Compression of 3D Triangular Meshes with Progressive Precision

Sébastien Valette, Alexandre Gouaillard, Rémy Prost

▶ To cite this version:

HAL Id: hal-00537022

https://hal.science/hal-00537022

Submitted on 17 Nov 2010

Compression of 3D triangular meshes with progressive precision

Sébastien Valette, Alexandre Gouaillard and Rémy Prost

CREATIS, Lyon, France

Abstract

In this paper we introduce a novel approach for progressive transmission of 3D triangular meshes. This algorithm is based on a new reversing approach of irregular mesh subdivision that enables a wavelet representation of any mesh geometry. In this paper, we show how to achieve progressive compression of 3D models by transmitting more and more wavelet coefficients computed from the original mesh vertices coordinates. The connectivity of the reconstructed mesh remains the same as the original one, but its geometry is progressively refined by means of Bitplane encoding. This approach processes directly floating point coordinates which is the most common representation for 3D meshes, and does not need quantization, which is a lossy transformation. Experimental results are given and demonstrate the efficiency of our encoding scheme versus other approaches.

Key words: triangular meshes, progressive compression, wavelets, irregular subdivision

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 bandwidth limitation. Data compression offers a solution for these problems by providing two different approaches: The model can be compressed either in a single resolution bitstream, or in a progressive bitstream. Single resolution approaches (14; 20; 24) often give the best compression rate, but progressive schemes

(1; 5; 7; 9; 11; 16; 19; 27) 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 may be useful for huge models.

In this paper we propose a new approach for progressive transmission of 3D models based on the decomposition of the meshe geometry on a Geometrical Wavelet Basis. The main contribution proposed in this paper is a progressive transmission of the mesh geometry from the lowest to the highest resolution level while keeping its connectivity unchanged. During reconstruction of the compressed model, the geometric refinement is obtained by progressively transmitting details (wavelet coefficients), and a good reconstruction is performed with only a small proportion of the wavelet coefficients set. In sharp contrast with other work, our approach can directly compress the vertices coordinates, without any quantization step. The paper is organized as follows : in section 2 we briefly discuss previous works related to 3D mesh compression. Section 3 explains why quantizing the vertices coordinates provides a tradeoff between efficiency and accuracy. Section 4 describes the main parts of our algorithm. Experimental results are shown, and a comparison is made versus other approaches in section 5. Finally, section 6 concludes the paper and some perspectives are given.

2   Related Works

Single-resolution compression schemes for 3D meshes usually create a single bitstream, which can be split into two parts : the connectivity bitstream (which describes the mesh connectivity graph) and the geometry bitstream (the vertices coordinates) (see figure 1.a). Progressive transmission of meshes involves splitting both bitstreams into several components. The connectivity bitstream usually contains a base mesh which is further refined by reading the successive bitstreams. The geometry bitstream is also decomposed into a base geometry and several geometrical refinements. Figure 1.b depicts how these bitstreams are interleaved during transmission.

Progressive transmission for meshes, introduced by Hoppe (9), is based on successive mesh simplification by means of edge contractions which offer optimal granularity and linear complexity, but non-linear storage and transmission cost. Several approaches derive from Progressive Meshes: Pajarola and Rossignac implemented an improved version of Progressive Meshes (19), encoding the mesh connectivity reconstruction by edge split batches, with an average coding cost of 7.2 bits per vertex. Karni et al (11) improved the edge contraction sequence and the geometry coding to enhance both the progressive transmission rate-distortion tradeoff and the rendering speed of the processed meshes. Cohen-Or et al. (5) 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 removal of high valence vertices. Alliez and Desbrun (1) improved this technique by removing only vertices with valence below 7 and using a new retriangulation approach that maintains 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 (12; 13) 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. The main drawback of this scheme is the need to calculate eigenvectors of the mesh connectivity graph which computational complexity is not linear to the number of triangles, but to its cube. This drawback was solved partially in (13) where precomputed eigenvectors are used, using a planar mapping of the mesh connectivity graph on a regular graph. In (27) Valette and Prost introduced the Wavemesh approach, which uses an irregular subdivision inversion scheme (26). This scheme aims at processing an inverse 1-to-4 subdivision (17) on the input mesh whenever it is possible. The geometric approximations are performed by wavelet decomposition, which enhances the approximation quality of the lower resolution levels.
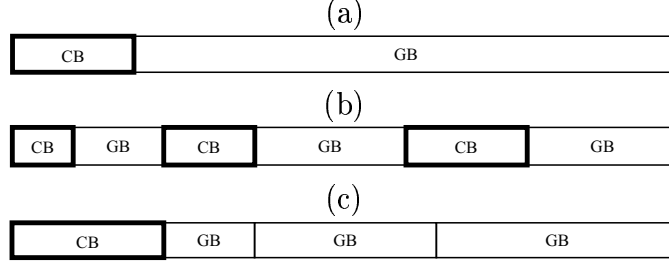
Fig. 1. Different bistreams for different approaches (a) mono-resolution approach (b) progressive approach (connectivity+geometry) (c) progressive approach (geometry only). CB=Connectivity bitstream GB=Geometry Bitstream.

