HIGH RATE COMPRESSION OF 3D MESHES USING A SUBDIVISION SCHEME

Guillaume Lavoué, Florent Dupont and Atilla Baskurt

LIRIS UMR 5205 CNRS
43, Bd du 11 novembre, 69622 Villeurbanne Cedex, France
phone: + (33) 4 72 44 83 95, fax: + (33) 4 72 43 15 36, email: glavoue@liris.cnrs.fr
web: http://liris.cnrs.fr/guillaume.lavoue

ABSTRACT

In this paper we present a new framework, based on subdivision surface approximation for efficient compression and coding of 3D models represented by polygonal meshes. Our algorithm fits a piecewise smooth subdivision surface to the input 3D mesh, aiming at getting close to the optimality in terms of control points number and connectivity of the subdivision control polyhedron. Our method is particularly suited for meshes issued from mechanical or CAD parts; indeed in these cases the research of the optimality is quite relevant. The found control polyhedron is much more compact than the original mesh and visually represents the same shape after several subdivision steps. This control polyhedron is then encoded specifically to give the final compressed stream. We obtain very promising results in terms of compression.

1. INTRODUCTION

Advances in computer speed, memory capacity, and hardware graphics acceleration have highly increased the amount of three-dimensional models being manipulated, visualized and transmitted over the Internet. In this context, the need for efficient tools to reduce the storage of this 3D content, mostly represented by polygonal meshes, becomes even more acute, particularly to reduce the transmission time for low bandwidth applications. Many efficient techniques have been developed for encoding polygonal meshes but fundamentally, this representation remains very heavy in terms of amount of data (a large points set, on top of the connectivity have to be encoded). Other models exist to represent a 3D shape: NURBS surfaces or subdivision surfaces. These models are much more compact. A subdivision surface is a smooth (or piecewise smooth) surface defined as the limit surface generated by an infinite number of refinement operations using a subdivision rule on an input coarse control polyhedron (see Figure 1). Hence, it can model a smooth surface of arbitrary topology (contrary to a NURBS model which needs a parametric domain) while keeping a compact storage and a simple representation (a coarse polygonal mesh). Moreover it can be easily displayed to any resolution. Subdivision surfaces are now widely used for 3D imaging and have been integrated to the MPEG4 standard [15]. For all these reasons, we have developed a new algorithm, based on subdivision surface fitting for efficiently compressing 3D meshes, for low bandwidth transmission and storage. Section 2 details the related work about mesh compression and subdivision surface fitting, while the overview of our method is presented in section 3. Sections 4 and 5 detail our subdivision surface approximation method and the associated coding scheme. Finally, results are presented in section 6.

2. RELATED WORK

2.1 Mesh compression

A lot of work has been done about polygonal mesh compression. A good review can be found in [3]. This representation contains two kinds of information: *geometry* and *connectivity*, the first describing coordinates of the vertices in the 3D space, and the later describing how to connect these positions. The connectivity graph is often encoded using a region growing approach based on faces [4], edges [7] or vertices [18]. Others techniques consider progressive approaches which encode a base mesh and then vertex insertion operations [1]. Less efforts have been done about geometry compression which is often simply performed by predictive coding and quantization. Other researches have put more efforts on geometry driven mesh coding, using wavelets [10, 19] or spectral compression [9]. On the whole, better mesh compression methods give between 1 and 2 bytes per vertex; although this represents an excellent result, the output bit stream remains large for complex objects because of the high number of vertices to encode. That is why we have chosen to approximate input meshes with subdivision surfaces (see Figure 1), of which control polyhedrons should contain much lesser faces to store or transmit, knowing that after several refinement steps, the subdivision surface will visually represents the shape of the original mesh (of which original connectivity will not be kept).

2.2 Subdivision surface approximation

Several methods already exist for subdivision surface fitting, most of them take as input a dense mesh and obtain the subdivision control mesh connectivity by simplification [13, 14]. With these simplification based approaches, the control mesh connectivity strongly depends on the input mesh and therefore can gives quite bad results if the input mesh is very irregular. Our algorithm remains independant of the target mesh connectivity by using boundaries and curvature information. Some algorithms [17, 8] also remain independent of the target mesh, by iteratively subdividing and shrinking an initial control mesh toward the target surface. Unfortunately this method fails to capture local characteristics for complex target surfaces. To our knowledge, the optimality in terms of control points number and position represents a minor problematic in the existing algorithms but is particularly relevant for mechanical or CAD objects. Only Hoppe et al. [6] optimize the connectivity (but not the number of control points) by trying to collapse, split, or swap each edge of the control polyhedron. Their algorithm produces high quality models but need of course an extensive computation time. Our algorithm optimizes the connectivity of the control mesh by analyzing curvature directions of the target surface, which reflect the natural parameterization of the shape.

