A multiple B-Spline representation for progressive 3D mesh compression

KHALED MAMOU, TITUS ZAHARIA, and FRANCOISE PRETEUX

Groupe des Ecoles des Télécommunications / Institut National des Télécommunications
Unité de projets ARTEMIS
9, rue Charles Fourier 91011 Evry, France

Abstract: - This paper proposes a new progressive compression scheme for 3D triangular meshes, based on a multi-patch B-Spline representation. First, the mesh is segmented into multiple patches. Each patch is then parameterized, and approximated by a B-Spline surface. The B-Spline control points are stored into 2D images and compressed using optimized still image encoders. The initial mesh topology is lossless encoded by applying the Touma and Gotsman (TG) algorithm. The performances of the proposed compression scheme show significant gains (30% on average) when compared to those of the Spectral Compression (SC), and outperforms TG and MPEG-4 encoders especially within the range of low bitrates (less than 8 bits per vertex).

Key-Words: - 3D mesh compression, geometry images, B-Spline, parameterization.

1 Introduction

Graphic content is today extensively exploited within various, both professional and general public application domains, including CAD, medical imagery, video gaming, synthetic film production…

Graphics cards currently available on most general public devices (PCs, PDAs…) are optimized for real-time rendering of 3D meshes, which are today the most popular way of representing 3D content. However, such 3D mesh representations require a huge amount of both bandwidth and memory resources for ensuring their transmission and storage on different networks and terminals. The challenge of elaborating efficient and compact 3D data representations has been extensively addressed within the last decade resulting in a rich literature of which some comprehensive overviews are presented in [11], [25].

Let us first mention the monoresolution, connectivity driven compression approaches. Here, a deterministic traversal of the mesh makes it possible to implicitly represent and lossless encode the adjacency relations between vertices, edges and faces. The geometry is then differentially encoded by exploiting the neighborhood relations and the above-mentioned traversal order, within the framework of a prediction scheme. As representative of this family, let us mention the so-called "Topological Surgery" [8] method, adopted within the MPEG-4 standard, and the Touma and Gotsman (TG) [7] coding scheme, long time considered as the state of the art of monoresolution compression techniques.

Ensuring scalable transmission of 3D content requires the elaboration of progressive compression schemes that require multiresolution representations with multiple levels of detail.

Within this framework, let us first mention the spectral compression (SC) technique introduced in [17], which combines lossless encoding of the connectivity [7] with progressive geometry compression. Here, the geometry signal is decomposed into the basis of the eigenvectors of the Laplacian operator, which extends for 3D meshes the 2D DCT transform. The spectral representation is highly efficient especially for low bitrate compression of smooth 3D models. However, the major limitation comes from the expensive computational complexity (cubic with the number of vertices) of the decoding process, which makes it un-tractable in practice.

Recent approaches completely modify the initial mesh connectivity in order to obtain semiregular [9], [10] or regular [5], [16] topological representations well-suited for compression purposes. The principle consists of deriving, from the initial mesh a low-distortion parameterization. The mesh geometry is then represented as a 2D image (so-called geometry image), obtained by re-sampling the parametric domain, which can be efficiently compressed with standard, optimized still image encoders such as JPEG, JPEG2000… Here again, the underlying smoothness and sampling density assumptions are essential for guarantying the compression efficiency.

Such methods are fully progressive and lead to high compression rates, since the topological information is totally discarded and the still image encoders ensure optimized compression performances. However, they suffer from several limitations, related to the remeshing of the parametric domain. First, regularly resampling the parametric domain may lead to degenerated triangles with abnormal shape aspects, resulting in visually unpleasant artifacts [4], [5]. Second, when multiple patch parameterizations are considered [4], the heuristic stitching procedures involved do not guaranty the smoothness of the decoded mesh at the level of patch borders. Third, accurately capturing the details present in the original mesh requires a dense re-sampling [5], [10], which leads to huge decoded meshes (*e.g.* 256.000 vertices for usual 512 x 512 geometry images), unpractical for real-time rendering purposes. Fourth, certain applications do not admit the alteration of the initial connectivity, as in the case dynamic, animated mesh coding recently considered within the framework of MPEG-4/AFX standard [27].

