, 4, 9, 11, 5, 3, 6, 18] offer more flexibility : for a given model, several resolution levels are constructed so that progressive transmission and reconstruction are possible. Also, lower resolution levels can be rendered faster which can be useful when the model size becomes a problem.

In section 2, we briefly mention several progressive compression approaches for 3D triangular meshes, and review the Wavemesh approach which we improved. Section 4 explains how Wavemesh is improved to achieve optimal performance when processing subdivision connectivity meshes. Section 5 illustrates the good behavior of the new approach and a conclusion follows.

2. RELATED WORKS

Progressive transmission for meshes was introduced by Hoppe [5], based on successive mesh simplification by edge contractions, which offer optimal granularity and linear complexity cost, but non-linear storage and transmission cost. Several approaches derive from Progressive Meshes: Pajarola and Rossignac implemented an improved version of progressive meshes [11], encoding the mesh connectivity reconstruction by edge splits batches, with

an average coding cost of 7.2 bits per vertex. Karni and Gotsman [6] improved the edge contraction sequence and the geometry coding to enhance both progressive transmission rate-distortion trade-off and rendering speed of the processed meshes.

Cohen-Or et al. [3] propose a progressive transmission based on successive vertex removal followed by deterministic retriangulation. Vertices are removed according to their valence and their geometric properties. The deterministic retriangulation leads to an average connectivity compression of 6 bits per vertex, but the simplified mesh becomes less and less regular due to the Z-retriangulation used for large valence vertices. Alliez and Desbrun [1] improved this technique by removing only vertices with valence bellow 7 and using a new retriangulation approach that keeps the mesh regularity along the simplification. The vertices can also be removed according to a geometric criterion to improve the quality of the approximations. This approach compresses the mesh connectivity to an average of 3.69 bits per vertex.

Karni and Gotsman [7] developed spectral geometry compression, where the geometry is projected on an orthogonal vector space, constructed with the eigenvectors of the mesh connectivity laplacian matrix. This scheme provides good mesh approximations, even with few transmitted coefficients. However, this algorithm is not fully progressive, as the mesh connectivity remains the same, only the geometry of the mesh changes with the resolution.

In [9], Khodakovsky et al. present a pure geometry coder, where the input model is remeshed and provides the best rate-distortion tradeoffs so far, when the user does not need to keep the original connectivity of the 3D mesh. Single-rate lossy compression that resample the surface in a semi-regular pattern have also been proposed [14, 2].

In [18] we introduced a new approach, Wavemesh which uses an irregular subdivision inversion scheme [16]. This scheme aims at processing an inverse loop subdivision [10] on the input mesh when it is possible. The geometric approximations are performed by wavelet decomposition, which enhances the approximation quality of the lower resolution levels. Although Wavemesh performs very well (and better than any other progressive approach), it can be improved, both in terms of complexity and compression efficiency, as shown in the following sections.

3. TECHNICAL BACKGROUND ON WAVEMESH

In order to achieve progressive transmission, Wavemesh simplifies the original mesh by reversing an irregular subdivision scheme [16]. The simplification is repeated until the resulting mesh cannot be simplified anymore (for meshes homeomorphic to a sphere, the simplest mesh is a tetrahedron). We obtain a hierarchy of meshes, from the simplest one M^0 to the original mesh M^J . For each mesh

M^j we call v^j its number of vertices. Following [17], the wavelet decomposition can be applied to the geometrical properties of the different meshes which are linked by the following matrix relations:

$$C^{j-1} = A^j \cdot C^j \quad (1)$$

$$D^{j-1} = B^j \cdot C^j \quad (2)$$

$$C^j = P^j \cdot C^{j-1} + Q^j \cdot D^{j-1} \quad (3)$$

where C^j is the $v^j \times 3$ matrix representing the coordinates of the vertices of M^j , D^{j-1} is the $(v^j - v^{j-1}) \times 3$ matrix of the wavelet coefficients at level j , A^j and B^j are the analysis filters, P^j and Q^j are the synthesis filters. As both the connectivity simplification and the geometrical approximation are reversible, we achieve an effective progressive transmission when transmitting only the lowest resolution mesh M^0 , and by iteratively transmitting the subdivision information from M^j to M^{j+1} and the wavelet coefficients D^j where $0 \leq j < J$ [18]. The connectivity simplification scheme used by Wavemesh attempts to inverse Loop subdivision [10] on the mesh wherever it is possible, that is, on semi-regular regions. For irregular meshes, this simplification is impossible, and Wavemesh merges faces not only four by four, but also in groups of three or two faces, or leaves some faces unchanged, and performs edge flips when needed. In [18] we showed that to obtain the best compression ratio, we must simplify a mesh M^{j+1} to a mesh M^j having the lowest possible number of vertices.

4. IMPROVING THE INVERSE PROBLEM SOLVER

This section describes how we improve the behavior of wavemesh for meshes with or without subdivision connectivity.

4.1. Meshes with subdivision connectivity

The original simplification scheme is based on a region growing algorithm, which expands a simplified mesh set, starting from a group of four seed faces which will be merged into one face in the simplified mesh. As explain in [15], without any supplementary assumption, our algorithm could find the optimal simplification for subdivision connectivity meshes in only 25% of the cases. Actually, this hit rate depends on the choice of the four chosen seed triangles. Figure 1 shows the example of a simple mesh which was obtained by applying Loop subdivision on a coarse mesh. In figure 1(b), the chosen seed faces are the dark greyed ones. With these seed triangles, the simplification could not find the optimal simplification; only the greyed faces were merged four by four. This issue is due to (1) the black marked vertex which has valence 5, and (2) the mesh boundaries. On the other hand, choosing another group of seed triangles in figure 1(c) gives the optimal simplification, as all the faces were merged four by four. The optimally simplified mesh is shown in figure 1(c). The difference between these two cases is that the first seed triangles do not have any irregular vertex (that is a vertex with valence different from 6) on their boundaries, and the second group has an irregular vertex on one of its corner. Let a triangular base mesh M^0 with genus 0 with or without boundaries and regularly subdivided 1:4 up to the resolution j . A necessary and sufficient condition to find the exact solution of the inverse problem of the regular subdivision by 4:1 face merging of M^j is to choose a seed group of four faces (first merging) having a vertex with a valence different of six on one of its corner. Proof : In the direct subdivision problem a vertex v_i

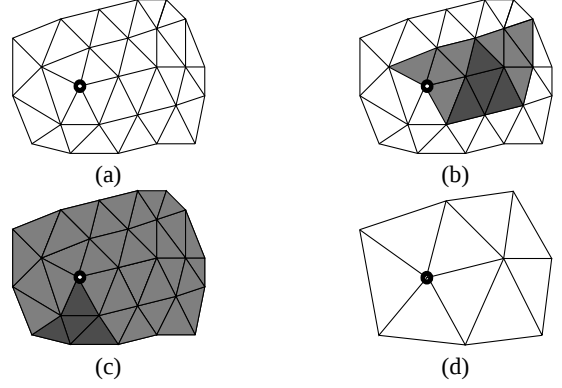


Fig. 1. Seed faces selection : (a) subdivided mesh (b) unlucky seed triangles choice; (c) lucky triangles choice (d) original coarse mesh

of valence val_i of the base mesh M^0 has a constant invariant valence for each subdivision. It follows that a vertex v_k of M^j with a valence different from six is a vertex of M^0 since every vertex created by a 1:4 subdivision has a valence of 6. A necessary condition for the reversible subdivision is that such vertices should be preserved at each merging step. Clearly this is a sufficient condition because merging 4:1 is exactly the inverse of subdivision 1:4 if we consider the same seed group.