3. FRAMEWORK AND PRELIMINARY WORK

Our framework is the following: firstly the target 3D objects are segmented into surface patches. Then, for each patch, a local approximating subdivision surface is constructed, associated with a control mesh. The final control mesh defining the whole surface is then created assembling every local control meshes, and encoded. Our algorithm adapts the connectivity of the control mesh to the natural parameterization of the target model by using a specific quad-triangle subdivision scheme [16].

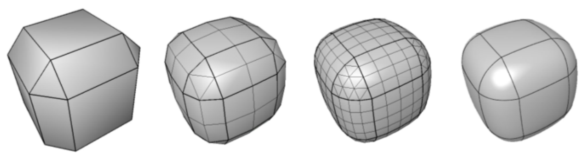

Figure 1: Example of Quad-Triangle subdivision.

Decomposition into patches: The used method is based on the curvature tensor field analysis and presents two distinct complementary steps: a region based segmentation which decomposes the object into near constant curvature patches, and a boundary rectification based on curvature tensor directions, which corrects boundaries by suppressing their artefacts or discontinuities. This method is detailed in [11]. Resulting segmented patches, by virtue of their properties (constant curvature, clean boundaries) are particularly adapted to subdivision surface fitting (see Figure 2.a).

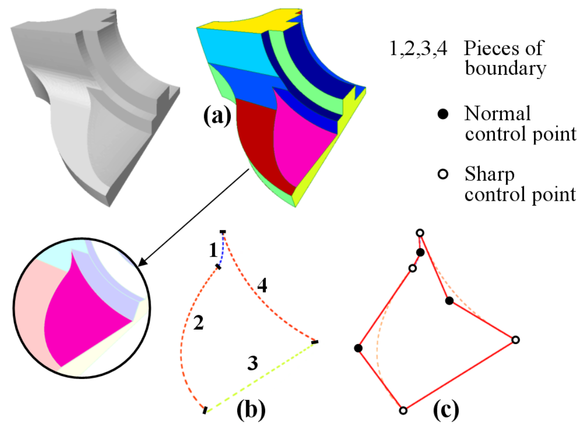

Figure 2: Illustration of segmentation (a), boundary extraction (b) and subdivision curve approximation (c).

Local subdivision surface approximation: For each patch, the mechanism for subdivision surface fitting is the following: First, the boundary of the patch is extracted. In order to prevent our model from cracks, for each patch the boundary is divided into pieces of boundary corresponding to the different adjacencies with its neighbouring regions (see Figure 2.b). Then the boundary is approximated with a piecewise smooth subdivision curve following the algorithm detailed in [12]. According to subdivision properties, the associated control polygon (Figure 2.c) will represent the boundary of the control polyhedron of the approximating subdivision surface. Then our process will attempt to connect control points of the control polygon, in order to create the optimal set of facets that will represent our final control polyhedron.

Final control polyhedron construction and coding: The final control polyhedron for the whole object is then created by assembling local control polyhedrons and is encoded specifically to give the compressed output binary stream.

4. SUBDIVISION SURFACE APPROXIMATION

4.1 Edge score definition

Once the boundary control polygons have been extracted, the purpose is to create edges and facets by connecting the control points in such a way that the corresponding created subdivision surface is the better approximation of the target surface for these given control points. For this purpose, we consider the lines of curvature of the original surface, represented by local directions of minimum and maximum curvature. Control lines of a subdivision surface are strongly linked to the lines of curvature. Indeed the topology of a control polyhedron will strongly influence the geometry information of the associated limit surface, which is also carried by lines of curvature [2]. This coherency between control lines and lines of curvature is shown in the example presented in Figure 3.

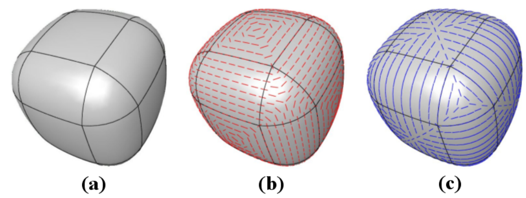

Figure 3: The coherency between control lines (a), minimum (b) and maximum (c) directions of curvatures.