In order to overcome such limitations, this paper introduces a novel approach for progressive, low bitrate compression of smooth and densely sampled large 3D meshes such as those obtained with 3D scanners.

As in [17], our approach combines lossless encoding of the connectivity with progressive geometry compression.

The lossless connectivity constraint relates to the industrial application of the French National SEMANTIC-3D project [28], that partially financed this study.

As in [5], [16], the mesh geometry is represented as a 2D image. Our approach exploits a connectivity driven multiple patch parameterization and a B-Spline surface approximation. The preservation of the original connectivity makes it possible to re-sample the geometry images at parametric coordinates of initial vertices, and thus to avoid the above-mentioned problems related to degenerate triangles, patch border stitching, and excessive oversampling.

The next section describes the proposed compression method and details the different stages involved. The performances of the new compression approach are objectively evaluated and discussed in Section 3. Finally, Section 4 concludes the paper and opens perspectives of future work.

2 Multiple B-spline encoding scheme

The proposed encoding scheme is presented in Figure 1. The mesh geometry is converted into a regular representation (2D images) using a B-Spline approximation [15], [16]. Unlike the approaches from [16] where a single B-Spline surface is used for representing the entire model, we propose a multiple patch representation which makes it possible to accurately represent 3D shapes of arbitrary complexity. In order to determine the patches, a segmentation procedure is first applied. Each patch is parameterized using the Tutte's algorithm [1] with the additional optimization introduced in [18], and then approached by a B-Spline surface. The B-Spline control points are finally quantized and stored into three grey-scales images one for each *x*, *y* and *z* control point coordinates. This B-Spline geometry images are finally compressed using a still image encoder. In this paper, we have selected the JPEG 2000 encoder, for its recognized compression performances.

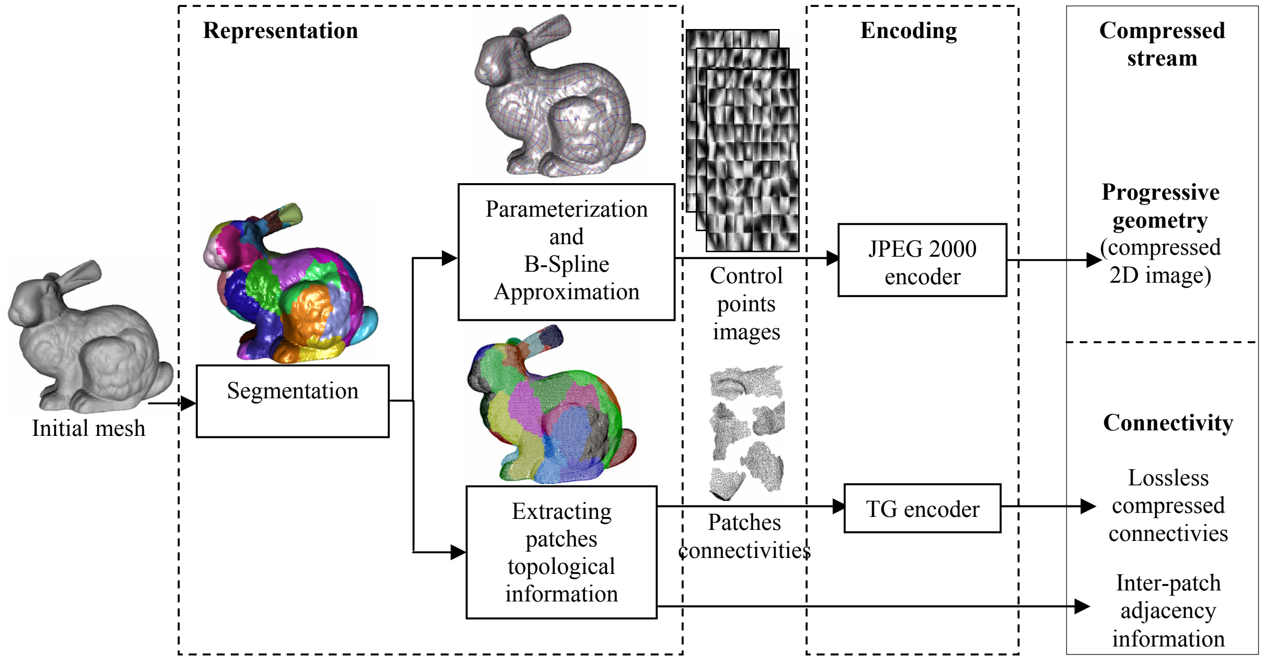


Figure 1: Multiple B-Spline representation and compression scheme.