A problem occurs when the mesh does not have any irregular vertex, but these cases are very rare, since for example, a mesh homeomorphic to a sphere must have at least four irregular vertices. We can prove that by using Euler's formula for 2D polygons with genus 0 :

$$f^j + v^j - e^j = 2 \quad (4)$$

where e^j and f^j are respectively the number of edges and the number of faces of the mesh M^j . Introducing the constraints for triangle meshes [8]:

$$\sum_i val_i = 2e^j \quad (5)$$

$$2e^j = 3f^j \quad (6)$$

where val_i is the valence of the vertex i . Combining 4 and 6 results:

$$2e^j = 6v^j - 12 \quad (7)$$

Let us now assume that M^j has n_1 valence six vertices and n_2 other vertices. We also assume that these n_2 vertices are labelled from 0 to $n_2 - 1$. Equation 5 combined with 7 now becomes:

$$6n_1 + \sum_{i/val_i \neq 6} val_i = 6(n_1 + n_2) - 12 \quad (8)$$

Considering that any vertex must have a valence at least equal to three, it comes :

$$n_2 \geq 4 \quad (9)$$

4.2. Irregular meshes

Another improvement can be made by choosing several seed groups for one given mesh : we first choose a group of seed triangles as explained earlier, and grow the seed region by only merging faces four by four. When the region cannot grow anymore, we choose another group of seed triangles in the set of non merged

triangles and grow this region still by merging only groups of four triangles. This step is repeated until no new seed group can be found. The last step consists in growing the merged regions with the irregular inverse scheme. This approach will be more effective on irregular meshes having large subdivision connectivity subregions. Figure 2 shows a comparison between the original simplification algorithm and the improved one on the Mannequin mesh (11703 vertices). Although this mesh is irregular, its top part has subdivision connectivity. Its bottom side may have been triangulated in order to have a closed mesh. Figure 2(a) shows the results obtained by the original algorithm. the triangles colored in grey were merged four by four, the black and white ones were not. We can see that the original approach did not find the optimal simplification for the upper side of the mesh. Figure 2(b) shows the results obtained with the new approach : all the triangles in the upper side were merged four by four; then the algorithm was able to find at the first try the optimal simplification for the subdivision connectivity region.

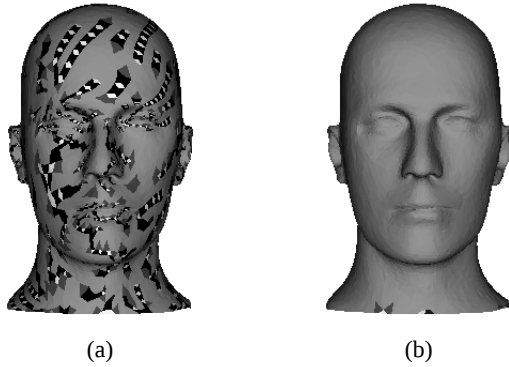


Fig. 2. Improved inverse problem solver : (a) original approach: even if the mesh has subdivision connectivity regions, misalignment results in non optimal simplification (different colors), (b) optimized approach : subdivision connectivity regions are optimally simplified (uniform coloring)

4.3. Progressive connectivity coding

When operating on a mesh that is the result of at least one stage of a regular subdivision, our progressive compression algorithm can now automatically detect that the mesh has regular subdivision connectivity and restore the connectivity of the mesh before the last subdivision stage. We encode such situations with a 1-bit flag, thus avoiding the need to transmit any connectivity information for the corresponding refinement. Based on this flag, the decompression module will correctly refine the connectivity of the previously decoded mesh using regular subdivision. This process may be repeated to similarly undo prior stages of subdivision, until the coarsest mesh is reached.

When simplifying the coarsest mesh that was the initial stage of a subdivision process, or when simplifying a mesh that is not the result of a subdivision process, our algorithm removes a subset of the vertices and encodes information that will enable a semi-regular refinement process to restore them, along with the correct connectivity.