In (16), Khodakovsky et al. present a pure geometry coder, where the input model is remeshed into a semi-regular one. This approach is very efficient, as it removes the connectivity information of the input mesh when the user does not need to keep the original connectivity of the 3D mesh. Single-rate lossy compressions that resample the surface in a semi-regular pattern have also been proposed (23; 2).

In (8), Guskov et al. describe a new 3D representation. A Normal mesh is a mesh in which geometry can be represented mostly by normal components and is then suitable for progressive compression. Again, this approach provides very good results, at the cost of a remeshing of the compressed model. This approach was recently improved in (15).

Note that remeshing progressive approaches (16; 8) do not retain the original connectivity of the mesh, and the compressed bitstream mainly consists of geometrical refinements (except the base mesh connectivity) as shown in figure 1.(c)

3   Mesh Coordinates Coding : Quantization Issues

In general, the vertices coordinates of a mesh consist of three floating point values. This is an issue for compression schemes which often use entropy coding to remove statistical redundancies between the coded values (e.g. difference values for predictive coding (24), spectral coefficients (12)). As a consequence, the vertices coordinates are quantized before the encoding stage. The quantization step depends on the desired precision on the reconstructed model (usually geometry is quantized to 8 to 12 bits per coordinate). Hence, if one wants to have n precision levels(corresponding to different quantization steps), then n independent compressed bitstreams will be generated (see (10) for an example). Figure 2 shows the progressive compression efficiency of Wavemesh(27), for different quantization steps (8, 10, 12 and 14 bits). PSNR was measured in dB as : $PSNR = -20.log_{10}\left(\frac{D_h}{BB_d}\right)$ where D_h is the Hausdorff's distance measured by METRO (4) and BB_d is the bounding box diagonal length of the mesh. We can see that each bitstream attains its maximum PSNR depending on the applied quantization step. More generally (3), uniform quantization of uniformly distributed floating-point coordinates to integers coded in q bits results in a final PSNR equal to:

$$PSNR \approx 6.q \quad dB \tag{1}$$

In figure 2 we can see that Wavemesh is a lossless coder (for a given quantization step): for each curve, the maximum PSNR matches equation 1 (i.e. 52db for 8 bits quantization, 64dB for 10 bits 75dB for 12 bits and 88dB for 14 bits). Moreover, with high quantization (i.e. integers coded with few bits) the PSNR will not be much greater than defined by equation (1) and a low quantization (i.e. integers coded with many bits) results in a rate-distortion curve
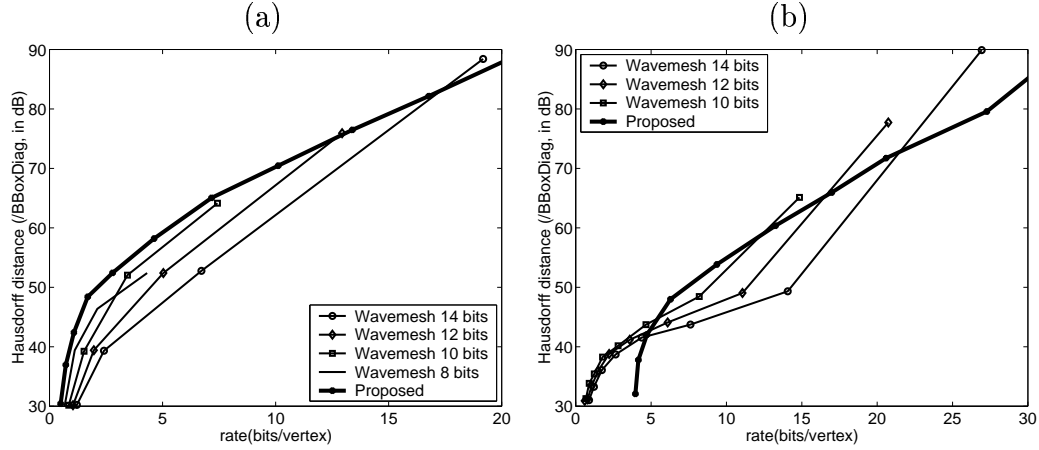
3

Fig. 2. Rate-distortion efficiency comparison : Wavemesh vs proposed approach : (a) Venusbody (b) Horse

with low performances at low bitrates : For both the Venubody and the Horse mesh, at low bitrates, 10 bits quantization is much more efficient than 14 bits quantization. As a consequence, there is a tradeoff between final accuracy (when the bitstream is fully transmitted) and performance at low bitrates.