The connectivity of the patches is lossless encoded separately using the TG approach [7]. An additional topological information describing the inter-connexions between patches is also integrated into the binary stream.

At the decoder side, the connectivities of the patches are first recovered. This information makes it possible to recompute, for each patch the Tutte parameterisation, which does not require the knowledge of any geometric properties. The patches are then stitched together with the help of the additional topological information transmitted. In this way, the initial connectivity is totally recovered.

The geometric information is decoded by reconstructing B-Spline surfaces from the transmitted control points and then sampling them at the parametric coordinates associated to the mesh vertices.

Let us now further detail each of the stages involved within the proposed coding scheme.

2.1 Mesh segmentation into patches

The objective of the segmentation stage is to obtain a partition of the initial mesh into mutually disjoint patches. Each patch should be homeomorphic to a disc and well-suited for low distortion parameterizations (*cf.* Section 2.2).

Existing techniques [3], [4], [5], offers several tradeoffs between quality of parameterization and computational complexity. In [3], the proposed segmentation procedure works well for texture mapping purposes, but the resulting patches might be too tiny, which is inappropriate when considering compression applications. On the contrary, the patches obtained with the algorithm described in [5] are too complex for enabling low distortion parameterizations. This drawback is overcome by the algorithm proposed in [4], which extends of the Lloyd-Max quantizer, but at the price of a prohibitive computational complexity.

In this paper, we introduce a rapid and automatic segmentation algorithm applicable to any manifold mesh of genus 0. The proposed approach makes it possible to:

- uniformly distribute the Gaussian curvatures of the mesh over the patches,
- privilege mesh segmentation along crease features,
- generate compact boundaries by minimizing the number of patch border edges,

at a low computing cost (linear complexity with the number of the mesh faces).

Let F denote the set of faces of M and E the set of its internal edges (note that for a manifold mesh an internal edge has exactly two incident faces). The goal is to determine a partition $\Pi = \{F_1, F_2, \ldots, F_N\}$ of F such that the quantities:

$$\sum_{f \in F_i} w_{gauss}(f) \qquad (1)$$

are equal and the total cut cost

$$C = \sum_{e \in E^{border}(\Pi)} w_{angular}(e) \qquad (2)$$

is minimal. Here,

- $w_{gauss}(f) = \sum_i \left| k(v_i^f) \right|$, with $k(v_i^f)$ denoting the Gaussian curvature of the mesh evaluated at vertex v_i^f, $\{v_i^f\}_i$ the set of vertices of face f, and $| \ |$ the absolute value of a real number,
- $w_{angular}(e) = \dfrac{1}{\left\| n(f_1^e) \times n(f_2^e) \right\|}$, with $n(f)$ being the unit normal vector of face f, f_1^e and f_2^e the two faces incident to edge e, $\| \ \|$ the L_2 norm, and $\times$ the vector product of two vectors,
- $E^{border}(\Pi)$ denotes the set of edges located at the frontier between different patches of the partition.

In order to resolve this NP hard optimization problem, we have adopted the heuristic graph partitioning algorithm described in [13]. In order to reduce the computational complexity, the mesh is massively simplified by applying a sequence of edge collapse operations [26], performed on the dual graph of the mesh. During this simplification process, the weights $w_{gauss}(f)$ and $w_{angular}(e)$ are consequently accumulated for taking into account the gaussian curvature and crease angle distributions of the initial mesh. After solving the optimization problem described above on the coarse mesh, the segmentation of the initial mesh is obtained by reversing the simplification process and propagating the partition labels from coarse to fine levels.

The different patches thus obtained are finally cut into separate connected components, by duplicating the frontier vertices and the edges. The adjacency relations of the patches are totally described by these cut operations (*cf.* Section 2.4).