Thus, for each couple of control points from the boundary control polygon, a Coherency Score (*SC*) is calculated, taking into account the coherency of the corresponding potential control edge with the lines of curvatures of the corresponding area on the target surface. The mechanism is illustrated on Figure 4: For each potential edge E, we consider its vertices P_0, P_1 and their respective limit positions P_0^∞, P_1^∞. Then we calculate the pseudo geodesic path, between these limit positions, to simulate the control line, by applying the Dijkstra algorithm on the vertices of the original surface. Finally we consider the curvature tensors of the n vertices V_i of this path, and particularly their curvature directions. The coherency score SC for this potential edge E is:

$$SC(E) = \frac{min(\sum_{i=1}^{n} \theta min_i, \sum_{i=1}^{n} \theta max_i)}{n} \tag{1}$$

where θmin_i (resp. θmax_i) is the angle between the minimum (resp. maximum) curvature direction of the vertex V_i and the segment $P_0^\infty P_1^\infty$. This score $SC \in [0,90]$ is homogeneous to an angle value in degrees. Two special cases are taken into account, concerning the nature of vertices V_i belonging to the path:

- If V_i owns an isotropic curvature tensor (plane or spherical region), hence the directions of curvature do not carry information. In these cases θmin_i and θmax_i are set to 45, to not influence the final score.
- If V_i is on a boundary (while not being the beginning or the end of the path), then a penalty is introduced, because if the corresponding potential edge represents a correct control edge, thus it should not cross or touch a boundary. Therefore in these cases, θmin_i and θmax_i are set to 90.

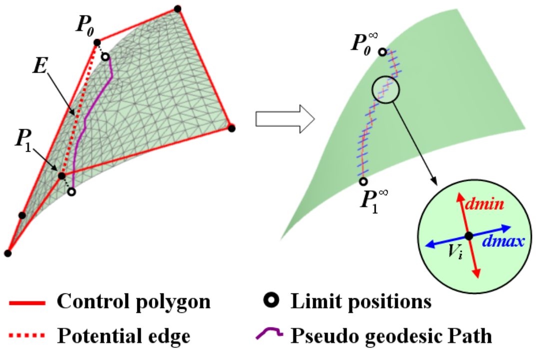

Figure 4: Mechanism for edge score definition.

4.2 Local subdivision surface construction

Our algorithm is the following: we extract a single contour from the boundary control polygons, that we call the Topologic Contour. In the case of a single boundary target surface, it is automatic. In the case of a multiple boundaries target surface (a cylinder for example), we have several control polygons, hence we link them by creating edges and doubling certain control points. For n boundaries, we create $(n-1)$ edges, by choosing those associated with smallest scores *SC*. Once the topological contour has been extracted, our algorithm is quite simple. We consider the potential edge associated with the smallest score *SC* (dotted segments in Figure 5), and we cut the contour along this edge, creating two sub-contours. This algorithm is repeated recursively on sub-contours until it remains only plane contours (see contours 1,2,3 on Figure 5). Then for each plane contour, we check its convexity; if it is convex, we create a facet, and if not, we decompose it into convex parts, using the algorithm from Hertel and Mehlhorn [5]. By assembling created facets we obtain our initial polyhedron of which limit surface represents in most case a quite good approximation of the original surface (see Figures 4 and 5).

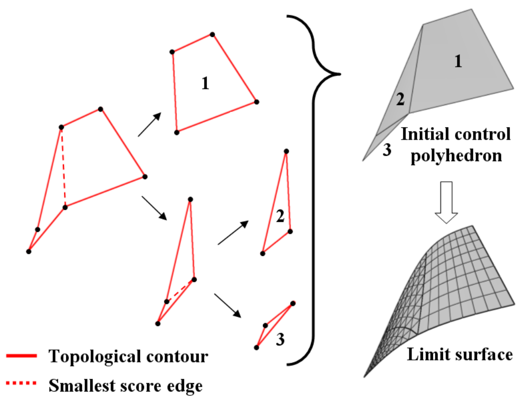

Figure 5: Local control polyhedron creation mechanism.