4 Our Approach

In this paper, we propose a progressive compression of the mesh geometry. The connectivity of the reconstructed mesh is kept unchanged. The geometry is processed by an irregular Wavelet Transform (26) and the wavelet coefficients are transmitted by order of magnitude, to improve the rate-distortion efficiency of the approach.

We encode the wavelet coefficients with a bitplane approach (a zerotree coder, as used in (16; 21)). bitplane encoding can be applied directly to floating point coordinates, thus avoiding the quantization tradeoff explained in section 3. Indeed, bitplane encoding entails progressively refining the reconstructed values. With these algorithms, one can encode any floating-point value in a scalable manner. Figure 2 compares the performance of our approach with (27). As our approach does not need quantization, only one curve for each model is given. We can see that avoiding quantization gives good performances in the whole range of proposed bitrates (except low bitrates for the Horse model). For the smooth Venusbody mesh, our approach performs better than Wavemesh for all quantization steps. Combining a zerotree coder such as used in (21) and the Wavelet Transform for irregular meshes provides a good means of progressively compressing meshes, as described in the following sections.

4.1 Wavelets for irregular meshes

The wavelet decomposition is strongly related to subdivision when considering 3D meshes. Indeed a subdivision scheme inherently creates nested vector spaces defining the 3D surface mesh, which is mandatory in order to apply the wavelet decomposition. A mathematical framework has been built in (18), allowing Multiresolution analysis on semi-regular meshes. In order to apply the wavelet decomposition on irregular meshes geometry, Valette and Prost (26) proposed to reverse an irregular subdivision scheme (25) to coarsen the input mesh. The coarsening is repeated until the resulting mesh cannot be further simplified (for meshes homeomorphic to a sphere the simplest mesh is a tetrahedron). A hierarchy of meshes is constructed, from the coarsest, i.e. M^0 to the original mesh M^J. For each mesh M^j, v^j
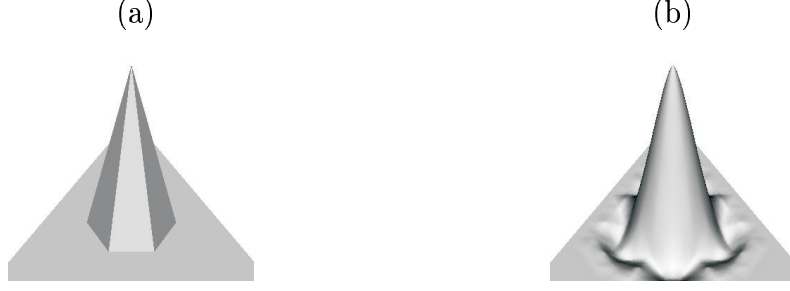
Fig. 3. Scaling functions : (a) lazy filterbank (b) lifted filterbank

is its number of vertices. Following (26), the wavelet decomposition can be applied to the different mesh geometries which are linked by the following matrix relations:

$$C^{j-1} = A^j.C^j \tag{2}$$

$$D^{j-1} = B^j.C^j \tag{3}$$

$$C^j = P^j.C^{j-1} + Q^j.D^{j-1} \tag{4}$$

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 and P^j and Q^j are the synthesis filters. Both the connectivity coarsening and the geometrical approximation are reversible. The connectivity coarsening scheme used in this scheme attempts to reverse 1:4 subdivision (17) on the mesh wherever possible, that is, on semi-regular regions. For irregular meshes, this coarsening is impossible, and faces are merged not only four by four, but also in groups of three or two faces; some faces are left unchanged, and some edge flips are performed in order to make the simplification easier. It is shown in (27) that to obtain the best compression ratio, a mesh M^{j+1} should be simplified to a mesh M^j having the lowest possible number of vertices. The matrix filters A^j, B^j, P^j and Q^j are constructed using the lifting scheme, starting from an easy-to-build filterbank, i.e. the "lazy" filterbank. Note that in our experiments we used two types of lifting; to orthogonalize the lazy wavelets (hat functions), and the lazy scaling functions (wider hat functions):

- Primal lifting : changes the shape of the wavelet functions to enhance the quality of processed approximations during analysis. Details on primal lifting can be found in (18; 26)
- Dual lifting : changes the shape of scaling functions. Hence it is an interpolating lifting, leading to smooth reconstruction even with few transmitted coefficients. For dual lifting we used the butterfly lifted wavelets scheme described in (22). Figure 3 shows an example of smooth scaling function (b) created by applying dual lifting to a non-smooth lazy scaling function (a).