Let us note that the obtained patches may present multiples boundaries. In order to obtain patches with a single boundary, the different boundaries are further interconnected by performing a supplementary cut along the shortest path between the two boundary loops [24].

Moreover, in order to avoid patches with tubular shape aspects, a final refinement is performed, which consists of interconnecting (by additional cut operations) the boundary loop with the internal vertex of highest gaussian curvature.

The resulting patches are then independently parameterized, as described in the following section.

2.2 Patch parameterization

By definition, a parameterization of a surface S is a bijective mapping f between a 2D domain and the surface S, as expressed in the following equation:

$$f : D \subset \mathrm{IR}^2 \rightarrow S \subset \mathrm{IR}^3$$
$$(u,v) \in D \mapsto S(u,v) = (f_x(u,v), f_y(u,v), f_z(u,v)) \qquad (3)$$

The issue of determining "good" parameterizations of triangulated surfaces has been intensively studied in the literature [1], [2], [12], [14], existing techniques integrating both topological and geometric information.

However, within our compression scheme, the parameterization needs to be determined uniquely from the connectivity information, since it serves for determining geometric properties.

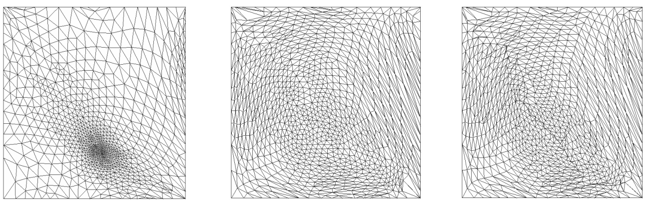
For this reason, we have retained the Tutte's method [1] which makes it possible to embed into IR^2 simply connected planar graphs, without edge self-intersections. This procedure consists of the following two steps:

- First, the boundary loop is mapped into a convex form of the plane (usually a 2D square).
- Then, the positions of the internal vertices are computed by supposing that each vertex is the barycentre of its neighbors.

Taking into account solely the topological information, the Tutte's method might introduce important geometric distortions, *i.e.* does not preserve metric properties such as length, area, angle between initial and parametric domain.

In order to minimize such distortions, the optimization procedure described in [18] is further applied, under the hypothesis that the mesh triangles have similar aspect ratios. Here, the idea consists of exploiting the analogy with a physical system constituted of masses located at the vertices of the mesh and connected with springs according to the edges. The Tutte parameterization is considered as a first guess solution. This initial parameterization is then iteratively enhanced. At each iteration, the springs elasticity coefficients are updated so that stretching induced by the parameterisation is minimized.

Figure 2 shows three different parameterizations, obtained with Tutte, optimized Tutte and Floater [2] algorithms, which integrates geometric information.



2.a. Tutte 2.b. Optimized Tutte 2.c. Floater.

Figure 2: Different mesh parameterizations.

We note that the optimized Tutte approaches the Floater results and thus enhances the initial Tutte parameterization.

Let us note that the parameterization needs to be recomputed by the decoder. This step is determinant for the complexity of the decoding algorithm. Here, we solve a linear system with $N \times N$ (N being the number of vertices per patch) sparse, symmetric, and positive defined matrices. By using the gradient conjugate [19] method, the computing times are around 0,3 seconds per patch of about few thousands faces (on a machine P4, 1.8GHz and 1Go of memory).

The obtained parameterization makes it possible to approximate in an elegant and computationally tractable manner the 3D mesh geometry by a B-Spline surface.

2.3 B-Spline approximation

Because of their attractive mathematical properties (smoothness, regularity, power of representation, local control, fast computation of geometric properties…) the B-Spline surfaces have been intensively used in various application domain, including CAD, modelling of virtual avatars…

A B-Spline surface (*cf.* [6], Chapter 3) is defined as:

$$\forall\ (u,\ v) \in [0,\ 1] \times [0,\ 1],$$
$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} P_{ij} N_{i,p}(u) N_{j,q}(v)\,, \qquad (4)$$

where (u,v) are the parametric coordinates, $P_{ij} \in \mathrm{IR}^3$ the surface control points and $N_i^p(u)$, $N_j^q(v)$ the B-Spline basis functions of degree p and q, defined over knot vectors $U = \left\{ \underbrace{0,...0}_{p+1}, u_{p+1} ..., u_i, ..., u_n, \underbrace{1,...,1}_{p+1} \right\}$ and $V = \left\{ \underbrace{0,...0}_{q+1}, v_{q+1} ..., v_j, ..., v_m, \underbrace{1,...,1}_{q+1} \right\}$.