Regions where the connectivity is semi-regular are simplified to a coarser mesh from which they can be reproduced through reg-

Model	v	Ve	AD01	WM(I)	WM(II)
Mann.	11703	0.36	3.58 9.98	2.71 9.65	0.58 8.07
Venus	11362	0.32	3.59 10.15	2.77 8.63	0.27 7.14
Fandisk	6475	1	4.99 12.34	3.2 10.03	3.02 9.83
Tiger	2738	0.057	2.67 12.67	1.72 12.34	0.31 11.74
Horse	19851	2.21	4.61 16.24	3.89 16.75	3.88 16.75

Table 1. Comparison of lossless compression results. Numbers are : connectivity coding (above) and geometry coding (below), both in bits per vertex. All the models were quantized to 10 bits per coordinates, except for the horse (12 bits)

ular subdivision. Thus, the main objective of our encoding is to tell the subdivision process which regions are regular and how to refine the irregular ones. It does so primarily by indicating which edges of the simplified mesh must be split. Additional bits are necessary to resolve triangulation ambiguities in cases where only two edges of a triangle in the simplified mesh are split and also, in rare cases, to convert the result of the semi-regular subdivision to the desired connectivity by flipping specific edges.

5. RESULTS

Table 1 Shows a comparison of different algorithms for lossless compression, for 5 different meshes. The second row is the number of vertices of the considered mesh. The third row is the vertices valence entropy, which gives us a good approximation of the efficiency of the best existing single rate encoders [8, 13]. The three next rows give results obtained with [1], the original Wavemesh implementation [18] and the proposed improved approach. On the very irregular Horse mesh, the new Wavemesh approach performs as good as the previous one. For the fandisk mesh, which has some subdivision connectivity regions, the improved version performed better than the old one. On meshes with large subdivision connectivity regions (Venus, Tiger and Mannequin), the new method dramatically reduces the connectivity cost below 1 bit per vertex. Also, for the venus mesh, our approach leads to a connectivity encoding cost of 0.28 bits per vertex, which is lower than the vertices valence entropy, which means that for this mesh, Wavemesh performs better than single resolution approaches. In figure 3, we compare the rate distortion efficiency of our new implementation versus different schemes [1, 6, 7, 18] We can see that our new approach outperforms all other lossless progressive algorithm, and is on pair with Spectral Compression, which is neither a fully progressive nor a lossless approach.

6. CONCLUSION AND PERSPECTIVES

We proposed in this paper an improvement of the Wavemesh approach, higher efficiency for meshes with regular regions, which outperforms other lossless progressive approach both in terms of rate-distortion and lossless compression. Further work could address genus reduction, large models processing, and non-manifold meshes handling.