5. FINAL CONTROL POLYHEDRON CONSTRUCTION AND CODING

Once each patch has been fitted with a subdivision surface, the final control polyhedron for the whole object is created by assembling local control polyhedrons while marking local boundary control edges as sharp (specific subdivision rules which respect sharpness of the edges) (see red edges on Figure 6). This control polyhedron containing triangles, quadrangles, higher order polygons and marked edges is then encoded. Concerning connectivity information, we have chosen to implement the Face Fixer [7] algorithm (see Section 2.1) seeing that this encoding scheme is based on edges and allows to process arbitrary polygonal meshes and not just fully triangulated ones. In addition this scheme, which provides quite good compression rates, is able to encode easily face groupings which can be useful, in a perspective way, to transmit the segmentation results within the object. This algorithm encodes the connectivity graph by a list of n labels (among ≈ 10, depending on the maximum face degree), with n the number of edges. The corresponding bit stream is created using an arithmetic coder which achieves quite good results. Concerning geometry encoding, which remains in progress, a 10 bit quantization is performed. Once the positions of three vertices of a planar face are known, we have to encode only 2 coordinates for the remaining vertices. Flags on the edges (sharp or not) are represented by a n sized binary vector, encoded with a run length algorithm. Thus the total size of the compressed stream is the sum of the connectivity (C), geometry (G) and flags (F) sizes (see examples in Figure 6, C, G and F are given in bytes).

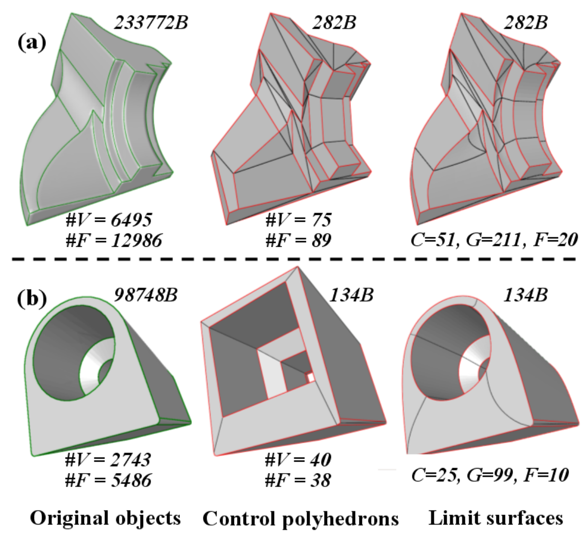

Figure 6: Results of our subdivision surface fitting algorithm, and corresponding file sizes (in bytes).

6. RESULTS AND DISCUSSIONS

Our compression method was tested on several different objects; figure 6 shows the number of vertices and faces of the original objects and of the corresponding found control polyhedrons for two mechanical models: Fandisk (a) and Swivel (b). Control polyhedrons have widely less faces and vertices compared with initial surfaces and the approximation errors remain very low (limit surfaces are very close from original objects). Mean L1 and L2 errors are shown on Table 1, they are calculated between the original object and the subdivision surface after 4 refinement steps. Table 1 shows original binary sizes (*BS*), compressed sizes (*CS*) and compression rates ($CR = BS/CS$) for both objects, after encoding. Table 2 shows a comparison, for the Fandisk object, with different state of the art algorithms: Alliez and Desbrun progressive encoding [1] and the wavelets based algorithms from Khodakovsky et al. [10] and Valette et al. [19]. Our algorithm achieves drastically better compression rates (≈ 800), while keeping a low geometric error. Coders from Alliez and Valette are lossless thus the geometric error is limited to the quantization error *QE* (a 10 bits quantization, like ours).

	$BS\,(Bytes)$	$CS\,(Bytes)$	CR	$L1/L2\,(10^{-3})$
Fandisk	233772	282	829	$0{,}99/0{,}012$
Swivel	98748	134	737	$3{,}03/0{,}044$

Table 1: Original binary sizes (*BS*), compressed sizes (CS), compression rates (*CR*) and L1 and L2 errors.

	Alliez	*Valette*	*Kodakovsky*	*Ours*
$Size\,(Bytes)$	14075	10603	6063	282
CR	17	22	39	829
$L2\,(10^{-3})$	*QE*	*QE*	$0{,}045$	$0{,}012$

Table 2: Compressed sizes, associated compression rates and L2 errors for several approaches applied to the Fandisk object.