After parameterization, a couple $(u,v) \in \mathrm{IR}^2$ of parametric coordinates is associated to each mesh vertex. In this way, the mesh vertices irregularly sample the parametric domain. Approximating a set of irregular points by a B-Spline is of course possible, by using scattered data interpolation techniques such as those described in [20], [21], [22]. However, these non-linear optimisation procedures are highly complex and do not guaranty theoretically the convergence of the algorithms.

For this reason, we have adopted a simple, yet efficient solution [6] consisting of:

- computing a grid of points $\{Q_{k,l}\}$ on the mesh by resampling the parametric domain according to a uniform sampling grid $\{(\overline{u}_{k,l}, \overline{v}_{k,l})\} \subset [0,1] \times [0,1]$,
- determining the control point matrix $P = \{P_{i,j}\}$ of the B-Spline surface which best fit (in the mean square error sense) the data points $\{Q_{k,l}\}$.

Matrix P is determined by applying the cross-sectional procedure described in [6], which guarantees in all cases the existence of the solution, obtained by solving a set of well-conditioned linear systems.

Being defined on a rectangular lattice, the control points $P_{i,j}$ can be stored in three grey-scales images, one for each x, y, and z coordinate $P^x_{i,j}$, $P^y_{i,j}$, and $P^z_{i,j}$. Each image is finally compressed by using the JEPG 2000 progressive encoder, adopted because of its highly optimal performances.

The above-described procedure makes it possible to determine for each individual patch the geometry information. As the per-patch connectivity is also available, the last step of our algorithm concerns the stitching of different patches.

2.4 Patch stitching

For being able to stitch the different patches at the decoder side it is necessary to integrate into the bitstream an additional topological information describing the cuts operated on the mesh (Section 2.1).

For each cut c, the following overhead topological information is transmitted:

- the two neighbouring patches, F_1^c and F_2^c, to be connected,
- the start vertices of v_s^1 *et* v_s^2 within each patch boundary loop,
- the arrival vertex v_e^1 of the patch F_1^c, and
- the senses of traversal on the boundaries s^1 and s^2 of each patch.

The patch stitching algorithm is illustrated in Figure 3. Considering these elements, the decoder is capable of recovering the original connectivity.

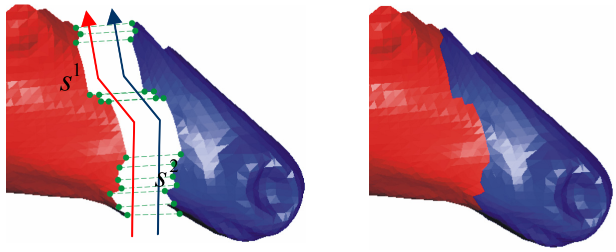


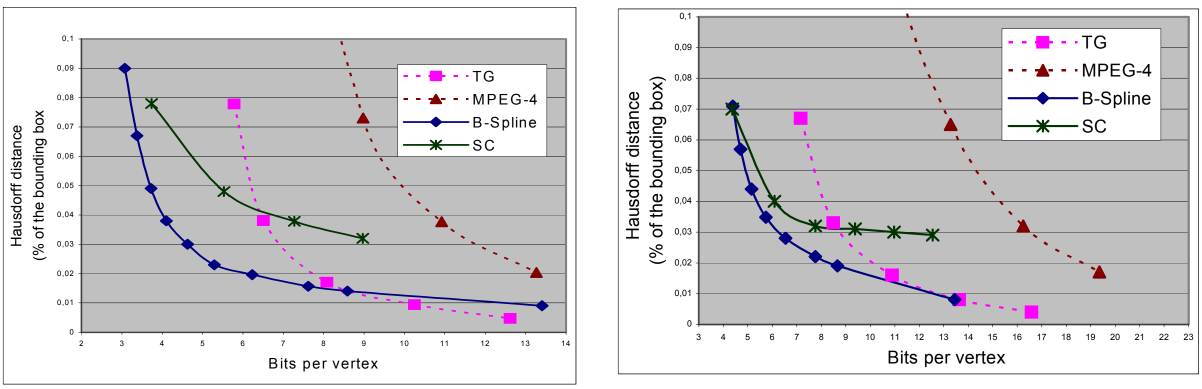
Figure 3: Merging two distinct patches together.