4.2 Connectivity coding

There are many algorithms for compressing the connectivity graph of a manifold mesh, in a single-resolution approach (see for example (14; 20; 24)), or in a progressive approach (1; 5; 7; 9; 11; 19; 27). We chose to encode the mesh connectivity using (27), because this approach provides a way of applying the Wavelet Transform to irregular meshes, which is very efficient for geometry compression. Currently, valence-driven compression schemes such as (24) give results very close to the optimal (14). To optimize the compression efficiency of our approach, we could encode the connectivity graph with valence-driven coders, but this would make the decoder more complex. Indeed, applying the Wavelet Transform to irregular
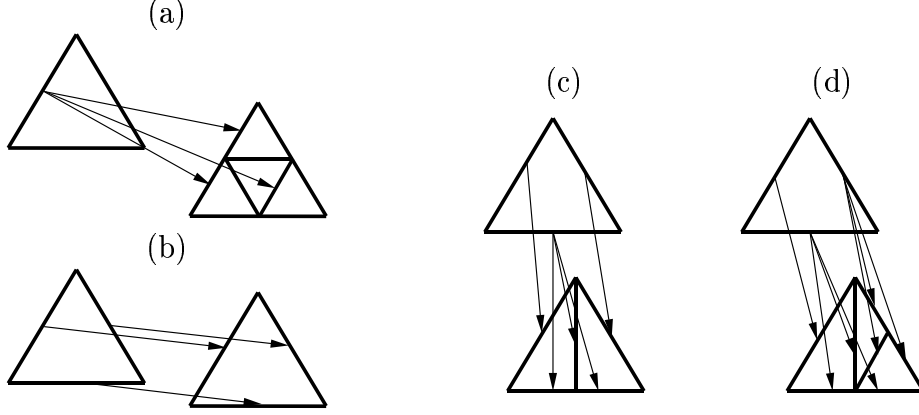
Fig. 4. Edge-based Hierarchies for zerotree coding with irregular subdivision : (a) 1:4 subdivision (b) unchanged face (c) 1:2 subdivision (d) 1:3 subdivision

meshes requires the construction of a hierarchy of meshes, which is the inverse problem of subdivision. By coding the mesh connectivity using the connectivity coder embedded in Wavemesh (27), the hierarchy of meshes can be built in a straightforward manner by the decoder without the need to solve the inverse problem.

4.3 Progressive Geometry Coding with a Zerotree

Based on the properties of orthogonal transformations, an efficient encoding in terms of PSNR is to code the highest order bits of the largest magnitude coefficients first. Zerotree coders, as used in (16; 21) provide efficient progressive compression by coding and refining values from the most significant bits to the least significant ones. For each dataset $D = \{w_i\}$ created by wavelet decomposition, a tree is built which gives efficient hierarchical access to all the coefficients, for both the coder and the decoder. At the decoder end, the values are reconstructed with a decreasing reconstruction error. In fact, zerotree coding is a type of bitplane encoding, since it encodes the binary representation or the values, assuming that a large number of coefficients will have low magnitude and that there is a spatial correlation of coefficient magnitude across the resolution levels. A coefficient w_i is said to be significant if its magnitude is higher than a specified threshold T. Highest magnitude coefficients (which are significant according to a threshold $T = T_{max}$ ($T_{max} = 2^q$ where $q = \lfloor log_2(max|w_i|) \rfloor$) are coded first. Successive refinement passes occur, which involve the localization of newly significant coefficients and the refinement of significant coefficients, until the desired precision is reached. Note that between each refinement pass, the threshold T is divided by two, thus increasing the number of significant coefficients. This encoding scheme offers minimal granularity, as the encoded bitstream consists only of "0" or "1" codes which are either significance codes or refinement codes. The main difficulty in our case is building the tree used to localize the wavelet coefficients. To be brief we are not providing here the detailed algorithm, but the reader can refer to (21) for details. Inspired by (16), we chose the edges of the mesh to build a tree, as wavelet coefficients have a one to one association with the mesh edges. Then we had to define relationships linking parent and child edges in the tree. The irregular subdivision scheme is a good framework for this process. Figure 4 depicts the parent-child links for each case of subdivision. For compression efficiency, we decided that an edge which is not bisected during subdivision only has one child edge (its corresponding edge in the subdivided mesh). Figure 4.a (1:4 subdivision) only shows parent-child relationship for one parent edge, for visibility purposes. Relationships for the two other parent edges can be recovered by rotation. Note that in these figures, bisected edges all have three child edges, but in the general case, an interior edge is adjacent to two faces. As a consequence, interior parent edges will have four child edges and boundary parent edges will have only three child edges attached.
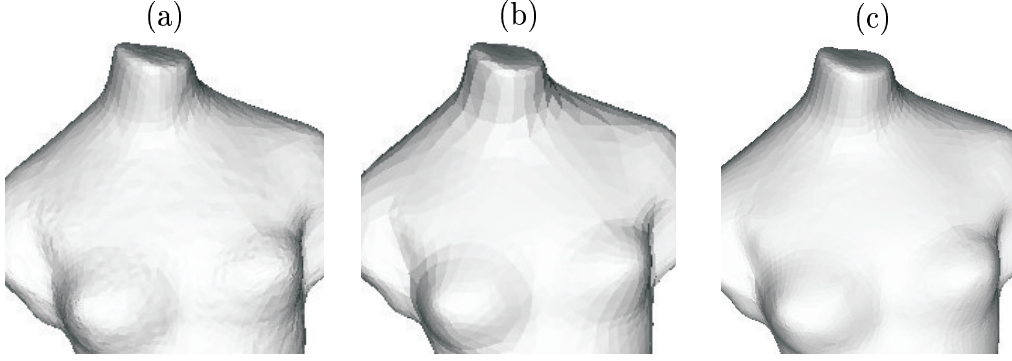
6