7. CONCLUSION

We have presented a new framework for compression and coding of 3D models. Our approach, particularly adapted for mechanical objects, is based on subdivision surface fitting. Our approximation algorithm aims at optimizing the connectivity of the generated subdivision control polyhedron which is then encoded specifically. After a segmentation step, the 3D object is divided into surface patches of which boundaries are approximated with subdivision curves which lead to the subdivision control polyhedrons by linking control points of the boundary control polygons. These edges are created with respect to the lines of curvature, to preserve the natural parameterization of the target surfaces. The final control polyhedron containing triangles, quadrangles, higher order polygons and marked edges is then created by assembling local subdivision control polyhedrons and encoded using an efficient edge based algorithm followed by an entropic coding for the connectivity and a 10 bit quantization for the geometry. Results show quite impressive compression rates compared with state of the art algorithms, even if our approach still presents some limitations which represent interesting perspective works: results are effective for mechanical models since they present large constant curvature regions which are particularly adapted for subdivision inversion; our method is not suited for natural objects. An other limitation is the difficulty to adjust the resulting error, seeing that local subdivision surface construction mainly relies on boundary approximation. As future work we wish to introduce a surface optimisation scheme to better manage this resulting error. Finding a way to treat natural noisy objects is also of interest.

Acknowledgements

This work is supported by the French Research Ministry and the RNRT (Reseau National de Recherche en Telecommunications) within the framework of the Semantic-3D national project (http://www.semantic-3d.net).

REFERENCES

[1] P. Alliez, M. Desbrun, Progressive Encoding for Lossless Transmission of 3D Meshes, *ACM Siggraph Conference Proceedings*, pp. 198–205, 2001.

[2] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Levy, M. Desbrun, Anisotropic polygonal remeshing, *ACM Transactions on Graphics*, vol. 22 (3), pp. 485–493, 2003.

[3] C. Gotsman, S. Gumhold, L. Kobbelt , Simplification and compression of 3D-meshes, *Tutorials on multiresolution in geometric modeling*, A. Iske, E. Quak, M. Floater (eds.), Springer, 2002.

[4] S. Gumhold, W. Strasser, Real time compression of triangle mesh connectivity, *ACM Siggraph Conference Proceedings*, pp. 133–140, 1998.

[5] S. Hertel, K. Mehlhorn, Fast triangulation of simple polygons, *Lecture Notes In Computer Science, Proceedings of the International FCT-Conference on Fundamentals of Computation Theory*. vol. 158, pp. 207–218, 1983.

[6] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, W. Stuetzle, Piecewise smooth surface reconstruction., *ACM Siggraph Conference Proceedings*, pp. 295–302, 1994.

[7] M. Isenburg, J. Snoeyink, Face Fixer: Compressing Polygon Meshes with Properties, *ACM Siggraph Conference Proceedings*, pp. 263–270, 2001.

[8] W.K. Jeong, H.C. Kim, Direct reconstruction of displaced subdivision surface from unorganized points, *Journal of Graphical Models*, vol. 64 (2), pp. 78–93, 2002.

[9] Z. Karni, C. Gotsman, Spectral compression of mesh geometry, *ACM Siggraph Conference Proceedings*, pp. 279–286, 2000.

[10] A. Khodakovsky, P. Schröder, W. Sweldens, Progressive Geometry Compression, *ACM Siggraph Conference Proceedings*, pp. 271–278, 2000.

[11] G. Lavoue, F. Dupont, A. Baskurt, A new cad mesh segmentation method, based on curvature tensor analysis, *Computer Aided Design*, 2005, In press.

[12] G. Lavoue, F. Dupont, A. Baskurt, A new subdivision based approach for piecewise smooth approximation of 3D polygonal curves, *Pattern Recognition*, 2005, Accepted for publication.

[13] A. Lee, H. Moreton, H. Hoppe, Displaced subdivision surfaces, *ACM Siggraph Conference Proceedings*, pp. 85–94, 2002;

[14] W. Ma, X. Ma, S. Tso, Z. Pan, A direct approach for subdivision surface fitting from a dense triangle mesh., *Computer Aided Design*, vol. 36 (16), pp. 525–536, 2004.

[15] MPEG4, ISO/IEC 14496-16. Coding of Audio-Visual Objects: Animation Framework eXtension (AFX), 2002.

[16] J. Stam, C. Loop, Quad/triangle subdivision, *Computer Graphics Forum* vol. 22 (1), pp. 79–85, 2003.

[17] H. Susuki, Subdivision surface fitting to a range of points, *Pacific graphics proceedings*, pp. 158–167, 1999.

[18] C. Touma, C. Gotsman, Triangle mesh compression, *Graphic Interface Conference Proceedings*, pp. 26–34, 1998.

[19] S. Valette, R. Prost, A Wavelet-Based Progressive Compression Scheme For Triangle Meshes : Wavemesh, *IEEE Transactions on Visualization and Computer Graphics*, vol. 10 (2), pp. 123–129, 2004.