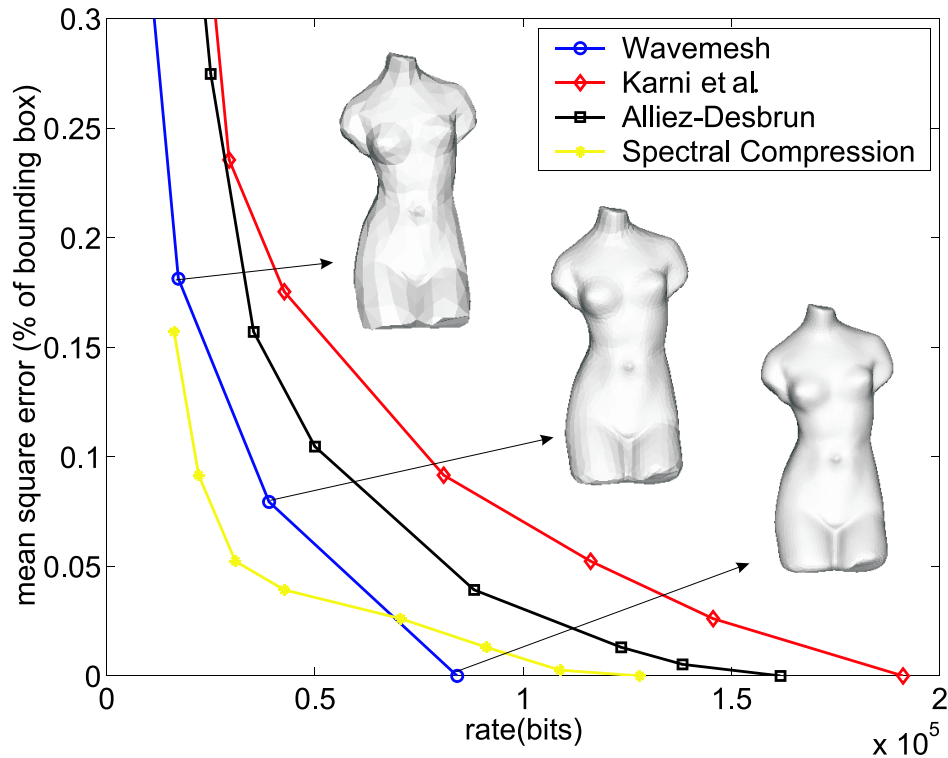


Fig. 3. Rate-distortion curve for the venus mesh

7. REFERENCES

- [1] P. Alliez and M. Desbrun, "Progressive encoding for lossless transmission of 3d meshes," in *ACM Siggraph Conference Proceedings*, 2001, pp. 198–205.
- [2] M. Attene, B. Falcidieno, M. Spagnuolo, and J. Rossignac, "Swingwrapper: Retiling triangle meshes for better compression," *ACM Transactions on Graphics*, vol. 22(4), 2003.
- [3] D. Cohen-Or, D. Levin, and O. Remez, "Progressive compression of arbitrary triangular meshes," in *IEEE Visualization 99*, 1999, pp. 67–72.
- [4] O. Devillers and P.-M. Gaudoin, "Progressive and lossless compression of arbitrary simplicial complexes," in *ACM Siggraph Conference Proceedings*, 2002.
- [5] H. Hoppe, "Progressive meshes," in *ACM Siggraph 96 Conference Proceedings*, 1996, pp. 99–108.
- [6] Z. Karni, A. Bogomjakov, and C. Gotsman, "Efficient compression and rendering of multi-resolution meshes," in *Visualization Conference Proceedings*, 2002, to appear.
- [7] Z. Karni and C. Gotsman, "Spectral Compression of Mesh Geometry," in *ACM Siggraph 00 Conference Proceedings*, 2000, pp. 279–286.
- [8] A. Khodakovsky, P. Alliez, M. Desbrun, and P. Schröder, "Near-optimal connectivity encoding of 2-manifold polygon meshes," *The Journal of Graphical Models/special issue*, 2002.
- [9] A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive Geometry Compression," *ACM Siggraph Conference Proceedings*, pp. 271–278, 2000.
- [10] C. Loop, Ed., Smooth subdivision surfaces based on triangles, *Masters thesis. Dept. of Mathematics, University of Utah*, 1987.
- [11] R. Pajarola and J. Rossignac, "Compressed Progressive Meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6(1), pp. 79–93, 2000.
- [12] J. Rossignac, "EdgeBreaker : Connectivity Compression for Triangle Meshes," *IEEE Transactions on Visualization and Computer Graphics*, 1999.
- [13] C. Touma and C. Gotsman, "Triangle Mesh Compression," *Graphics Interface 98 Conference Proceedings*, pp. 26–34, 1998.
- [14] A. Szymczak, J. Rossignac, and D. King, "Piecewise regular meshes: Construction and compression," *The Journal of Graphical Models/special issue on Processing of Large Polygonal Meshes*, 2002.
- [15] G. Taubin, "Detecting and reconstructing subdivision connectivity," *The Visual Computer, Special Issue on Subdivision*, 2001.
- [16] S. Valette, Y. S. Kim, H. J. Jung, I. Magnin, and R. Prost, "A multiresolution wavelet scheme for irregularly subdivided 3d triangular mesh," in *IEEE Int. Conf on Image Processing ICIP99*, October 1999, vol. 1, pp. 171–174.
- [17] S. Valette and R. Prost, "Wavelet based multiresolution analysis of irregular surface meshes," *submitted to IEEE Transactions on Visualization and Computer Graphics*, 2001.
- [18] S. Valette and R. Prost, "A wavelet-based progressive compression scheme for triangle meshes : Wavemesh," *submitted to IEEE Transactions on Visualization and Computer Graphics*, 2001.