Fig. 5. Visual aspect at equivalent bitrates : (a) Wavemesh (10 bits quantization) (b) Wavemesh (14 bits quantization) (c) Proposed approach

5   Results

We have tested our algorithm on several reference models. Our implementation is fast. As an example, the compression of the Bunny mesh with 34834 vertices takes less than 10 seconds on an Intel Pentium II laptop computer running at 400Mhz with 128Mb RAM. In figure 6, we compared the rate-distortion efficiency of our approach when applying several different lifting schemes : only primal lifting, only dual lifting, or both primal and dual lifting. For the regularly sampled Venusbody (11362 vertices, figure 6.(a)), using both primal and dual lifting significantly improves the efficiency of our approach. On the irregular Bunny (34834 vertices, figure 6.(b)), dual lifting does not improve the quality of the results; moreover, at high bitrates, it penalizes the global performances of our approach. We believe that this issue is caused by the Butterfly Lifted Wavelets which were initially created for regular 1:4 subdivision. Adaptation of Butterfly lifted wavelets to irregular subdivision schemes should be considered, as explained in (6).

Figure 2 compares our approach with results obtained by the improved Wavemesh(28) using several quantization steps, on the Venusbody and the Horse (19851 vertices). Note that for each model, only one curve is drawn for our approach, as quantization is not needed. On the Venusbody, the proposed approach outperforms all results obtained by Wavemesh. Good performances result from the fact that this model is very smooth and its connectivity is highly regular. On the other hand, the Horse model has very irregular connectivity. As a consequence, encoding the connectivity of the mesh costs much more than for the Venusbody. That is why the rate-distortion curve begins above 4 bits/vertex (this is the encoding cost for the connectivity). For bitrates above 5 bits/vertex, our approach achieves a good rate/distorsion tradeoff. As an example, applying Wavemesh with 14 bits/coordinate quantization gives the best results above 20 bits/vertex, but performs poorly for low bitrates. On the other hand, our approach achieves relatively good compression for a wider range of bitrates. Closeup views of the Venusbody mesh reconstructed by different approaches at equivalent bitrates ($\approx 7.2bits/vertex$) are shown in figure 5 : (a) Wavemesh, 10 bits quantization : the model is fully reconstructed (11362 vertices) so the connectivity information reaches its maximum. But quantization of coordinates to 10 bits integers introduces visible noise. (b) Wavemesh, 14 bits quantization : the model was reconstructed to a mesh with 2842 vertices; most of the data was transmitted for the mesh geometry. But the reconstruction is not smooth, because of the low number of vertices. (c) Our approach : the model is reconstructed to its highest connectivity level, and no quantization noise affects the visual aspect of the mesh.