3 Experimental results

The experiments have been carried on a set of a dozen of meshes usually used for evaluation purposes in the literature [4], [5], [7], [10], [17]. Such meshes exhibit smooth and densely sampled geometries and were obtained with 3D scanners. The original data are available on the Stanford and Cyberware repositories [29], [30]. The performances of the encoder are expressed in terms of rate/distortion curves. The rates are expressed in bit per vertex (bpv). The distortions are measured using the Hausdorff distance between the original and the compressed mesh estimated with the MESH software described in [23] and available at http://mesh.epfl.ch.

All the results were obtained by subdividing the meshes into 50 patches and using second order B-Spline surfaces with 24×24 control points.

Figure 4 presents the rate/distortion curves for the "Venus" and "Horse" models obtained using our B-Spline approach, the TG compression scheme [7], the MPEG-4 encoder [8], and the spectral compression method [17]. For [17], the meshes were segmented into 120 patches and the spectrum was quantized on 14 bits.



4.a. "Horse" model. 4.b. "Venus" model.

Figure 4: Rate/distortion curves for "Horse" and "Venus" models.

The proposed compression schema leads to the best results within the range of low bitrates (< 8 bpv).

For bitrates higher than 8 bpv, the TG encoder provides slightly better rate/distortion ratios. However, the obtained gains are minimal: the associated distortion being negligible (about 0,02% of the diagonal on the bounding box of the mesh). At such error levels, the visual quality of the generated meshes becomes quasiconstant!

The SC technique also proves to be highly efficient, especially at very low bit rates. However, within the range from 4 to 10 bps, our encoder performs significantly better than the SC approach, with an average gain of 30%.

Figures 5 and 6 illustrate some compressed meshes at different bitrates and with several methods. The visual quality of the B-Spline compressed meshes is obviously higher than the one obtained using the TG and MPEG-4 encoders. This is can be explained by the B-Spline representation which guaranties smooth surfaces and thus visually graceful compressed models all along the transmission process, starting from very low bitrates.