Figure 7 compares our approach with that of other progressive compression schemes: Alliez and Desbrun (1), Karni et al. (11), Spectral Compression(12) and Wavemesh (27), for the Venusbody model (11362 vertices, 10 bits quantization) and for the Venus head (8268 vertices, 12 bits quantization). It can be seen that our approach performs similarly or better than Spectral Compression, for a much lower computational complexity. Note that results given for our approach were obtained without quantizing the mesh, thus guaranteeing that we can reach higher precision than other works. Figure 10 and 8 show different reconstruc-
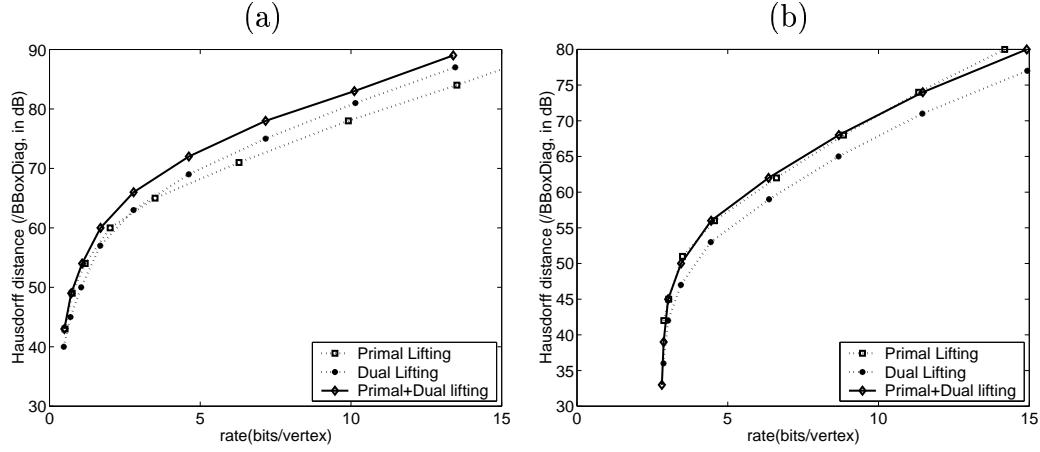
Fig. 6. Comparison between different lifting schemes: (a) Venusbody (b) Bunny
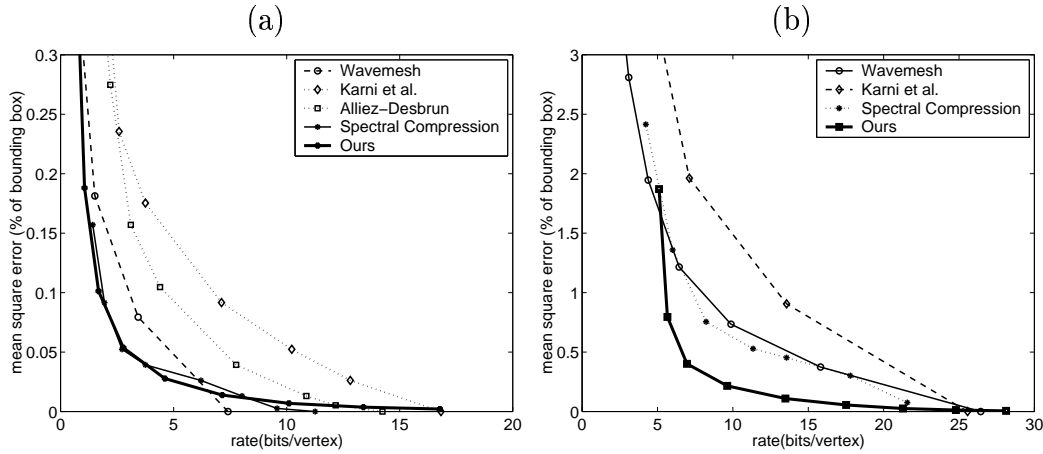

Fig. 7. Rate-distortion efficiency comparison with several approaches : (a) Venusbody (b) Venus head

tion levels for the Venusbody and Venus head meshes. Clearly, the Venus head mesh is not smooth at low bitrates, due to its very irregular sampling. On the other hand, the Venusbody mesh is well reconstructed even at low bitrates. Finally, we compare our approach with the Progressive Geometry Coder (PGC) of Khodakovski et al. (16) in figure 9. This figure shows the rate-distortion efficiency on the Venus model (50002 vertices) for PGC, and two curves are given for our approach : the first one results from compressing the original Venus model, and the second one results from compressing the remeshed model used by PGC. Results clearly show that the remeshed model is significantly better compressed than the original one. As expected, remeshing is effective when one wants to keep only the geometry of 3D models.

6   Conclusion

We have proposed in this paper a novel approach for progressive compression of 3D triangle meshes. Our approach can be applied directly to irregular meshes without applying any remeshing or quantization step, which makes it an asymptotically lossless coder for floating-point vertices coordinates. The algorithm applies a Wavelet Transform on vertices coordinates, which are further encoded with a zerotree coder. Experimental results show that our algorithm achieves good performances, with a low computational complexity. Our
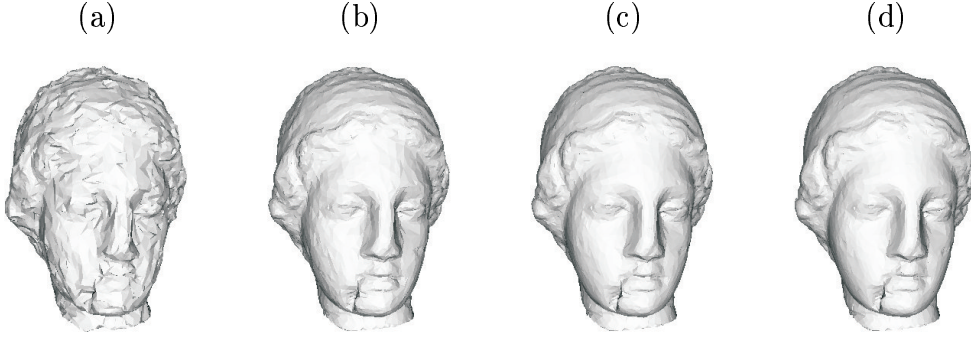
Fig. 8. Progressive transmission for Venus : (a) 9.55 bits/vertex (b) 17.4 bits/vertex (c) 21.2 bits/vertex (d) 28 bits/vertex
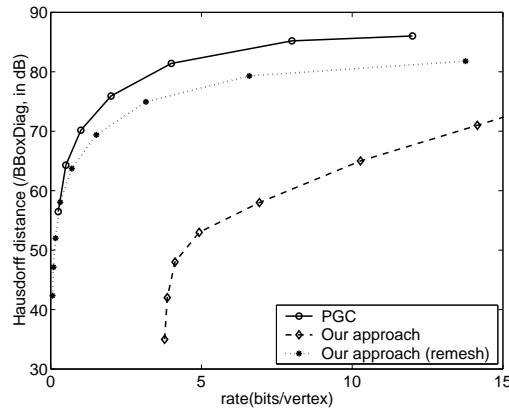

Fig. 9. Comparison with Progressive Geometry Compression : Venus head
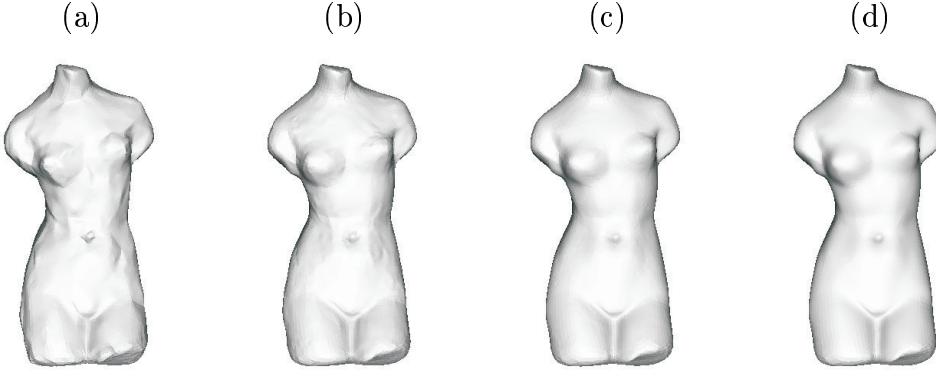

Fig. 10. Progressive transmission for Venusbody : (a) 1.05 bits/vertex (b) 2.78 bits/vertex (c) 7.15 bits/vertex (d) 13.3 bits/vertex

approach is based on progressive transmission of wavelet coefficients with zerotree encoding. An irregular subdivision scheme reconstructs the original mesh connectivity graph. Further work could address the construction of an improved Butterfly interpolating scheme adapted to the irregular subdivision scheme, in order to improve the efficiency of the proposed approach for highly irregularly sampled models or models with sharp features.

9

References

[1] P. Alliez, 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, J. Rossignac, Swingwrapper: Retiling triangle meshes for better compression, ACM Transactions on Graphics 22(4).

[3] T. Berger, Rate Distortion Theory, Englewood Cliffs, N.J. : Prentice-Hall, 1971.

[4] P. Cignoni, C. Rochini, R. Scopigno, Metro : Measuring error on simplified surfaces, Computer Graphics Forum (17(2)) (1998) 167–174.

[5] D. Cohen-Or, D. Levin, O. Remez, Progressive compression of arbitrary triangular meshes, in: IEEE Visualization 99, 1999, pp. 67–72.

[6] I. Daubechies, I. Guskov, P. Schröder, W. Sweldens, Wavelets on irregular point sets, Phil. Trans. R. Soc. Lond. A 357 (1760) (1999) 2397–2413.

[7] P.-M. Gandoin, O. Devillers, Progressive and lossless compression of arbitrary simplicial complexes, in: ACM Siggraph Conference Proceedings, 2002, pp. 372–379.

[8] I. Guskov, K. Vidimce, P. Schröder, W. Sweldens, Normal meshes, ACM Siggraph Conference Proceedings (2000) 95–102.

[9] H. Hoppe, Progressive meshes, in: ACM Siggraph 96 Conference Proceedings, 1996, pp. 99–108.

[10] M. Isenburg, P. Alliez, Compressing polygon mesh geometry with parallelogram prediction, in: Visualization conference proceedings, 2002.

[11] Z. Karni, A. Bogomjakov, C. Gotsman, Efficient compression and rendering of multi-resolution meshes, in: Proceedings of IEEE Visualization, 2002.