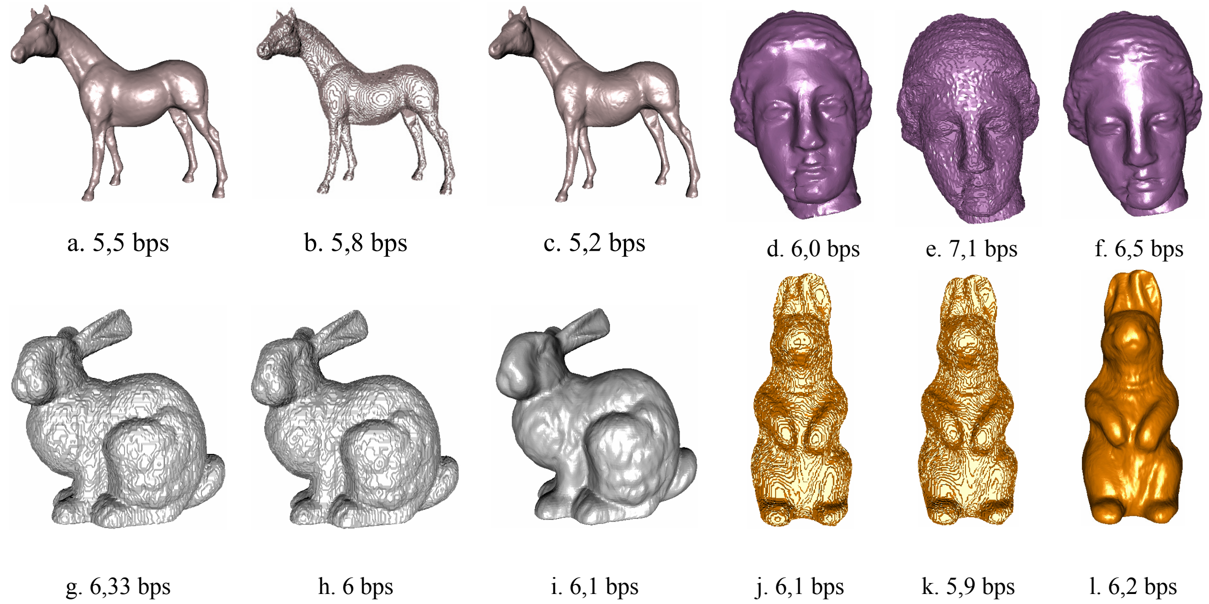


Figure 5: Compressed meshes for “Horse”, “Venus”, “Bunny”, and “Rabbit” models obtained with: SC (a and d) MPEG-4 (g, and j), TG (b, e, h, and k) and our approach (c, f, i, and l).

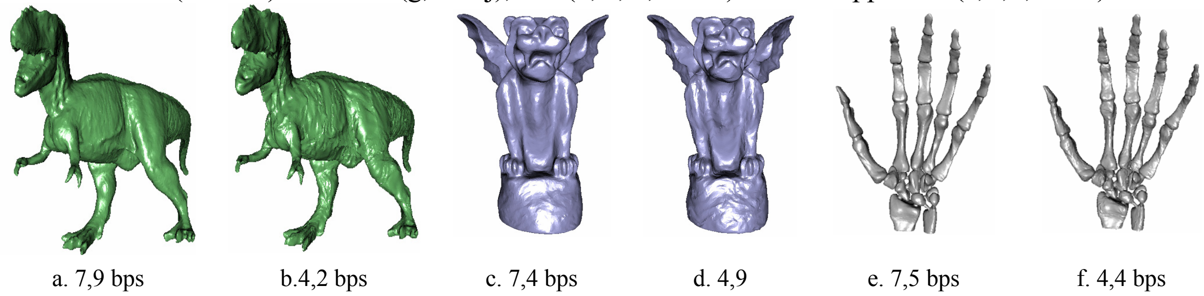


Figure 6: Compression results for different meshes (“Tyra”(a,b), “Gargoyle”(c, d), “Hand”(e, f)) at several bitrates.

The presented results show that the proposed B-Spline approach is particularly suited for low bitrate, progressive compression, with lossless connectivity.

4 Conclusion and future work

This paper introduced a new 3D mesh compression scheme, based on a multiple B-Spline representation and dedicated to densely sampled, smooth mesh models.

The geometry is progressively encoded, while preserving the initial mesh connectivity, which is lossless compressed.

The several stages necessary for building the novel representation were described in details.

Experimental results show that the proposed compression scheme outperforms the other connectivity preserving approaches, such as MPEG-4, Touma and Gotsman and spectral compression, for low bitrates (less than 8 bpv).

Future work, will address the issue of adapting the proposed encoder to the compression with loss of connectivity. The generation of a triangulation inter-patches with no artefacts and degenerate triangles will be particularly studied. Within this framework, a more efficient parameterization technique, integrating geometric metric properties will be considered.

5. Acknowledgment

This work was partially supported by the French Research Ministry and the RNRT (*Réseau National de Recherche en Télécommunications*) within the framework of the SEMANTIC-3D national project.

References:

[1] W.T. Tutte "How to draw a graph", *Proc. London Math. Soc. 13*, pp. 743–768, 1963.

[2] M.S. Floater, "Parametrization and smooth approximation of surface triangulations", *Computer Aided Geometric Design*, 14 (3):231-250, 1997.

[3] B. Lévy, S. Petitjean, J. Maillot, "Least squares conformal maps for automatic atlas generation", *SIGGRAPH'2002*, pp. 362-371, 2002.

[4] P. Sander, Z. Wood, S. Gortler, J. Snyder, H. Hoppe, "Multi-chart geometry images" *Eurographics Symposium on Geometry Proc. 2003*, pp. 146-55, 2003.