[12] Z. Karni, C. Gotsman, Spectral Compression of Mesh Geometry, in: ACM Siggraph 00 Conference Proceedings, 2000, pp. 279–286.

[13] Z. Karni, C. Gotsman, 3d mesh compression using fixed spectral bases, Proceedings of Graphics Interface (2001) 1–8.

[14] A. Khodakovsky, P. Alliez, M. Desbrun, P. Schröder, Near-optimal connectivity encoding of 2-manifold polygon meshes, Graphical Models 64 (3–4) (2002) 147–168.

[15] A. Khodakovsky, I. Guskov, Compression of normal meshes, Geometric Modeling for Scientific Visualization, to appear .

[16] A. Khodakovsky, P. Schröder, W. Sweldens, Progressive Geometry Compression, ACM Siggraph Conference Proceedings (2000) 271–278.

[17] C. Loop (Ed.), Smooth subdivision surfaces based on triangles, Masters thesis, Dept. of Mathematics, University of Utah, 1987.

[18] M. Lounsbery, Multiresolution Analysis for Surfaces of Arbitrary Topological Type, PhD thesis, Dept. of Computer Science, University of Washington, 1994.

[19] R. Pajarola, J. Rossignac, Compressed Progressive Meshes, IEEE Transactions on Visualization and Computer Graphics 6(1) (2000) 79–93.

[20] J. Rossignac, EdgeBreaker : Connectivity Compression for Triangle Meshes, IEEE Transactions on Visualization and Computer Graphics (1999) 47–61.

[21] A. Said, W. Pearlman, A new fast and efficient image codec based on set partitioning in hierarchical trees, IEEE Transactions on Circuits and Systems for Video Technology 6(3) (1996) 243–250.

[22] P. Schröder, W. Sweldens, Spherical wavelets: Efficiently representing functions on the sphere, Computer Graphics Proceedings (SIGGRAPH 95) (1995) 161–172.

[23] A. Szymczak, J. Rossignac, D. King, Piecewise regular meshes: Construction and compression, The Journal of Graphical Models/special issue on Processing of Large Polygonal Meshes 64 (2002) 183–198.

[24] C. Touma, C. Gotsman, Triangle Mesh Compression, Graphics Interface 98 Conference Proceedings (1998) 26–34.

[25] S. Valette, Y. S. Kim, H. J. Jung, I. Magnin, R. Prost, A multiresolution wavelet scheme for irregularly subdivided 3d triangular mesh, in: IEEE Int. Conf on Image Processing ICIP99, Vol. 1, 1999, pp. 171–174.

[26] S. Valette, R. Prost, Wavelet based multiresolution analysis of irregular surface meshes, IEEE Transactions on Visualization and Computer Graphics, to appear.

[27] S. Valette, R. Prost, A wavelet-based progressive compression scheme for triangle meshes : Wavemesh, IEEE Transactions on Visualization and Computer Graphics, to appear.

[28] S. Valette, J. Rossignac, R. Prost, An efficient subdivision inversion for wavemesh-based progressive compression of 3d triangle meshes, in: IEEE Int. Conf on Image Processing ICIP03, 2003.
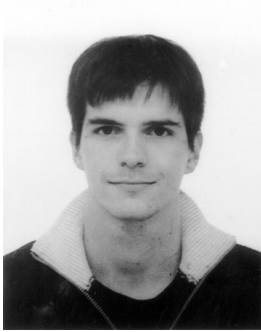
Sébastien Valette Was born in France, in 1975. He recieved the M.S. Degree and the Engineering Diploma from the Electrical Engineering Department, at the National Institute for Applied Sciences (INSA) of Lyon, France, in 1998. He obtained the PhD Degree at INSA of Lyon in 2002. His research interests include 3D processing, wavelets, progressive compression and multiresolution analysis.

Alexandre Gouaillard received in 2002 a Master degree of Electrical Engineering and a Master Degree of Image Processing, both from the National Institute for Applied Sciences of Lyon (INSA-Lyon, France). Currently Ph. D. student, he is spending half his time in the multiresolution Group of Creatis Laboratory (France) and half in Kanai's Computer Graphics Laboratory (Japan). His main research topics are: wavelet multiresolution meshes and their application to compression and segmentation of medical images.This includes parameterisation and remeshing of 3D surface meshes.

Rémy Prost received his doctorate degree in Electronics Engineering and his "Docteur es Sciences" degree from Lyon University and the National Institute of Applied Sciences (INSA), Lyon, France, in 1977 and 1987 respectively. He is currently a professor in the Department of Electrical Engineering at INSA-Lyon. Both his teaching and research interests include digital signal processing, inverse problems, image data compression, multiresolution algorithms, wavelets, and meshes processing. He leads the ' Volume (3D) Image Processing ' project in the CREATIS Laboratory (CNRS 5515) at INSA-Lyon.