[5] X. Gu, S. Gortler, H. Hoppe, "Geometry images", *ACM SIGGRAPH 2002*, pp. 355-361, 2002.

[6] L. Piegl, W. Tiller "The NURBS Book", Springer-Verlag, New York, NY, 1997.

[7] C. Touma, C. Gotsman, "Triangle mesh compression", *Proc. of the 24th Conference on Graphics Interface*, pp. 26-34, San Francisco, CA, USA, 1998.

[8] G. Taubin, J. Rossignac, "Geometric Compression through Topological Surgery", *ACM Trans. Graphics*, pp. 84–115, 1998.

[9] J. Szymczak, Rossignac, D. King, "Piecewise Regular Meshes: Construction and Compression". *Graphic Models*, pp. 183-198, 2002.

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

[11] P. Alliez, C. Gotsman, "Recent Advances in Compression of 3D Meshes", *Proc. of the Symposium on Multiresolution in Geometric Modeling*, 2003.

[12] K. Hormann, M.S. Floater, "Surface Parameterization: a tutorial and Survey", *Cambridge workshop on multiresolution in geometric modelling*, Sept. 2003.

[13] G. Karypis, V. Kumar, "Multilevel *k*-way Partitioning Scheme for Irregular Graphs", *Technical Report 036*, Minneapolis, MN 55454, May 1996.

[14] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, D. Dobkin, "MAPS: Multiresolution Adaptive Parameterization of Surfaces", *Computer Graphics Proceedings (SIGGRAPH 98)*, pp. 95-104, 1998.

[15] V. Krishnamurthy, M. Levoy, "Fitting Smooth Surfaces to Dense Polygon Meshes", *Proc. SIGGRAPH '96*, 1996.

[16] F. Losasso, H. Hoppe, S. Schaefer, J. Warren, "Smooth geometry images", *Eurographics Symposium on Geometry Processing*, pp. 138-145, 2003.

[17] Z. Karni, C. Gotsman "Spectral Compression of Mesh Geometry", *Siggraph 2000 Computer Graphics Proceedings*, pp. 279-286, 2000.

[18] Shin Yoshizawa, Alexander G. Belyaev, Hans-Peter Seidel "A fast and simple stretch-minimizing mesh parameterization" *Int. Conf. on Shape Modeling and Applications*, pp. 200-208, 2004.

[19] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, "Numerical Recipes in C++: The Art of Scientific Computing". Cambridge Univ. Press, 2002.

[20] B. F. Gregorski, B. Hamann, K. I. Joy, "Reconstruction of B-Spline Surfaces from Scattered Data Points", *Proc.of the Int.l Conf. on Computer Graphics*, pp.163, 2000.

[21] J. Haber, F. Zeilfelder, O. Davydov, and H.-P. Seidel, "Smooth approximation and rendering of large scattered data sets", *Proc. of IEEE Visualization*, pp.341-347, 571, IEEE Computer Society, 2001.

[22] M. Eck, H. Hoppe, "Automatic Reconstruction of B-Spline Surfaces of Arbitrary Topological Type", *ACM SIGGRAPH*, pp. 325-334, 1996.

[23] N. Aspert, D. Santa-Cruz, T. Ebrahimi, "MESH: Measuring Errors between Surfaces using the Hausdorff distance", *Proc. of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 705-708, 2002.

[24] E.W. Dijkstra "A note on two problems in connection with graphs", *Numerische Mathematik*, pp. 1:269-271, 1959.

[25] K. Mamou, T. Zaharia, F. Preteux, "3D Mesh Coding Techniques Applied to CAD Data: A Comparative Evaluation", *Proc. 2nd Int. Conf. on Intelligent Access to the Multimedia Documents and on the Internet* (MEDIANET 2004), pp. 17-34, Tunisia, Nov. 2004.

[26] H. Hoppe, "Progressive meshes", *Proc. ACM SigGraph*, pp. 99-108, 1996.

[27] The MPEG-4 International Standard, Part 16 – Animation Framework eXtension, ISO/IEC 14496-16, Geneva, Switzerland, 2003.

[28] www.telecom.gouv.org/RNRT

[29] http://graphics.stanford.edu/

[30] www.cyberware.com